Kony for SAP

Technical Library

Release 24.04

May, 2015

Revision History

Date	Document Version	Description of Modifications/Release
05/29/2015	1.6	Updated HTTP Services filter options in *Resource Service* for SAP 24.04 Release
02/17/2015	1.5	Added new HTTP header options for buffering and Media services, updated UI for HTTP Test and Trace Utility for SAP 24.04 Release
11/27/2014	1.4	Added enhancements in *HTTP Services Test Tool*, and *Developing HTTP Services* for SAP 24.04 Release
09/02/2014	1.3	Added *Installing the MOM Utility* and *Running a Performance Test* for Kony for SAP 24.04 Release
07/25/2014	1.2	Added *Text description maintenance for Dictionary Workbench and an Integrated Object Workbench (IOB)*, *HTTP Trace linking from Local database Workbench (LDB) and Data Object Workbench (DOB)*, *IOB linking from DOB and LDB workbenches*, *new HTTP Headers*, *HTTP Test Tool enhancements*, *HTTP Services Batch processing*, *Notification Manager enhancements*, *HTTP Services Batch processing*, *Computer Center Management System Monitor Utility*, *Solution Manager Alert Configuration*, *Support Report*, *Server Log Report*, *On demand binary feature*, *Deprovisioning users* for Kony for SAP 24.04 Release

Date	Document Version	Description of Modifications/Release
05/12/2014	1.1	Added *Group Maintenance*, *User and Sessions Report*, *Text Objects*, *Performance Test Workbench*, *HTTP Services Trace Utility*, *HTTP Services Test Utility*, *HTTP Services* for Kony for SAP 24.04 Release
09/18/2013	1.0	Document Release for Kony for SAP

1. Preface

The SAP Add-in Framework is:

- A certified solution developed specifically for SAP
- A complete mobile technology framework that enables to deploy business applications efficiently and effectively, that are fully integrated with SAP.

Its primary objectives are to provide a run anywhere, view anywhere infrastructure that enables the rapid development and deployment of "light weight" low cost mobile applications, without the need for complex external middleware and infrastructure.

The add-in provides a single integrated framework that is used to deploy and manage applications on all major mobile platforms including Windows, BlackBerry, Android, and IOS (iPhone/iPad). It is a complete proven mobile infrastructure built specifically for SAP, supporting both packaged and customized applications. Its unique architecture and full utilization of the SAP technology provides a clear advantage over other typical mobile solutions, making it incredibly cost effective and easy to implement and manage.

1.1 Purpose

This document describes how to use SAP add-in framework.

1.2 Intended Audience

This document is intended for Administrators, end users, Architects, and Developers.

1.3 Formatting Conventions

Following are the formatting conventions used throughout the document:

Convention	Explanation
`Monospace`	■ User input text, system prompts, and responses ■ File path ■ Commands ■ Program code ■ File names.
Italic	■ Emphasis ■ Names of books, and documents ■ New terminology.
Bold	■ Windows ■ Menus ■ Buttons ■ Icons ■ Fields ■ Tabs.
URL	Active link to a URL
Note	Provides helpful hints or additional information
Important	Highlights actions or information that might cause problems to systems or data

1.4 Contact Us

We welcome your feedback on our documentation. Write to us at techpubs@kony.com.

For technical questions, suggestions, comments or to report problems on Kony product line, contact prodsupport@kony.com.

1.5 The Add-in Release 24.04

Important: When you upgrade your SAP add-in to 24.04 and run the check install (YECI), review the licence information carefully. The add-in license checking changed with the v24.02 release and is backward compatible with your previous license definition. However, on the rare chance that you are impacted by the number of users and /or interface runs, contact Kony to generate a new license definition.

1.5.1 Installation and Upgrade

The Installation and Upgrade section is for system administrators and others tasked with deploying the Kony for SAP solution. There are three base components: the Secure Container that is deployed onto the device, The Access Gateway that is typically deployed behind the firewall in a DMZ, and the MEAP server that is implemented inside SAP or on its own Netweaver instance.

1.5.2 Development and Customization

The Development and customization section is for developers creating and maintaining applications developed using the SAP Add-in. It starts with the basic concepts, the IDE and then moves into more complex components such as ABAP exits and Java exits.

1.5.3 System Management and Administration

The System Management and Administration section is for system administrators. It is broken down into sections on the administration of the Secure Container, management of the MEAP Server, options for tracing and logging, tuning as well as security.

2. Disclaimer

Table of Contents

3. Installation and Upgrade

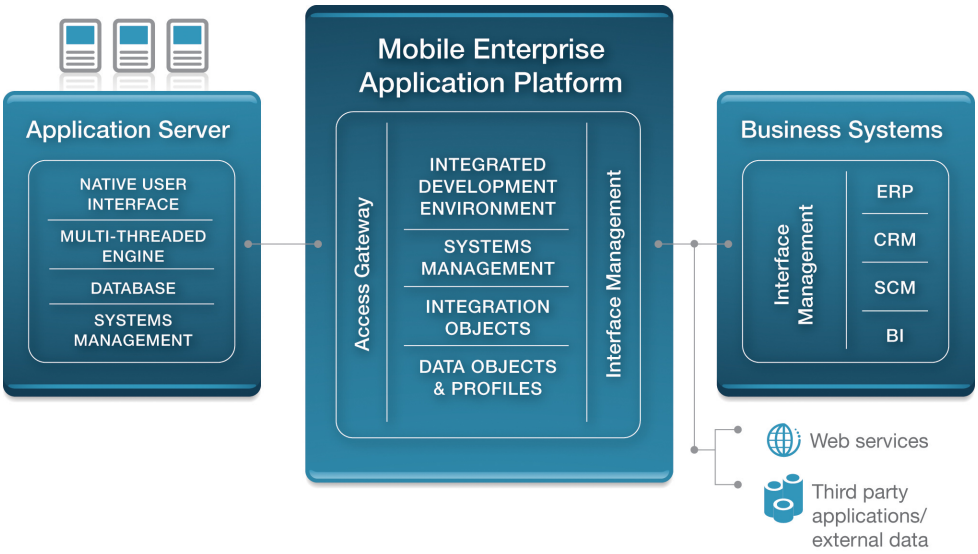
The section is aimed at administrators who are responsible to install Kony for SAP. This guide is specific to the Kony for SAP product, and contains information pertaining to installation on the variety of platforms that Kony currently supports. It describes the three main components of the software, namely: the Secure Container that is deployed onto the device, the Access Gateway that typically sits behind the firewall in the DMZ and the MEAP Server that is implemented inside SAP or on its own Netweaver instance. The Secure Container is designed to be automatically provisioned (that is have its configuration downloaded and automatically implemented) from the central MEAP Server using the Server Profiler functionality. This process occurs when you first start it. You can also configure it manually and bypass the provisioning process altogether.

3.1 Key Topics

Summary of components
SAP MEAP Server
Access Gateway
Secure Container

3.2 Summary of Components

Kony for SAP provides a complete framework to support the full life cycle of mobile applications through design, build, deploy and operation. The framework supports a variety of operating systems and device types, effectively providing the entire infrastructure required for a MEAP (Mobile Enterprise Application Platform). The following diagram provides a high level overview of all the components. There are three main technology layers, a Secure Container that runs on the device that communicates with a central Access Gateway that in turn communicates with a central MEAP Server. All communication between the container on the remote device and the gateway is over TCP/IP using the sockets or Http(s) protocol. Communication between the gateway and the MEAP server in SAP is through the standard SAP RFC protocol (as per SAP certification requirements). For more information on networking options and architectures, refer to management networking.



An integrated development environment (IDE) is provided in the MEAP Server that is used to build and deploy the mobile application. The application definition is stored as meta-data; that is interpreted at runtime by the Secure Container on the device. This avoids having to implement compiled code and thus makes the applications portable to other environments. This technique allows for the easy deployment of application definitions out to remote users. The Workbench is available to customers to perform their own developments and maintenance if required. The mobile application synchronizes applications and data with the MEAP Server through a combination of push/pull mechanisms through the Access Gateway. Data profiles are configured that determine the rules by which data objects are

selected and transmitted to specific, or groups of, devices. All data transfers are compressed and encrypted (as required). The MEAP Server holds the central repository of all business data, document, and images. You should view this repository as a central staging area for all back end systems; information is copied to the staging area and then transmitted to either the mobile device or integrated with the back end. The MEAP Server provides a complete integration layer through which all interface definitions are configured, monitored and managed. This integration layer also provides the filtering and load balancing functionality to prevent overloading of, and subsequent impact to the back end systems.

3.2.1 Secure Container

The Secure Container provides a "virtual application environment" within which you execute all applications and store all data on the mobile device. You can implement the Container on all major mobile platforms, IOS, Android, BlackBerry and Windows. It is a fully multi-threaded engine and you can scale to virtually any size within the constraints of the operating system. This enables all data synchronisation tasks, network management, and peripheral device connections to run automatically in the background without user interaction. In addition, you may configure the application to run stand alone with a local database on the device. In this mode, network connectivity is attempted in the background and the user is notified when a connection is re-established or lost. A tick/cross graphic on the user interface also clearly indicates whether the device is connected or not. Secure Container has no impact or reliance on any other component on the device. Standard APIs are used throughout and no specific patches are required.

The whole concept of the secure container is to:

- Abstract applications away from the device and OS specifics; enable a "configure once, run anywhere" capability and thus avoid fragmentation
- Totally secure all business applications and data on the device
- Segregate business applications away from private applications and data
- Enable a rich native user interface and full integration with native applications and peripheral devices

3.2.2 Access Gateway

The Access Gateway is used as a pivotal point of communication between the Secure Container on the device and the central MEAP Server. It performs high-speed compression and complex encryption of data transferred over the network. You can implement multiple gateways to provide high availability failover; if the primary gateway is not available, all network traffic automatically "switches over" to the secondary gateway. The Access Gateway may be implemented on either a Windows or Unix server operating system and requires a JVM (Java Virtual Machine) to be installed. Communication between the container of the remote device is through TCP/IP using the sockets or Http(s) protocol. Communication between the gateway and the MEAP server in SAP is through the SAP RFC protocol. In the case of RFC, the gateway manages a shared connection pool that is configurable as to the minimum and maximum number of connections. If connections are not used, that is, idle for a period of time, they are scaled back to the minimum number that is configured. Connections are kept open for as long as possible for efficiency reasons as re-establishing connections and re-logging is a overhead on the SAP system. You may run the gateway logs onto SAP using a special CPIC/Service user and SAP transactions under the actual SAP user credentials through "user switching".

3.2.3 SAP MEAP Server

Written in ABAP, the certified MEAP Server add-in is portable to all 4.x, ECC, CRM, APO, BW SAP releases (including Unicode). It provides a complete application workbench to centrally design, construct and deploy mobile applications. It is also responsible for the scheduling, execution and monitoring of all SAP interfaces and a central console, is provided to monitor, analyse and safely restart failed interface runs. The add-in also provides a SAP emulator to execute mobile applications from within the SAPGUI. The SAP add-in is installed as a standard SAP transport using a unique /SKY/ namespace. No modification (repair) to standard SAP is required. An important component of the SAP add-in component is the profiling configuration that is responsible for routing the relevant information between mobile devices and the back end SAP host. These rules are configured inside SAP using the SAP Add-in workbench.

> ***Important:*** The SAP certified MEAP Server runs "inside SAP" in its own /SKY/ namespace. You may implement on any SAP instance that supports a ABAP stack, including a stand alone NetWeaver instance. Because it runs inside SAP, it has complete access to all the SAP sub-systems and therefore can provide complete end-to-end integration and enables existing SAP skills to be re-used.

3.3 Software Release Management Overview

Kony releases a complete software release approximately every quarter (three month period). This release contains all the software, documentation, and papers. Each quarterly release is given a unique release number, for example, release 24.01.00. In this way, all components are easily traceable. In between the quarterly releases, you may provide patch releases to address problems with the software, for example, 24.01.01. The latest software releases are available either from Kony Developer Portal, or from public application stores such as iTunes, Android Marketplace, and BlackBerry App World.

The format of the release code is RR.SS.PP where:

- RR is the major release number
- SS is the incremental sub-release number
- PP is the patch or fix level

Approximately every quarter, Kony provides a:

- full software release of software (from either the app stores or from Kony Developer Portal)
- update of the technical online help and tutorials
- technical newsletter outlining major functionality and relevant articles.

Kony for SAP is generally fully backward compatible with previous releases. However, Kony recommends that you read the release information in the online technical help prior to upgrading. The release information in the technical online help, documents all the enhancements and fixes in detail and provides an impact analysis with the preceding three releases and a upgrade guide. It is also possible to run previous releases of components with newer releases of others, for example, an older version of a Secure Container with the latest MEAP Server. For consistency, Kony recommends upgrading all components during an upgrade. All the necessary information is available on Kony Developer Portal.

3.4 Obtaining a License

If you require more devices than that the free version allows, you need to request a license from Kony. Licenses come in the form of a file, usually called "license.dat". They are typically sent through email to the customer who then installs them into the SAP environment using the software management utility YECI (/n/sky/yeci). The same process applies to upgrade or extending existing licenses.

3.4.1 Backward Compatibility

As of release 24.01, License checking is changed to allow for "free" versions and a more flexible subscription based model, based on the number of users and interface runs. Previous license definitions are supported, but on the rare chance that Customers are impacted, you need to generate a new license definition. For more information on this, click here.

3.5 Downloading Kony for SAP Software

You can download Kony for SAP software from Kony Developer Portal (Downloading the SkyMobile software) or you can download the Secure Container component from the Apple, Android or BlackBerry "app stores".

> ***Note:*** Refer to the Kony Developer Portal for details on how to access and navigate the SDN site. If you cannot access this for any reason, contact Kony support with your request: support@kony.com.

As well as using the public application stores and manually implementing the Kony for SAP software using the Kony Developer Portal, you may utilize your own private Enterprise applications stores or specialized device management software such as Airwatch, Afaria, and MobileIron. It is the view of Kony in this respect that you can use the MDM (mobile device management) software to implement the Secure Container, but the Add-in framework handles all the applications and data profiling.

3.6 Performing Upgrades

3.6.1 SAP MEAP Server

Upgrading the SAP MEAP server software, is exactly the same as installation procedure. The import of the new transport and check install can be performed over the previous release without any impact on applications, data objects, and configuration. The same is true if you need to "roll back" to a previous release.

> ***Note:*** If you have installed a license for a specific release, then you have to request a new license for the new release that you are upgrading to, for example, from 24.00.03 to 24.01.01. Usually, licenses are generated for any release and still work fine when the software is updated.

3.6.2 Access Gateway

The upgrade process for Access Gateway is the same as installation. You can either remove the previous release or implement the new release in a separate directory and switch any dependencies. Sky recommends that you perform a side by side implementation in development/QA using the same configuration and then when testing is successful, totally replace the existing production gateway(s).

3.6.3 Secure Container

While upgrading the secure container software, you must re-provision the device from scratch, that is the local database data is lost and is re-downloaded as part of provisioning.

3.7 Installing a SAP MEAP Server

This section describes the process to implement a MEAP Server into an SAP environment. This can be either a ERP, CRM, BW system or a stand alone NetWeaver instance. Standard SAP transports practices are used and all objects are implemented using the SAP certified /SKY/ namespace. There is no impact to existing SAP or Customer objects.

> ***Important:*** Separate SAP transports exist for ECC (kernel 620+) and legacy R/3 (all kernel releases prior to 620). If you are unsure, contact Kony for assistance.

3.7.1 Installation Procedure

1. Obtain a valid SAP Add-in license.
2. Download the MEAP Server software from Kony Developer Portal.
3. Unzip and copy the SAP transport files.
4. Check the prerequisites.
5. Shut down any connected Access Gateways and Secure Containers (upgrade only).
6. Import the MEAP transport.
7. Run the check install process in each client where the MEAP Server is used.
8. Import the license.
9. Import the shortcuts transport (optional but recommended).

3.7.2 MEAP Server Prerequisites

Component	Requirement
SAP release	• ECC+ (ECC release, 640+ kernel)
File system storage	• Ensure that there is sufficient free space (20mb) in the `/usr/sap/trans` directory

SAP database storage	• Use SAP transaction DB02 or equivalent tool.
	• Oracle tablespaces:
	  ◦ PSAPUSER1D (data)
	  ◦ PSAPUSER1I (index)
	• SQL Server data file has room to grow

Important: Ensure that there is sufficient free space (200MB+) in the SAP customer (user) database areas to physically create the /sky/ database tables. As a rule of thumb, the index free space should be approximately 1/3 of this, for example, 70mb.

3.7.3 MEAP Server Installation

The SAP certified MEAP Server software is downloaded from Kony Developer Portal and is provided in the form of a zip file containing a standard SAP transport. Typically, two transport files (`K######.SKP` and `R######.SKP`) are provided in the zip, you must copied them to the standard SAP transport library (for example, `/usr/sap/trans/cofiles` and `/usr/sap/trans/data`) in binary mode.

You then use the standard SAP transport system (for example, TP, STMS, 3rd Party software) to import the software into the relevant SAP environment. Normal SAP Basis support should be able to perform this. The prime thing to remember, is that these are 'external' transports and you must import using U12 (unconditional) to avoid problems with replacing original objects.

> ***Important:*** You must copy the transport files to the SAP trans libraries in binary mode to avoid possible corruption. In UNIX, you may need to chmod each file so that the SAP TP utility can read the contents.

3.7.3.1 Recommended Upgrade process

If you upgrade an existing SAP MEAP Server, you are recommended the following process:

- Disable all existing Access Gateway host connections to the SAP system (or shut the gateways down).
- Perform the SAP import (as per the usual process)
- Run the check install process
- Re-enable (or re-start) all Access Gateways.

All releases of the SAP MEAP server are designed to be upward and backward compatible, that is, you need not modify all your existing definitions. The typical "back out" strategy is to re-import the previous SAP add-in release. You should fully test this strategy in a QA environment prior to production.

3.7.3.2 Manual Installation

To import the transport manually, add the transport files to the appropriate directories, then use the SAP tp utility.

From the command line (for example, UNIX)

```
cd /usr/sap/trans/bin
tp addtobuffer SKPK###### <SID>
tp import SKPK###### <SID> client<nnn> U12
```

where replace <SID> with the SAP system ID of the target system, and replace <nnn> with the target client number.

3.7.3.3 Using STMS

To import using the transport management system (STMS), perform the following steps using a user id with sufficient authorization.

- Start transaction STMS.
- Select the target system <SID>.
- Select Extras > Other Requests > Add, and enter the transport request, for example, SKPK######.
- Highlight the new request, and select Request > Import
    - Enter the target client <nnn>
    - Select additional options (expert mode)
    - Select all options.

> ***Note:*** The STMS options slightly vary between SAP releases, but the general process is the same.

3.7.3.4 Troubleshooting

The following section describes some of the problems that could occur with the SAP transport import and the appropriate actions to take.

Insufficient Space in SAP

Transport Errors

Request does not Match Component Version

Insufficient space in SAP:

If the import of the transport fails due to insufficient space it can leave some of the /SKY/ namespace tables in an inconsistent state. This particularly seems to be a problem with Oracle databases. If this happens, the inconsistent tables need to be dropped and then recreated. The process to do this is:

- Log in to SAP as a user with a developer key
- Create a Z* program with the code shown below
- Run the program
- Run transaction SE11
- Reactivate each table that was dropped

Creating a Z* program

```
EXEC SQL.
drop table "/SKY/<TABLE_NAME>"
ENDEXEC.
```

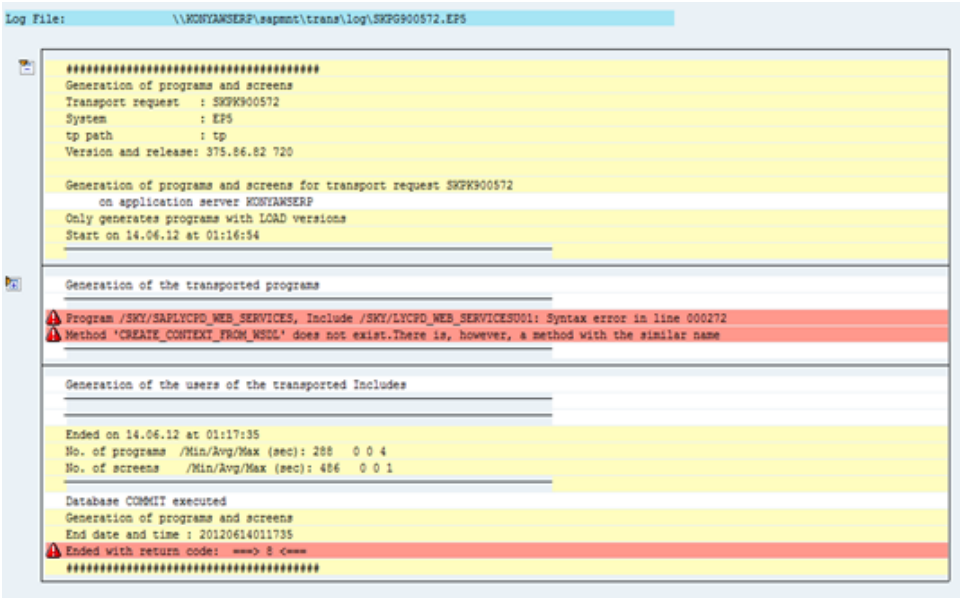
> ***Note:*** You need to replace <TABLE_NAME> with the name of the inconsistent table

Transport errors:

In some cases, there is a functionality that is not supported in your release of SAP. We have been careful to abstract these areas into their own function groups so as not to affect the core system. Typical examples of this are the newer SAP web service, HTTP and XML APIs that we bundle with Kony for SAP that do not exist or changed in your release of SAP.

The following transport errors are acceptable:

Syntax errors in `/SKY/SAPLYCPD_WEB_SERVICES` (as per below) indicate that this functionality is not supported in your SAP system. No action is required on your behalf, it just means that you cannot utilize the web services interfaces in your SAP release.



"Request does not match component version" warning:

As of SAP kernel release 702, you may get a warning generated that the transport was generated using a older version of the TP utility. This is just a warning and does not affect the import of Kony for SAP. Kony packages Kony for SAP using the most forward and backward SAP ECC version so that transport requests are compatible with any system 620+. You may also turn off this check. For more information, search "Request does not match component version" in Google, or refer http://service.sap.com:

1581638 - component checks no longer working in TMS

1577714 - Transport requests in Check requirements phase

3.7.4 Shortcut Transaction Codes

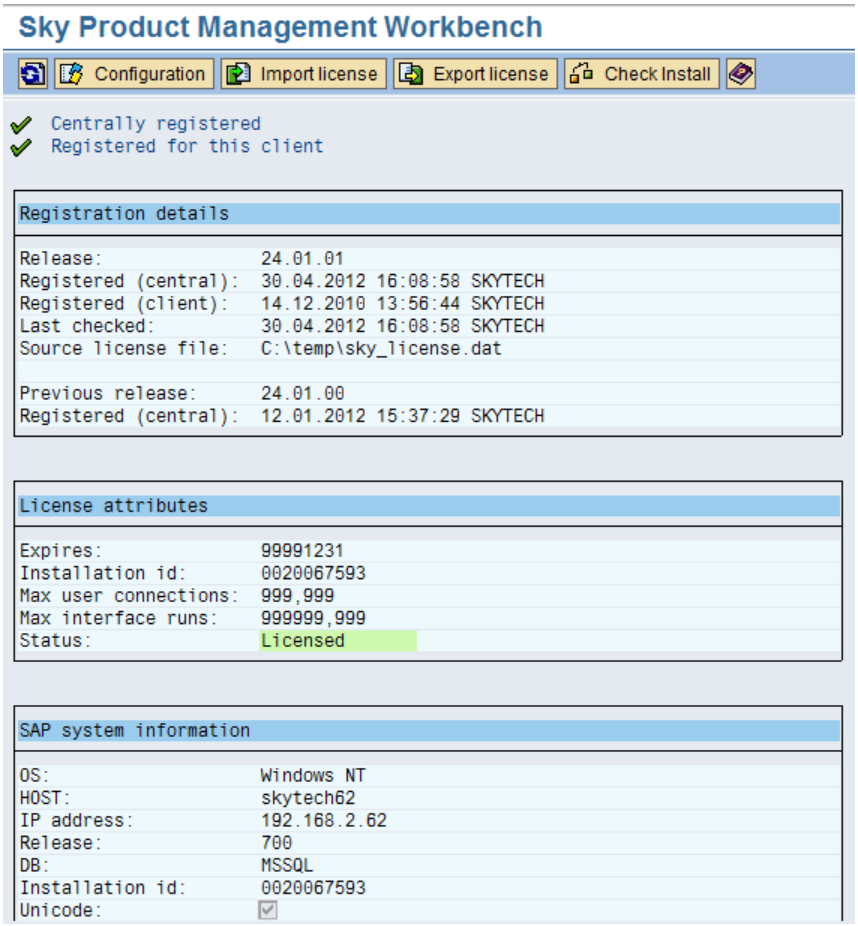
By default, you need to prefix the transactions used to start the SAP add-in functionality with the namespace, for example, `/n/sky/yvtw`. This can be very tedious. Kony provides an optional transport of shortcut transaction codes that are a lot easier to use, for example, YVTW. The reason that we do this as a separate transport is that you need to implement as non-namespace objects and as such there is a remote chance that they may clash with existing customer transaction codes. They are prefixed with "Y" which should only be used for third party utilities by convention, that is customized transactions should start with "Z".

To implement the shortcut transaction codes, you need an additional transport that is downloaded from Kony Developer Portal. The procedure is the same as importing the Sky namespace SAP add-in.

> ***Important:*** Before you implement the shortcut transaction codes, first check for any existing custom transactions that could be potentially overwritten. The check install process highlights any potential conflicts. If in doubt, run the check install again and review the short cut transaction code analysis.

3.7.5 Product Registration and Check Installation

After the SAP Add-in software is imported into the SAP system and before you can be use, you must register it. Sky provides a generic utility to do this. To start the utility, use the transaction /SKY/YECI or just YECI if using short cut transactions. A screen similar to the following appears, showing the product registration details, licensing attributes and SAP system information. If you upgrade, SAP Add-in automatically detects the new release and prompts you to perform a check installation process that you can run at any time.

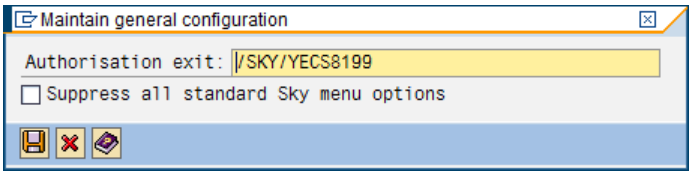


From this screen, you may also import and export Kony for SAP license definitions; and perform a check install. You must run the check install process in every SAP client where you want to use the Kony for SAP software. The process automatically verifies all the software components and sets up the run time environment. Once you do this, the Kony for SAP software is ready for use. If there is no imported license definition, the Kony for SAP software defaults to "Free" mode, meaning that the software runs in a restricted state as per the terms of the "Free software download" license agreement.

There are also some configuration settings that you may use to control authorizations and on-line help.

3.7.5.1 Product Management Console Configuration

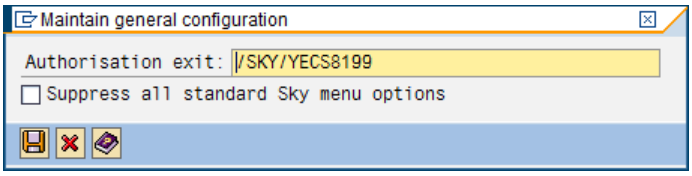
There are a number of configuration options that you can customize in the product management console. These are mainly to do with global authorizations.



Option	Description
Authorization exit	This is a custom ABAP program that you may install to further vet authorization requests.
Suppress menu options	If you check this, then all the standard SAP Add-in menu options are suppressed. This is an easy way to help close up a system in terms of navigating between SAP Add-in components.

3.7.5.2 Product Management Console Configuration

There are a number of configuration options that you can customize in the product management console. These are mainly to do with global authorizations.



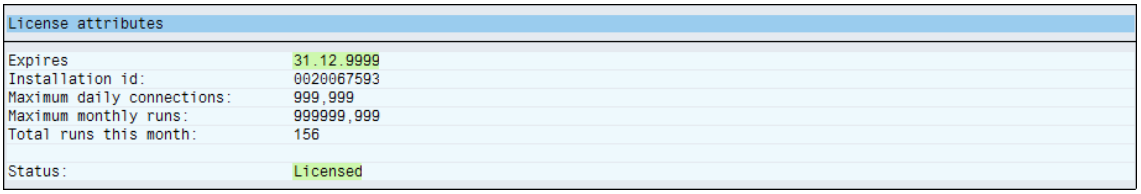
Option	Description
Authorisation exit	This is a custom ABAP program that may be installed to further get authorization requests.
Suppress Sky menu options	If you check this, then all the standard menu options are suppressed. This is an easy way to help close up a system in terms of navigating between MEAP components.

3.7.6 Importing a License

Before you may use the MEAP Server in a production sense, you must have obtained a valid license file from Kony for SAP for your installation. Kony for SAP may not operate optimally unless a valid license key is imported, refer Obtaining a License. If you have installed a "free" version of the Kony for SAP software, you are allowed a notional number of free device connections and interface runs before restrictions start to apply. Refer free licensing for more information. As of release 24.01 of the SAP add-in, the license checking changed to support the "free" versions of the software and implement new maximum unique devices per day and maximum interface runs per month checks. Older license definitions prior to 24.01 still work, but Kony for SAP encourages customers to check their license attributes carefully when they upgrade to 24.01+ or import a new Kony for SAP license definition (see below).

3.7.6.1 License Attributes and Checking

The license attributes section in the software management workbench list (YECI) highlights the current restrictions and usage:



License attributes	
Expires	31.12.9999
Installation id:	0020067593
Maximum daily connections:	999,999
Maximum monthly runs:	999999,999
Total runs this month:	156
Status:	Licensed

Depending on your Service Level and Maintenance Agreement with Kony for SAP, an expiry date, maximum device connections per day and maximum interface runs per month apply. The current usage is also displayed and is highlighted in red if exceeded. If you have reached the maximum number of unique device connections per day, then any additional connections are refused. If you exceed the maximum number of interface runs per month, then any additional ECS process runs start in a failed state, and you need to manually re-start.

3.7.6.2 Importing a New Sky License Definition

To import a license definition, use transaction /SKY/YECI or just YECI if using the shortcut transactions, click **Import License** button on the application toolbar. A confirmation prompt appears, after which you are prompted for the location of the license file that Kony for SAP provided you with. Select the file and click the **Save** button. The license file is now installed.

> ***Note:*** If the product license status is still not valid, check your license definition. The installation ID must match your SAP installation number and the expiry date must be valid. Check your license attributes carefully.

3.7.6.3 Exporting a License Definition

You may export your license definition at any stage to a file on your desktop. You may choose to do this as a backup or to migrate the license file to your other SAP systems. To perform an export, click **Export license** on the application tool bar.

3.7.7 Creating a SAP CPIC User and Authorization Profile

The SAP RFC client requires a CPIC (interface) user to configure in SAP and specified in its configuration file. Before the user is created, you may need to create an appropriate security profile for Kony for SAP usage. This enables Kony for SAP to execute the SAP add-in functions. Note that a CPIC (interface) user cannot log directly onto SAP; the external interface programs may only use using the RFC protocol. Kony for SAP does not provide any security profile or authorization objects with the installation because these configurations and policies differ considerably from customer to customer.

3.7.7.1 Example Authorization Profile

Authorization Object	Attribute	Value
S_RFC: Authorization Check for RFC Access	ACTVT: Activity	16: Execute
	RFC_NAME: Name of RFC to be protected	/SKY/Y* SYST
	RFC_TYPE: Type of RFC object to be protect	FUGR
S_TCODE: Transaction Code Check at Transaction Start	TCD: Transaction Code	SESSION_MANAGER
S_BTCH_NAM: Background Processing: Background User Name	BTCUNAME: Background User Name for Authorization Check	*

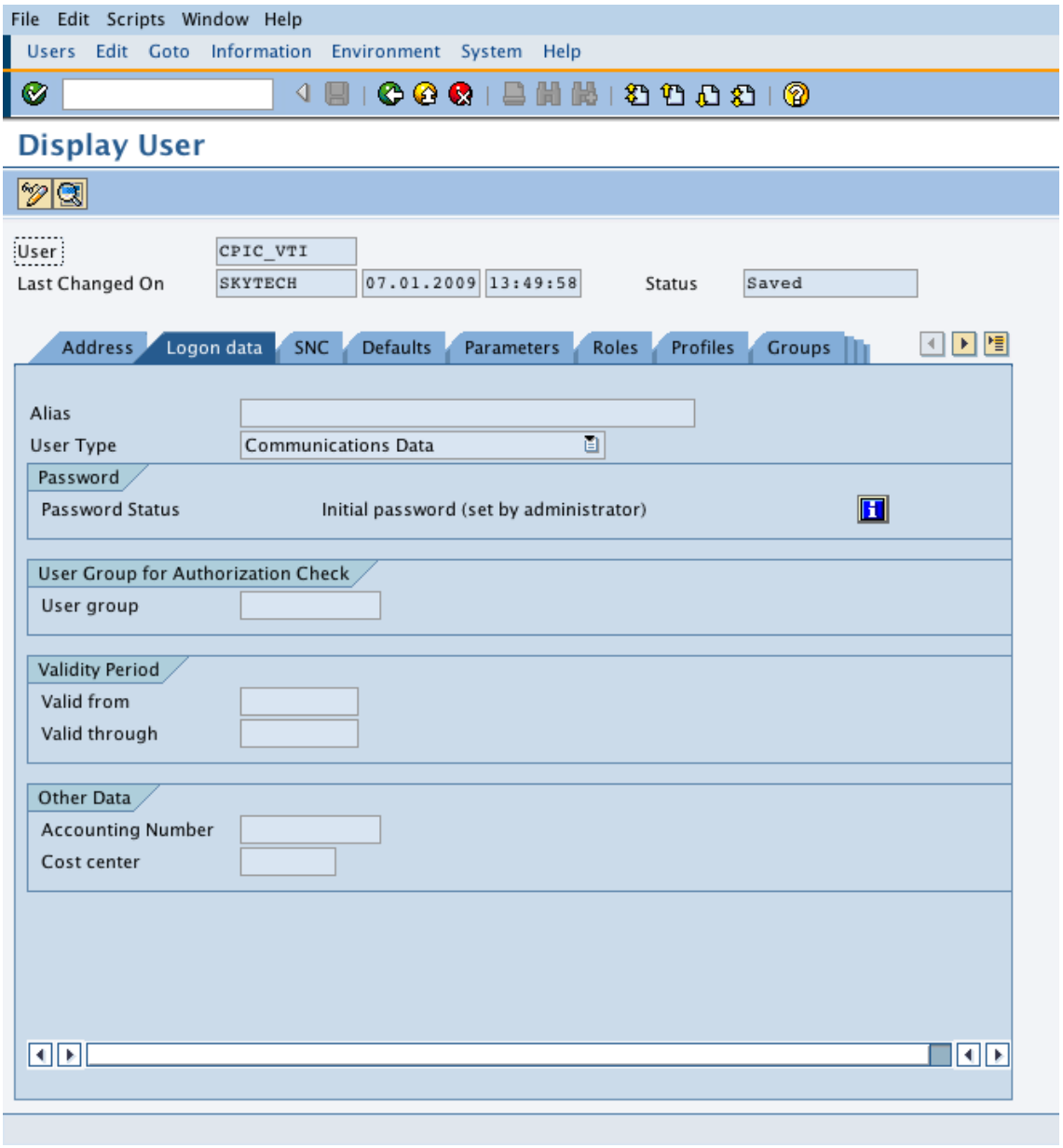
Authorization Object	Attribute	Value
S_BTCH_JOB: Background Processing: Operations on Background Jobs	JOBACTION: Job operations	• DELE: Delete Background jobs • LIST: Display Spool Requests Created by Job • PLAN: Copy or Repeat Jobs • PROT: Display Job Processing Log • RELE: Release Jobs (Released Automatically When Scheduled) • SHOW: Display Job Queue
	JOBGROUP: Summary of jobs for a group	*
S_DATASET: Authorization for file access	ACTVT: Activity	• 06: Delete • 33: Read • 34: Write • A6: Read with filter • A7: Write with filter
	FILENAME: Physical file name	*

Authorization Object	Attribute	Value
	PROGRAM: Program Name with Search Help	*
S_GUI: Authorization for GUI activities	ACTVT: Activity	• 02: Change • 04: Print, Edit Messages • 60: Import • 61: Export
S_LOG_COM: Authorization to execute logical operating system commands	COMMAND: Logical command name	*
	HOST: R/3 System, Name of Application server	*
	OPSYSTEM: R/3 System, Operating System of application server	*

Go back to the top.

3.7.7.2 Example CPIC User

Use SU01 in SAP to create a CPIC user for RFC connections or the Gateway to use. Allocate the security profile (see above) to manage access using this user ID. A CPIC user cannot logon directly to SAP and indicates an 'interface' category of access. CPIC users are not required to periodically change their passwords.



Go back to the top.

3.7.8 MEAP Server Configuration

Once you have implemented the MEAP Server and run the automated 'check install' procedure, it is now ready for use. There are various configuration options that are defaulted by the check install process that in most cases do not need to be changed. If required, you may review these and change using various configuration utilities. Review:

- MEAP Server Management
- Container Management Console Configuration
- Product Management Console Configuration

3.7.9 Making Local SAP Repairs

In exceptional cases, you may need a repair to perform locally to the SAP MEAP Server directly in a Customer system. To be able to do this, perform the following steps:

- Add the SKY Namespace Key
- Make the Code Repair
- Transport the Code Repair

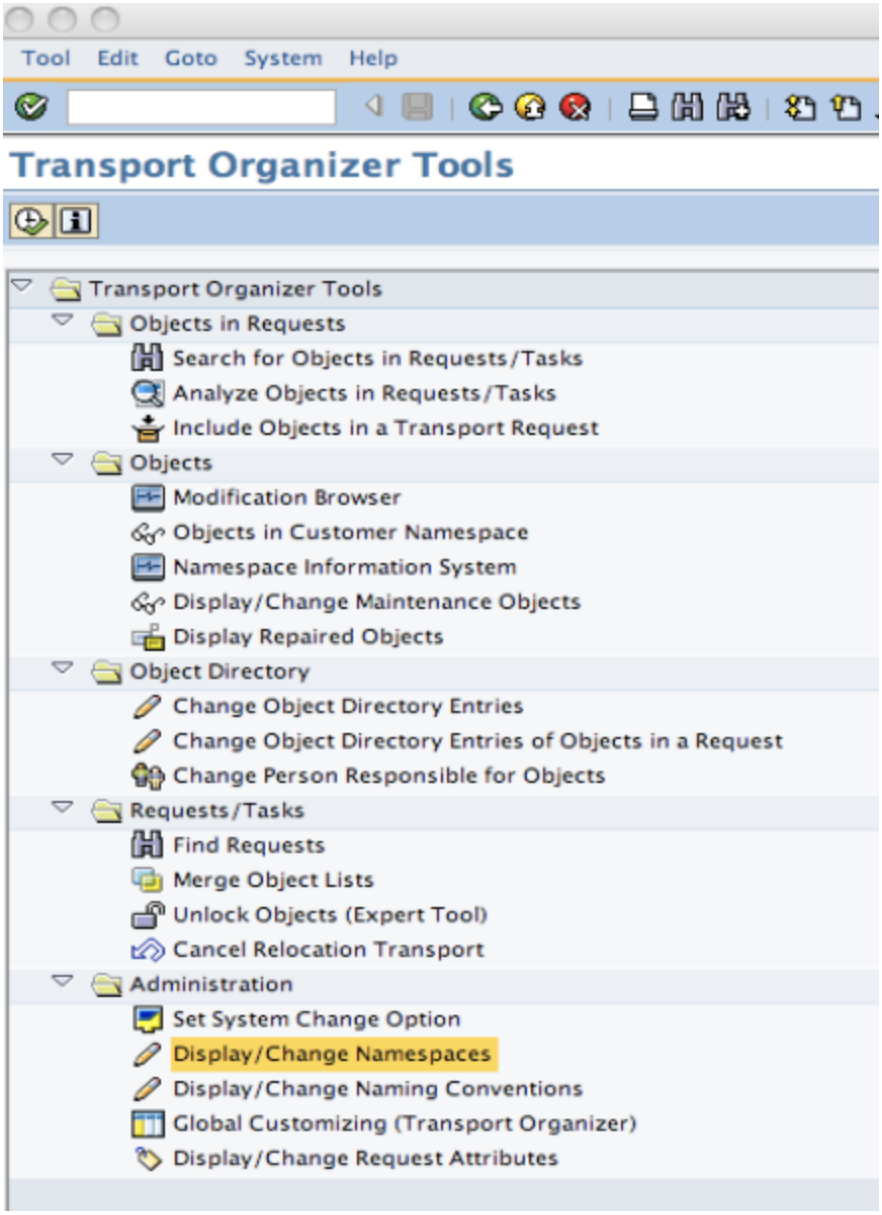
> *Important:* This is only allowed for special circumstances that Kony support confirms and authorizes.

3.7.9.1 Adding the SKY Namespace Key

To add the Sky namespace key, perform the following steps:

- Run transaction SE01
- From the menu, select **Goto** > **Transport Organizer Tools**

- Select the option **Administration** > **Display/Change Namespaces**

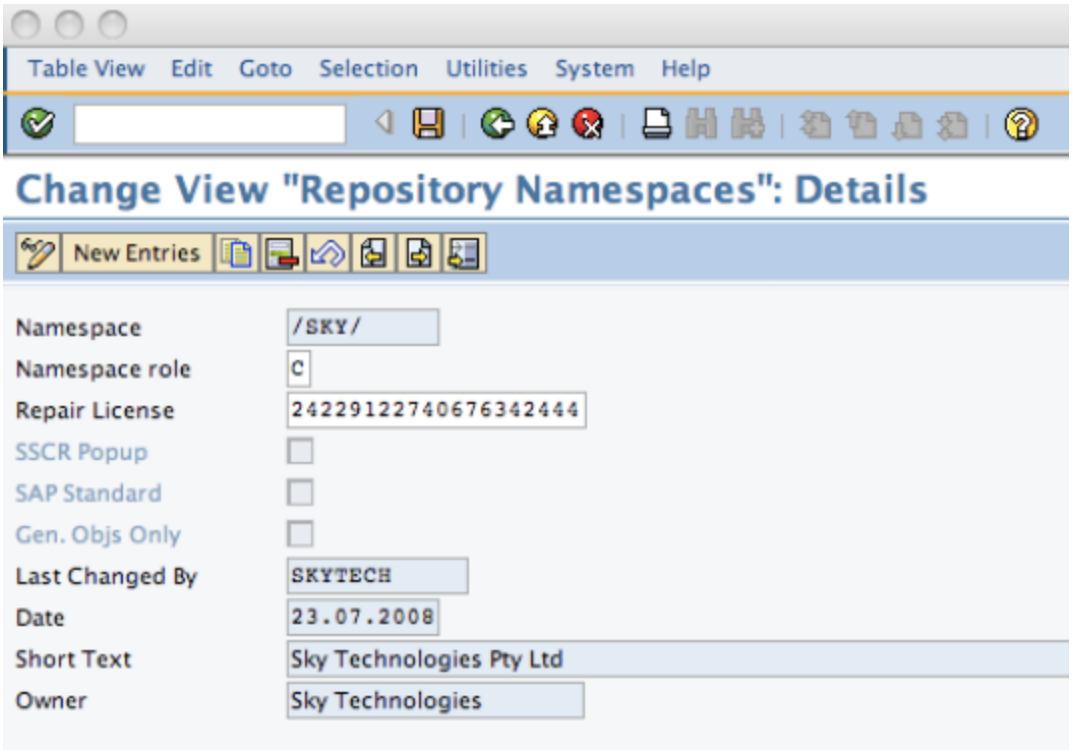


- Switch into Change mode

> ***Important:*** In a client system, normally there should not be an entry /SKY/. That is what you are about to add.

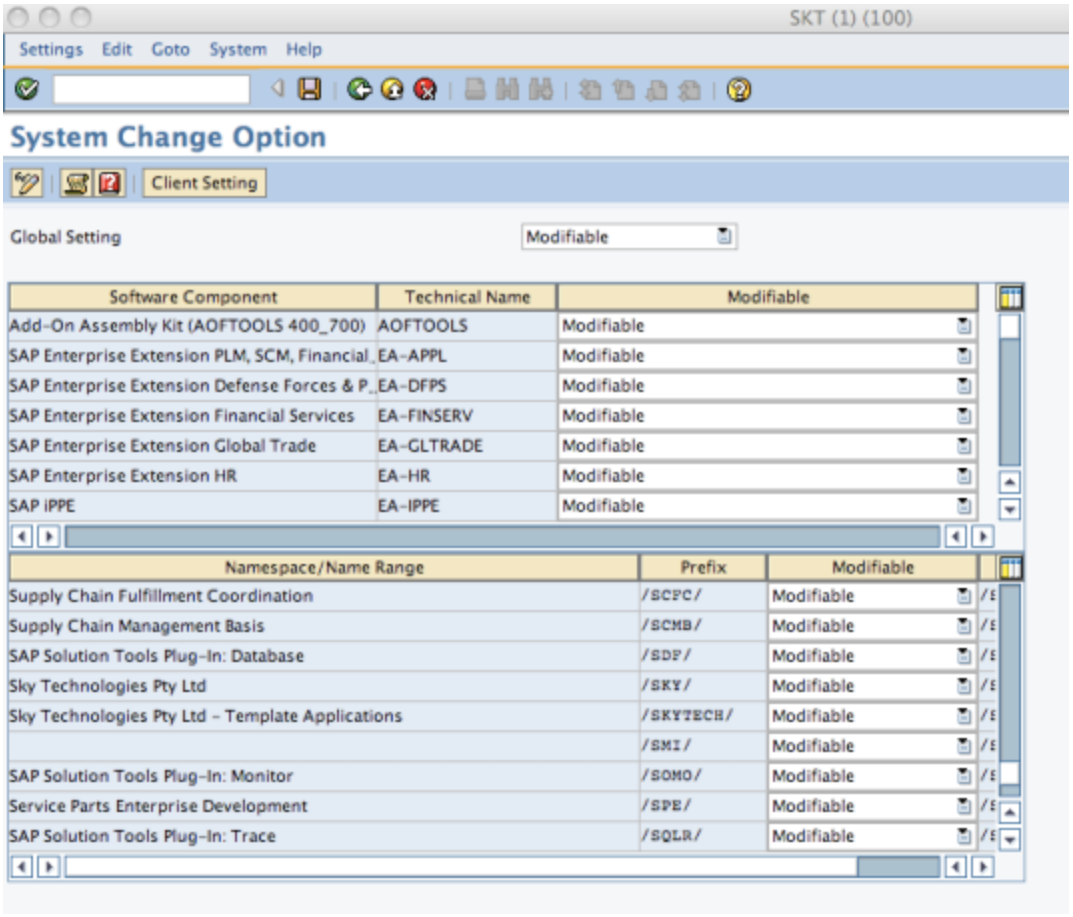
- Create a new entry with the following details:

Field	Value
Namespace	/SKY/
Namespace Role	C (Consumer)
Repair License	24229122740676342444
Short Text	Sky Technologies



- Navigate back to **Transport Organizer Tools**
- Select **Administration** > **Set System Change Option**

- In the bottom **Namespace** pane, locate the **/SKY/** namespace. Ensure the **Modifiable** flag is configured to `Modifiable`.



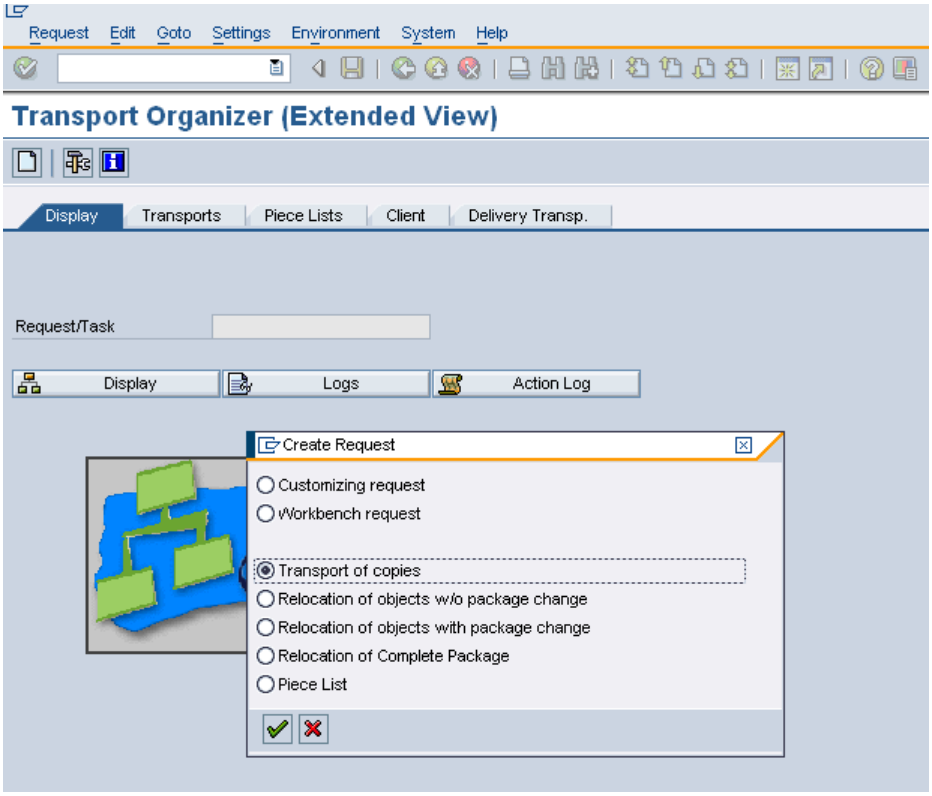
3.7.9.2 Make the Code Repair

Once the **/SKY/** namespace is modifiable, change the ABAP code as instructed by Kony support. This invokes the SAP modification assistant that you may turn off using **edit** > **modification operations** > **switch off assistant**. A **local** transport is created that you cannot transport to other SAP systems. To do this, create a transport of copies (see the next section).

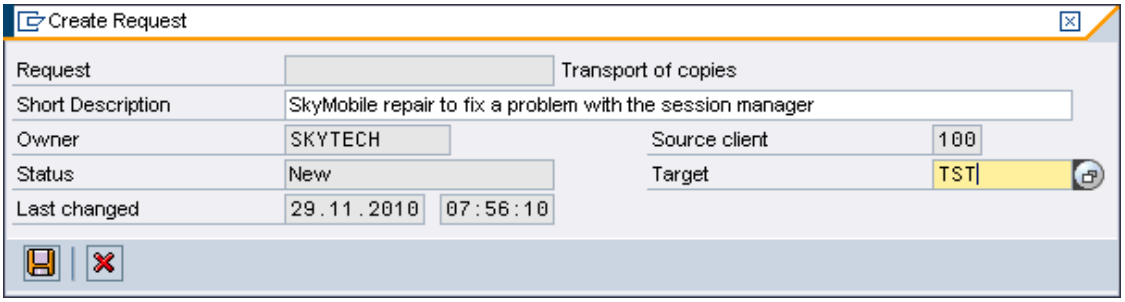
3.7.9.3 Transport the Code Repair

Any repairs to /SKY/ results in a local transport. To transport the repair to other SAP systems:

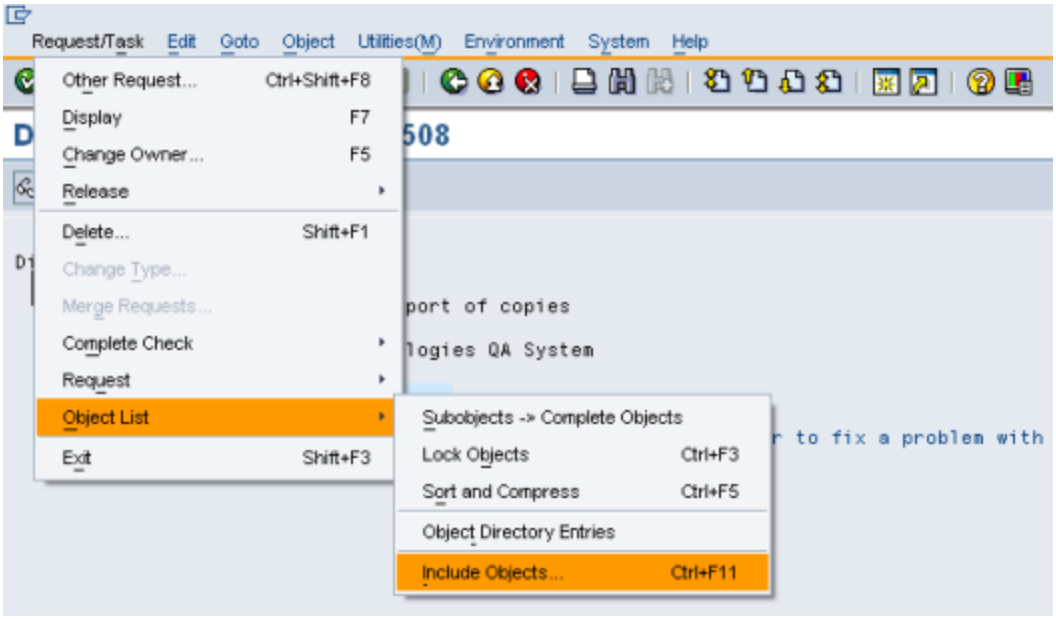
- First you must release the local transport, noting the name
- Create a transport of copies using SE01. Click the create icon and select **Transport of copies** option.



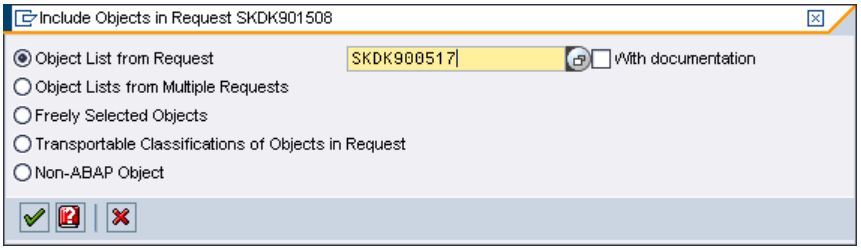
- In the pop-up, give a meaningful description and select the initial target SAP environment, usually the test/QA consolidation system.



- A transport is created and you are directed to the SE10 display request screen. Position your cursor on the transport name and select **Include Objects**:



- Select the **Object list from request** option and type in the name of the local repair that was released previously. This copies in the contents of the local repair transport into the new transport of copies.



- Once completed, you can treat the transport of copies just like any other transport; released and imported into other SAP systems.

3.8 Installing an Access Gateway

Overview

The Access Gateway option is required for all Secure Containers running on remote devices. It is essentially an implementation of a Secure Container on either a UNIX or Windows server, but with just a gateway port configured. Make sure that the operating system is appropriate to support multiple TCP/IP port connections, for example, Windows 2008 server edition. Some operating system versions are designed only to support a limited number of connections, for example, Windows 7 professional. The Access Gateway is usually implemented behind the firewall in the DMZ with one external port exposed and a host connection pool to a back end system. It essentially acts as a single point of entry into the WAN/LAN. All communication between the Access Gateway and remote Secure Containers on the device is through TCP/IP sockets or HTTP(s) and you should configure to compress and encrypt. Multiple gateways are supported with automatic fail over capability. For more information on this, refer to the gateway switchover section in the management guide. Usually the Access Gateway does not require a lot of hardware resources, but if the application is transferring large quantities of data object and/or binary data, Kony recommends that you properly size the target server(s) to enable optimum performance and allow for growth.

Each Access Gateway manages a shared RFC connection pool with configuration for the minimum and maximum number of connections to manage. A SAP user ID and password is also provided to establish the RFC connections. This SAP user is set up as either a CPIC or service type user and must have SAP access to the /SKY/YVTI function group as a minimum.

Host interfaces

You need to configure at lease one host interface definition to configure to define how the gateway implements and manages connections with the back end host system. In the case of SAP, this is a standard RFC connection pool. Click here for an example.

> ***Note:*** You may configure multiple back end host connections linked to corresponding gateway ports. The Secure Container can only connect to a single gateway port.

3.8.1 Access Gateway Ports

Configure the XmlGatewayPort section in the configuration file. You may specify multiple Access Gateway ports, each with a unique id and port. Click here for full documentation of all the available options. Each gateway port definition corresponds directly with a single host interface section through the HOSTINTERFACE option. Click here for a typical implementation.

3.8.2 Secure Container Configuration

Once you start the Access Gateway, its gateway ports are active and it is connected to the back end host system, you may configure Secure Containers to connect to it through their "SERVER.HOSTINTERFACE" configuration section. Usually this is defined centrally in the MEAP Server Profiler and automatically pushed out to the device at provisioning time. Click here for a typical implementation.

3.8.2.1 Installation Process

In summary, the process to implement a Access Gateway is as follows:

- Check that the server you have in mind has enough capacity
- Set up a SAP CPIC or service type user with the appropriate authorizations for the gateways initial SAP RFC logon. You require a SAP service type user if you intend to perform dynamic user switching in SAP (recommended).
- Make sure the nominated server has a JVM (Java Virtual Machine) or JRE (Java Runtime Environment) installed, See the recommendations under Windows Server and UNIX installations.
- Install a Secure Container as per the Windows or UNIX instructions. The windows installer automatically checks and generates a default configuration for you.
- Manually edit and install the `skymobile.cfg` configuration file (if required):

- identity management (optional)
- network encryption (optional)
- back end host interface(s)
- gateway port(s).

Important: If you intend to use dynamic SAP user switching for interfacing, then you must configure the Access Gateway SAP user as a "service" type user.

Note: A sample gateway profile is automatically implemented as part of the MEAP Server check install process with all the parameters populated. You may easily export this to a configuration file format for use in the install. The Windows installer automatically generates a default configuration, but you should review for optimal performance and security.

3.8.3 Windows Gateway Installer

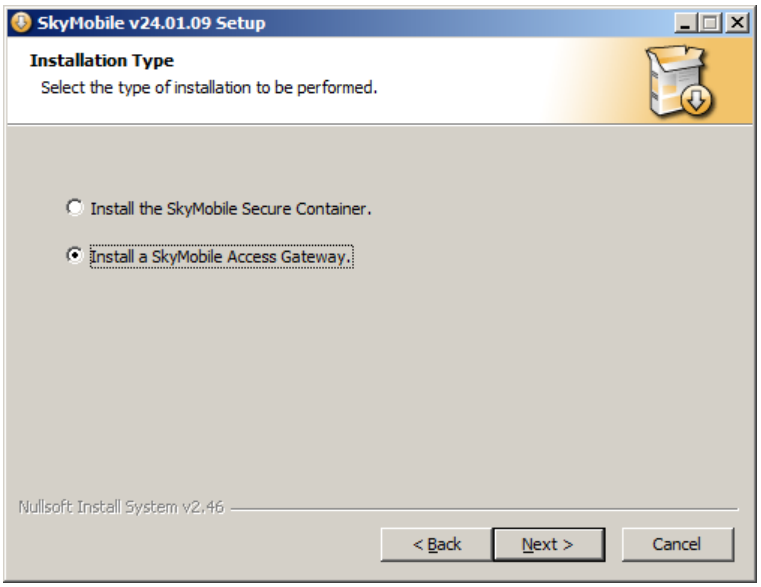
As of release 24.01.09+, Kony for SAP provides an automated Windows installer that implements and configures an Access Gateway. This is the easiest way to get started. The process is as follows:

1. You need to install a Java JVM (Java Virtual Machine) or JRE (Java Runtime Environment) onto the server. You can get this free of charge from http://www.java.com/en/download/manual.jsp#win make sure you download the correct 64 or 32 bit version, depending on the Windows version. The installer tries to check this. Most Windows servers are 64bit these days.
2. Download the Windows installer from Kony Developer Portal.
3. Run the installer on the machine where you want to host the gateway (this should be a Windows server edition OS).
4. Select the gateway option from the prompt and fill out the target SAP system attributes (these are checked).
5. Once complete, check that the service is running and is connected to SAP (see post installation checks).

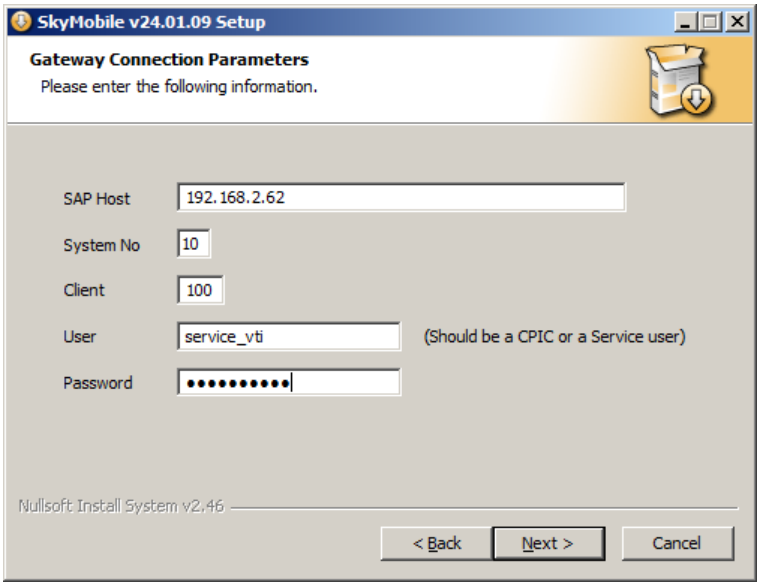
3.8.3.1 Installation Process

The following screens appear:

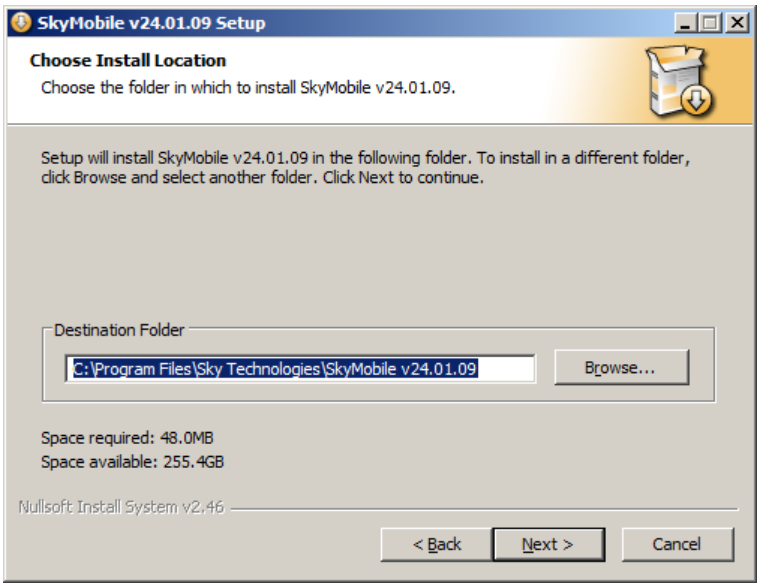
1. In the pop-up that appears, select **Install a Sky Access Gateway** option.



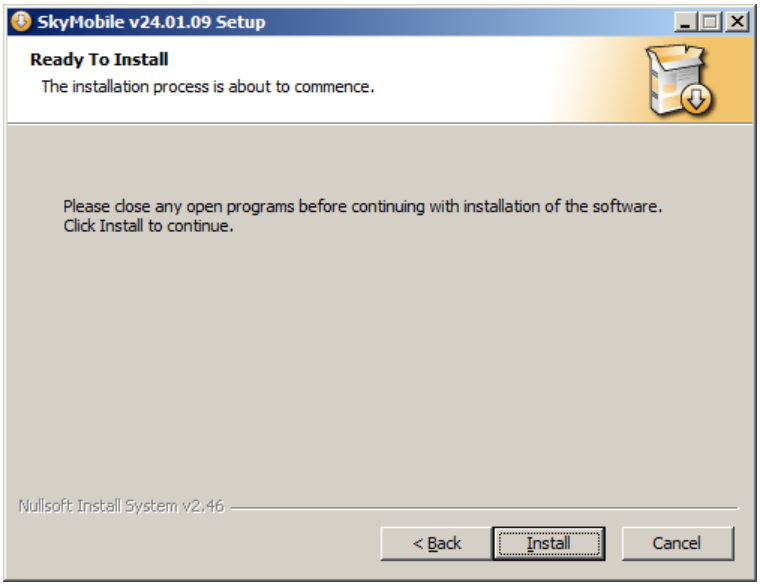
2. Fill out the target SAP system attributes. A check is performed to make sure these are valid. Checks are also performed to ensure that there is a Java JVM installed and 32/64 bit compatibility.



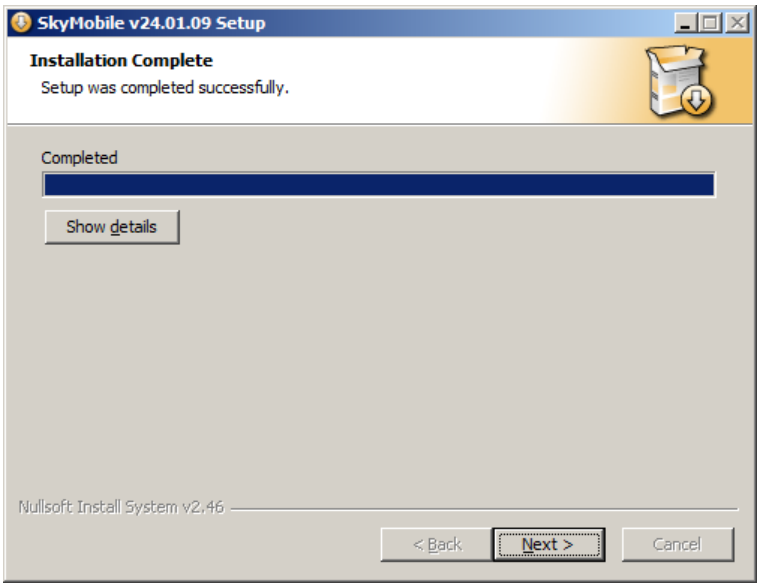
3. Once the SAP attributes are validated, you are prompted for the target directory.



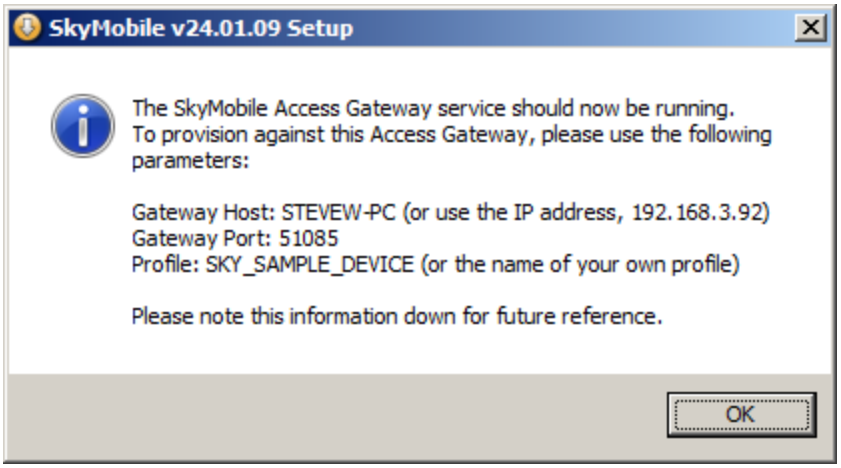
4. You are then prompted to install the software. You should close any open programs that could possibly conflict Click **Install**.



5. The installation of the software now commences. This takes not more than 10-30 seconds.



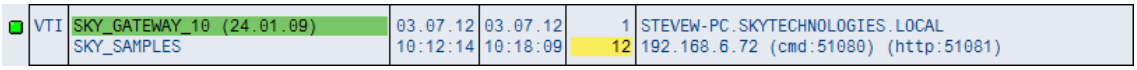
6. This step automatically creates and starts an Access Gateway service called "SkyMobile_Access_Gateway" (SKY_GATEWAY_{SAP sysno}).
   A confirmation pop-up appears that lists the Gateway host and port details that you require for the Secure Containers to connect to:



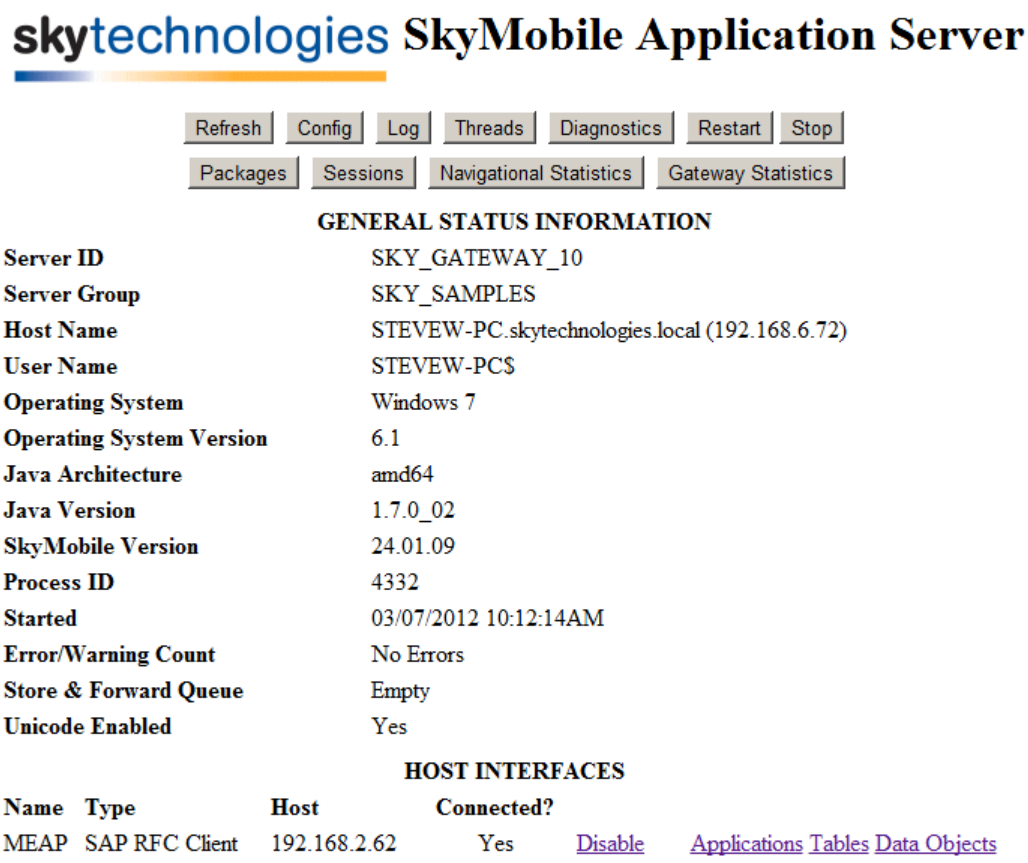
3.8.3.2 Post Installation Checks

Check your Windows services (control panel - Administrative Tools - Services) to make sure that the service is started. If it has not, you need to check the gateway log for possible reasons. To do this, navigate to the installation directory that you chose, for example, `{directory}\SkyMobile v24.01.09\vti\log`. The log is a text file that lists all start up and operating messages.

You should also check that the gateway is connected to the SAP system. To do this, log onto SAP and run the /n/sky/yecj or YECJ transaction code. Execute the selection screen and your gateway should appear in the list (as below). If it is not, then a SAP host connection is not established and you have to check the SAP attributes and userID/password.



Start the gateways web status page to view its host connection, configuration, log. You can do this from any browser using: {gateway ip address}:5{SAP sysno}81, for example, localhost:51081. The following details appear:



From here, you can check whether the host interface is connected, view the log information, and configuration.

3.8.4 Example Access Gateway Configuration

The following is a sample configuration file for an Access Gateway. Obviously, you need to review and change some settings depending on the environment. The more obvious mandatory changes are highlighted. You should be able to cut and paste this configuration into `skymobile.cfg` after initial installation and use it as a basis. This configuration implements a gateway port of 10000 and standard control ports: 15070 (command), 15071 (http) and 15076 (relay).

> ***Note:*** This example configuration does not implement an identity management service. For details on how to do this, refer to the identity management security section in the management guide. You should also review the gateway port encryption options and how to implement network encryption if the port is exposed to external access.

A sample gateway profile is automatically implemented as part of the MEAP Server check install process with all the parameters populated. You may easily export this to a configuration file format for use in the install.

Example Access Gateway Configuration

```
SERVER.APPLICATIONCACHE.APPLICATIONREFRESHINTERVAL = 300
SERVER.APPLICATIONCACHE.FUNCTIONCACHESIZE =
SERVER.APPLICATIONCACHE.MAINTAINHOSTCONNECTIVITY = false
SERVER.APPLICATIONCACHE.REFRESHBYFUNCTION = true
SERVER.BACKUPRESTORE.AUTOBACKUPTIMES =
SERVER.BACKUPRESTORE.BACKUPDIRECTORY =
SERVER.BACKUPRESTORE.BACKUPITEMS =
SERVER.BACKUPRESTORE.COMPRESSIONALGORITHM = GZIP
SERVER.BACKUPRESTORE.USECOMPRESSION = true
SERVER.BINARYFILES.DEFAULTBINARYGROUP =
SERVER.BINARYFILES.DOWNLOADDIRECTORY = ./downloads
SERVER.BINARYFILES.ERRORRETRYINTERVAL =
SERVER.BINARYFILES.FILELOCATION:SOUNDS = ./sounds
SERVER.BINARYFILES.HEXMODEENCODING =
SERVER.BINARYFILES.MAINTAINHOSTCONNECTIVITY = false
```

```
SERVER.BINARYFILES.MEMORYRETENTIONPERIOD =
SERVER.BINARYFILES.PACKAGEREFRESHINTERVAL = 300

SERVER.BINARYFILES.UPDATEHOSTMD5HASHES =
SERVER.COMMANDPORT:PORT15070.PORT = 15070
SERVER.GENERAL.AUTORESTART = false
SERVER.GENERAL.DATEFORMAT = MM/DD/YYYY
SERVER.GENERAL.RESTRICTTOHOST =
SERVER.GENERAL.RESTRICTTOUSER =
SERVER.GENERAL.SERVERGROUP = gateway server group
SERVER.GENERAL.SERVERID = gateway server id
SERVER.GENERAL.TIMEFORMAT = HH:MM:SSAA
SERVER.GENERAL.UNICODEENABLED = true

SERVER.HEARTBEAT.HEARTBEATCONNECTEDINTERVAL = 60
SERVER.HEARTBEAT.HEARTBEATHOSTADDRESS =
SERVER.HEARTBEAT.HEARTBEATHOSTNAME =
SERVER.HEARTBEAT.HEARTBEATINTERVAL = 30
SERVER.HEARTBEAT.HEARTBEATLOSSSEVERITY = I
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:APPLICATION =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:BINARYFILES =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:HEARTBEAT =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:LOCALDATABASE =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:PACKAGES = true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:SESSIONMANAGER
= true
SERVER.HOSTINTERFACE:SAP.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAP.JNILOGGING = false
SERVER.HOSTINTERFACE:SAP.MAINTAINHOSTCONNECTIVITY = true
```

```
SERVER.HOSTINTERFACE:SAP.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAP.MINIMUMNUMBERHANDLERS = 2
SERVER.HOSTINTERFACE:SAP.SAPCLIENT = client number
SERVER.HOSTINTERFACE:SAP.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAP.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAP.SAPHOST = SAP Host IP Address
SERVER.HOSTINTERFACE:SAP.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAP.SAPLBGROUP =
SERVER.HOSTINTERFACE:SAP.SAPLBHOST =
SERVER.HOSTINTERFACE:SAP.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SAP.SAPPASSWORD = SAP CPIC password
SERVER.HOSTINTERFACE:SAP.SAPSYSTEM = SAP system number
SERVER.HOSTINTERFACE:SAP.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAP.SAPUSELB = 0
SERVER.HOSTINTERFACE:SAP.SAPUSER = SAP CPIC Userid
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWCONFIGCHANGESFROMWEBPAGE =
true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWFILEENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWFILEUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWLDBUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.AUTHORISATIONPASSWORD =
SERVER.HTTPCONTROLPORT:PORT15071.AUTHORISATIONUSERNAME =
SERVER.HTTPCONTROLPORT:PORT15071.HTMLBUTTONLOCATION =
SERVER.HTTPCONTROLPORT:PORT15071.HTMLPAGESIZE =
SERVER.HTTPCONTROLPORT:PORT15071.PORT = 15071
SERVER.HTTPCONTROLPORT:PORT15071.USEHTMLCOMPRESSION =
SERVER.JAVA.JAVAARGUMENTS =
SERVER.JAVA.JAVACOMMAND =
SERVER.LOCALDATABASE.ERRORRETRYINTERVAL = 300
SERVER.LOCALDATABASE.MAINTAINHOSTCONNECTIVITY = false
SERVER.LOCALDATABASE.MAXIMUMNUMBERUPDATETHREADS = 5
SERVER.LOCALDATABASE.MINIMUMNUMBERUPDATETHREADS = 1
```

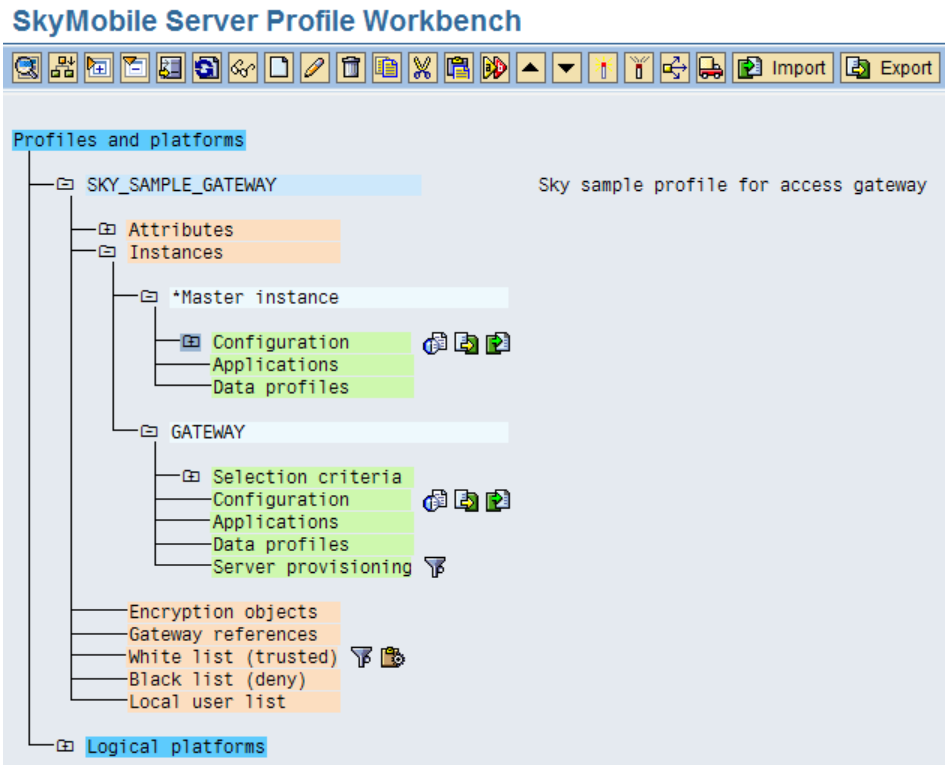
```
SERVER.LOCALDATABASE.TRANSFERBUFFERSIZE = 500000
SERVER.LOG.EXCLUDELOGMESSAGESEVERITIES =
SERVER.LOG.EXTRALOGGINGOPTIONS =
SERVER.LOG.LOGFILEDIRECTORY = ./log
SERVER.LOG.LOGSTRATEGY = P
SERVER.LOG.MAXIMUMNUMBERLOGFILES = 5
SERVER.LOG.SHOWMILLISECONDS = false
SERVER.LOG.STOREDPROCDUMP = false
SERVER.LOG.STOREDPROCLOGGING = false
SERVER.MONITOR.CLEARRFCTRACEFILES = true
SERVER.MONITOR.GARBAGECOLLECT = true
SERVER.MONITOR.LOGMEMORYUSAGE = true
SERVER.MONITOR.LOGTHREADCOUNT = true
SERVER.MONITOR.LOGTHREADNAMES = false
SERVER.MONITOR.MONITORINTERVAL = 60
SERVER.MONITOR.RFCTRACEEXPIRYDAYS = 3
SERVER.MONITOR.USEMONITOR = true
SERVER.NETWORK.ACCEPTTIMEOUT =
SERVER.NETWORK.BINDTIMEOUT =
SERVER.NETWORK.INVALIDIPADDRESSES =

SERVER.NETWORK.LISTENBACKLOG =
SERVER.NETWORK.NETWORKERRORSAMPLESIZE =
SERVER.NETWORK.NETWORKERRORTHRESHOLD =
SERVER.NETWORK.READTIMEOUT =
SERVER.NETWORK.USENATIVESOCKETS =
SERVER.NETWORK.VALIDIPADDRESSES =
SERVER.RELAYCONNECTIONS.ALLOWCONFIGCHANGESFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWFILEENQUIRYFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWFILEUPDATESFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWLDBUPDATESFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.AUTHORISATIONPASSWORD =
```

```
SERVER.RELAYCONNECTIONS.AUTHORISATIONUSERNAME =
SERVER.RELAYCONNECTIONS.HTMLBUTTONLOCATION = Bottom
SERVER.RELAYCONNECTIONS.HTMLPAGESIZE = 15
SERVER.RELAYCONNECTIONS.RECONNECTSECONDS =
SERVER.RELAYCONNECTIONS.RELAYHOST =
SERVER.RELAYCONNECTIONS.RELAYPORT = 5063
SERVER.RELAYCONNECTIONS.SHOWLDBROWCOUNTS = true
SERVER.RELAYCONNECTIONS.SUPPRESSLOGO =
SERVER.RELAYCONNECTIONS.USEHTMLCOMPRESSION =
SERVER.RELAYPORT:PORT15076.IDLETIMEOUT =
SERVER.RELAYPORT:PORT15076.PINGINTERVAL =
SERVER.RELAYPORT:PORT15076.PORT = 15076
SERVER.RELAYPORT:PORT15076.RELAYREQUESTTIMEOUT =
SERVER.STORAGE.STORAGECOMPRESSONSHUTDOWN = true
SERVER.STORAGE.STORAGECONSISTENCYCHECK = true
SERVER.STORAGE.STORAGEDATABASEFILENAME = ./db/vti.db
SERVER.STORAGE.STORAGEINDEXDEFRAGONSHUTDOWN = true
SERVER.STORAGE.STORAGEINDEXREBUILD = false
SERVER.STORAGE.STORAGEINTERNALIZESTRINGS = true
SERVER.STORAGE.STORAGEINTERNALLOGGING = false
SERVER.STORAGE.STORAGELOCKTIMEOUT = 10000
SERVER.STORAGE.STORAGEPAGECACHESIZE =
SERVER.STORAGE.STORAGETYPE = SkyDB
SERVER.XMLGATEWAYPORT:PORT10000.COMPRESSIONALGORITHM = GZIP
SERVER.XMLGATEWAYPORT:PORT10000.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.XMLGATEWAYPORT:PORT10000.ENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORT10000.ENCRYPTIONKEYFILE =
SERVER.XMLGATEWAYPORT:PORT10000.ENCRYPTIONKEYHEXENCODED =
SERVER.XMLGATEWAYPORT:PORT10000.HOSTINTERFACE = SAP
SERVER.XMLGATEWAYPORT:PORT10000.PINGCHECKIDLESECONDS =
SERVER.XMLGATEWAYPORT:PORT10000.PINGCHECKTIMEOUT =
SERVER.XMLGATEWAYPORT:PORT10000.PORT = 10000
SERVER.XMLGATEWAYPORT:PORT10000.USECOMPRESSION = true
```

```
SERVER.XMLGATEWAYPORT:PORT10000.USEENCRYPTION = false
SERVER.XMLGATEWAYPORT:PORT10000.XMLDEBUG = true
```

3.8.5 Sample Access Gateway Profile

The MEAP Server check installation process automatically creates a sample profile containing configuration for a access gateway. You can export this configuration and use as the basis for a Access Gateway installation. To do this, invoke the server profile workbench and select the SKY_SAMPLE_GATEWAY profile. The following appears:



Position your cursor on the "*Master instance" configuration node and click the export icon in the application tool bar. You are prompted for the target file name on your desktop and then the export format to use. Select the defaults and then a gateway configuration file is created.

3.8.6 Sizing an Access Gateway

You can use the following basic table as a guideline for minimum memory requirements for an Access Gateway. The gateway is multi-threaded and is usually only constrained by available memory if large sets of data (database and binary) is de-compressed. You can control the amount of data handled for a device at any one time using the "transfer buffer" configuration. Large binaries associated with data objects should use the "load asynchronously" data object option. Once implemented, you can use the gateway statistics page to highlight bottlenecks for both Container connections and SAP host processing.

Total devices	Concurrent Devices	Memory (MB)
10	3	150
50	5	200
100	10	500
500	15	1000
1000	30	2000

The amount of disk space is constant at 100mb and assumes a minimum processor speed of 1.66GHz+. Most organizations have a standard SOE for servers and in general, Kony recommendation is a “low end server” typically with dual 1.66GHz processors, 2-4 GB of memory and 80GB of disk. This does seem like overkill given the table above, but it does give scope for growth and the ability to run multiple gateways and/or a central server (if required).

3.9 Installing a Secure Container

3.9.1 About

One of the key advantages of Kony for SAP is its ability to run on many different platforms. The following sections provide details on each platform on how to install and configure the Secure Container software, trouble shoot problems and utilize the vendor infrastructure.

For detailed information on systems management of Secure Containers, refer System Management and Administration.

3.9.2 Key Topics

Deploying on Windows

Deploying on BlackBerry

Deploying on Android

Deploying on iOS

Deploying HTTP Applications

Deploying on Unix

Configuring the Secure Container

3.9.3 Secure Container Supported Platforms

The following table outlines the supported device operating systems. If you have a release that is not listed, check with Kony.

OS	Release	Memory+	Notes
Windows Mobile	5.x, 6.x	128 Mb	Windows 7 is not currently supported. Support for Windows 8 metro is currently in progress. You require a Java JVM.
Windows Desktop/Tablet	XP, Vista, 7, 2003, 2008	1 Gig	Both 32 and 64 bit options are supported. You require a Java JVM.
BlackBerry Phones/Tablets	4.x, 5.x, 6.x, 7.x	64 Mb	Devices running BBX 10+ or QNX (Playbook) are not directly supported. You may run Secure Container on these platforms using BlackBerry Java or Android emulation. Contact RIM for more details.
IOS	4.x, 5.x	N/A	iPhone, iPad, iTouch 4.3+, 5.x. ***Note:*** 6.x is currently being tested.
Android	2.1+, 3.x, 4.x	N/A	4.x devices are supported as of release 24.01.11.
Unix	AIX, HP-UX, Linux, and Solaris	1 Gig	Application Server and Access Gateway only (no client). You require a Java JVM.

OS	Release	Memory+	Notes
Max OSX	11+	1 Gig	Application Server and Access Gateway only (no client). You require a Java JVM.

3.9.4 Windows Platforms

Secure Container runs on any 32 bit, 64 bit and mobile Windows environment. Separate implementation processes are provided for the desktop/server and the mobile environments. For any of the Windows platforms, you must install a Java Runtime Environment, for example, Sun for desktop/server, Creme for mobile. Currently, there is limited public application store support by Microsoft, so Kony provided specialized installers and CAB files to mimic an app store download. After installation, the provisioning process is the same as all the other platforms.

3.9.4.1 Windows Desktop and Server

The Sky Secure Container installation for Windows guide provides detailed steps and prerequisites that you need to ensure that the Container installs and executes correctly on the Windows platform. Ensure that the requirements are met before attempting an install.

Installation Options

Sky provides two installers for Windows:

- a basic, automated installer that performs as much checking as possible that a Java JVM is installed, and 32/64 bit compatibility.
- an expert installer that consists of a zip file

> ***Note:*** The expert installer is designed for people who are very experienced with Kony and need to perform extensive configuration of the Secure Container.

Requirements

Before commencing the installation, ensure that the following prerequisites are addressed:

- The Sky SAP add-in components are installed and configured in the SAP system.
- A target installation directory must be selected.
- The appropriate Sky Application Server Windows installation file must be obtained from Sky Technologies.
- A Java JVM must be installed.
- Depending on organizational security policy, a Windows Service Account with appropriate privileges may be required.

> ***Important:*** Ensure that you are installing on a Server version of Windows (for example, Windows Server 2003) if you plan to support more than a handful of users from a single Windows machine. The TCP/IP stacks included with operating systems such as Windows XP and Windows 7 professional do not scale well in such scenarios, and this can lead to problems later on.

Target Directory

You can install the Sky Application Servers anywhere on the Windows host, but typically they are installed in a directory called "Sky Technologies" under Program Files. Select an appropriate location based on the conventions of your organization. Throughout the remainder of this documentation, this directory is referred to as `<<SKY HOME>>`.

Java Runtime Environment

A Java Runtime Environment (JRE) is required to run the Sky Application Server. For the Windows platform, Sky Technologies recommend using the Sun JRE environment available from the Sun web site (http://java.sun.com). The Sky Technologies software is compatible with any version of Java from 1.1.8 onwards. In a practical sense this means virtually any version of Java currently available, as it is now very uncommon to see versions of Java earlier than 1.1.8 still in use. You may use either 32bit or 64bit versions of the JRE on 64 bit Windows platforms. However if using a 32 bit JRE on a 64 bit OS, it is necessary to install the 32 bit version of SkyMobile.

Windows Service Account

By default, the Sky Application Server services run as the Local System account. If required, you can change this to a dedicated Windows Service Account.

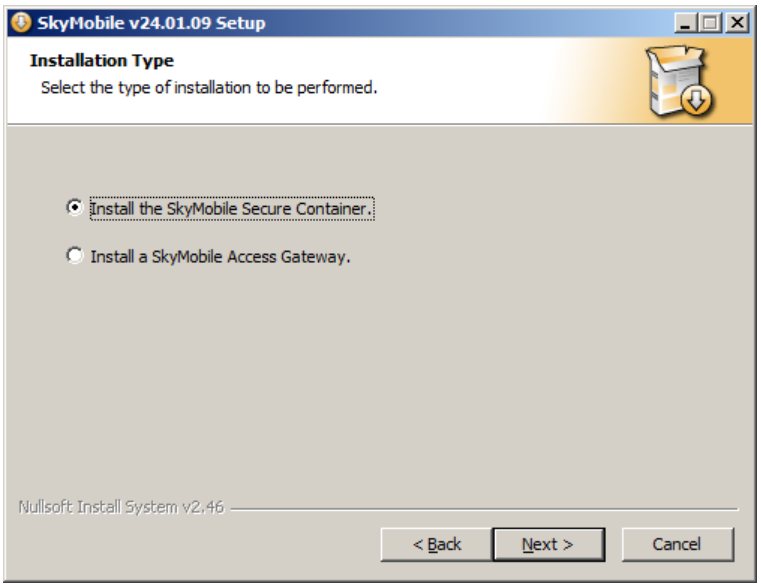
Click here to go back to the top.

Windows Basic Installer

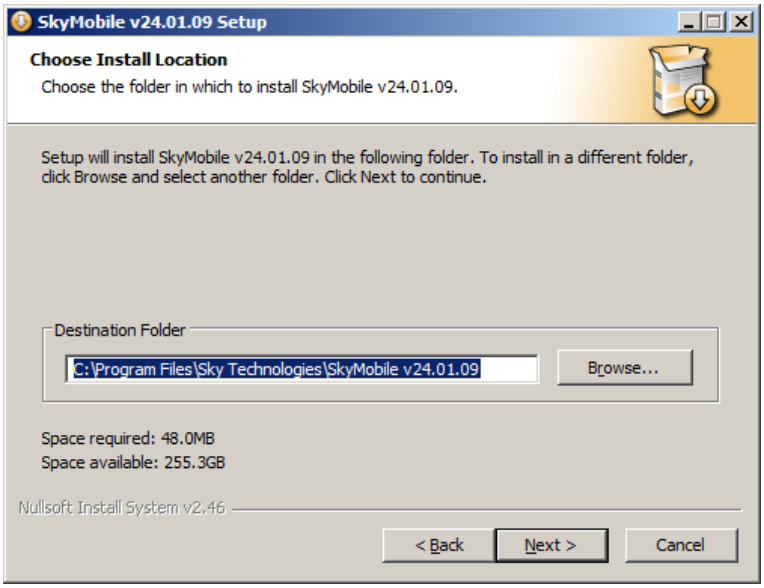
Kony for SAP provides a simple installer for Windows that caters for both 32 and 64 bit requirements. The installer also checks that a Java JVM is available and is compatible in terms of 32 or 64 bit modes. If you did not do so already, download a Java JVM for Windows free of charge from www.java.com. Download the SkyMobile Windows installer from either Kony Developer Portal. Then follow the prompts:

1. Click **Next**.

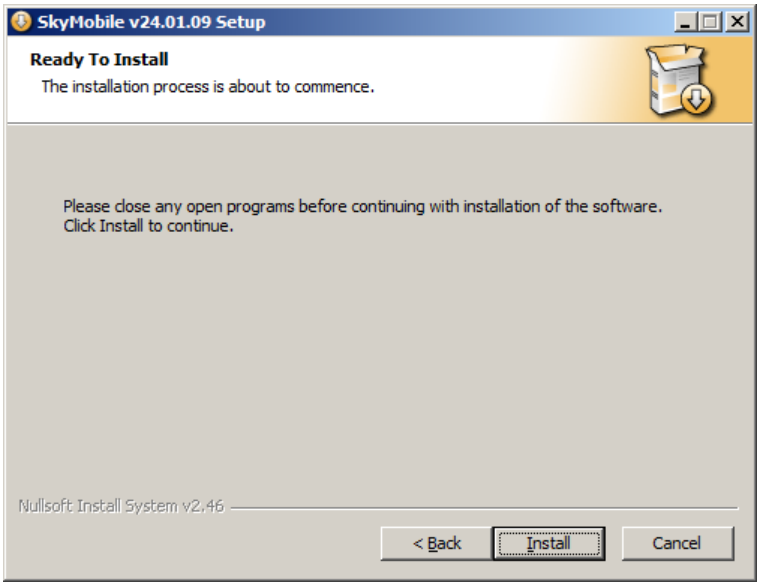
2. Select the **Install the SkyMobile Secure Container** option, and click **Next**.



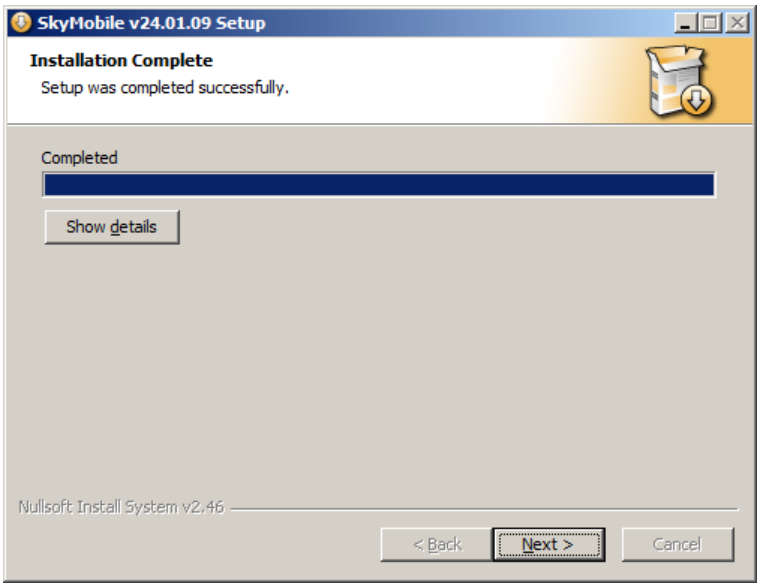
3. Select the installation directory.



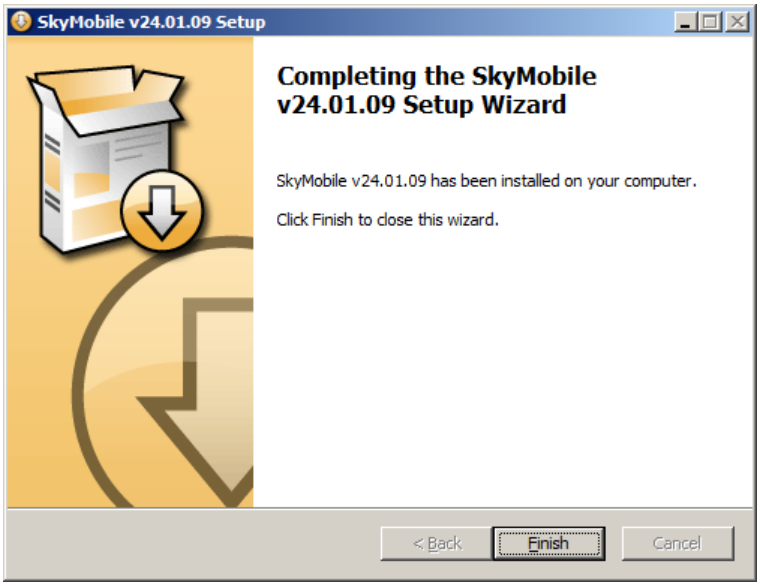
4. Close any open programs that may conflict with the install.



5. Click **Install**. The installation takes only five -10 seconds.



The installation finishes and a desktop short cut is created.



6. Click **Finish**.

When you start the Secure Container for the first time, you go through the standard device provisioning process to properly profile your device and user.

Test the Sky Application Servers

At this point, you can try launching the Sky Application Server to see whether it is successfully configured and installed. You can launch SkyMobile from a batch script in the `home/system` directory. For example, `<<SKY_HOME>>\v##.##.##\vti\vti.bat`.

Check the server log file (in the "log" directory) to see whether the server started successfully, and whether it is able to connect to the Access Gateway and MEAP Server.

If no log file is produced, there are likely two explanations.

- Invalid Java command line. Ensure that the SKY_JAVA_CMD environment variable is set and exported, or alternatively set the "JavaCommand" configuration parameter in the configuration file. Make sure that the value you assign is the correct path to the Java executable.
- The path to the Java executable defined in `SKY_JAVA_CMD` or `JavaCommand` exceeds the DOS limitation of eight characters and has something like `Program Files\Java\...` rather than `Progra~1\Java\...`

SkyMobile Windows Service

Following a successful launch of SkyMobile, the next step is to address how to have the installed services automatically launch when the machine boots, and how to restart them in the event that they are manually killed or go offline.

- Creating a Windows Service
- Configuring the Service
- Configuring Service Dependencies

3.9.4.2 Creating a Windows Service

You are provided with a batch script (`service_install.bat`) to create a Windows service for each of the Sky Application Servers and a corresponding script (`service_uninstall.bat`) to remove the service.

To create the service:

1. Log on as a user with permissions to create services.
2. Navigate to the appropriate Application Server directory under `<<SKY_HOME>>\v##.##.##`.
3. Run the batch script `service_install.bat`.
4. Start the Windows services applet.
5. Confirm that the new Application Server service is present.

> ***Note:*** If you need to run multiple instances of an Application Server as Windows services, you can install from separate directories by supplying different arguments to the `service_install.bat` script. You can achieve either by either changing the default argument in each copy of the `service_install.bat` script, or by invoking them from the DOS command line with different arguments.

3.9.4.3 Configuring the Service

When created, the Windows service for each Sky Application Server is defaulted to manual startup and to logon using the Local System account. To ensure automatic startup, restart and authorization you may want to change the values as indicated in the table:

Option	Value
Startup Type	Set to automatic to have the service start automatically when the server is started
Log on as	Local System Account if no dedicated service account. Otherwise specify the service account.
Recovery	First and Second Failure - set to "Restart the service" to automatically restart the service if it fails.

3.9.4.4 Configuring Service Dependencies

If you configured the Application Server to use database storage you should specify a dependency on that database service to ensure that the Sky Application Server starts only once the database is available. You can achieve this using a service dependency rule.

To do this:

1. Start the registry editor (regedit or regedt32).
2. Navigate to the following registry hive:
   HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/SKY_{product}
3. Create a new REG_MULTI_SZ value with the name: DependOnService

You must configure the data portion of the registry value to the name of the database service (for example, MSSQLSERVER in the case of Microsoft SQL Server). This ensures that the database server launches before the Sky product does, allowing it to connect to it properly.

Windows Environment Variables

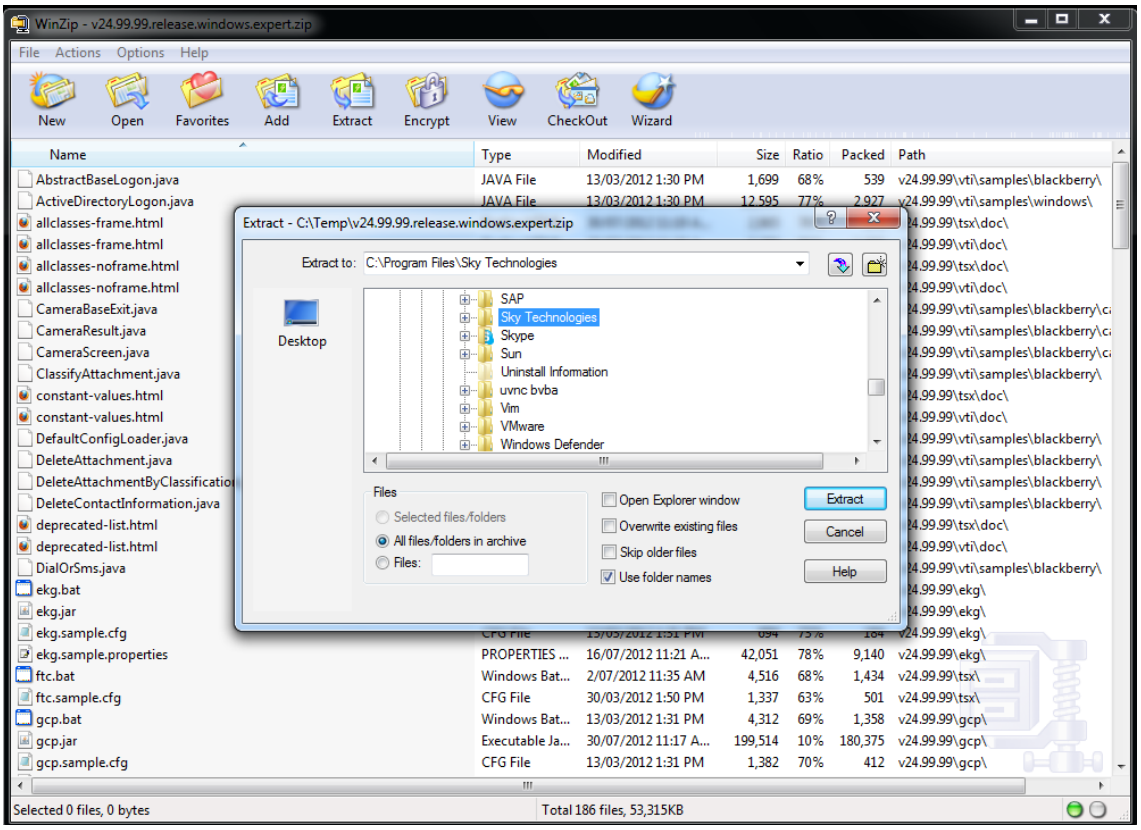
Kony for SAP makes use of a number of environment variables that control various facets of its behaviour. It may be appropriate in some circumstances to consider making use of these. However, in most situations you do not require.

Environment Variable Name	Function
SKY_JAVA_CMD	This environment variable specifies the path to the Java virtual machine. It is used if no value is specified for the "JavaCommand" parameter under the [Java] section of the relevant configuration file. If no value is obtained from either source, the Java command is assumed to be "java", that only works if the Java binary lies in the current path of the user.
SKY_JAVA_ARGS	This environment variable specifies any arguments that you can use in confunction with the SKY_JAVA_CMD environment variable.
SKY_JAVA_HOME	Sets the directory for the SkyMobile installation directory.
SKY_VTI_HOME	Sets the directory for the SkyMobile Application server directory.

Windows Expert Installer

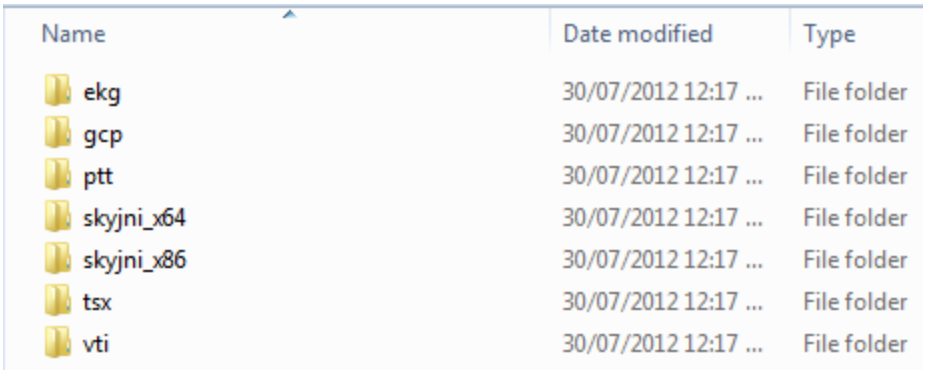
Sky provides an expert installer for Windows that is delivered as a zip file. The expert installer is available on Kony Developer Portal. To use this installer:

1. Download a copy of the zip file to the Windows machine
2. Using a tool like Winzip, extract all of the files to the appropriate location, ensuring you retain the folder names.



3. In Windows Explorer, navigate to the directory where you extracted the files.

   You see two JNI folders - one for 32bit and a second for 64bit Windows



4. If running on a 32bit Windows platform with a 32bit version of Java, rename `skyjni_x86` folder to `skyjni`.
5. If running on a 64bit Windows platform with a 64bit version of Java, rename the `skyjni_x64` folder to `skyjni`.

The SkyMobile container is now ready for configuration.

> ***Note:*** If you try and run with a JNI version that does not match your Windows version and/or JRE version (for example running Windows 32bit with a 32bit JRE and 64bit JNI libraries) you get the following error in your SkyMobile server log.

```
30/07/2012 12:26:03PM ERROR: [VtiHeartbeat] Error while attempting to
open a connection to host.
java.lang.UnsatisfiedLinkError: C:\Temp\v24.99.99\skyjni\skyrfc.dll:
Can't load AMD 64-bit .dll on a IA 32-bit platform
at java.lang.ClassLoader$NativeLibrary.load(Native Method)
at java.lang.ClassLoader.loadLibrary1(Unknown Source)
at java.lang.ClassLoader.loadLibrary0(Unknown Source)
at java.lang.ClassLoader.loadLibrary(Unknown Source)
at java.lang.Runtime.loadLibrary0(Unknown Source)
at java.lang.System.loadLibrary(Unknown Source)
at
au.com.skytechnologies.ecssdk.saprfc.SapRfcConnection.loadLibrary
```

```
(Unknown Source)
at au.com.skytechnologies.ecssdk.saprfc.SapRfcClientConnection.<init>
(Unknown Source)
at au.com.skytechnologies.ecssdk.saprfc.SapRfcClientConnection.<init>
(Unknown Source)
at au.com.skytechnologies.vti.VtiHostSapRfcConnectionMaker.runThread
(Unknown Source)
at au.com.skytechnologies.ecssdk.thread.EnhancedThread.run(Unknown
Source)
30/07/2012 12:26:03PM ERROR: [VtiHeartbeat] Heart beat connection was
NOT established for host interface MEAP.
```

Troubleshooting on Windows

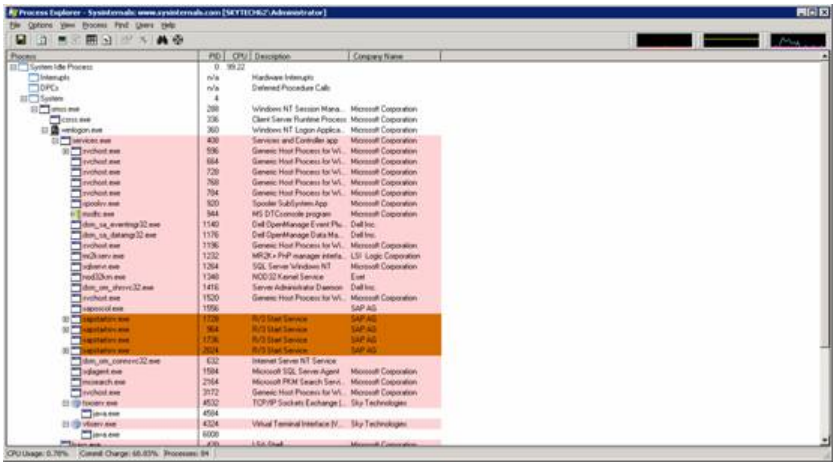
This section lists some of the more common problems. If you cannot resolve the issue yourself, contact Kony for SAP support.

Issue	Process
SkyMobile does not start and do not produce a log.	The most common cause of this problem is that there is no Java Virtual Machine available. Check to ensure that the value specified in either the Sky product configuration file JavaCommand option or the SKY_JAVA_CMD environment variable points to a valid Java Virtual Machine executable. Try typing "java" at the command line to see if Java is installed. If Java is installed and is located under the `Program Files` directory, check that you have specified `Progra~1` in the JavaCommand or environment variable.
SkyMobile does not start and produces a log.	The log file should contain details of the error that is preventing the product from launching successfully. Examine the log file in a text editor and this should allow appropriate corrective action. The most common cause of this scenario is a missing license file.
SkyMobile Windows Service stops for no apparent reason.	The most likely explanation for this situation is a known bug in Sun Java Virtual Machine under Windows. Windows incorrectly targets Java processes started from a Windows service as user processes, so when a user logs out they are shut down by the operating system, regardless of the fact that as services they are meant to be user independent. Until Sun and/or Microsoft manage to resolve this problem, the current workaround is to append a "Xrs" at the end of the JVM command line specified in either the Sky product configuration file JavaArguments option, or the SKY_JAVA_CMD environment variable. This instructs the Sun JVM to accept a reduced signal set, and prevents the problem from occurring. If the above explanation is not relevant in your case, examine the Windows event log and the VTI log for further information on why the service stops.

Issue	Process
OutOfMemoryException Problems	If the Java Virtual Machine (JVM) is running out of memory, you are probably using one that allocates an arbitrary upper limit to the amount of memory it can use. Many JVMs, including those that Sun supplied, do this as a safeguard to protect Java from consuming too many resources. In most cases, you can resolve the situation by increasing the upper memory limit assigned to the Java virtual machine when it is launched. You need to modify either the Sky product JavaArgument or the SKY_JAVA_CMD environment variable so that the increased memory ceiling is specified as an argument to the JVM executable. The appropriate syntax to use depends on the JVM that you are using; consult relevant JVM documentation if you are unsure. As an example, the following setting is appropriate under the Sun JVM, to increase the maximum memory ceiling to 128 megabytes: -Xmx128m
Issues accessing a network drive or printer queue	This problem generally arises when Java product runs as a Windows service and is configured to run as "Local system". Because you cannot link the service to a specific user, network resources such as mapped drives and print queues are not accessible to it. The solution in this scenario is to configure the Windows service to run as a specific user rather than as "Local system". You can achieve this change through the Windows services applet.
Loading incorrect `skymobile.cfg` on Windows 7.	Windows 7 protects the `Program Files` directory by creating a separate directory that it uses for extra files created. There are two ways to fix this problem. 1. Run `wpc.exe` as administrator and it ignores this extra directory. (Or) 2: Navigate to `C:\Users\<username>\AppData\Local\VirtualStore\Program Files\Sky Technologies\xxx` to see the files it uses.

Monitoring SkyMobile in Windows

The Sky Application Server applications run continuously as Java JVMs. Typically each instance of a JVM runs as a separate Windows process. You can view the processes using Task Manager or Process Explorer.



The Java Virtual Machines show up as either `java` or `jre` (if executing Java runtime environment).

3.9.4.5 Windows Mobile

SkyMobile can run in two modes on a handheld device:

Container

In this mode, the complete Application Server is installed on the handheld device (along with the presentation client). This enables both offline and real-time applications. This is the Sky recommended implementation.

Thin Client

In this mode, the only software installed on the handheld device is the presentation client (WPC). In this mode, only real-time applications are possible.

Configuration

In order to complete the installation of your Sky Technologies software, you may need to change the standard configuration files provided with the software. To prepare for this configuration, ensure that you still have an ActiveSync connection from your PC to your handheld device. From the PC, browse to the `\Program Files\Sky Technologies\VTI` directory. In this directory, you see a file named `skymobile*.cfg*`. This file contains all the configuration settings for the Application Server and Presentation Client (WPC).

Copy the file back to your PC. They are standard text files and you can edit using Notepad or Wordpad. The Configuration Options describe all the meanings and possible values of the various settings within the configuration file.

Once you make any necessary configuration changes, copy the file(s) back to the `\Program Files\Sky Technologies\VTI` directory on the handheld device.

You are ready to start your application.

Secure Container Installation

Before commencing the installation, ensure that you have the following:

- A PC running Microsoft Windows 2000, 2003, XP, Vista or 7. The PC requires a USB or serial port to which you can connect the handheld device on which the software is installed.
- Microsoft ActiveSync must be installed on the PC that you use. ActiveSync is often shipped with many handheld devices. You can also download from http://www.microsoft.com.
- A Java Virtual Machine (JVM) CAB file. Sky recommends the use of NSI CrEme JVM.
- An installation file for the Sky Technologies software. Sky Technologies provides this, and is normally called `v#.##.release.exe`, where #.## is the software version number.
- The configuration guide for your application (if Sky Technologies provided).

Recommended JVM Vendors and Configuration

To run SkyMobile on Windows Mobile platform, you must first install a suitable JVM. All the following JVMs have different features that you need to consider when selecting the appropriate JVM for your project. Unfortunately, there are known issues with all of the following JVMs. Sky recommends using CrEme but under certain circumstances, you may need to use one of the others.

JVMs

- CrEme
- JBED
- J9

> ***Note:*** When copying text from this document, ensure that your signs are ASCII code 2D. Word quite often substitutes this character for ASCII code 96 that the JVM does not recognise. A simple way to avoid this is to copy and paste the text into your text editor, and replace the signs by manually typing over them.

3.9.4.6 CrEme

Detail	Notes
Supported by:	NSICom
Purchase from:	http://nsicom.com
Java supported:	J2ME CDC Personal Profile 1.0 JSSE Swing 1.1.1 javax.comm

Detail	Notes
Features:	Memory usage up to 64MB (Standard is 32MB) Remote thread/memory monitor utility Java extensions API for some Symbol devices
Issues:	As of November 2008, the JIT module caused a JVM crash under certain circumstances. This appeared to be primarily triggered by application refreshes. "--nojit" disables JIT that avoids the crash, but slows down the application. This error appeared to be fixed in later versions of CrEme.

Useful options:

Option	Details
-mi	Run multiple instances of CrEme.
-ml x	Sets memory limit to x KB. Max is 64000.
-mon	Allow the debug monitor to connect
-nojit	Disable the JIT compiler
-Of	Redirect console output to jscpout.txt
-sp:0	Turn off the CrEme splash screen
Recommended `skymobile.cfg`	Server.Java.JavaCommand = `\Windows\CrEme\bin\CrEme.exe` Server.Java.JavaArguments = -cp '&classpath&' -sp:0 -Of -ml 64000

3.9.4.7 JBED

Detail	Notes
Supported by:	Esmertec
Purchase from:	http://handango.com (Esmertec Jbed CDC JVM)
Java supported:	J2ME CDC Personal Profile 1.0
Issues:	Some issues are noticed with respect to socket connections intermittently hanging upon startup. You can resolve this by using the SkyMobile native sockets implementation.

Useful options:

Option	Details
-Xnowinceconsole	Hides console window
-Xnoprogressbar	Hide the splash screen
-Djeode.evm.memory.stackSize=x	Configure the stack size to x bytes. If you receive a stackOverflowException in the log file, try setting this value to 98000 or greater.
Recommended `skymobile.cfg`	Server.Java.JavaCommand = `\Windows\evm.exe` Server.Java.JavaArguments = -cp "&classpath&" -Xnowinceconsole --Xnoprogressbar Server.Network.UseNativeSockets = true

3.9.4.8 J9

Detail	Notes
Supported by:	IBM
Purchase from:	http://handango.com (IBM Websphere Everyplace Micro Environment for Windows Personal Profile 1.1)
Java supported:	J2ME CDC Personal Profile 1.1
Features:	Java Debug Wire Protocol (JDWP) Supports Smartphone
Issues:	Quite slow

Useful options:

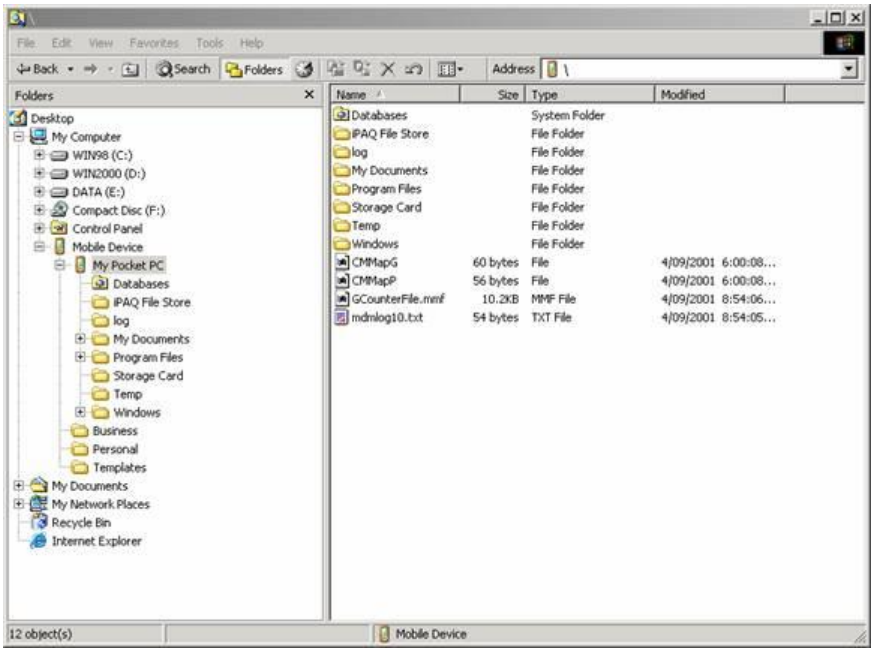
Option	Details
-debug	Enable JDWP
-jcl:xxx	Specify which class library to use (foun11) or (ppro11)
-noverify	Disable class file verification
-Xint	Run interpreted only
-Xjit[:count=x, code=y] Eg. --Xjit -Xjit:count=0 -Xjit:code=1024	Enable JIT. Specify count to force JIT compilation after x invocations. Specify code to configure the size of the JIT code cache (in KB).

Option	Details
Recommended `skymobile.cfg`	Server.Java.JavaCommand = `\Program Files\J9\PPRO11\bin\j9w.exe` Server.Java.JavaArguments = -jcl:ppro11 -cp "&classpath&"

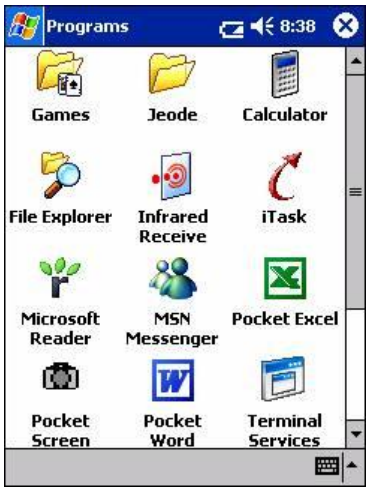
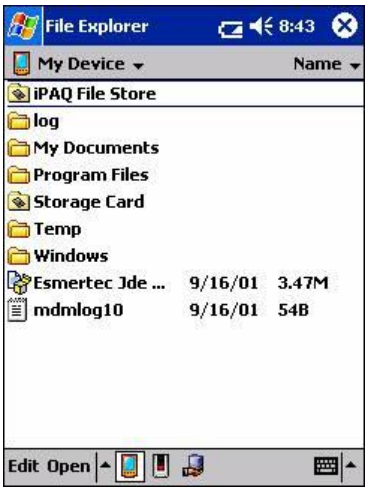
Installing JVM

To install Java Virtual Machine on the handheld device, perform the following:

1. Locate the JVM CAB file. Once you obtain the Java Virtual Machine to install, you should be able to locate a CAB file.
2. Copy the JVM CAB file onto the handheld device. Underneath the `My Computer` directory, you can find an item called 'Mobile Device' (this is installed by Microsoft Activesync). By opening this item, you can browse the file system on the handheld device. Copy the `JVM CAB` file from its current location on your PC, and paste it into the `My Pocket PC` folder.



3. Launch File Explorer on the handheld device. To do this, press the **Start** button and select the **Programs** option from the menu. In the window that appears, tap the 'File Explorer' icon to launch File Explorer.

	
Tap on File Explorer to browse the device	Tap on the file to install the JVM

4. Install the JVM. Use File Explorer to locate the **JVM CAB** file that you copied onto the device. Tap on the file to install it.

Installing SkyMobile

To install the Sky Technologies VTI software on the handheld device, perform the following:

1. Unpack the Sky Technologies software onto the PC. The Sky Technologies Windows Mobile software is provided in the form of a "zip" file. You can access the files inside a "zip" archive on Windows XP and later versions of Windows, by simply doubleclicking the file. If you are running an earlier version of Windows, you may need to use a tool such as WinZip to unpack the archive. You can extract the files inside the "zip" archive into any directory you choose; the location is not important.
2. Locate the `VTI CAB` file. This file is called `vti.cef.cab`, and it should be present in the directory into which you unpack the Sky Technologies software.
3. Copy the `VTI CAB` file onto the handheld device, as described for the `JVM CAB` file.
4. Launch File Explorer on the handheld device, as described for the `JVM CAB` file.
5. Install VTI, as described for the `JVM CAB` file.

Installing Thin Client

Before commencing the installation, ensure that you have the following:

- A PC running Microsoft Windows 95, 98, ME, 2000, 2003, or XP. The PC requires a USB or serial port to which you can connect the handheld device on which the software is installed.
- Microsoft ActiveSync must be installed on the PC that you use. ActiveSync is often shipped with many handheld devices.You can also download from http://www.microsoft.com.
- An installation file for the Sky Technologies software. Sky Technologies provides this , and is normally called `v#.##.release.exe`, where #.## is the software version number.
- The configuration guide for your application (if provided by Sky Technologies).

> ***Note:*** A Java Virtual Machine (JVM) is not required on the handheld device when you use WPC in thin client mode.

Installation Process

Installation of the Sky Technologies Windows Presentation Client (WPC) software on the handheld device requires the following steps.

1. Unpack the Sky Technologies software onto the PC. The Sky Technologies Windows Mobile software is provided in the form of a "zip" file. You can access the files inside a "zip" archive on Windows XP and later versions of Windows, by simply double-clicking the file. If you are running an earlier version of Windows, you may need to use a tool such as WinZip to unpack the archive. You can extract the files inside the "zip" archive into any directory you choose; the location is not important.
2. Locate the `WPC CAB` file. This file is called `wpc.cef.cab`, and it should be present in the directory into which you unpack the Sky Technologies software.
3. Copy the `WPC CAB` file onto the handheld device, as described for the JVM CAB file in this section.

4. Launch File Explorer on the handheld device, as described for the JVM CAB file in this section.

5. Install WPC, as described for the `JVM CAB` file in this section.

Starting the Application

If you are using a local Application Server configuration, be sure that you are in a situation where the SkyMobile server is able to connect to a Access Gateway when starting your application for the first time. You can connect through a GPRS connection, 802.11, 3G network, or by sharing your PC network connectivity through ActiveSync. This is necessary as the application needs to download from the central MEAP Server. Once this step is successfully completed, connectivity is only necessary for application functions that access SAP directly. If you use a thin client configuration, your handheld device always requires network connectivity to be able to access the SkyMobile Application Server delivering your application. Again, this connectivity can be through a GPRS connection, 802.11, 3G network, or by sharing your PC network connectivity through ActiveSync.

From this point, you should are able to begin using your application.

Creating a Customized CAB File

You are provided with a CAB file that installs SkyMobile on to a Windows mobile device as part of each release. However, if your application contains java exits, print templates, or you want to bundle up your skymobile.cfg file as part of the installation, then you have to build your own CAB file. To build your own CAB file, follow these steps:

1. Steps required to configure the `makecab.bat` script
2. Running the `makecab.bat`

What is a CAB File?

A CAB file, short for cabinet file, is a group of files compressed into one larger file to conserve disk space. CAB files are often used to distribute applications. During installation, the Setup program extracts files from the CAB file and copies them to the specified locations on the hard disk.

They are essentially an archive file - like a zip or jar file. In this case, the cab file also contains instructions about where to place the unzipped files. It can even create directories and registry keys. When you install products on a PDA device, they are copied across to the device either manually or by using the `setup.exe` utility (Refer Sky guide on automatically loading CAB files from a PDA storage card). Once there, if copied manually, simply click the cab file and it automatically installs applications such as VTI on the device. If using the `setup.exe` utility, the add/remove programs of active sync is activated, the cab file is copied across to the device and is automatically installed.

What is Required?

1. On your desktop PC, create a directory called `makecab`. Place the following files in that directory. These are the Microsoft products required to create the cab file and you can find in the `../Windows CE Tools/wce420/POCKET PC 2003/Tools` directory of the Pocket PC 2003 SDK for Embedded Visual C++ 4.0.

File
CabWiz.exe
MakeCab.exe
Cabwiz.ddf

2. The following files use the activesync link to copy the cab files across to the device, activates the `winload exe` on the device and installs the application. Place these files in the same directory.

File
CabWiz.exe
MakeCab.exe
Cabwiz.ddf

3. Create a file called `MakeCab.bat` in `makecab` directory using any text editor, include the following lines and save. - This is for convenience. It provides the `cabwiz.exe` with the correct parameters to make the cab file.

   MakeCab.bat

```
@Echo off
echo Attempting to create cab file(s)
cabwiz "VTI.inf" /err VTIerror.txt /cpu CEF
echo You may now close this window.
```

4. Create a file named `VTI.inf` using a text editor in the same directory. This is the configuration file that you need for VTI installation.

   Copy and paste from the sample `VTI.inf` from appendix 2.1

   The following describes some of the section from `VTI.inf` file that needs customization.

   [SourceDisksNames]

   This gives the location of all files to be archived or installed. In this case, it is in a directory called `Shared`. '1' refers to the all files equal to '1' in SourceDisksFiles section. (See the sample).

   Sample

   For example: 1 = ,"Common files",,Shared

   [SourceDisksFiles]

   These are all the files that you should install and all referencing to Shared directory - '1'.
   In this case:

   Sample

   skymobile.cfg = 1
   vti.jar = 1
   wpc.cfg = 1
   vti.lnk = 1
   vti_server.lnk = 1
   ecssys.dll = 1
   wpc.exe = 1

[DestinationDirs]

This is where to put the files once archived.

Sample

[Files.Application]
[Files.Shared]
[Files.ShortCut]

These sections are not strictly necessary, but in this case, is a neater way of setting up the install process.

5. Now create a directory called `Shared` in the `makecab` directory. Place the following files in the new `shared` directory.

File
skymobile.cfg
ecssys.dll
vti.jar
vti_server
wpc.cfg
wpc.exe

Note: You need to include any other SkyMobile specific configuration files, such as user exit jars and print templates.

6. So far in your `makecab` directory, you should have the following seven standard files plus a directory called `Shared`.

   Cabwiz.exe
   Makecab.exe
   Cabwiz.ddf
   setup.exe
   install.ini
   makecab.bat
   VTI.inf
   Shared (directory only)

Creating the CAB file

Click the `MakeCab.bat` file.

If everything is correct you should have a CAB file created in your `makeCab` directory. If there are any problems with creating cab file, it reflects in the txt file `Vtierror.txt`.

> ***Note:*** Warning messages are just warnings and you not need take any action to correct.

> ***Important:*** Each time you commence making a new cab file, be sure to delete the old one first or move it to a different directory so as to not lose it.

Sample .vtf file

VTI.inf

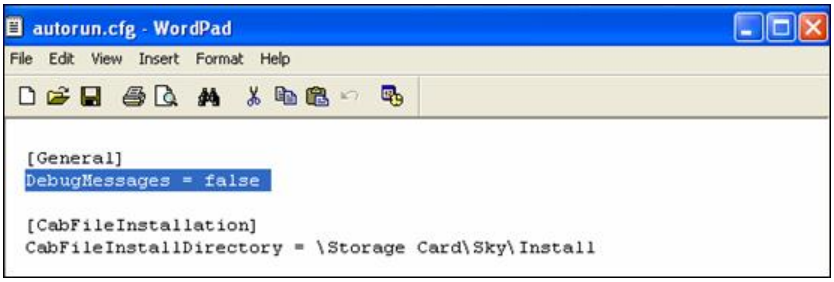
```
[Version]
Signature = "$Windows NT$"
Provider = "Sky Technologies"
CESignature = "$Windows CE$"
[CEStrings]
AppName="VTI"
InstallDir=%CE1%\Sky Technologies%AppName%
[CEDevice]
UnsupportedPlatforms = "Jupiter","HPC" ; Does not support pltfrm1
VersionMin = 3.0
VersionMax = 100
[CEDevice.CEF]
ProcessorType = 0
[DefaultInstall]
CopyFiles = Files.Application, Files.Shared, Files.ShortCut,
Files.PrintTemp, Files.Dll
[SourceDisksNames]
1 = ,"Common files",,Shared
[SourceDisksFiles]
skymobile.cfg = 1
vti.jar = 1
wpc.cfg = 1
vti_server.lnk = 1
ecssys.dll = 1
wpc.exe = 1
[DestinationDirs]
Files.Application = 0,%InstallDir%;\Program Files\Sky Technologies\vti
Files.Shared = 0,%InstallDir%;\Program Files\Sky Technologies\vti
Files.ShortCut = 0,%CE17%\Programs ;\Windows\StartMenu\Programs
Files.PrintTemp = 0,%InstallDir%\print_templates;\Program Files\Sky
```

```
Technologies\vti\print_templates
Files.Dll = 0,\Windows ;\Windows
[Files.Application]
"vti.jar",,,0x00000001 ; rename, and warn if skipped
[Files.Shared]
"skymobile.cfg",,,0x80000000; rename, and mark as shared
"vti.evm",,,0x80000000; rename, and mark as shared
"wpc.exe",,,0x80000000; rename, and mark as shared
"wpc.cfg",,,0x80000000; rename, and mark as shared
[Files.ShortCut]
"vti_server.lnk",,,0x80000000 ; rename, and mark as shared
[Files.PrintTemp]

[Files.Dll]
"ecssys.dll",,,0x80000000 ; rename, and warn if skipped Installing
using a SD card
```

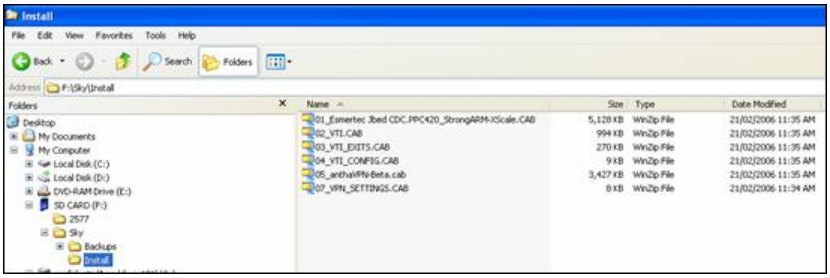
Once cab files have been generated and need to be deployed to a Windows Mobile device an approach may be use a pre-configured SD card to perform the initial install of the application. The idea being that on a reboot of the device, the software automatically re installs from the cab files on the SD card.

3.9.4.9 Preparing the SD card

On the SD card make a directory called 2577 from the root directory. When a device is cold booted or when the card is installed into a running device, Windows Mobile looks here first to execute any items recorded. In this place two files, autorun.exe and "autorun.cfg" (autorun.exe is supplied by Sky Technologies in the wince/arm directory). Below is a sample "autorun.cfg", and the instruction being given is where to find the SkyMobile cab files on the SD card. Copy the relevant cab files to this location. The SD card is now ready.



The autorun.exe script executes the cab files onto the Windows Mobile device. Remember the destination of the files within the cab file is part of the cab file itself. The autorun.exe script executes the cab files in sequence so be careful to name the files according to the sequence they should be install. As seen below the easiest way to achieve this is to use a numbering convention.



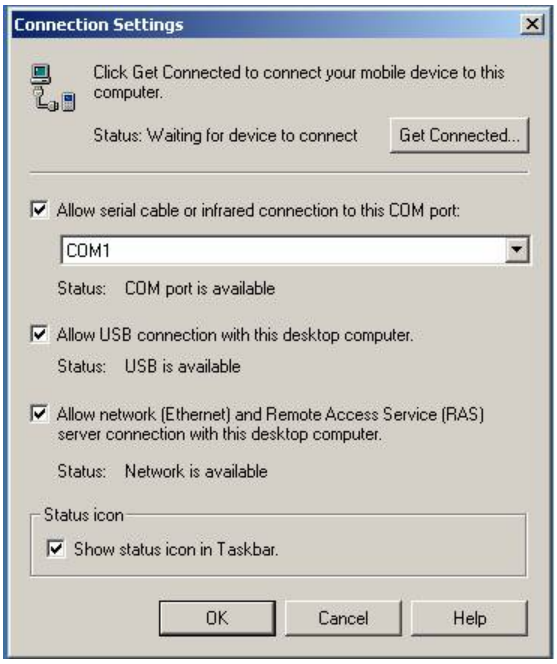
This procedure does not retrieve data that may have been on the device before the reinstallation process. Once the SkyMobile software is reinstalled a SkyMobile backup would then be required to retrieve data stored back to the last time a backup was completed on the device.

How to Install and Use Active Sync

1. Download and install ActiveSync on to your desk top and Windows Mobile
   http://www.microsoft.com/windowsmobile/addons/default.mspx

   You may also have been provided with ActiveSync as a resource along with the Windows Mobile device. Once you install, open Active Sync on your desktop PC (Double-click Active Sync on your desk top), then go to **File** > **Connection Settings**, and select the following check boxes,

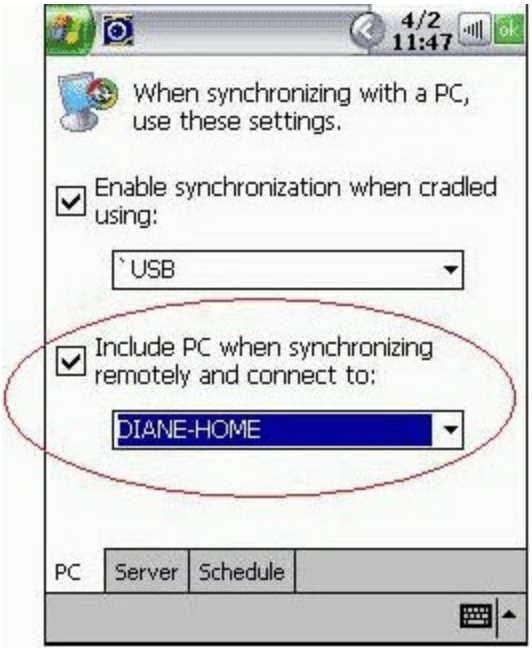
   - **Allow serial cable or infrared connection to this COM port**.
   - **Allow USB connection with this desktop computer**.
   - **Allow Network (Ethernet) and Remote Active Services (RAS) server connection with this desktop computer**. Below the check box, it should say, "Status: network is available".



2. Connect the device cradle using USB/ Serial Port.
   The docking cable comes with two connectors, USB and Serial. Use either to connect the

device to your PC. You need to tell the pocket device manually the type of connector you used. (that is USB/ Serial port.) by tapping > **Start** > **Programs** > **ActiveSync** > **Tools** > **Options**. You see two drop-down lists as shown below. On the first drop-down list, select one of the following:

- **115200 Default** if the connection is Serial Port.
- **USB** if the connection is USB.



3. On your PC, run ActiveSync (Double-click) ActiveSync to open and then click the file > Get Connected to synchronize with the Windows Mobile device. If the connection established, you see the word 'Connected' on your Windows Mobile and on the Microsoft ActiveSync window on your PC. If the connection is not established, you see 'Not Connected'. You may have to do the above step a few times in order to establish a connection.
4. Once connected, on your PC, double-click **My Computer** > **Mobile Device** > **My Pocket PC**. This shows you the contents of the `My Pocket PC` folder on your Windows Mobile.

5. You have to place the CAB file for the product you wish to install into this folder. Details about CAB files and how you can create one are explained later in this document. You can either drag it from another Explorer window or you can Copy and Paste it.

6. On the Windows Mobile device, run File Explorer by selecting **Start** > **Programs** > **File Explorer** from the menu button.

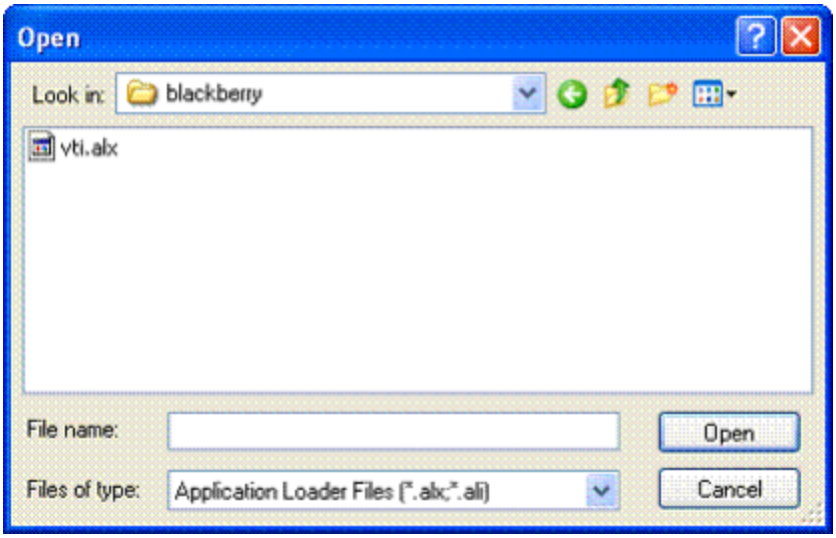
7. Select **My Device** from the selected folder drop-down list. If you are viewing the directory system from your PC, the name of the directory is `My Pocket PC`.

8. Click the CAB file to install the application.

> ***Note:*** You do not actually see the `.CAB` file extension because the Windows Mobile File Explorer hides file extensions.

Synchronising Files with the Device

A Windows Mobile usually comes with a cradle that you can attach to the desktop PC through a serial/USB cable. Microsoft provides a program called ActiveSync that manages the transfer of data between the desktop and Windows Mobile device. You can configure ActiveSync to synchronize e-mail, files, applications and more.

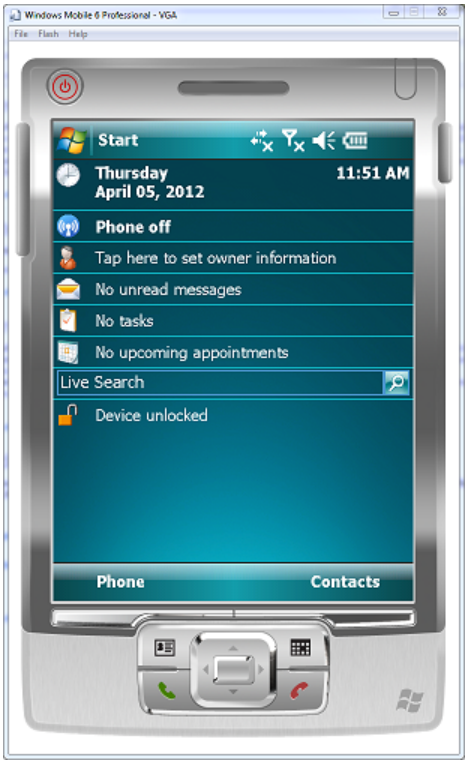


The Windows Mobile Emulator

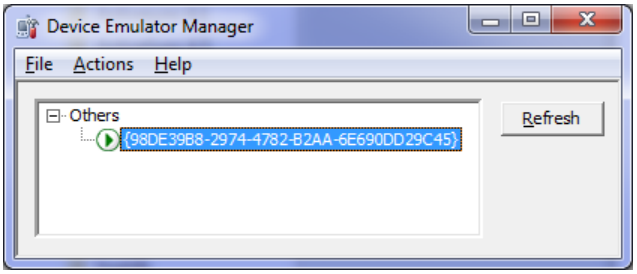
Microsoft provides emulators for the Windows Mobile Phone platforms. You can find these in the following locations:

- WM 5.0 Emulators: PPC and Smartphone
- WM 6.0 Emulators
- WM 6.1 Emulators

Once you download and install, shortcuts for a variety of different sized emulators are placed on your start menu. Once you open, you can interact the emulators with in the same way as a real device.



To simulate events such as connecting through USB (cradling) or resetting the device, you can start the "Device Emulator Manager" that lists all available Windows emulators. The various commands are available upon right-clicking the correct emulator instance in the list.



3.9.5 BlackBerry Platforms

This section describes the installation of an Application Server onto a BlackBerry device. You may download the Server automatically from the BlackBerry World "app store" and then automatically provision. The following sections describe the options available to manually package and install the Server.

> ***Note:*** The minimum operating system version required on the BlackBerry device is OS v4.2.1+ (Devices running OS v4.1 and below are not supported anymore). The maximum operating system version of the BlackBerry device is v6.

3.9.5.1 To Check your Device Model and Version

1. Find and select the **Options**. You see the **Device Options** screen.
2. Select **About.** The BlackBerry model number appears at the top of the **About** screen, directly under that, you see the platform version.

If you have lower than OS v4.2.1 and wish to upgrade your device OS to v4.2.1 or later, visit http://na.blackberry.com/eng/support/downloads/download_sites.jsp for more instructions.

BlackBerry JDE Version

> ***Note:*** Applications built in the BlackBerry JDE are forward-compatible with newer BlackBerry Device Software versions, but they are not backward-compatible with older versions.
> For example, an application built in the 4.2.1 BlackBerry JDE version 4.2.1 runs on a BlackBerry device running BlackBerry Device Software 4.2.1 and later. It does not run on a BlackBerry device running BlackBerry Device Software 4.1.

3.9.5.2 Blackberry App World

You must have the BlackBerry app world application and an account.

From the BlackBerry device:

- Select the app world application
- Search for SkyMobile
- Select the SkyMobile application and download.

3.9.5.3 Manual Installation

The following sections document how to manually install the SkyMobile Application Software for BlackBerry.

Requirements

Before commencing the installation, ensure that you have the following:

- A PC running Microsoft Windows 2000, 2003, XP, Vista or 7. The PC requires a USB or serial port to which you can connect the BlackBerry device on which the software is installed.
- A BlackBerry device running version 4.2.1 (or later) of the BlackBerry OS. You can find out which version of the BlackBerry OS your device uses by selecting the "Options" icon from the Applications Menu on your device, and then choosing the "About" entry.
- You must install a version of the BlackBerry Desktop Software on the PC that you use. The BlackBerry Desktop Software is usually shipped with most BlackBerry devices. You can also download from http://na.blackberry.com/eng/services/desktop/. Select the 'Download' option. You are presented with a list, select a version of the Desktop software that is applicable to your devices OS. Newer versions of the Desktop Software should support older models of devices. Sky recommends downloading the latest version of the Desktop Software that is available and applicable to your device.
- An installation file for the Sky Technologies BlackBerry software. Sky Technologies provides this, and is normally called v##.##.##.release.blackberry.zip, where ##.##.## is the software version number.

Manual Installation Process

To install the Kony for SAP software on a BlackBerry device, follow these steps:

> ***Note:*** The version of the BlackBerry Desktop Software that appears here is v4.2. These screens are different on later versions, but these options are still present.

1. Unpack the Sky Technologies software onto the PC. The Kony for SAP software for BlackBerry is provided in the form of a "zip" file. In Windows XP and later versions of Windows, the files inside a "zip" archive can be accessed by simply doubleclicking on the file. If you are running an earlier version of Windows, you may need to use a tool such as WinZip to unpack the archive. The files inside the "zip" archive can be extracted into any directory you choose; the location is not important.

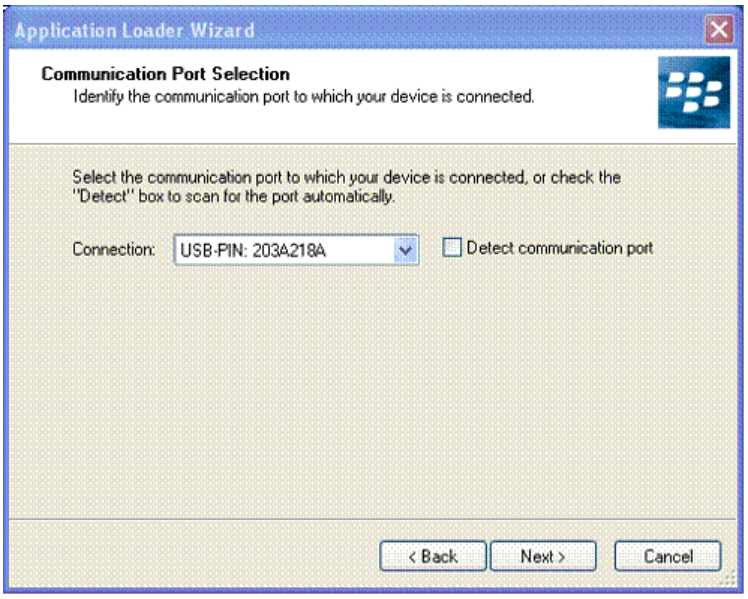
2. Connect the BlackBerry device to the PC. In most cases this is through a USB connection. However, connection through a serial cable is also possible.

3. Launch the BlackBerry Desktop Software. The simplest way to do this is from the Windows start menu. The BlackBerry Desktop Software is normally listed under Start > Programs > BlackBerry > Desktop Manager.

4. Double-click the "Application Loader" icon shown in the BlackBerry Desktop Manager to invoke the BlackBerry Application Loader.

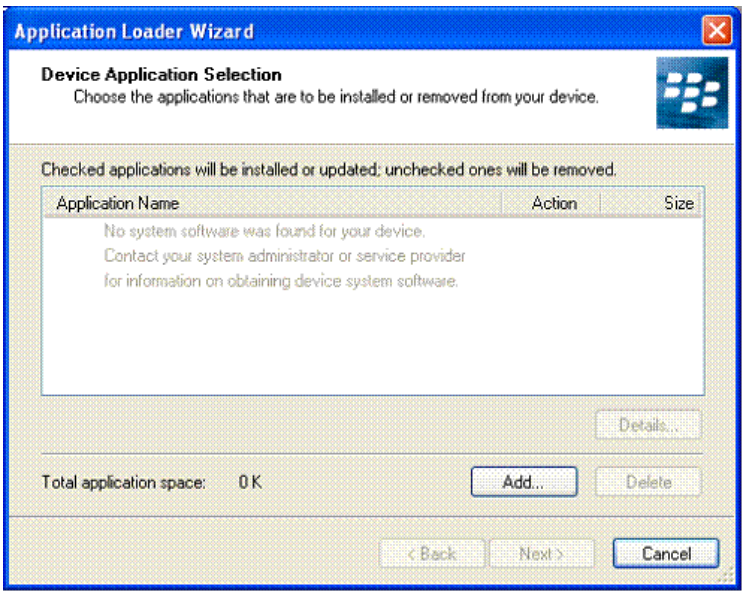
5. Examine the software currently installed on the BlackBerry device. The BlackBerry Application Loader Wizard performs this step after navigation past the initial introductory screens.

6. Click **Next**.

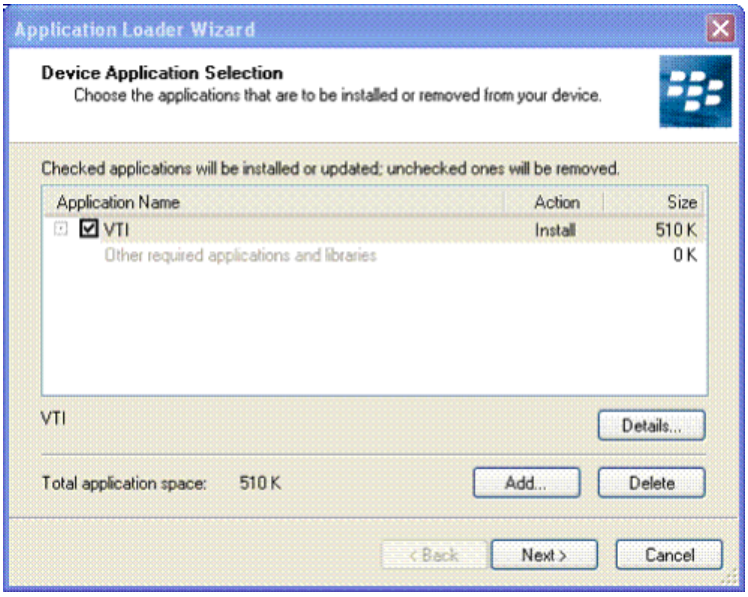


7. Click **Next**.

8. Click **Add** to add the SkyMobile software to the application list.

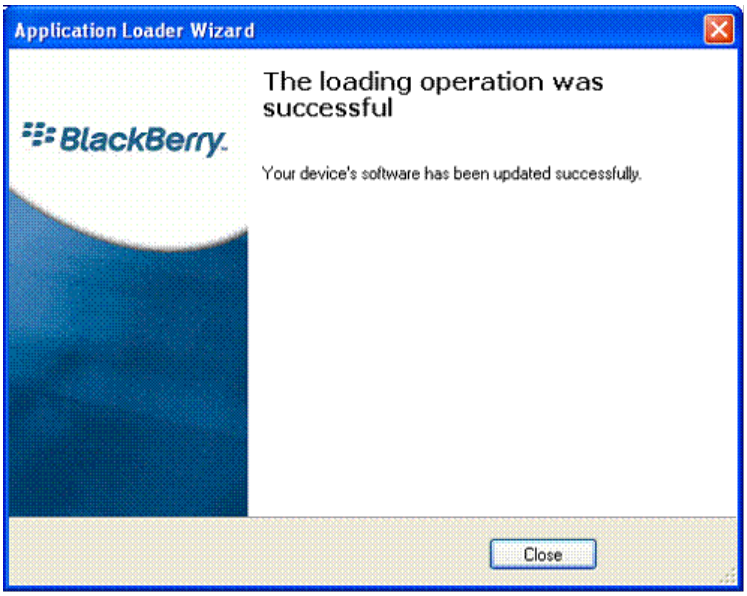


9. Check that the SkyMobile application is now listed. Your screen should look like the one shown below. If everything looks correct, click **Next**.



10. Complete the installation. Click **Finish**. The BlackBerry Application Loader installs the software onto your BlackBerry device.

11. The following screen appears at the completion of the installation process. Click **Close** to shut down the BlackBerry Application Loader.



At this point, the Kony for SAP software is fully installed on the device and is now ready to configure for use. You may close the BlackBerry Desktop Manager window and decouple the BlackBerry device from the PC.

> *Important:* It is noticed on BlackBerry devices running newer operating systems, the devices may not automatically restart after installation. If this is the case, ensure that you switch off the device, and then switch it back on. Certain post install operations are designed to run when the device starts up and the need to restart the device.

Testing the Connection

By default, the Blackberry client is installed with no configuration and requires the device to provision to gain access to an application. Click the Sky icon on the applications menu to test the functionality that presents you with the default provisioning process.

3.9.5.4 Wireless Deployment

As opposed to manually installing applications from a PC using the desktop manager, you may deploy the applications "Over the Air" (OTA) using the BlackBerry Enterprise Server (BES). This is the preferred method to support changes to remote devices once they are in the field.

Requirements

The administrative staff of the BlackBerry infrastructure within an organization is responsible for Wireless application deployment to BlackBerry devices. This section therefore assumes that the reader plays such a role, and has a reasonable working knowledge of systems management in a Windows environment.

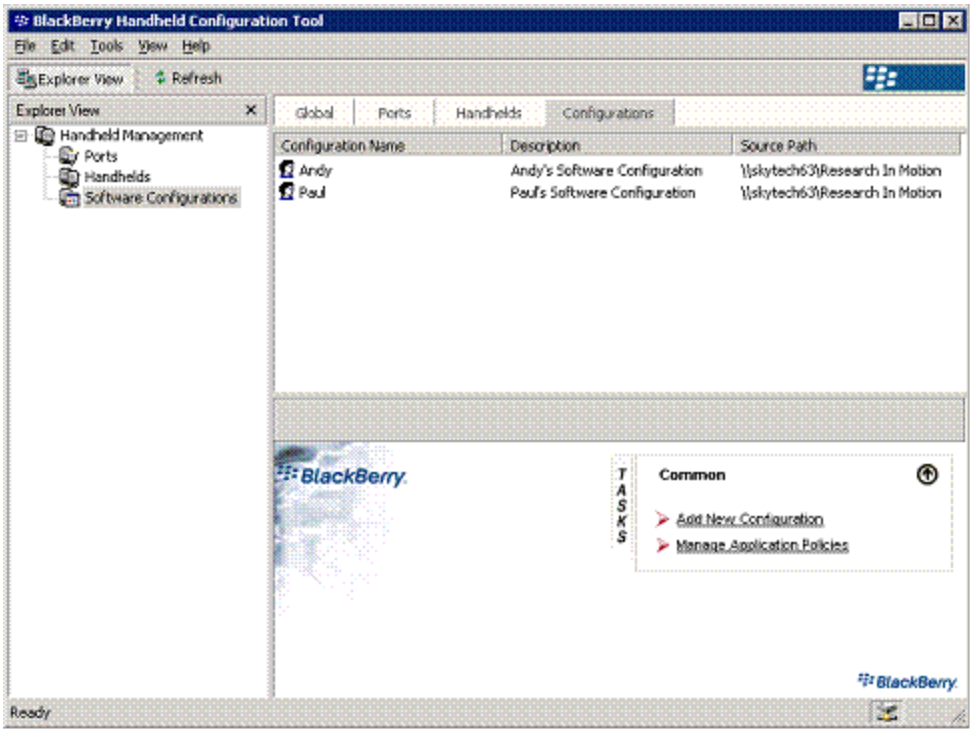
In order to facilitate the wireless deployment, ensure that you have the following.

- Your organization uses a BlackBerry Enterprise Server (BES) that is of version 4.0 or later (as the wireless application deployment feature is not available in earlier releases of the BES).
- Full access to the configuration tools associated with the BES, particularly the BlackBerry Handheld Configuration Tool.
- An installation file for the SkyMobile BlackBerry software. The Sky Technologies provides this, and is normally called v##.##.##.release.blackberry.zip, where ##.##.## is the software version number.
- One or more BlackBerry handsets running version 4.2.1 or later of the BlackBerry OS that you should link to the BES used for deployment.

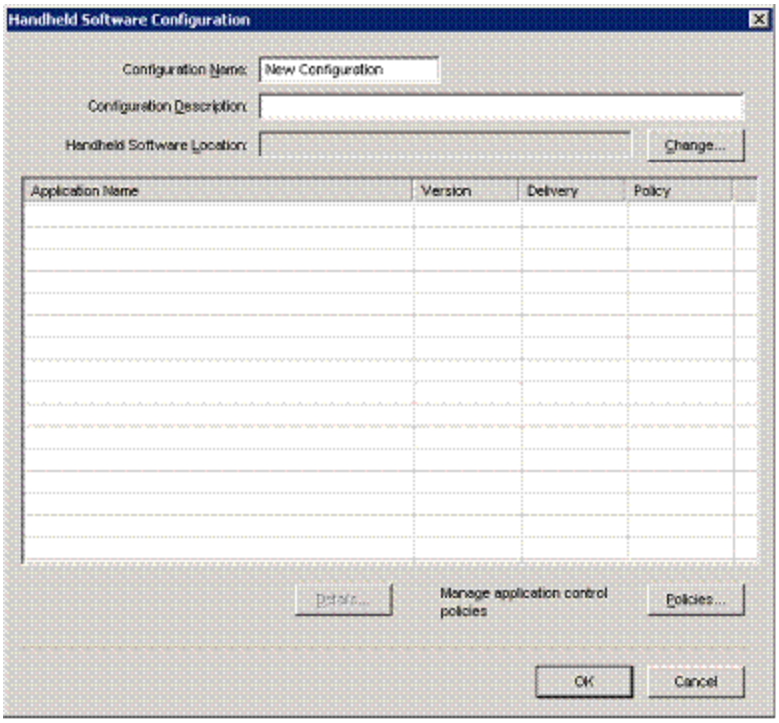
Wireless Deployment Process

Wireless deployment of the SkyMobile software for BlackBerry devices through the BES requires the following steps.

1. Invoke the BlackBerry Handheld Configuration Tool, and identify (or create) the target software configuration(s). The BlackBerry Handheld Configuration Tool is found on the machine on which the BES is installed on. You should be able to list the software configurations currently defined by selecting the **Software Configurations** entry in the left-hand pane of the window.



   If your organization has pre-existing software configurations present, identify to include which (if any) of these SkyMobile software. Take careful note of the UNC "source path" associated with each one.
   If your organization does not have any pre-existing software configurations present, you have to create one to receive the SkyMobile software. Click the "Add New Configuration" link in the lower right-hand pane of the window to achieve this.



Select an appropriate configuration name and configuration description. You must then enter a UNC directory path to the handheld software location. This is usually a shared directory that maps to `C:\Program Files\Common Files\Research In Motion` on the target server. Note this UNC directory name carefully, once you enter.

2. Unpack the Sky Technologies software onto the PC. The SkyMobile BlackBerry software is provided in the form of a "zip" file. In Windows XP and later versions of Windows, double-click the file to access the files inside a "zip" archive. If you are running an earlier version of Windows, you may need to use a tool such as WinZip to unpack the archive. You can extract the files inside the "zip" archive into any directory you choose; the location is not important.

3. Copy the relevant Sky Technologies files into the correct directory location(s) for your BES. The files you need to copy are located in the directory to which you unpack the SkyMobile BlackBerry software. The five files you require are `vti.alx`, `vti.cod`, `vti_lib.cod`, `vti_sdk.cod`, and `vti_sof.cod`.

   You should copy these five files into a specific directory structure set up beneath the UNC

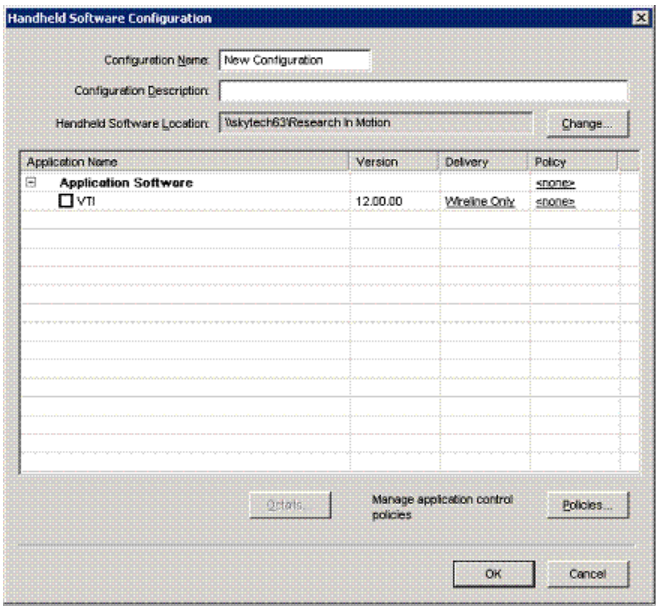
directory (or directories) identified in step 1. For each such directory, you require a path to a lowerlevel directory called `<UNC directory>\Shared\Applications\SkyMobile for BlackBerry`. Create any directories that do not already exist in order to ensure that this path is valid. Copy the `vti.alx`, `vti.cod`, `vti_lib.cod`, `vti_sdk.cod`, and `vti_sof.cod` files into the `SkyMobile for BlackBerry` directory at the bottom of this directory hierarchy. (Incidentally, the `SkyMobile for BlackBerry` directory is not mandatory, and you can call something different if this is preferred).

4. Rebuild the application loader software index. You should perform this step once for each UNC directory identified in step 1. Open a DOS command window and navigate to the following location `<UNC directory>\AppLoader`. Once there, enter the following command:

   loader.exe /reindex

   This causes the application software index to be rebuilt and makes the BES aware that the SkyMobile software is now potentially available for wireless deployment.

5. Invoke the BlackBerry Handheld Configuration Tool, and add the SkyMobile software to the target software configuration(s). You can view the SkyMobile software in the list of available software.

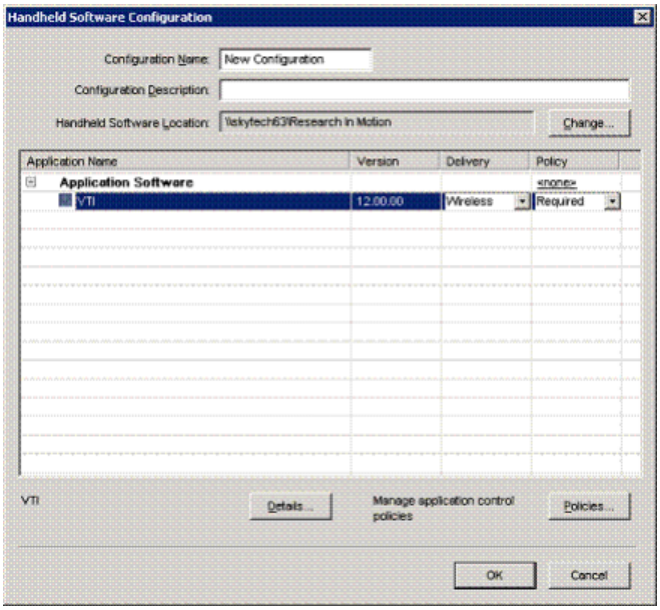


You need to make the following changes to ensure that the SkyMobile application is correctly configured for wireless deployment:

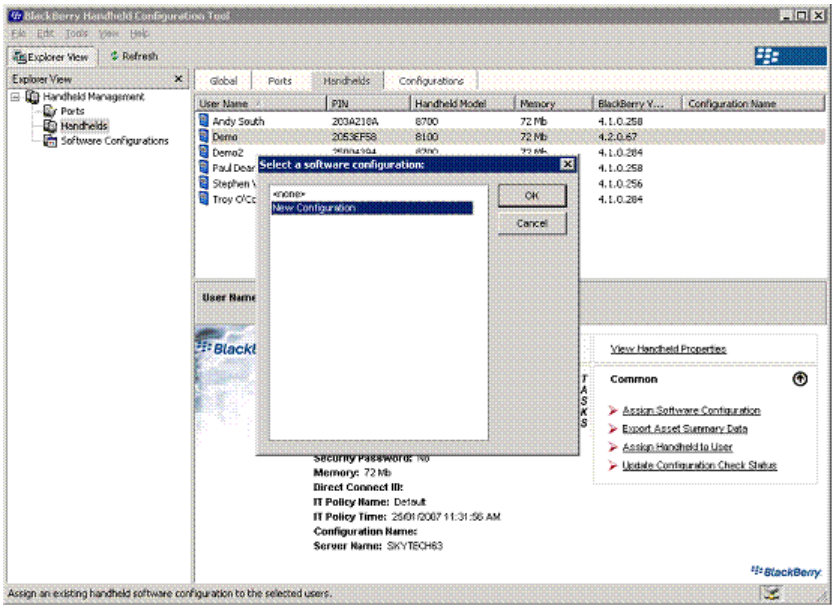
- Select the tick box to the left of the SkyMobile application name.
- The delivery mode for the SkyMobile application should be set to "Wireless" rather than "Wireline Only".
- The SkyMobile application needs to be linked to a policy with a disposition of "Required". If you don't already have an appropriate policy set up, you can create one by clicking the **Policies** button towards the lower right of the screen. Create a policy called **Required** and set its disposition to **Required**. Leave all other fields unchanged and then save the policy.

  Once you are finish, the SkyMobile entry in the software configuration should look like

this:



6. Invoke the BlackBerry Handheld Configuration Tool, and associate the software configuration (s) containing the SkyMobile software with the handhelds to be updated. On entering the BlackBerry Handheld Configuration Tool, select **Handhelds** from the lefthand pane of the window. Select the handheld(s) to be updated, and then click **Assign Software Configuration** button in the lower righthand portion of the window. Assign the software configuration containing the SkyMobile software to the handheld(s), as below:



7. Wait for the BES to deploy the SkyMobile software out to the targeted handheld(s). This can take some time, because a typical BES is normally configured to deploy out new applications every four hours.

   It is possible to decrease the waiting time involved by using regedit to add the following DWORD registry key to the machine upon which the BES is installed:

   HKLM\SOFTWARE\Research In Motion\BlackBerry IT Admin Server\AppProvisioningPollInterval

   The DWORD value is the number of minutes between polling intervals – for example, a DWORD value of 15 means that new applications are pushed out once every 15 minutes. Note that setting a too low value here has a significant performance impact if the BES is managing a lot of handsets.

   After you change the application provisioning poll interval, you should stop the BlackBerry Policy Service and restart (through BlackBerry Manager, or the Services control panel in Windows) in order to pick up the change.

Troubleshooting Wireless Deployment

If you experience difficulties with wireless deployment of the SkyMobile application out to a handset, some things you can do to help resolve the problem are as follows.

- Ensure that the BlackBerry Policy Service is currently running on the machine where the BES is installed (through BlackBerry Manager, or the Services control panel in Windows).
- Check the Windows Application event log to see whether there are any relevant messages from the BlackBerry Policy Service.
- Check the BES logs. The BlackBerry Policy Service is of most interest in this context. It writes to the "POLC" log.
- Ensure that the BES "understands" the handset to which the deployment is taking place. The BES may not necessarily know how to handle handset models that come out after it was put in place. There is a file called `device.xml` that resides in the application loader directory: `<UNC directory>\AppLoader`. This file contains an entry for each type of handset understood by the BES. If you are attempt to deploy to a handset that is not listed in this file, the deployment does not succeed. In such a case, you should contact Research In Motion and ask for assistance in modifying this file in such a way that you can handle the handset in question properly.

3.9.5.5 Blackberry Simulator Installation

Requirements

Before commencing the installation, ensure that you have the following:

- A PC running Microsoft Windows 2000, 2003, XP or Vista.
- The Sun Java runtime (version 1.5 or later of the J2SE runtime) installed on the PC that you use. This is necessary to ensure that the RIM BlackBerry Device Simulator software runs correctly. You can download from http://www.java.com.
- Version 4.2.1 (or later) of the RIM BlackBerry Device Simulators installed on the PC you are using. You can download from http://na.blackberry.com/eng/developers/resources/simulators.jsp. Select the option for downloading a BlackBerry Device Simulator. This should take you to another screen, wherein you can select a product. Select any option for downloading providing the operating system version is at minimum v4.2.1 or greater, that is, 4.2.1 through 6. Sky recommends downloading the device simulators that correspond to the devices you wish to deploy or test SkyMobile upon. You have to enter personal details on the blackberry website, after which you are given an option to save the file to your computer. Once that is completed, execute and complete the installation for the device simulator.
- An installation file for the Sky Technologies software. The Sky Technologies provides this, and is normally called `v##.##.release.exe`, where ##.## is the software version number.

Installation Process

To install the Kony for SAP software on the BlackBerry device simulator, follow these steps:

1. Unpack the Sky Technologies software onto the PC. The SkyMobile BlackBerry software is provided in the form of a "zip" file. In Windows XP and later versions of Windows, you can access the files inside a "zip" archive by simply double-clicking the file. If you are running an earlier version of Windows, you may need to use a tool such as WinZip to unpack the archive. You can extract the files inside the "zip" archive into any directory you choose; the location is not

important. The archive contains folders applicable to different versions of the BlackBerry OS. Each of these folders contains the five files; `vti.cod`, `vti_bpc.cod`, `vti_lib.cod`, `vti_sdk.cod`, and `vti_sof.cod` concerned with installation.

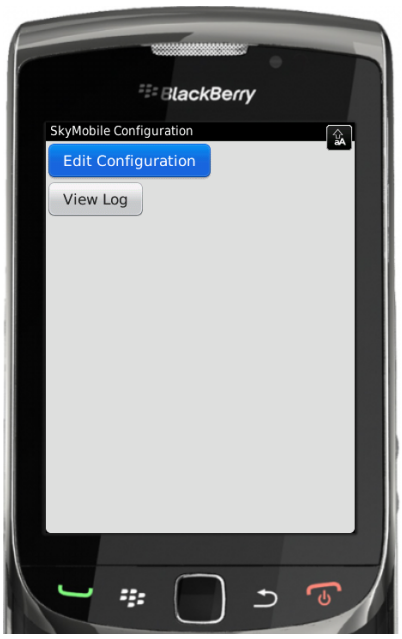
2. Launch your chosen BlackBerry device simulator. Once this is completed, launch the simulator, from the Windows Start Menu, under `Research In Motion\BlackBerry Device Simulators xxx\`. Once the simulator completes initializing; on the simulator menu, select the option to 'Load Java Program' that is found under the **File** menu option. Navigate to the directory where you unpacked the SkyMobile software. Under this folder navigate to the folder that is lower or equal to your simulators version of the OS, that is, if your simulator has 4.7 installed, use application files within the `460` folder. If your simulator has OS 4.5, then navigate to the `421` folder. Select the five cod files and click Open. Wait till the simulator again displays the home screen. You should now be able to select the Sky icon on the Applications screen.

> ***Note:*** On the BlackBerry Device Simulator v4.6, the Sky icon is sometimes located on the Application Screen > Downloads.

3.9.5.6 Configuring the Software

The SkyMobile BlackBerry software is configured upon the BlackBerry device itself (or within the simulator, if a simulator is being used). In order to do so, first select the **Options** item from the master menu, then **Third Party Applications**. Assuming your software installation is successful, you should be able to see an item entitled "SkyMobile".

Select this item and you are presented with the option to access SkyMobile configuration or view the application server logs. Choose **Edit Configuration**.



You are now presented with the main configuration screen for the SkyMobile BlackBerry software.

From this point, you have access to edit the client, server and general configuration options.

> ***Note:*** From the SkyMobile options screen, you have access to purge any stored data currently held by the SkyMobile server and/or reset the SkyMobile configuration to its original state.

How to Set Up a Direct Connection on your Device

This section describes the Connection Mode option "Direct" that is specified in the Server configuration options under Relay Connections and the Access Gateway.

To ensure that SkyMobile connects to the internet "directly" and not through the "BES/MDS", follow these steps:

1. Upgrade your BlackBerry OS.
   You need to have a BlackBerry Operating system version 4.2 or later. If this is not the case, you should install it on your BlackBerry first.

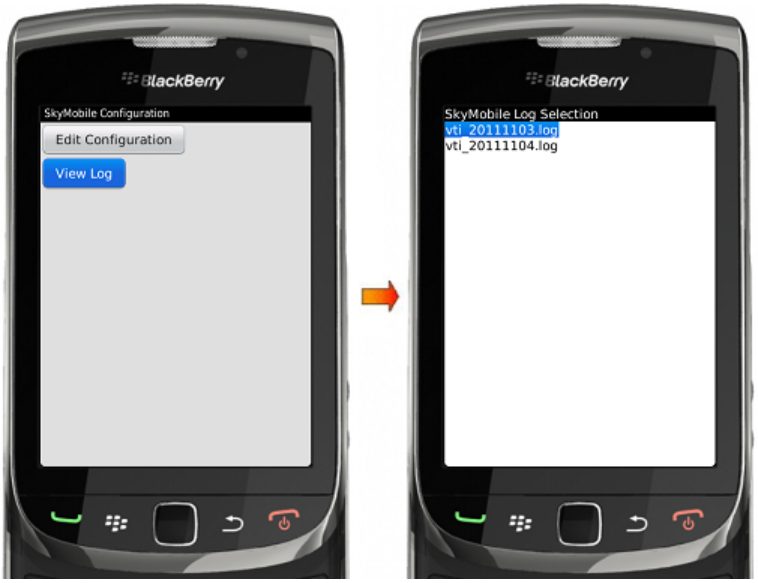
2. Configure your BlackBerry and Your Carrier
   On the BlackBerry handheld, go to **Options** > **Advanced Options** > **TCP**.
   There is a new option in BlackBerry OS 3.8 and later. Here, you have to fill in details regarding the Access Point Name (APN). Enter details that are appropriate for your carrier. If no username and password is provided, leave these blank. Also, ignore the Gateway IP address, unless your BlackBerry device specifically has a textbox to enter it in. Once you entered the APN information, save it and then reboot your BlackBerry device.

3. Run your BlackBerry Internet Software
   Once you completed the above two steps, you can now connect directly to the Access Gateway and/or the relay port, without having to go through the BES/MDS.

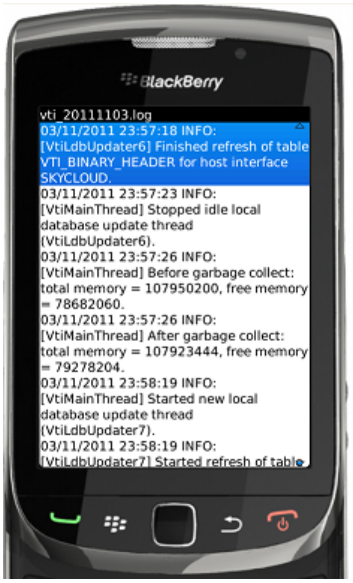
Viewing the Application Server Log File

If a local Application Server is enabled, it creates a log file each time it runs. The log file contains information about potentially noteworthy events that take place in the background while the SkyMobile application is in use. It often also contains further information about any errors that may occur.

In order to view the log file of the local SkyMobile server, bring up the menu on the main configuration screen by clicking the thumbwheel. Then, select the "View Log" option. The SkyMobile Log Selection screen appears.



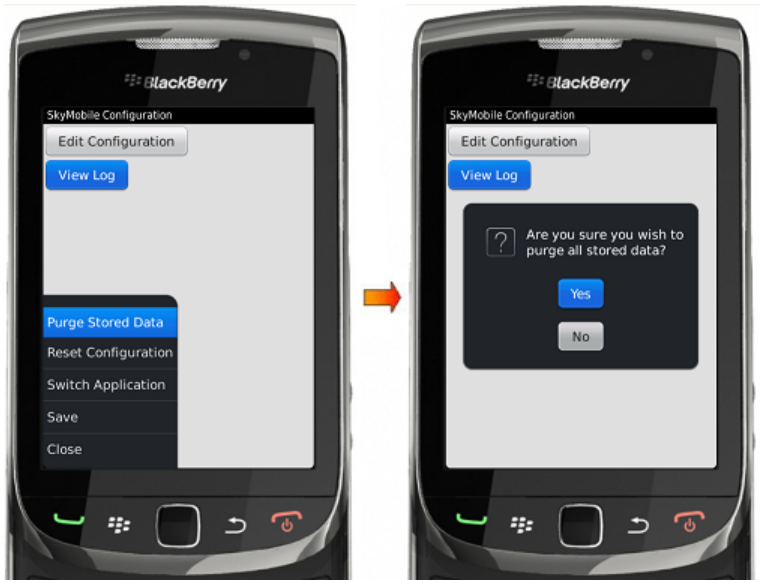
The default log configuration settings for the local SkyMobile server is such that the log file is overwritten each time SkyMobile runs, meaning that only one log file is normally present. However, it is possible to retain a log file history, in which case, many log files can potentially be present. You can delete the unwanted log files from this screen by selecting the "Delete" menu item whilst the relevant log file is highlighted. The relevant log file is immediately deleted. You can select the log files for viewing, by pressing either the enter or space key whilst the relevant item is highlighted. A "Select" menu item is also available that has the same effect. If a log file is selected for viewing, its contents appear.



It is possible to navigate up and down through the log file currently displayed by using the thumbwheel. When you finished viewing the file, you can either press the escape key or select the "Close" menu item.

Purging Stored Data held by the Local Application Server

In some unusual circumstances, it may become necessary to purge stored data held by the local SkyMobile server. The SkyMobile server stores information relating to the definition of the applications it has available, as well as a repository of cached data that is continually synchronized with the back-end host. This information is automatically deleted if the SkyMobile software is removed (uninstalled) from the BlackBerry handheld. It does, however, present an overhead on the storage capacity of the device while the SkyMobile software remains installed. In the event that a purge of the stored data held by the SkyMobile server becomes necessary, it can be initiated from the main configuration screen. Bring up the menu by clicking the thumbwheel, and then select the "Purge Stored Data" option.



A confirmation appears to ensure that a purge of all stored data is in fact required. If confirmed, the purge proceeds and a message indicating its success (or otherwise) appears.

> ***Note:*** A purge of data stored by the local server may result in loss of any locally stored data that is not yet synchronized with the backend host system. Application definitions are also removed from the local cache, and therefore you need to download these from the backend host the next time is runs. As this process can be lengthy (depending on the size of the application), you should undertake a purge of locally held data where absolutely necessary.

Resetting the Configuration back to its Default State

It is possible to reset the SkyMobile configuration settings back to their "default" values at any time. You can achieve this from the main configuration screen, by bringing up the menu by clicking the thumbwheel, and then selecting the "Reset Configuration" option. A confirmation appears to ensure that a configuration reset is in fact required. If confirmed, the configuration reset proceeds and a message indicating its success (or otherwise) appears.

? Unknown Attachment

After invoking this option, all configuration values revert to their default values. If site specific configuration values are pre-loaded (as described in section Pre-loading Configuration Settings, these are reloaded. Otherwise, the "generic" default configuration settings that the SkyMobile software uses are reloaded.

Blackberry Storage Options

Sky recommends using SkyDB on ExternalFileSystem with a class 10 sd-card for maximum performance.

You may also refer to Server configuration for configuration items and applicable values.

You can configure the database on the BlackBerry to operate in one of following three modes.

- Residing in the BlackBerry Persistence Storage within the OS somewhere
- SkyDB residing within some internal memory the device resides (not very reliable)
- SkyDB residing on an external memory card.

For the first option, you should leave the SERVER.STORAGE.STORAGETYPE blank or set to 'BlackBerry' to utilise the internal persistence storage mechanism built into the BlackBerry OS. The SERVER.STORAGE.STORAGEDABASEFILENAME configuration item is ignored when the database operates in this mode.

The remaining two options are utilized only when SERVER.STORAGE.STORAGETYPE is set to SkyDB.

For blackberry devices, the SERVER.STORAGE.STORAGEDABASEFILENAME configuration item does not specify the direct path to the database file as it does with other Operating Systems. Instead it is expecting one of two values for this parameter.

- InternalFileSystem, or
- ExternalFileSystem.

If you specify InternalFileSystem, the application looks for any root file system based on the blackberry internal 'store' memory, and secondly for any root file system based on the blackberry internal 'system' memory. InternalFileSystem is only available on devices that have NOT been locked down by a BES or original firmware disabling this option.

If ExternalFileSystem is specified, the application looks for any root file system based on an 'sdcard', secondly it looks for any root file system based on a 'cfcard', and thirdly it looks for any root file system based on a 'memorystick'. When using ExternalFileSystem, you are recommended to use the fastest micro sd-card, hopefully class 6 or class 10 or whatever you can get your hands on, but faster means less latency in reading, writing and seeking.

> ***Note:*** When the StorageType in use is SkyDB, the physical database filename `vti.db` is always called, no matter what, it is hardcoded that way, you may only choose which root storage mechanism this file resides within.

Each of these operating modes have their pros and cons.

For (1), the database is not encrypted by our code, it is encrypted by the OS, and this encryption is not privy to us. The access to this database is more security through obscurity. The database is only supposed to be accessible through a cod signed by our signatures, but this security mechanism may or may not have already broken.

For (2 and 3), you use the filesystems available to us directly. For InternalFileSystems, the speed is very bad (is a slow NAND chip as far as you know) and the size is VERY limiting. Sky cannot give accurate figures as device memory and processor speeds vary greatly between models, and database records vary depending on the amount of information they try to hold. For ExternalFileSystems, it resides on whatever external memory is available, so in this case, you are restricted by the speed of the memory card (class 10 micro sd-cards if you have a choice in the matter, and the device supports this), or slow compact flash, sony memory or slower class sd cards.

For (2 and 3) the BlowFish, AES and Triple-DES encryptions are offerred. BlowFish is used, by default, if encryption is enabled in the storage config.

As for storage capacities for each of the operating modes,

1. Using the OS persistent storage mechanism, the database can handle approximately <5000 records in a single table before breaking due to memory contraints of the OS.
2. Using SkyDB and InternalFileSystem, the database can handle >5000 records in a single table, but don't go over 10000, due to limited available internal memory in the device.

3. Using SkyDB and ExternalFileSystem, the database can handle record counts up to the available storage capacity of the memory cards. Capacity is less limiting and more user controllable by selecting appropriate memory cards.

> ***Note:*** At present, you are not recommended to use SkyDB on the InternalFileSystem. If capacity is an issue, then you are recommended to use SkyDB on the ExternalFileSystem.

3.9.5.7 Running the Application

Assuming the initial SkyMobile software installation is successful, you should find a "Sky" icon on the master menu of the BlackBerry device.

If the software has not been configured, upon launching the application it allows you to complete the default provisioning process to configure the software.

Once the software is configured, select this item to invoke the application.

From this point, you can begin using your application. If you encounter problems, refer to troubleshooting or contact Sky Technologies support.

3.9.5.8 Java Exits on BlackBerry

As with other versions of the SkyMobile software, it is possible to incorporate custom-built Java code into your application. This permits precise control over the behaviour of an application, and also allows the application functionality to be extended beyond the scope of the SkyMobile application workbench. Because of the nature of the BlackBerry platform, there are some requirements peculiar to this particular environment. The process for creating and installing Java exits suitable for use on a BlackBerry handheld is therefore described below in some detail.

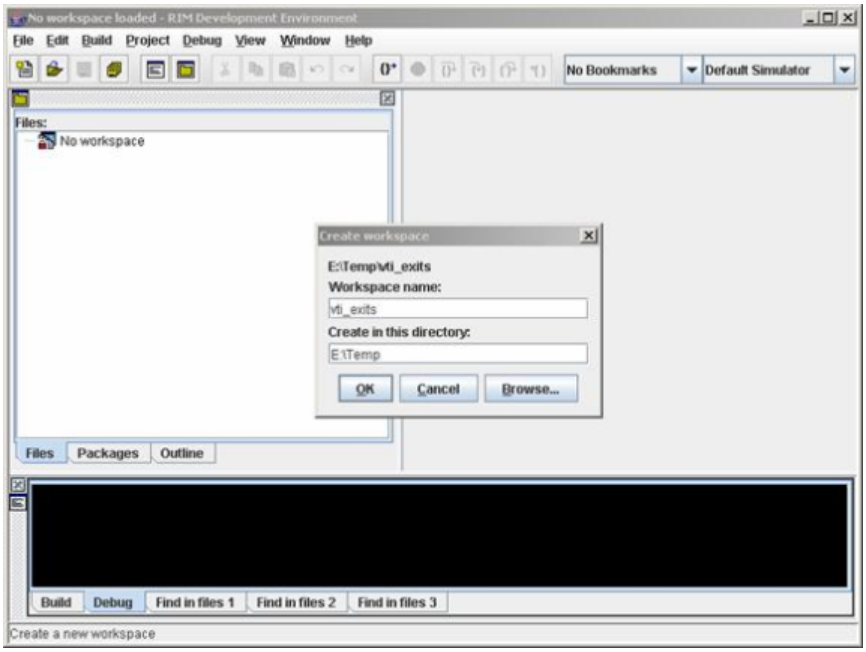
Requirements

To create Java Exits for the BlackBerry platform you require:

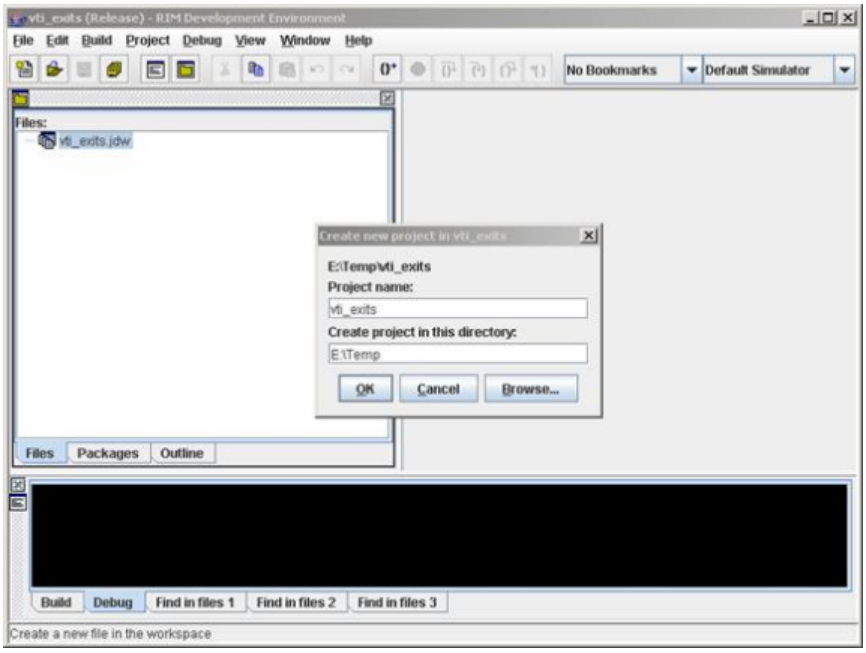
- A PC running Microsoft Windows 2000, 2003, XP or Vista.
- The Sun Java runtime (version 1.5 or later of the J2SE runtime) installed on the PC that you use. This is necessary to ensure that the RIM BlackBerry Java Development Environment (JDE) runs correctly. You can download it from http://www.java.com.
- Version 4.2.1 (or later) of the RIM BlackBerry Java Development Environment (JDE) installed on the PC that you use. You can download it from http://www.blackberry.net.
- The Sky Technologies software installed on the PC that you use, as described in the section addressing installation of software on your BlackBerry handset.
- A BlackBerry developer signing key purchased from Research In Motion and installed on the PC that you use. All VTI Java exits must be properly signed in order to work correctly. This is due to the way in which RIM licenses code to access certain controlled APIs. You can obtain further information on BlackBerry code signing process from the following URL: http://www.blackberry.com/developers/downloads/jde/api.shtml.

Exits Development Process

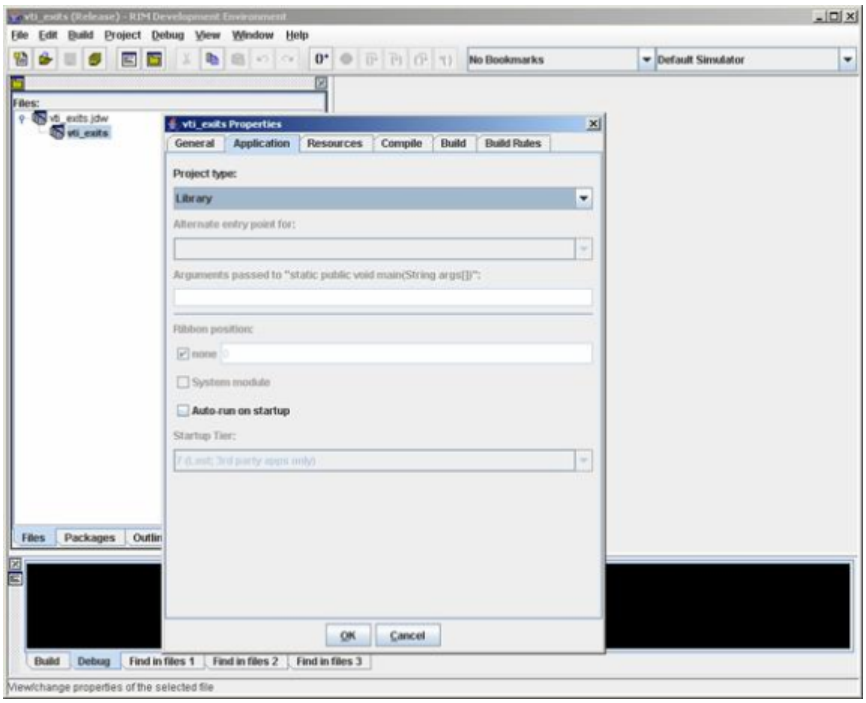
1. Create a new workspace. Start the RIM BlackBerry JDE and select **New Workspace**option from the menu. Choose a name and a directory location for your new workspace.



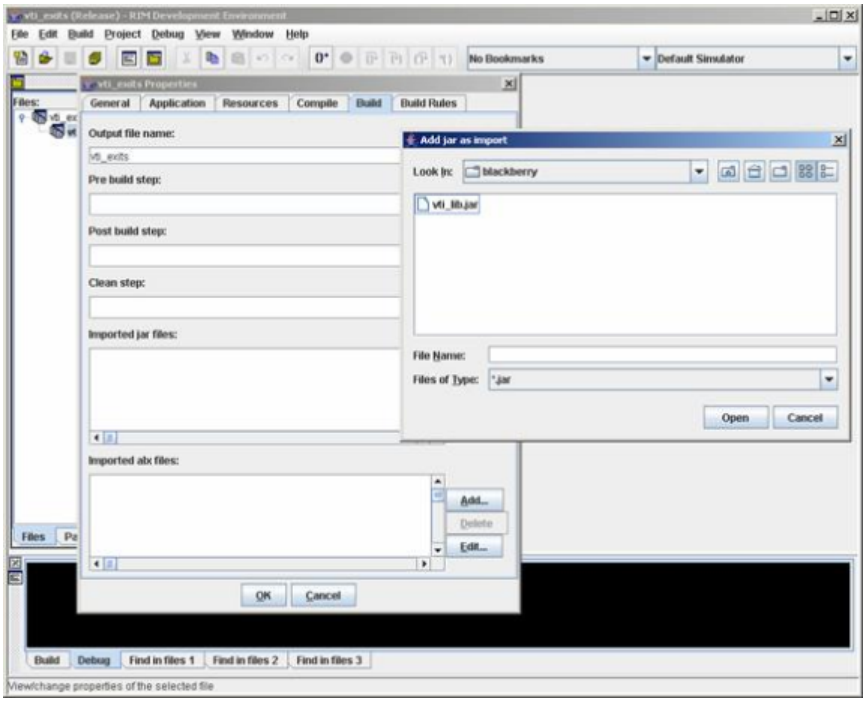
2. Create a new project. Right-click your new workspace and select **Create new Project in <project name>.jdw** option from the menu. Select a name and a directory location for the new project.



3. Configure the project to a library. Right-click the project you just created and select **Properties** from the menu. Select the **Application** tab. Change the **Project Type** field to **Library**.



4. Import the VTI Java classes from the VTI jar files. Still in the properties dialog for the project, select the **Build** tab. Under **Imported Jar Files**, click **Add**. Navigate to the directory in which you can find the Sky Technologies BlackBerry VTI software. Select the **vti_lib.jar** file and choose **Open**. Repeat the process for the `vti_sdk.jar` file. Click **OK** to dismiss the properties dialog.



5. Write the Java code. Use the BlackBerry JDE to develop the Java exits code exactly as you would for any SkyMobile application. The BlackBerry JDE is a fullblown Java development environment and RIM extensively documented it. For further assistance on how to write SkyMobile Java exits, refer to the VTIJava Programming Guide.

6. Compile and sign the Java code. Once you finished writing the Java code, you should cleanly compile and then sign using your developer signing key in conjunction with the RIM signing tool.

> ***Note:*** You must sign theVTI Java exits code in order to work on an actual BlackBerry handheld. You can use unsigned code successfully in the device simulator.

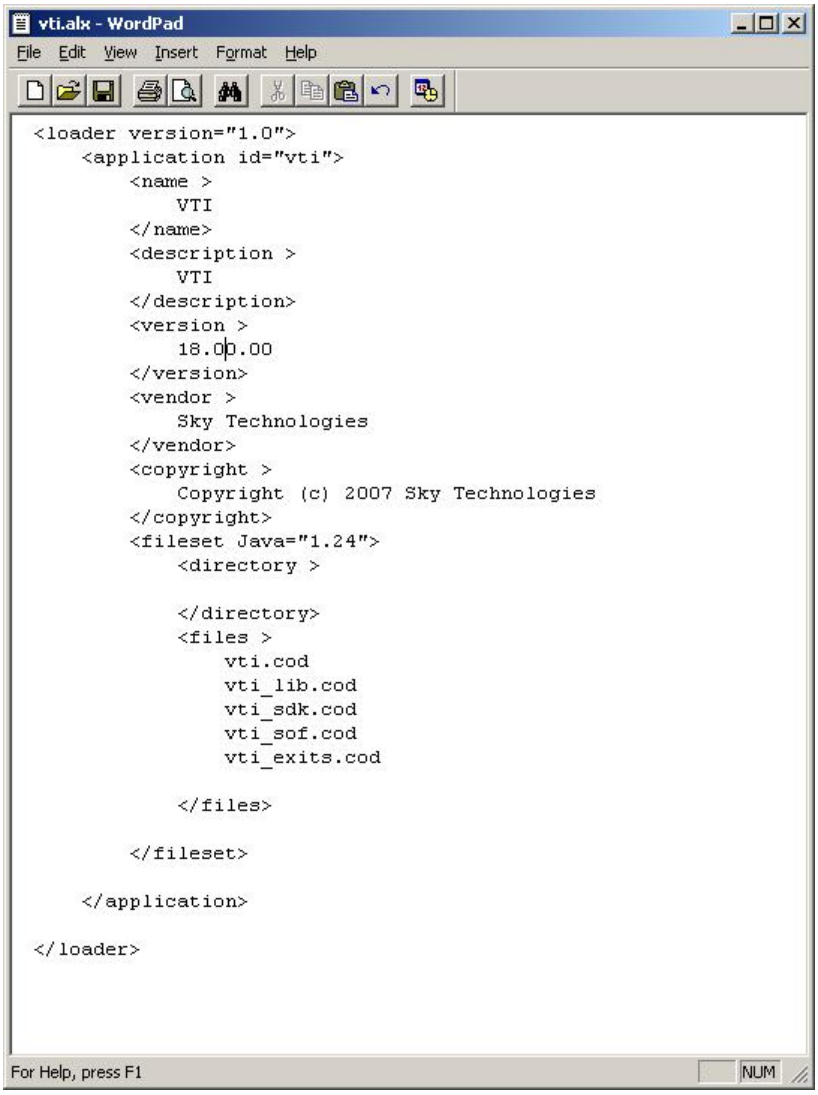
7. Right-click your project and select the **Generate ALX file** option from the menu. However, you really require this step only if you plan to deploy your VTI java exits as a separate application.

   Once you follow all these steps, the VTI Java exits code is ready to install.

Manual Installation of Java Exits

To install VTI Java exits manually, follow these steps:

1. Create a temporary directory.
2. Copy the `vti.alx`, `vti.cod`, `vti_lib.cod`, `vti_sdk.cod`, and `vti_sof.cod` files from their locations within the VTI release directory, into the temporary directory.
3. Copy your VTI exits jar code (that is in a file with a `.cod` suffix), placing it into the temporary directory.
4. Open the `vti.alx` file with Windows WordPad (or another suitable text editor). Edit the contents of the VTI fileset so that it now also contains your VTI exits jar code. For example, if your VTI exits code is contained in a file called `vti_exits.cod`, adding the following line to the `vti.alx` file, allows it to be included in the VTI application.



5. Install SkyMobile in the usual manner, using your modified `.alx` file rather than the original one bundled with the Sky Technologies VTI software.

Wireless Development of Java Exits

Wireless deployment of VTI Java exits is essentially the same as wirelessly deploying the VTI application itself, with the following minor differences:

- You should copy the file containing your VTI Java exits code (that is contained within a file with a `.cod` suffix) into the BES directory where the `vti.alx`, `vti.cod`, `vti_lib.cod`, `vti_sdk.cod`, and `vti_xai.cod` files are located (for example, `<UNC directory\Shared\Applications\SkyMobile for BlackBerry`).
- You should modify the `vti.alx` file to include the SkyMobile Java exits code in the files to deploy. Refer to Manual installation of Java exits for further information on how to achieve this.

Installing Java Exits on the Device Simulator

Installation of VTI Java exits on the BlackBerry device simulator is a very straightforward process. It merely involves copying the `.cod` file associated with your VTI Java exits, into the simulator cache directory, as described in section Manual installation process. No manipulation of `.alx` files is required.

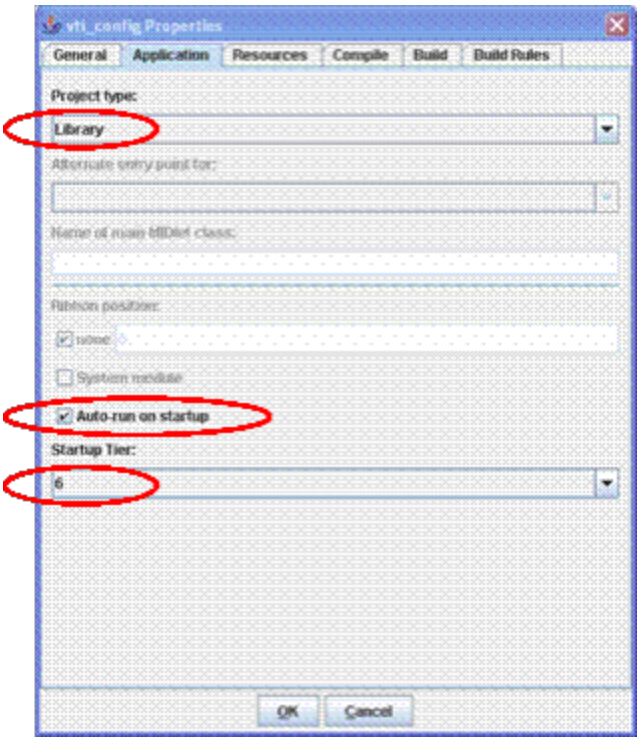
Preloading Configuration Settings

A relatively common requirement when implementing SkyMobile in a BlackBerry environment is the need to pre-configure handsets so that they are immediately ready for use following installation of the SkyMobile application. This removes the need for someone (whether an IT support staff member, or the owner of the handset) to go into the VTI Options and change the configuration settings to those needed for the target environment.

Pre-loading of SkyMobile configuration settings is achieved through a special type of VTI user exit. All the information that pertains to the creation and installation of VTI user exits is also true of the SkyMobile BlackBerry default configuration loaders. There are however, a couple of points worth noting.

1. The project containing the default configuration loader class must be defined as being a "Library".
2. The project must have the "Auto-Run at Startup" option enabled.
3. The default configuration needs to load before the SkyMobile application itself. It therefore should be set to run in Tier 6.

You can set all of these things through the BlackBerry JDE in the properties for the project, as shown:.



You can find some sample SkyMobile BlackBerry default configuration loading code in `VTI samples` directory. The `samples` directory is normally found in a directory called `samples` inside the Sky Technologies BlackBerry software release file. The relevant file is called `BBDefaultConfigLoader.java`. Feel free to use and modify this code as needed.

Deploying BlackBerry COD Files from Web

You can install all of the cod files through a web server in a number of ways. The preferred way to do this is through a `.jad` descriptor file that points your BlackBerry device to the required cod files. The bpc (BlackBerry Presentation Client) is within `vti.cod`. This bpc has two dependencies, the `vti_lib.cod` and `vti_sdk.cod` files. The sample demo application relies upon sof objects, and any other application you develop that relies upon sof objects also requires the `vti_sof.cod` file. So the example here contains four cods, the `vti.cod` for bpc, `vti_sdk.cod` and `vti_lib.cod` as dependencies, and `vti_sof.cod` for the demo application. You may add any number of cods to this `.jad` descriptor file to deploy all of your cods in a single installer. Simply append them to the end of the list of current cods, remembering to set the index to be the next available number,that is -5... -6 and so on for each of the parameters required within the `.jad` descriptor file. RIM-COD-Size-?: is specified as 12345, this is acceptable as the BlackBerry browser installer calculates the actual size of the cods by itself and ignores this parameter, although it needs to be present in the `.jad` descriptor file or it becomes an invalid descriptor, so 12345 is chosen for all cods. The BlackBerry/RIM JDE creates a `.jad` descriptor file for each cod file that it generates, so simply copy and paste the contents of this new `.jad` into the overall installation `.jad` file, update the RIM-COD-Size parameter, delete unnecessary parameters using the jad here as a guide as to what parameters are necessary, and reindex your cod file parameters to the next index. Deploy this new installation `.jad` file into your website along with all referenced cod files in the same folder under your web server, and deploy your bpc and prerequisite cod files through the single URL pointing to your installation `.jad` file.

Following is an example to deploy the demo application and all required cod files from a single `.jad` descriptor file, named `vti.jad`

vti.jad

```
Manifest-Version: 1.0
MIDlet-Name: vti
MIDlet-Version: 20.00.10
MIDlet-Vendor: Sky Technologies

MIDlet-Name-1: vti_lib
```

```
MIDlet-Version-1: 20.00.10
MIDlet-Vendor-1: Sky Technologies
RIM-COD-Size-1: 12345
RIM-COD-URL-1: vti_lib.cod
RIM-COD-Module-Name-1: vti_lib
RIM-COD-Module-Dependencies-1: net_rim_cldc,vti_sdk,net_rim_bbapi_
options,net_rim_os,net_rim_bbapi_menuitem,net_rim_pdap
RIM-Library-Flags-1: 3

MIDlet-Name-2: vti_sdk
MIDlet-Version-2: 20.00.10
MIDlet-Vendor-2: Sky Technologies
RIM-COD-Size-2: 12345
RIM-COD-URL-2: vti_sdk.cod
RIM-COD-Module-Name-2: vti_sdk
RIM-COD-Module-Dependencies-2: net_rim_cldc
RIM-Library-Flags-2: 2

MIDlet-Name-3: vti_sof
MIDlet-Version-3: 20.00.10
MIDlet-Vendor-3: Sky Technologies
RIM-COD-Size-3: 12345
RIM-COD-URL-3: vti_sof.cod
RIM-COD-Module-Name-3: vti_sof
RIM-COD-Module-Dependencies-3: vti_lib,net_rim_cldc,vti_sdk,net_rim_
pdap,net_rim_bbapi_invoke,net_rim_bbapi_browser
RIM-Library-Flags-3: 2

MIDlet-Name-4: vti
MIDlet-Version-4: 20.00.10
MIDlet-Vendor-4: Sky Technologies
RIM-COD-Size-4: 12345
RIM-COD-URL-4: vti.cod
RIM-COD-Module-Name-4: vti
RIM-COD-Module-Dependencies-4: net_rim_cldc,vti_sdk,vti_lib
```

To install a different configuration with your installations, refer to the "Pre-Loading Configuration Settings" section within Installation Guide/Client-Server Platforms/BlackBerry/Pre-Loading Configuration Settings. This details how to create a cod file specific to installing specific configuration settings, and how to deploy as per the process outlined above for adding an extra cod file to your single installation `.jad` descriptor file.

3.9.6 Android Platforms

This section explains the process to implement the Application Server onto an Android phone using the Android SDK toolkit. You may automatically install the Android Server directly from the Android Market app store and provision it without any manual intervention.

> ***Important:*** Android devices have a manufacturer configured maximum application limit. Initial devices are limited to 16MB that is easily reached when using large images and/or tables with images in the rows. Ensure that your devices have 64MB or greater to ensure satisfactory performance. Refer Android Deployment Considerations for details.

3.9.6.1 Installing from the Android Market

You must have a Google Gmail account.

> ***Note:*** The support for java exits and the launching of system commands is removed from the market version of the client due to security concerns. A version of the client with this functionality enabled, is available upon request.

From the Android device:

1. Launch the Android Market application.
2. Select the search icon.
3. Enter "SkyMobile".
4. Select the SkyMobile application by Sky Technologies.
5. Select Install.
6. Accept permissions.

3.9.6.2 Installing Manually using the .apk File

As well as automatically installing the SkyMobile Container from the Android Online Marketplace, you may also install it manually onto your phone or simulator.

Currently two versions of the SkyMobile application for Android are distributed:

1. `XXX.release.android.zip` that contains a 'safe' version of the application that has exits and system commands disabled.
2. `XXX.release.android.full.zip` that contains the full product.

Prerequisites:

Before commencing the installation, ensure that you have the following:

- A device running Android 2.1 or later.
- "Unknown sources" enabled under **Settings** > **Application Settings** on your phone.
- The Android SDK installed on your pc.
- The Android USB drivers (included in the SDK) installed.
- Your Android phone connected to your pc through the USB cable.

To Copy Files to the Android Device SD Card:

1. Ensure that the android usb drivers are installed and connected to your device through the usb cable
2. Expand the notification pane on the device (by wiping your finger downwards from the very top of the screen)
3. Select "USB Connected" and press "Mount"
4. You may now copy files to the SD card using windows explorer.

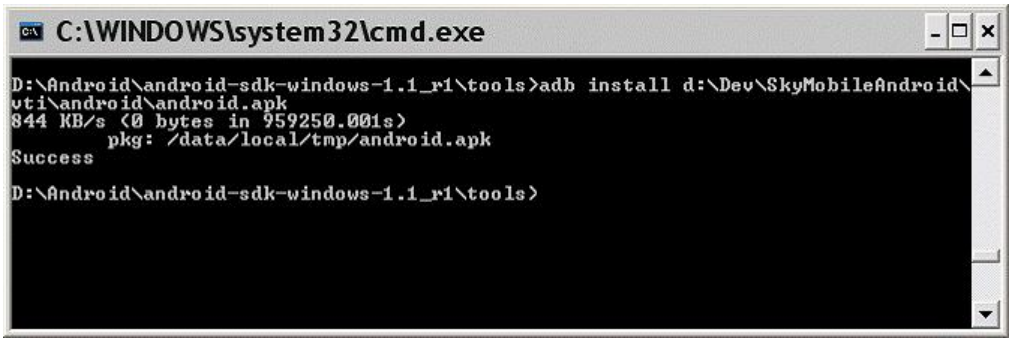
Installation Process

Open a command prompt on your computer and navigate to the android sdk `tools` directory.

To install: type `adb install [path]SkyMobile_APC.apk` (where "[path]" is the path of the SkyMobile apk file) and press enter.

To re-install: type `adb install -r [path]SkyMobile_APC.apk` to reinstall.

To uninstall: type `adb uninstall au.com.skytechnologies.apc`



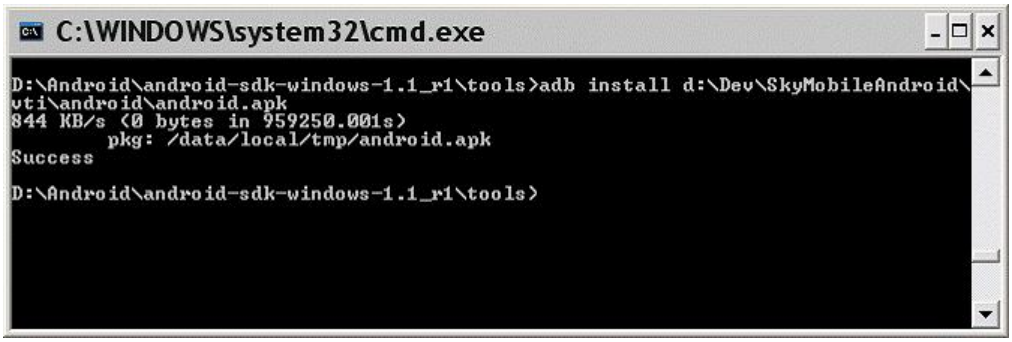
Installation Process

Open a command prompt on your computer and navigate to the android sdk "tools" directory.

To install: type `adb install [path]SkyMobile_APC.apk` (where "[path]" is the path of the SkyMobile apk file) and press enter.

To re-install: type `adb install -r [path]SkyMobile_APC.apk`.

To uninstall: type `adb uninstall au.com.skytechnologies.apc`



3.9.6.3 Launching the Android Client

Once Kony for SAP is installed under Android, three application icons are available under the applications menu:

SkyMobile

When launching SkyMobile for the first time, you are presented with the default provisioning screen. This guides you through downloading and running an application.

An icon appears in the task bar to notify the user that the SkyMobile application server is running. Click this icon to open a pane that allows the user to open or switch back to the client.

Sky Preferences

This application allows you to view the SkyMobile configuration settings. If enabled by your provisioning profile, you are also able to add or change existing settings.

Sky Log

This application allows you to view, send or delete log files that are stored in the protected area of the application. The log files appear in date order with the most recent at the top of the list.

> ***Note:*** To delete any existing data and configuration in order to return to the initial state you can go to **Settings** > **Applications** > **Manage Applications** > **SkyMobile** > **Clear Data**.

> ***Important:*** Applications are installed to a protected directory. In this case `/data/data/au.com.skytechnologies.apc/`. The application stores the files here, however are not accessible or visible by any other application.
>
> You may change the paths where files are stored to the SD card if you have one installed. An example of how you can specify this path is `/sdcard/SkyMobile/`.
>
> To modify the thin client preferences, use the preferences application.

3.9.6.4 Java Exits on Android

To begin, you need to install the following:

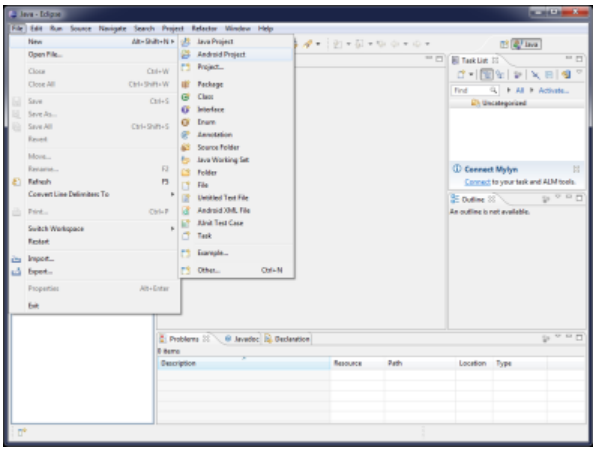
- Install eclipse IDE for Java Developers. http://www.eclipse.org/downloads/
- Install the Android SDK manager: http://developer.android.com/sdk/index.html
- Install the ADT Plugin for Eclipse: http://developer.android.com/sdk/eclipse-adt.html
- Add the following android components: http://developer.android.com/sdk/installing/index.html?pkg=tools
  - SDK Tools,
  - SDK Platform-tools,
  - SDK platform Android 2.1-update1)
- Obtain a copy of the android full release zip file.

The process for installing user exits on an Android device is unfortunately not straight forward. At the time of writing, due to limitations in the Android eclipse plugin, it is required that the `vti.jar` library is added to your project through another project rather than imported as an external library. This procedure guides you through creating the parent SkyMobile project, referencing this from your user exits project, and then installing this onto the device. This `vti.jar` is available as part of the Android "Full" `release.zip`.

1. Create a SkyMobile Project that points to the Android `vti.jar`.
2. Create a Project for your User Exit Code.
3. Export Project and Copy to the Device.

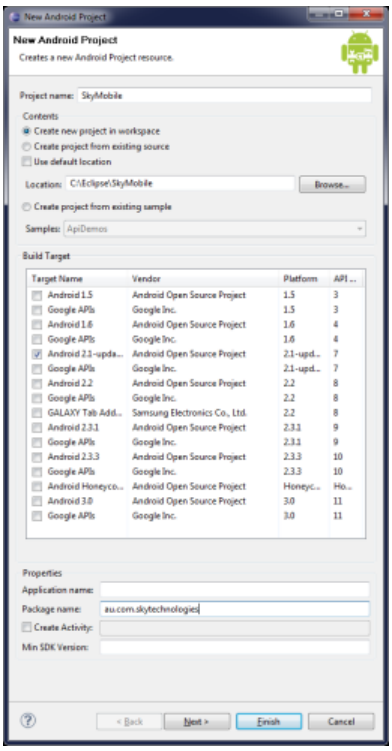
Create the SkyMobile Project

1. Create a new project by selecting **File** > **New** > **Android Project** from the menu.

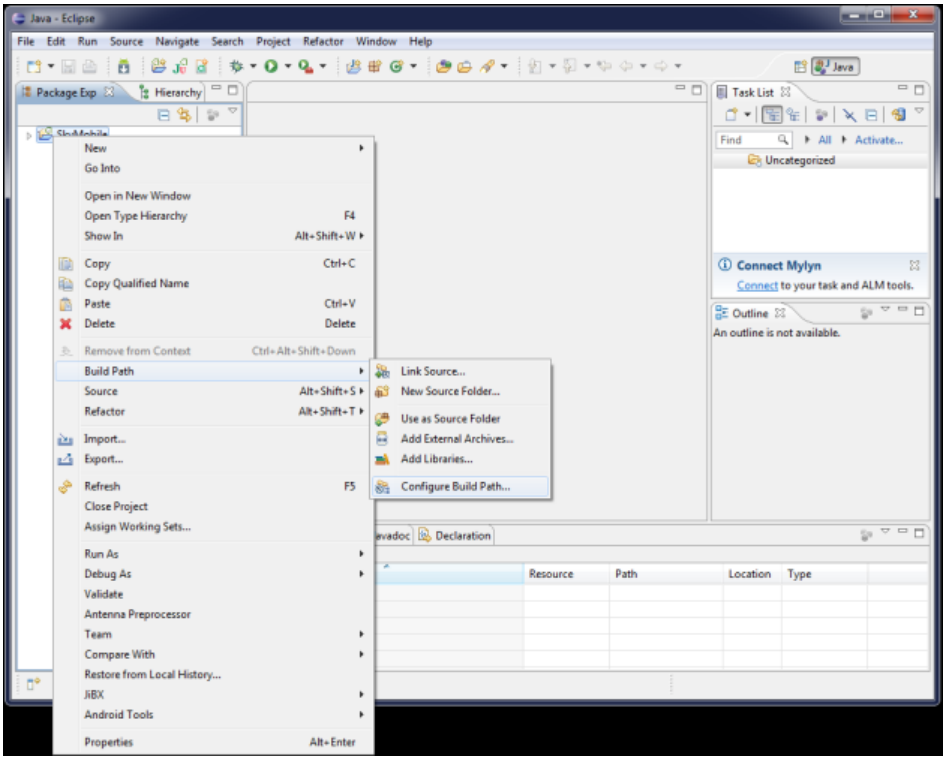


1. Enter the project name "SkyMobile" and define where you like the project files to be stored.
2. Select the base SDK that you are using. Enter the package name `au.com.skytechnologies`.
3. Deselect **Create Activity**.

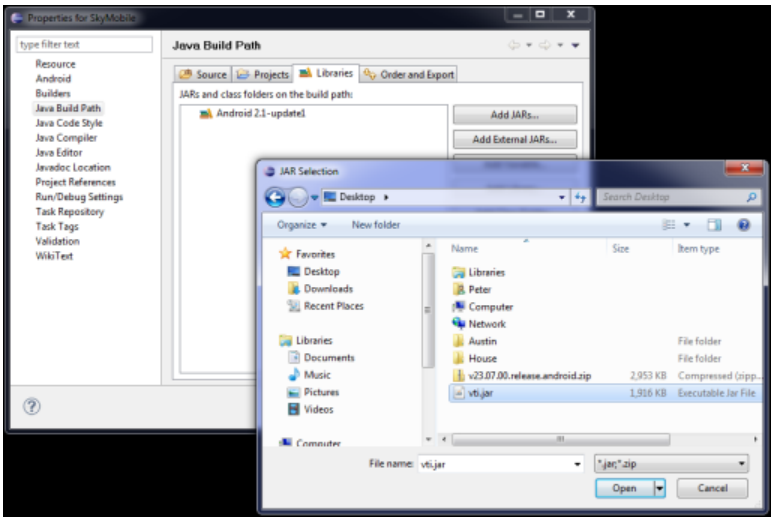
4. Click **Finish**.



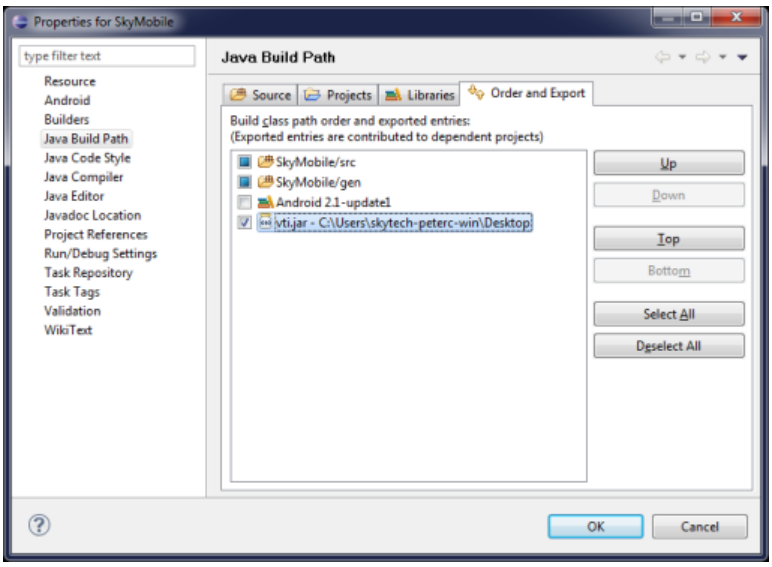
5. Right-click the name of the project you just created, then select **Build Path** > **Configure Build Path** from the menu.



6. Select the **Libraries** tab. Click **Add External JARs** and select the copy of the Android `vti.jar` file. (This is the file distributed alongside the `.apk` file in the android release zip file.)



7. Click the **Order and Export** tab and check the box on the left of **vti.jar**.

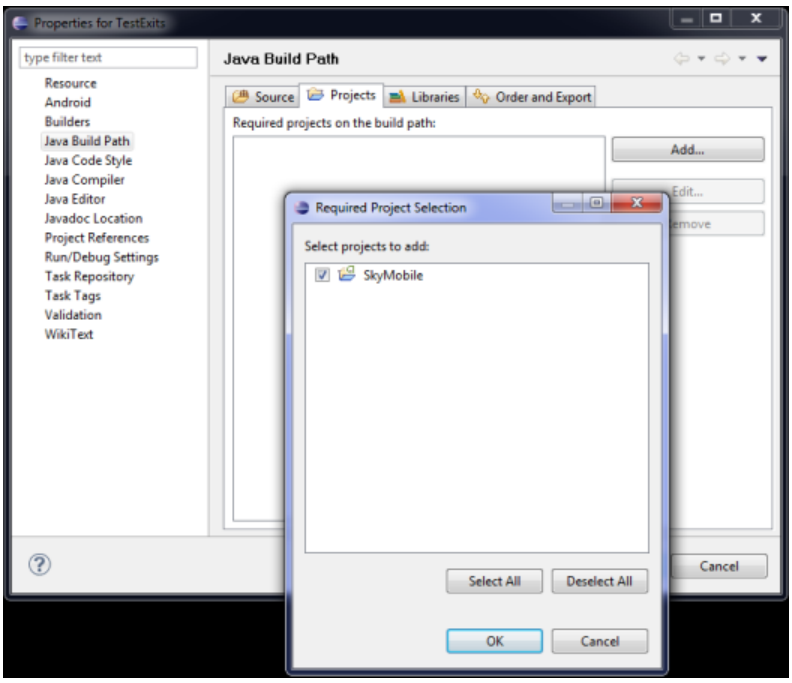


8. Press **OK** to close the window.

Create your User Exits Project

1. Select **File** > **New** > **Android Project** from the menu to create a new project.
2. Enter your desired project name and define where you want to store the project files. Select the base SDK that you are using.
3. Enter the package name `au.com.skytechnologies`.
4. Deselect **Create Activity**.
5. Click **Finish**.
6. Right-click the name of the project you just created, then select **Build Path** > **Configure Build Path**from the menu.

7. Click the **Projects** tab, click **Add** and check the box to the left of the **SkyMobile** project.

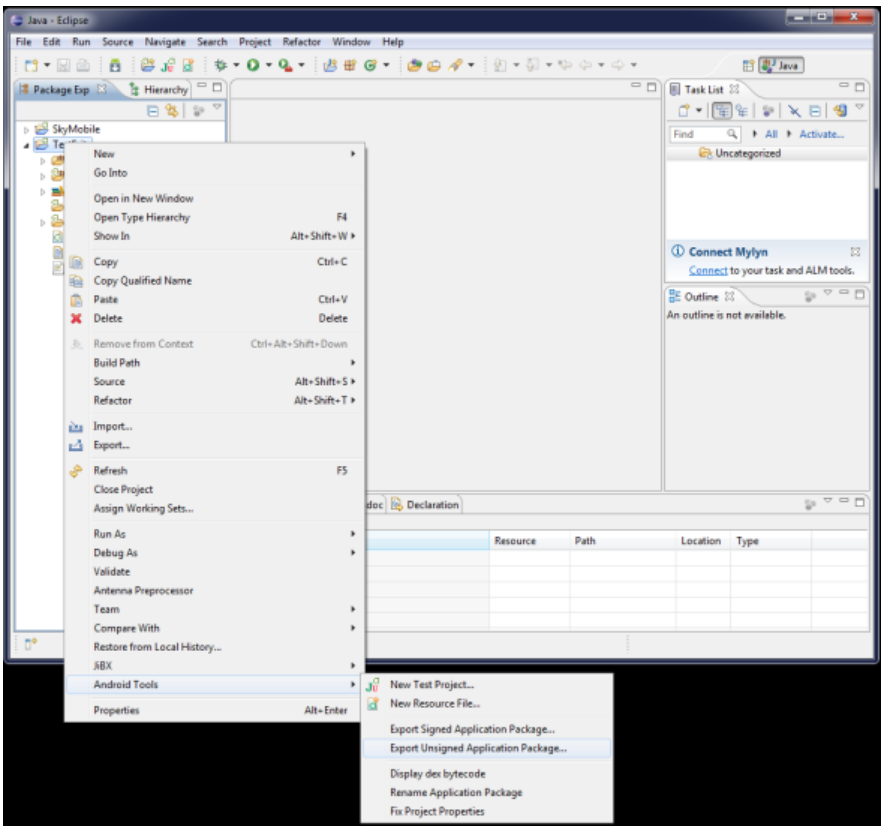


8. Click **OK** to close the window, and you are now ready to add whatever classes you want into your project.

Copy your User Exits to the Device

1. Right-click your user exits project and select **Android Tools** > **Export Unsigned Application Package**.

2. Name and save this file to an appropriate location.

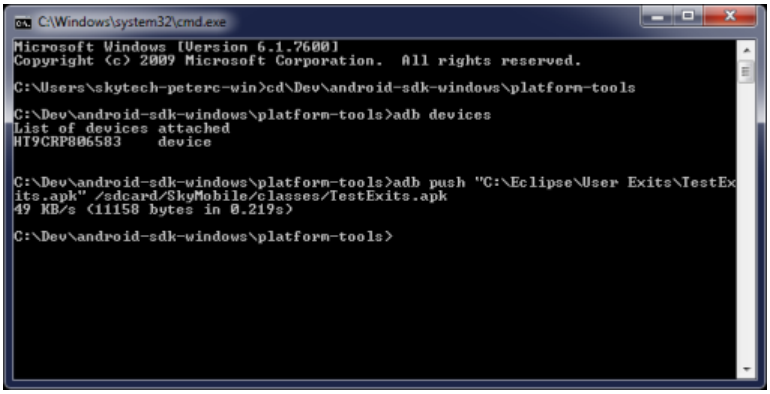


3. Open a command prompt and change directory to your android SDK's platform-tools directory.

4. Connect your device, or start the emulator and type 'adb devices' to ensure that one device appears.

5. Copy your user exits file onto the device/emulator using the command 'adb push <source> <target>'

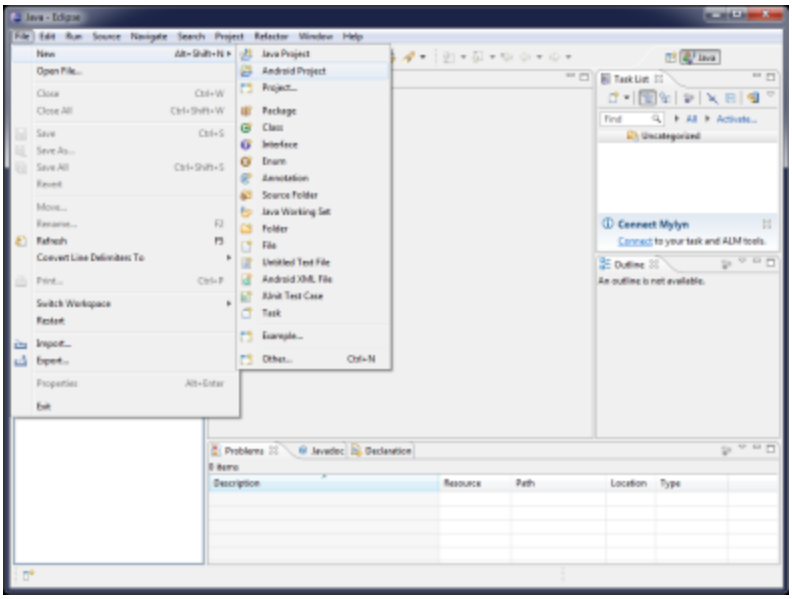
6. You are only allowed to write to the sdcard on the device. This can be referenced through `/sdcard/`.



7. Configure the Android client classes directory to point to the location of your exits `.apk` file. You can do through the SkyPreferences application, or through provisioning using the CLIENT.SERVER.CLASSESDIRECTORY parameter.

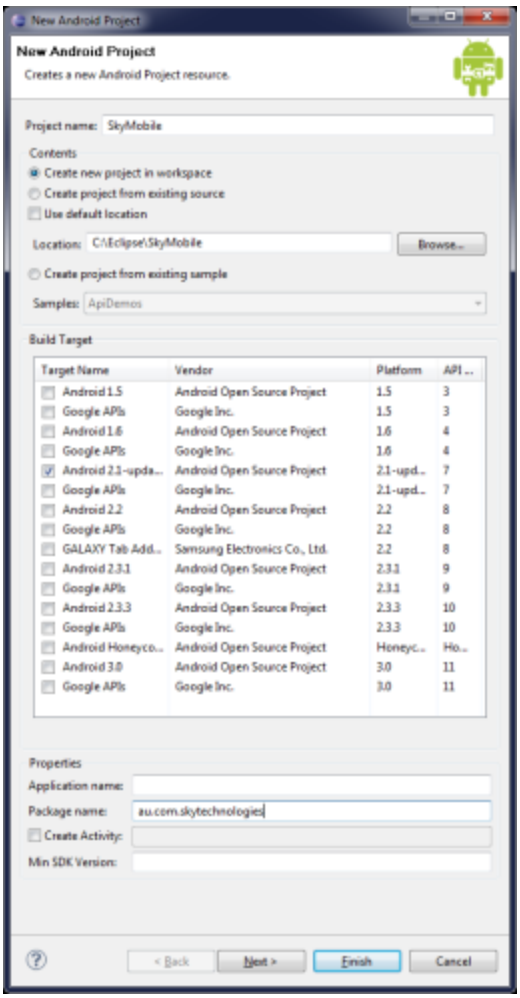
Creating the SkyMobile Project

1. Create a new project by selecting **File** > **New** > **Android Project** from the menu.

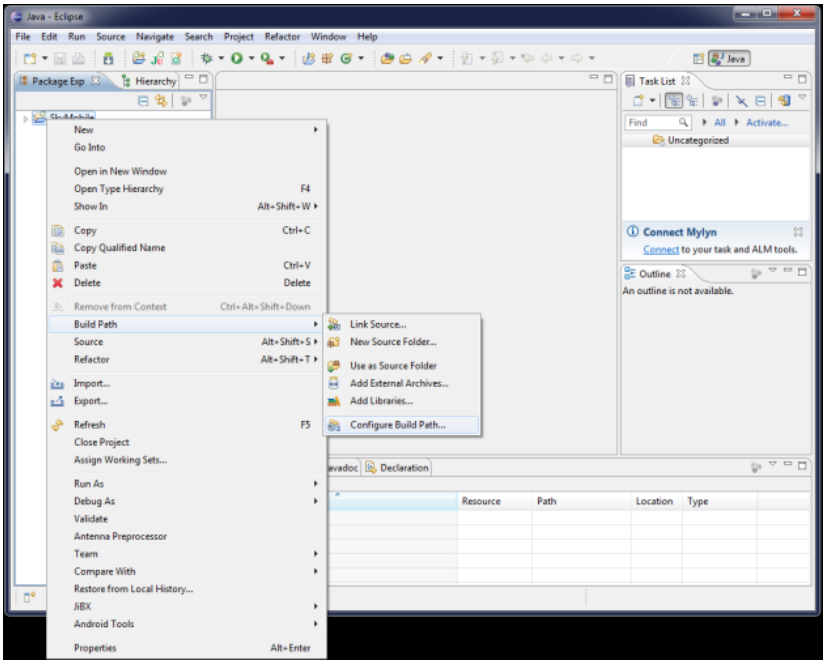


2. Enter the project name "SkyMobile" and define where you would like the project files to be stored.
3. Select the base SDK that you are using. Enter the package name "au.com.skytechnologies"
4. Clear the **Create Activity** check box.

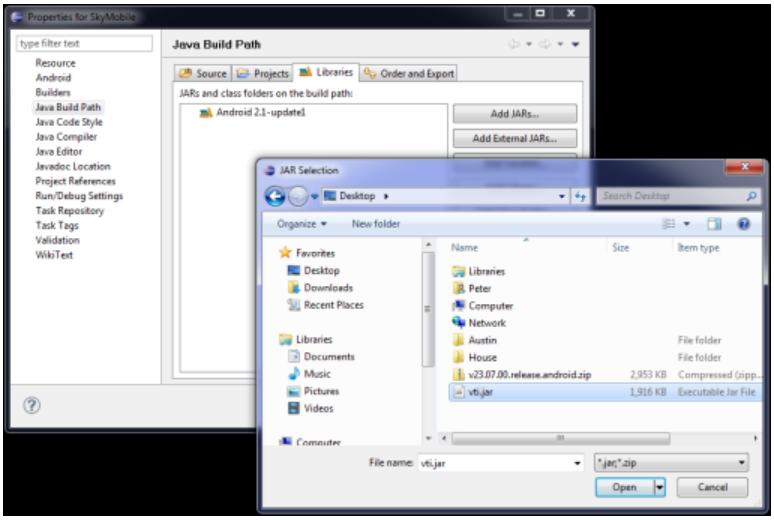
5. Click **Finish**.



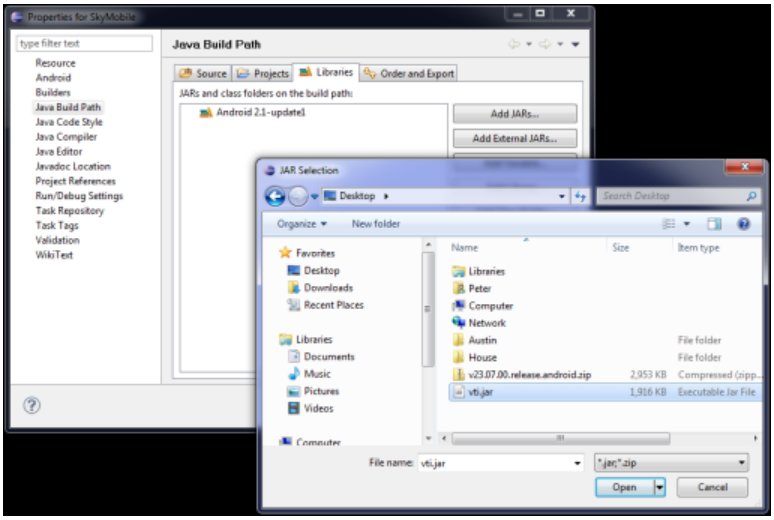
6. Right-click the name of the project you just created, then select **Build Path** > **Configure Build Path**... from the menu.



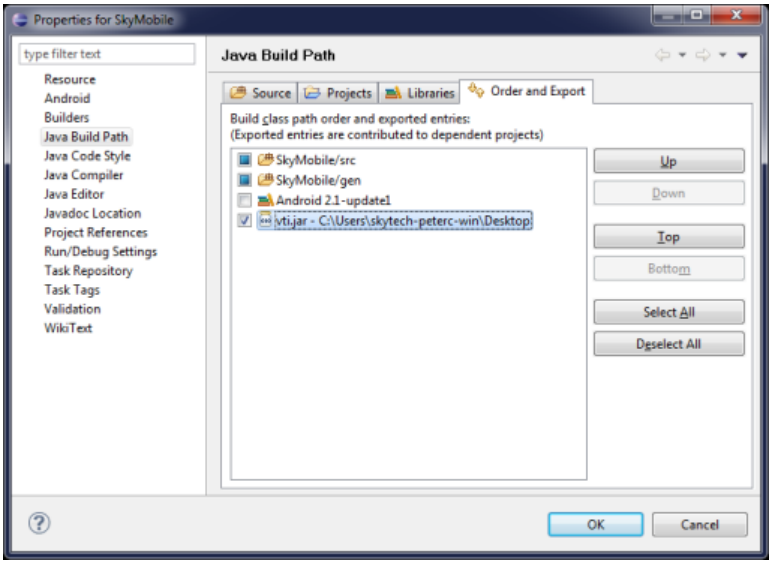
7. Click the **Libraries** tab, click the **Add External JARs**... button and select the copy of the Android vti.jar file. (This is the file distributed alongside the .apk file in the android release zip file.)



8. Select the **Libraries** tab, click the **Add External JARs**... button and select the copy of the Android vti.jar file. (This is the file distributed alongside the .apk file in the android release zip file.)



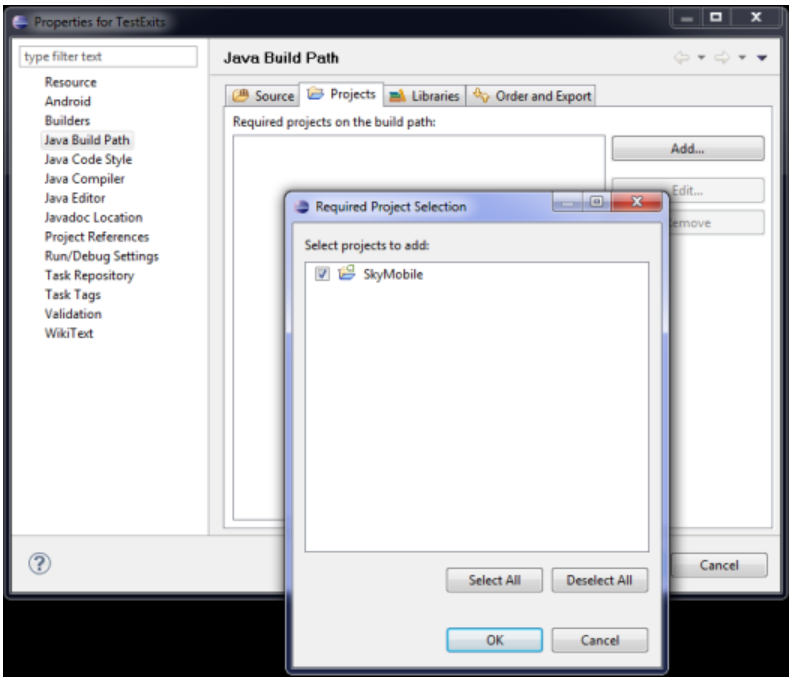
9. Click the **Order and Export** tab and select the **vti.jar** check box.



10. Click **OK** to close the window.

Creating your User Exits Project

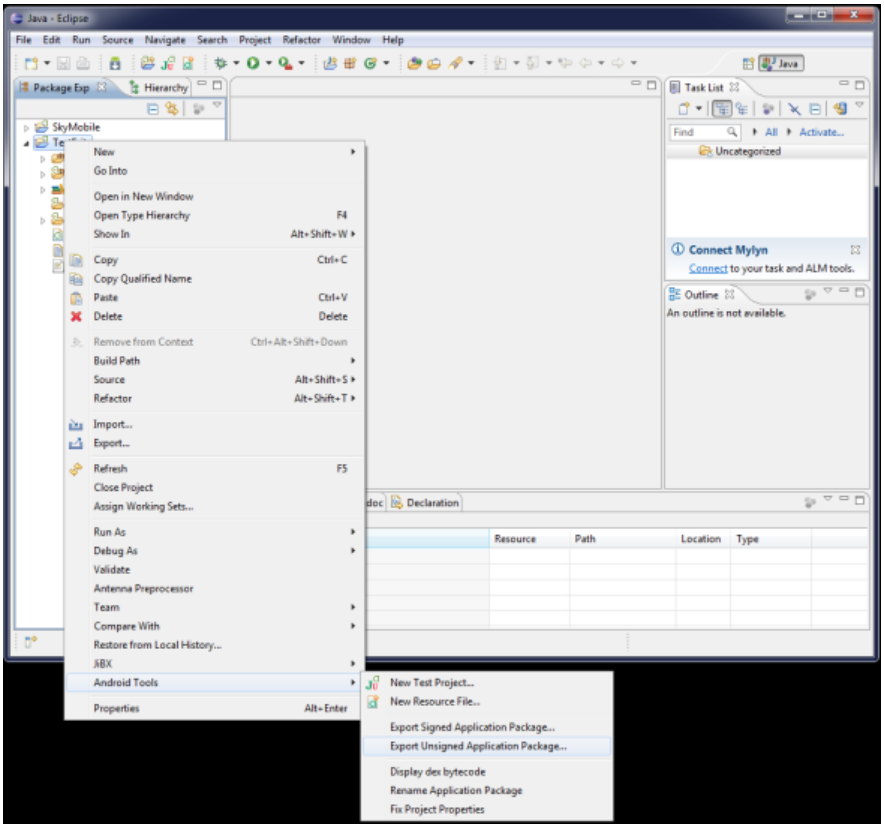
1. Create a new project by selecting **File** > **New** > **Android Project** from the menu.
2. Enter your desired project name and define where you would like the project files to be stored. Select the base SDK that you are using.
3. Enter the package name "au.com.skytechnologies".
4. Clear the **Create Activity** check box.
5. Click **Finish**.
6. Right-click the name of the project you just created, then select **Build Path** > **Configure Build Path**... from the menu.
7. Click the **Projects** tab, click the **Add**... button and select the **SkyMobile** check box.



8. Click **OK** to close the window. You are now ready to add whatever classes you want into your project.

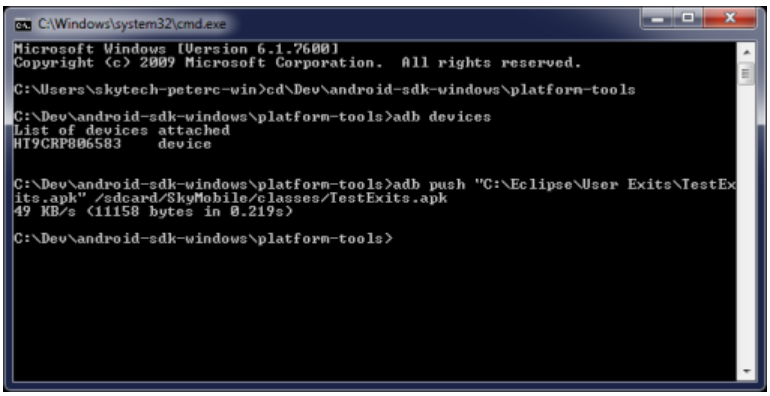
Copying your User Exits to the Device

1. Right-click your user exits project and select **Android Tools** > **Export Unsigned Application Package**...
2. Name and save this file to an appropriate location.



3. Open the command prompt and change directory to your directory of android SDK platform-tools.
4. Connect your device, or start the emulator and type 'adb devices' to ensure that one device is displayed.
5. Copy your user exits file onto the device/emulator using the command 'adb push <source> <target>'

6. You are only allowed to write to the sdcard on the device. You can reference through '/sdcard/'.



7. Configure the Android client classes directory to point to the location of your exits .apk file. You can do this through the SkyPreferences application, or through provisioning using the CLIENT.SERVER.CLASSESDIRECTORY parameter.

3.9.6.5 The Android Emulator

The android SDK includes an emulator that allows you to run the android client on a Windows or Mac PC.

Once started, the emulator acts like a device connected through the usb port. You need to install the Kony for SAP application manually on the device.

You can find the instructions on configuring an emulator and emulating an SD card at http://developer.android.com/guide/developing/devices/emulator.html.

3.9.6.6 Android Deployment Considerations

Android applications run in a sandboxed environment that provides limited access to the device resources. When deploying Kony for SAP on Android devices, it is necessary to understand these limitations and the various methods of ensuring Kony for SAP stays within them.

Memory

Each Android device specifies a maximum amount of memory that is available to each application. In early Android devices this was a very small 16MB, however in later devices (in 2012) this was generally around 64MB. The actual device manufacturer controlls this value and is unable to be changed. You should consider this memory ceiling when you decide which Android device to deploy on.

The two configuration options that primarily influence the maximum memory footprint that Kony for SAP uses are:

- SERVER.STORAGE.STORAGEPAGECACHESIZE
  Affects the overall memory footprint that the device uses and should be generally limited to 200 pages. If severely constrained, you can reduce to 50, however this value does affect database performance and you need to optimized through field tests to determine the most appropriate value for your device. (Refer to Database Configuration)
- SERVER.LOCALDATABASE.TRANSFERBUFFERSIZE
  Influences the maximum 'spike' in memory that occurs whilst sending to or receiving data from the gateway. (Refer toDatabase call recommendations)

Other influences are the number and size of graphics that the application uses, and the size and content of user exits and procedures.

Files

By default, all associated data files are installed in the protected directory `/data/data/au.com.skytechnologies.apc` that is only accessible by the Kony for SAP application. As a result, if the user has to access the files, you must do this through the Kony for SAP application itself. When Kony for SAP is running, you can do this through the web status page

and accessing the files page.

When a device is 'rooted' or 'jail broken', this rule is broken and the user is now able to access all files on the device. To protect sensitive data you are recommended to secure it using encryption.

Application Life Cycle

Unless the Kony for SAP application is configured to run in 'thin client' mode, when started, Kony for SAP launches the application server in "foreground mode". This means that the application server is seen as a critical service by the Android OS and is visible to the user. As a result, a notification icon appears in the notification bar for the entire duration that the Kony for SAP application server runs. Selecting this, notification launches the SkyMobile client.

Logging

A log viewer is provided to give easy access to any log files stored within Kony for SAP protected directories. However, if Android OS closes the application for some reason, then the reason is most likely does not show up in the application log file. Instead, the events are written into the Android OS system events log and you can view only using a third party program. One such program is "CatLog" in the Android Marketplace.

3.9.7 iOS Platforms

This section describes the process of installing a Kony for SAP secure container on iOS. It also addresses some things unique to the iPhone platform (for example, the way in which the Kony for SAP application integrates with standard iOS features such as the "Settings" menu).

3.9.7.1 Installation on iOS

You can install the Kony for SAP Secure Container for iOS in the following three ways:

- Installation on an iOS device using the Apple "app store"; or
- Installation on an iOS device using iTunes; or
- Installation on the iOS simulator.

Once the application has been installed, you automatically provision the configuration, applications and data profiles.

Installation on an iOS Device from the Apple App Store

Installing the SkyMobile Secure Container for iPhone is a relatively one straightforward if you are using the Apple "app store".

Prerequisites

- You must have an iTunes account.
- You must be using an iOS device (iPhone, iTouch, or iPad) running iOS version 4.3 or later.
- The iOS device must be connected to the Internet.

To install the SkyMobile Secure Container on an iOS device from the Apple App Store, follow these steps:

1. On the iOS device, launch the "App Store" application.
2. Search for "SkyMobile". The full name of the application is "SkyMobile - Enabling SAP on iPhone".
3. Select the SkyMobile application.
4. Once the application details appear, click the **Free** button. If you installed the application previously, this button may be titled **Install**.
5. If you are prompted to do so, enter the password for your iTunes account.
6. The SkyMobile application icon should then appear on your phone, along with a progress bar displaying how far along the application download progressed.
7. Once the SkyMobile application is fully downloaded, the progress bar disappears.
8. You may now launch the SkyMobile application, at which point the provisioning process commences.

Installation on an iOS device using iTunes

Installation from iTunes is somewhat more involved than installing the SkyMobile Secure Container from the Apple "app store". It is most commonly used if:

- Sky Technologies sends an "ad hoc" build for testing of features or bug fixes that are not yet released to the public at large; or
- Your organisation is making use of an MDM (Mobile Device Management) platform such as Afaria, Airwatch, or MobileIron to distribute the SkyMobile application.

Prerequisites

- You must use an iOS device (iPhone, iTouch, or iPad) running iOS version 4.3 or later.
- You require a Windows PC or Apple OS/X machine that has iTunes loaded onto it.
- An Apple iPhone/iPad/iTouch USB cable is needed to connect the iOS device to the machine running iTunes.
- The iOS device must be synced to the iTunes library on the machine running iTunes.
- You must have the SkyMobile application as either a `.zip` or a `.ipa` file.
- You also require a provisioning profile. This is a small file, normally with a `.mobileprovision` extension. The provisioning profile gives the SkyMobile application permission to run on your specific device. It is tied to your specific SkyMobile application build, and you should source from the same place you acquired the SkyMobile application itself, since without it, the SkyMobile application cannot run at all.

To install the SkyMobile Secure Container on an iOS device using iTunes, follow these steps:

1. If an older version of the SkyMobile application is already installed on your iOS device, uninstall it by clicking and holding any icon, then clicking the cross on the SkyMobile application when the icons start "wiggling". You can exit this mode when you are done by pressing the Home button underneath the screen.

2. Start iTunes and check under Library->Applications to see whether an older version of the SkyMobile application is present in your iTunes library. If it is present, delete it by selecting it and clicking the **Delete** key.
3. Connect the iOS device to the machine running iTunes using the Apple iPhone/iPad/iTouch USB cable.
4. Sync the device against your iTunes library. A sync may launch automatically when you first connect the device, in which case simply wait for it to finish. Otherwise, right-click your iOS device on the left-hand pane of your iTunes application window and select **Sync** from the menu.
5. Drag-and-drop the SkyMobile application file (`.zip` or `.ipa`) and the provisioning profile (`.mobileprovision`) onto Library >Applications in iTunes.
6. In iTunes, select your iOS device, choose the **Applications** tab. Make sure SkyMobile is present, and that the checkbox next to it is selected.
7. Sync your device again. Copy the new version of the SkyMobile application onto your device.
8. You may now launch the SkyMobile application, at which point the provisioning process commences.

> ***Note:*** The process of uninstalling an old version of SkyMobile destroys any associated data held on the device. Nevertheless, you are generally recommend to do this because it is difficult to verify the success or otherwise of an iTunes installation effort when the old version is still present, and this often leads to confusion.

Installation on the iOS Simulator

Unfortunately, due to restrictions put in place by Apple, configuring SkyMobile for use in the iOS simulator is far from straightforward. The iOS simulator only runs on OS/X, and is included as part of Apple XCode development environment. No standalone version of the iOS simulator currently exists.

Prerequisites

- You require an Apple machine running Snow Leopard (OS/X 10.6).
- You require a copy of XCode version 3.2.6 bundled with the iOS4.3 SDK. You can obtain the install file for this program only through the Apple Developer website, access to which requires registration as a developer with Apple. If you require access to the simulator and do not have a developer registration with Apple, you must go through their developer registration process in order to obtain access.
- You require a copy of the relevant SkyMobile release file for use on the iOS simulator. These files are normally named `vxx.xx.xx.release.iphone.simulator.zip`, where xx.xx.xx is the SkyMobile release number that you use. You can obtain such files from the Kony Developer Portal.

To install the SkyMobile Secure Container on the iOS Simulator, follow these steps:

1. Install XCode on the target machine.
2. Copy the SkyMobile iOS simulator release bundle onto the target machine.
3. Double-click the SkyMobile iOS simulator release bundle file. A new application called "SkyMobile (Simulator)" should appear in the same directory once it is decompressed.
4. Double-click the "SkyMobile (Simulator)" application. It should launch and run SkyMobile in the iOS simulator environment.

Known Limitations

- There is currently no way to launch the iOS simulator in "iPad" mode, so you are restricted to an iPhone-sized screen. Work on this limitation is ongoing.

- The simulator bundle attempts to run using the iOS SDK library it is compiled against. Because we currently compile SkyMobile for iOS under the iOS4.3 SDK, this is the version of the iOS SDK currently required. In the future, we may create simulator bundles to support other versions of the iOS SDK.

3.9.7.2 The iOS Startup Console

The **SkyMobile iOS Startup Console** allows you to perform some "behind the scenes" tasks associated with SkyMobile on an iOS device. Using it, you can:

- View or edit (depending on the policy of your organization) the configuration loaded onto the device;
- View any SkyMobile Application Server log file(s) stored on the device; or
- Reconfigure the application, removing all SkyMobile data and configurations from the device.

By default, the Startup Console is disabled when the SkyMobile application is installed. You can enable (or disable) as follows:

1. Open the **Settings** application.
2. Scroll down until you see an entry towards the bottom named "SkyMobile". Select this entry by touching it.
3. A screen, **Startup Console** that contains a single configuration appears, with a switch control to the right-hand side. Touch this control to toggle its state between *Off* and *On*.

Viewing or Editing the SkyMobile Configuration

The SkyMobile iOS configuration editor/viewer is invoked by touching the **Edit Configuration** button on the Startup Console.

If your organization has a policy of not allowing configuration editing (as specified when your server profile instance was configured), the **Edit Configuration** button is instead called **View Configuration**.

> ***Note:*** If there is currently no configuration (that is the iOS device is not yet provisioned), the **Edit/View Configuration** button on the Startup Console is disabled (greyed out).

If the configuration editor runs in "view" mode, you may scroll up and down to view the various configuration settings, but you are not permitted to make any changes.

If the configuration editor runs in "edit" mode, you may additionally perform any of the following tasks.

Adding a Configuration Key

To add a new configuration entry, touch the "+" button in the top-right corner of the screen. This invokes a second screen that allows you to enter a new configuration key and value.

> ***Note:*** You must enter the key in full, for example SERVER.GENERAL.UNICODEENABLED.

Once you have entered the new key and value, touch the **Save Changes** button to make them effective.

Modifying a Configuration Key

To modify an existing configuration entry, touch its entry in the scrolling list. This invokes a second screen that allows to enter a modified configuration key and/or value. Once you are happy with your changes, touch the **Save Changes** button to make them effective.

Deleting a Configuration Key

The configuration editor supports standard iOS "swipe to delete" behavior. If there is an entry you wish to delete, swipe along its entry in the scrolling list from left to right. A red **Delete** button should appear against the entry on the right-hand side. Clicking the **Delete** button removes the entry.

Viewing the Application Server Log

You can view the Application Server log file on the iOS platform by touching the **View Log** button on the Startup Console. This invokes the iOS log viewer.

> ***Note:*** The "View Log" button on the Startup Console is disabled (greyed out) if there are currently no log files available for viewing.

Log File Selection

On initial entry into the log viewer, you are prompted to select which of the available log files you wish to view.

If there is only one log file available, this step is skipped and the sole available log file is used.

Log File Viewing

Once you determine the log file to view, the contents of the log file appear in a scrolling table. You can scroll up and down using gestures in the usual manner. Errors are highlighted in red, warnings in yellow.

If your iOS device is configured for email, you can forward the selected log file to an email address of your choice by touching the **Email** button in the top right-hand corner. The log file is compressed to keep its size manageable, and then is emailed to the address that you specify.

Reconfiguring an iOS Device

You can reconfigure the SkyMobile application to its initial state by touching the **Reset Device** button on the Startup Console. If you do this, a confirmation message appears to double-check that this is in fact what you wished to do. If you press **OK**, the reset proceeds; if you press **Cancel**, it does not.

Resetting the device removes all SkyMobile application data and configuration settings from the device. This includes:

- The SkyMobile database in which downloaded applications and data are cached.
- Any Application Server log files resident on the device.
- All images, documents, sounds and other documents that the SkyMobile application downloads.
- The SkyMobile configuration data set up through provisioning.
- Any stored client-side certificates set up for use by SkyMobile during HTTPS authentication.

3.9.8 HTML Client

You may execute applications directly from a Web browser. In order to do this, you must define an HTTP port in the Application Server configuration file. Once you do this, the web browser connects to the TCP/IP address and designated HTTP port of the Application Server. In this mode, the Application Server effectively acts as a Web server, automatically generating HTML, accepting input and handling the application flow.

Example HTTP Application Port Configuration

```
SERVER.HTTPAPPLICATIONPORT:PORT15080.APPLICATION = 101
SERVER.HTTPAPPLICATIONPORT:PORT15080.CLIENTCANREQUESTAPPLICATION =
true
SERVER.HTTPAPPLICATIONPORT:PORT15080.DEVICE = 000
SERVER.HTTPAPPLICATIONPORT:PORT15080.HOSTINTERFACE =
SkyDemoRfcClient
SERVER.HTTPAPPLICATIONPORT:PORT15080.PORT = 15077
SERVER.HTTPAPPLICATIONPORT:PORT15080.SESSIONTIMEOUT = 60
SERVER.HTTPAPPLICATIONPORT:PORT15080.USEHTMLCOMPRESSION = true
SERVER.HTTPAPPLICATIONPORT:PORT15080.VERSION = 002
```

> ***Note:*** For a full description of these options, see the main Application Server configuration section.

3.9.8.1 Connecting to the Application Server

Start the web browser and issue the following command: http://{Application ServerTCP/IP address}:{Application Server port}

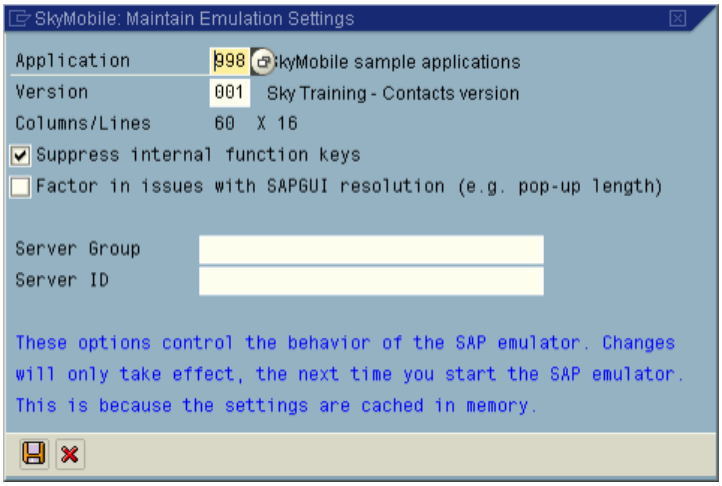
For example, http://localhost:15073 or http://192.168.2.110:15073

3.9.8.2 Specifying Parameters

You may use a query string to specify an application number, version number and/or device number when you connect. To do this type a '?' immediately following the port with options separated by '&'. For example: http://localhost:15073?appnum=996&vernum=003&devnum=001

3.9.9 SAPGUI Client

You may execute applications directly from the SAPGUI. The SAP add-in handles this user interface and has no dependency on the Application Server. This mode is useful for application development and testing, as a training tool, as a backup for equipment failure and to simplify SAP screen processing. Before executing the SAPGUI user interface, you must first configure the application and version you wish to execute and any trace options. You can do this using a popup that is configured either from the workbench or when the interface first starts.



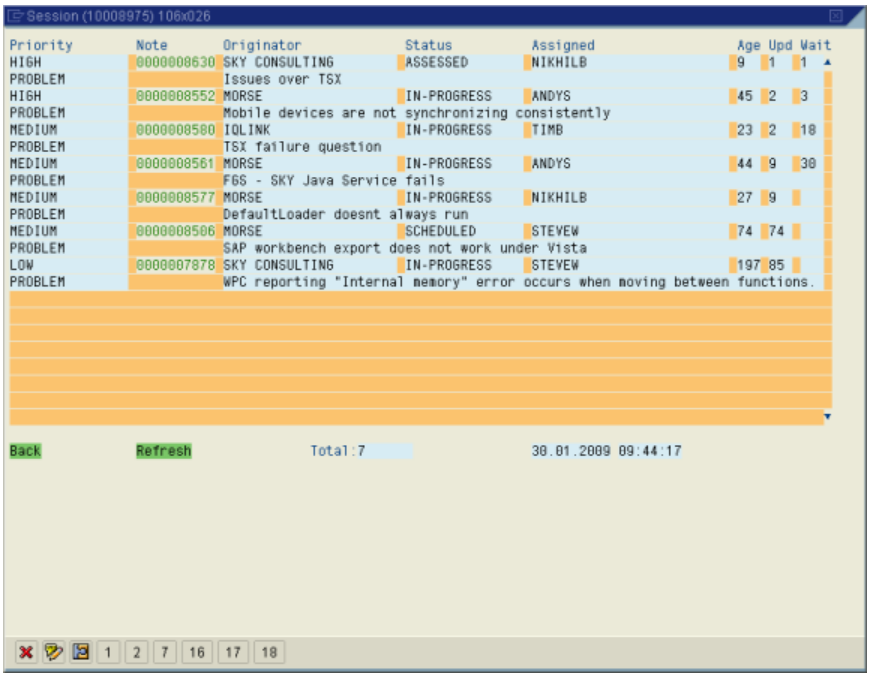
Option	Function
Application / Version	The application and version to execute
Suppress internal function keys	Do not include internally defined function keys, for example, switchboard.
Factor in SAPGUI resolution	The SAPGUI does not properly render the screen taking fonts into consideration. Because of this, blank lines may appear at the end of the SAP emulator screen. This option attempts to address this shortcoming to present a better window display.

Option	Function
Server Group/Id	The name of a Server Group and Id to use for the session. This may be necessary for data profiling, and tracing.

3.9.9.1 Starting the SAPGUI Emulator

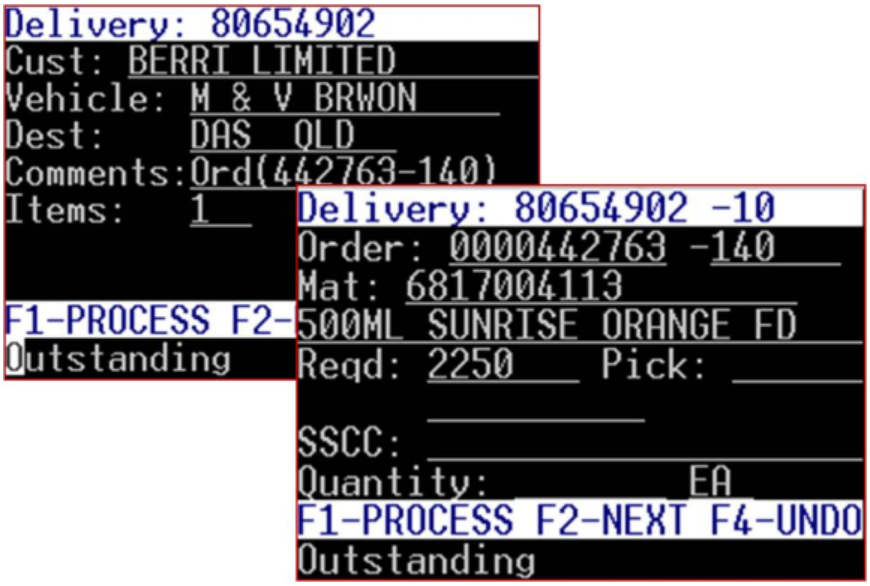
The SAPGUI emulator is normally started through the SAP Workbench through the icon adjacent to the version in the hierarchy list. You may also use the YVTE transaction.

The emulator tries its best to simulate a true GUI environment but is constrained by the functionality available in SAPGUI, for example, inability to display graphics and certain colours. However this is usually fine to test functionality.



3.9.10 VT220 Client

The Sky Secure Container fully supports VT220 text terminal type applications. You may execute through any appropriate Telnet server / client that support VT220 emulation. VT100 works, but has restricted functionality in terms of function keys and specialized escape sequences supported. You need not implement specialized client software, all that is required is a SkyMobile Application Server to connect to and the application/version to connect to.



3.9.10.1 Configuring a VT220 Terminal Port

You need to define a terminal application port in the Secure Container configuration `skymobile.cfg` referencing a SAP interface connection and the application to execute. Once you do this, you may use many telnet client that supports VT220 to connect directly to the TCP/IP address and designated port. In this mode, the SkyMobile Application Server acts as a telnet server, automatically formats the VT220 text terminal screens for display and handles all input and application flow. This mode is ideal for radio frequency (RF) devices or PC based telnet clients such as Reflection, Netterm, and QVTerm.

Example configuration (Highlighted values are mandatory)

```
SERVER.TERMINALAPPLICATIONPORT:PORT15072.APPLICATION = 998
SERVER.TERMINALAPPLICATIONPORT:PORT15072.CLEARFIELDVALUEONREENTRY =
true
SERVER.TERMINALAPPLICATIONPORT:PORT15072.CLIENTCANREQUESTAPPLICATION
= true
SERVER.TERMINALAPPLICATIONPORT:PORT15072.COLORENABLED = true
SERVER.TERMINALAPPLICATIONPORT:PORT15072.CONTROLCHARACTERSPERMITTED
=
SERVER.TERMINALAPPLICATIONPORT:PORT15072.DEVICE = 001
SERVER.TERMINALAPPLICATIONPORT:PORT15072.EXITDELAY = 2
SERVER.TERMINALAPPLICATIONPORT:PORT15072.HOSTINTERFACE =
SkyDemoRfcClient
SERVER.TERMINALAPPLICATIONPORT:PORT15072.LONGFIELDVALIDATIONMESSAGES
= false
SERVER.TERMINALAPPLICATIONPORT:PORT15072.PORT = 15072
SERVER.TERMINALAPPLICATIONPORT:PORT15072.SESSIONTIMEOUT = 0
SERVER.TERMINALAPPLICATIONPORT:PORT15072.TERMINALENQUIRY = true
SERVER.TERMINALAPPLICATIONPORT:PORT15072.TERMINALENQUIRYTIMEOUT = 10
SERVER.TERMINALAPPLICATIONPORT:PORT15072.TERMINALTYPE = VT220
SERVER.TERMINALAPPLICATIONPORT:PORT15072.TRACEINPUT = true
SERVER.TERMINALAPPLICATIONPORT:PORT15072.VERSION = 100
```

> ***Note:*** For a full description of these options, refer to configuration section.

3.9.10.2 Configuring an Answer Back String

An "answer back" string provides a mechanism to configure parameters on the device and pass these to the SkyMobile Application Server when a connection is made. This is a standard Telnet feature that is broadly available. This feature enables devices to nominate SkyMobile configurations when the device connects, for example:

Example

```
VTI[a=nnn,v=nnn,d=nnn,c=Y/N]
```

Field	Description
a=	SkyMobile numeric application identifier.
V=	SkyMobile numeric application version identifier.
D=	Device id (001-999). A free format string used to uniquely identify a device.
C=	Colour Yes/No indicator (Y/N). Indicates whether the device supports VT220 colour.

How answer back strings are configured varies per device and/or telnet emulator. Consult the vendor documentation for details. You must specify the TerminalEnquiry options for answerback to work.

3.9.10.3 Connecting to the Secure Container

All you need is the TCP/IP address and port of the Secure Container that you wish to connect to. Where this is configured varies per device and/or telnet emulator. Consult the vendor documentation for details. If you have problems, you may test this connection independently using a telnet emulator.

3.9.11 UNIX Platforms

The Sky Application Server Installation for UNIX guide provides detailed steps and prerequisites that you need to ensure that the Sky Technologies Application Server installs and executes correctly on the UNIX platform. For detailed information on the Application Server management, that is starting, stopping, and monitoring, refer to Systems Management.

3.9.11.1 Prerequisites

Overview

Before commencing the installation, ensure that you addressed the following prerequisites:

- The Sky SAP add-in components are installed and configured in the SAP system. See the SAP add-in section for more information.
- A suitable #Unix logon is configured.
- A target #Installation directory is created and its permissions are changed so that the installation user has full control of it.
- An ANSI #C compiler is installed on the UNIX host.
- A #Java Development Kit is installed on the UNIX host.
- A suitable version of the #SAP RFC libraries is obtained.
- The #Unix installation file is obtained from Kony.

> ***Note:*** Experience shows that these prerequisite tasks are usually the most time consuming part of the UNIX installation process. It is therefore appropriate to examine each in some detail.

Unix Logon

Kony recommends configuring a separate user that "owns" Kony software. By convention, this user is often known as "sky" or "skytech", but you can assign any desired name.

> ***Important:*** It is generally not appropriate to install or run the software as the root user, because this has undesirable security implications.

Installation directory

You can install the Sky Application Servers anywhere on the UNIX host, but typically they are installed in a directory called `sky` or `skytech` under `/opt`, `/usr/local`, or `/var`. Select an appropriate location based on the configuration of, and conventions adopted on, your Unix host.

Once the target directory is created, you should change its permissions to give the UNIX logon that "owns" the software full control of the directory (read, write, and execute).

Throughout the remainder of this documentation, this directory will be referred to as <<`SKY HOME`>>.

C compiler

You require a full ANSI C compiler to allow the Sky Application Server libraries to compiled on the target system. The table below details the recommended compiler for some common Unix Platforms.

Environment	Recommended C Compiler
AIX	AIX C Compiler or GCC
HP/UX	HP ANSI C Compiler or GCC. Please note that a "bundled" C compiler is installed by default with HP/UX. This is a cutdown C compiler that is primarily intended for use in kernel rebuilds, so it is NOT suitable for installation of the Sky Technologies software. The HP ANSI C Compiler is a separate product.
Linux	GCC
Solaris/SunOS	GCC

For details on how to install the chosen compiler, refer to the appropriate vendor documentation.

> ***Note:*** Once the Sky runtime libraries are compiled using the C compiler, you can generally copy them over to other machines with a similar configuration. Typically, therefore, you can install a C compiler on a development or test host machine and use to build libraries that you then copy over to the production host. This removes the need for any further compilation of code in the production environment.

Java Development Kit

You require a full Java Development Kit (JDK) for compilation of the Sky Application Server libraries. A Java Runtime Environment (JRE) is not sufficient for this purpose, although it is possible to compile the Sky runtime libraries using a JDK and then use them under a JRE in a production environment.

The table below details the recommended JDK vendor for some common Unix Platforms.

Environment	Recommended JVM Vendor	Download Web Site
AIX	IBM	http://www.ibm.com
HP/UX	Hewlett Packard	http://www.hp.com
Linux	Sun	http://java.sun.com
Solaris/SunOS	Sun	http://java.sun.com

If your platform is not listed, check the Sun web site (http://java.sun.com) to see if a JDK release is available for the environment that you use.

The Sky Technologies software is compatible with any version of Java from 1.1.8 onwards. In a practical sense this means virtually any version of Java currently available, as it is now very uncommon to see versions of Java earlier than 1.1.8 still in use. If you are downloading a JDK, note that you require the Standard Edition (J2SE) rather than the Enterprise Edition (J2EE) of Java.

The procedure to follow to install your chosen JDK varies depending on the vendor and the operating system in use. You should follow the installation instructions that the vendor of the JDK you selected provides.

SAP RFC Libraries

You require a copy of the SAP RFC libraries in order to build the Sky runtime libraries. You are recommended to the use v6.40, although earlier and later versions of the SAP RFC libraries are used successfully in the past.

You need to source the appropriate version of the libraries for the target environment based on the following factors:

- The flavour of Unix in use (Linux, Solaris, HP/UX, and AIX)
- The hardware chipset (SPARC, x86, PA-RISC, and Itanium)
- The operating system release (32 bit versus 64 bit).

SAP offers both a Unicode and a non-Unicode version of the SAP RFC libraries for each target environment. Currently only the non-Unicode version is supported under UNIX. This has no bearing on whether or not the target SAP system has a Unicode kernel, as it is still possible to communicate in non-Unicode mode with a SAP system running a Unicode kernel.

You can source the SAP RFC SDK libraries either directly from SAP site (http://service.sap.com) or from Sky Technologies. They are typically provided in CAR or SAR format (both compression schemes devised by SAP), and can be quite large (20MB+). Consequently you are recommended to source them directly from SAP wherever possible.

Unix Installation File

You must download a copy of the Unix installation file from SDN. This normally takes the form of a file called `v##.##.##.release.unix.tar.gz`, where ##.##.## is the software version number. This file is normally around 3-4 MB in size.

3.9.11.2 Unix Installation

You can summarize the installation steps for the Sky Application Server software as follows:

- Confirm installation of the C compiler and the Java Development Kit.
- Unpack the SAP RFC SDK.
- Unpack the Sky software.
- Configure the Makefile.
- Compile the Sky JNI libraries.
- Configure the Sky Application Servers.
- Test the Sky Application Servers.
- Configure server launch and monitoring.
- Tailor environment variables.

Confirm C Compiler and JDK Details

Compilation of the Sky JNI runtime libraries require access to a C compiler/linker, and a Java Development Kit (JDK).

Verify that both of these products are installed, and note the pathname to each component, as you use these later in the installation process.

There are a number of ways to confirm the path of the component:

- The "which" command (for example, "which cc").
- The "find" command (for example, "find / -name cc").
- The locate utility
- The man pages

The most appropriate method to use is dependent on the UNIX platform that is used, and your level of access to the target machine.

Unpack the SAP RFC SDK

To unpack the SAP RFC SDK libraries, follow these steps:

1. Log on to the Unix host.
2. Create a directory `<<SKY HOME>>/rfcsdk`.
3. Upload the SAP RFC SDK installation file to `<<SKY HOME>>/rfcsdk`. If you are using FTP to upload the file, ensure that you use binary mode when performing the transfer.
4. Change directory to `<<SKY HOME>>/rfcsdk`.
5. Unpack the file.

   The method you use depends to unpack the file is dependent on the format that is provided. If you have a CAR or a SAR file, you can unpack using the following command:

CAR/SAR command:

```
<<PATH_TO_SAP_EXE>>/sapcar -xvf ./<<RFC SDK ARCHIVE NAME.SAR>>
```

If you have what is known as a "tarball" (a `.tar.gz` or `.tgz` file), you can unpack by using the following commands:

"Tarball" command:

```
gzip -d <<RFC SDK ARCHIVE NAME.tar.gz>>
tar xvf <<RFC SDK ARCHIVE NAME.tar>>
```

Confirm that the contents of the archive were successfully extracted, and that no errors were reported during the process.

Unpack the Sky Java Software

The Sky software is provided in "tarball" format (a gziped tar archive). The filename of the archive includes the version number and has the format v##.##.##.release.unix.tar.gz.

The steps involved in unpacking the software are:

1. Upload the archive file to the Unix host to `<<SKY HOME>>`. If you are using FTP to upload the file, ensure that you use binary mode when performing the transfer.
2. Log on to the Unix host.
3. Change directory to `<<SKY HOME>>`.
4. gzip -d v##.##.##.release.unix.tar.gz
5. tar xvf v##.##.##.release.unix.tar

You should create a new sub-directory in the `<<SKY HOME>>` directory. For example, if you were to unpack `v##.##.##.release.unix.tar.gz` into `/usr/local/sky`, you expect to see a new directory created, `/usr/local/sky/v##.##.##`.

Confirm that the contents of the archive are successfully extracted, and that no errors are reported during the process.

Configure the Makefile

Configuring the Makefile is the first step in compiling the Sky JNI runtime libraries. These consist of a number of shared libraries that the Sky uses to perform certain lowlevel system functions, and to effect communications with SAP. The libraries are contained within a directory called `skyjni`, location immediately below the `<<SKY HOME>>/v##.##.##` directory for your version of the software (for example, `/usr/local/sky/v##.##.##/skyjni`).

You need to update the `Makefile` to reflect the setup on the installation machine. Using a text editor (such as vi), open the `Makefile` and go through it to make sure everything looks correct. You find that the `Makefile` consists of a number of sections that are as follows.

Site Dependent Options

Usually, you need to update everything in this section. You need to define the following things:

- The path to the JDK directory.
- The path to the C compiler binary.
- The path to the linker binary (usually the same as the C compiler).

Platform and Compiler Dependent Options

This section contains a number of different entries, one for each different combination of platform and compiler. All are commented out; you should select the one that applies to you and uncomment it by deleting the "#" characters from the start of each line.

If you find that your combination of platform and compiler is not listed, you may be compiling in a target environment where the software is not previously used. However, the Sky JNI code is extremely portable and is successfully compiled without modification on many different target platforms. If you find yourself in this situation, you can either contact Sky Technologies for further assistance, or (if you are more confident) synthesize an appropriate `Makefile` entry yourself from the examples shown. In the latter case, the most important factor to consider is that the code must be compiled as a shared library. Use whatever command line flags are needed to force the compiler and linker to recognise this fact.

Build Options

Uncomment any of the lines here that may be applicable to your situation. These are fairly self-explanatory and are documented in the Makefile itself.

Common Variables

These are secondary variables that are mostly derived from the information in the preceding sections. In the majority of cases, you need not change them.

Make Targets

These perform the actual work involved in building the libraries. The following make targets are defined:

Type	Description
all (default target)	Builds all the Sky JNI libraries
skydial	Builds the Sky dialup JNI library
skyrfc	Builds the Sky SAP RFC JNI library
skysys	Builds the Sky JNI system library
clean	Removes the Sky UNI libraries and any intermediate build files

Compile the Sky JNI Libraries

Once the `Makefile` is configured, you need to compile the Sky JNI libraries.

To compile the Sky JNI libraries, follow these steps:

1. Ensure that you are in the directory `<<SKY_HOME>>/v##.##.##/skyjni`.
2. Type "make".
3. Review the output for any errors.

It is common for portability issues between different flavours of Unix to give rise to warning messages during the compilation process. However, you need to address any errors before continuing. If all is as it should be, you should see three library files created in the `skyjni` directory: `libskydial`, `libskyrfc`, and `libskysys`.

Any errors that you are unable to resolve during this process, you should refer back to Kony.

Configure the Secure Container

Before you can use a Secure Container, you need to set up the configuration file. For information on the configuration options available and examples, refer to the general Secure Container Installation Guide.

Test the Secure Container

At this point, you can try launching one of the Sky Secure Containers to see whether it is successfully configured and installed. You can launch SkyMobile from a shell script, for example, `<<SKY HOME>>/v##.##.##/vti/vti.sh`. It is generally a good idea to use "nohup" to prevent the process from terminating when you log out, and should use the "&" sign to place the process into background mode. Thus, the command to launch the container from the VTI directory is as follows:

nohup ./vti.sh &

Check the server log file (in the `log` directory) to see whether the service started successfully, and whether it is able to connect.

If no log file is produced, there are two likely explanations.

- Invalid Java command line. Ensure that the SKY_JAVA_CMD environment variable is set and exported, or alternatively set the "JavaCommand" configuration parameter in the [Java] section of the relevant configuration file. Make sure that the value you assign is the correct path to the Java executable.
- Problems with file permissions. Check the file permissions on the directories and files you are working with.

Secure Container Launching and Monitoring

Following a successful launch of the Sky software, the next step is to address how to have the installed services automatically launch when the machine boots, and how to restart them in the event that they are manually killed or go offline. There are two main approaches.

Using GCP

Kony provides a tool called GCP (Generic Command Processor). GCP provides a simple command line interface for starting, stopping, and checking the status of Sky Application Servers. Setting up GCP is a fairly simple process.

GCP is located in the `<<SKY HOME>>/v##.##.##/gcp` directory. You need to set up a configuration file before you can use. Generally it is sufficient to simply copy the sample configuration file (`gcp.sample.cfg`) to create `gcp.cfg`. However, if you have configured customized command ports, or multiple instances of certain services, you may need to customize the configuration.

GCP is invoked through the shell script `<<SKY HOME>/v##.##.##/gcp/gcp.sh`. However, to allow different versions of the Sky software to co-exist (during an upgrade, for example), it is desirable to set up another script file that targets the desired version of the software. Sky therefore recommends that you create a wrapper script in `<<SKY HOME>>/gcp.sh`, as shown in the example below:

Example GCP Script

```
############################################################
#
# Script to start Sky Technologies Application Servers.
#
############################################################

SKY_JAVA_COMMAND=/usr/java/jdk1.6.0_02/bin/java
export SKY_JAVA_COMMAND
```

```
SKY_HOME=/opt/sky/v14.00.00
export SKY_HOME

$SKY_HOME/gcp/gcp.sh $*
```

You can invoke this script as necessary from many different places (for example, from within SAP). Upgrading to a new version of the software is then a matter of editing the value assigned to `SKY_HOME` once the new software is installed.

GCP is typically used from within SAP by setting up an ECS job to periodically check on the status of each server. The upside of this approach is that it gives SAP control over whether or not each server is running. The downside is that Java processes launched from SAP end up running as the SAP Unix user that can lead to problems with file permissions unless a consistent approach is adopted in terms of which user launches the services.

Using Standard Unix

An alternative to GCP is to configure the Unix operating system to launch the necessary services when the machine boots, and respawn them in the event that they die. The precise implementation of such scripts is dependent on the flavor of Unix that you use. Typically it involves writing an `rc.d` script. Consult your Unix documentation for further information on the best way to achieve this. However, take care to ensure that the Sky software is not launched as the root user. This is never a good idea from a security perspective, so make sure than any such solution includes a switch to the appropriate user.

Unix Environment Variables

SkyMobile makes use of a number of environment variables that control various facets of its behaviour. It may be appropriate in some circumstances to consider making use of these. However, in most situations they are not required.

Environment Variable Name	Function
SKY_JAVA_CMD	This environment variable specifies the path to the Java virtual machine. It is used if no value is specified for the "JavaCommand" parameter under the [Java] section of the relevant configuration file. If you can obtain no value from either source, the Java command is assumed to be "java" that only works if the Java binary lies in the current path of the user.
SKY_JAVA_ARGS	This environment variable specifies any arguments that you can use in confunction with the SKY_JAVA_CMD environment variable.
SKY_JAVA_HOME	Sets the directory for the overall SkyMobile installation directory.
SKY_VTI_HOME	Sets the directory for the SkyMobile Application server directory.
SKY_NICE	This environment variable allows SkyMobile processes to reduce their consumption of resources. To make use of this feature, you should set to a numeric value reflecting the number of "nice" categories you wish to reduce Sky processes by. Refer to the man page on the "nice" command for further information.
SKY_UMASK	This environment variable allows SkyMobile processes to change the default permissions assigned to files that are created (for example, log files). It works in exactly the same way as the "umask" command. Refer to the man page on the "umask" command for further information.

> ***Note:*** There are a number of options for introducing environment variable values into your environment.
> The strategies available depend on the platform and operating system in use.
> The preferences of Systems administration of your organization determines which of the available strategies is appropriate in your situation.

3.9.11.3 Troubleshooting on Unix

This section lists some of the more common problems. If you cannot resolve the issue yourself, contact Kony support.

Issue	Process
SkyMobile does not start and does not produce a log.	The most common cause of this problem is that there is no Java Virtual Machine available. Check to ensure that the value specified in either the Sky product configuration file JavaCommand option or the SKY_JAVA_CMD environment variable points to a valid Java Virtual Machine executable. Try typing "java" at the command line to see if Java is installed.
SkyMobile does not start and does not produce a log.	The log file should contain details of the error that prevents the product from launching successfully. Examine the log file in a text editor and this should allow appropriate corrective action.
OutOfMemoryException Problems	If the Java Virtual Machine (JVM) is runs out of memory, you are probably using one that allocates an arbitrary upper limit to the amount of memory it can use. Many JVMs, including those that Sun supplied, do this as a safeguard to protect Java from consuming too many resources. In most cases, you can resolve the situation by increasing the upper memory limit assigned to the Java virtual machine when it is launched. You need to modify either the Sky product JavaArgument or the SKY_JAVA_CMD environment variable so that the increased memory ceiling is specified as an argument to the JVM executable. The appropriate syntax to use depends on the JVM you use; consult the relevant JVM documentation if you are unsure. As an example, the following setting is appropriate under the Sun JVM, to increase the maximum memory ceiling to 128 megabytes: -Xmx128m.

Issue	Process
Cannot Execute error messages	This indicates that your current user does not have appropriate execute permissions on one or more of the shell scripts from which SkyMobile is launched. If your user owns the files in question, you can type `chmod +x <script filename>`to grant yourself execute permission on the script. If you do not own the files, you need to contact your system administrator to help you resolve the problem.
Cannot access log or other output files	This situation relates to UNIX file permissions. It most commonly occurs where SAP launches an application service. Because the SAP user is usually very strictly controlled, any files that it creates by default may not be accessible to other users. As a first way to resolve this problem, try setting the environment variable SKY_UMASK to 000 (three zeroes) in the profile of the SAP user. This should have the effect of ensuring that all files that the SAP user created while executing SkyMobile processes are accessible to all users. If the problem persists despite this, you need to talk to your system administrator, who can devise a solution that is compatible with your site's security policy. Possible approaches include: • Granting your user access to the SAP user group. • Changing the "umask" on the SAP user to something less restrictive. • Writing a shell script that does a "chmod" and/or "chown" on the files in question. You can invoke this script from SAP (as the SAP user) to grant other users access the files.

UnsatisfiedLinkException Problems

If the product is writing UnsatisfiedLinkException stack traces to the application logs, this means that the Sky JNI layer is not invoked successfully. There can be a number of reasons for this. The most common are listed below.

Potential Problem	Solution
The JNI libraries are not compiled.	Compile the JNI libraries.
The JNI libraries are copied from another machine that is not object code compatible.	You must recompile the JNI libraries on the new host.
The SAP RFC library is moved, deleted, or for some other reason is not present in the location it was in when the JNI libraries are compiled.	Restore the SAP RFC library to its correct location. Alternatively, you may recompile the JNI libraries with the SAP RFC library in a new location.
32-bit/64-bit incompatibility exists.	Refer to 64-bit Compatibility Issues.

Controlling CPU Usage

For very busy UNIX systems with limited resources, SkyMobile has capability to limit their consumption of resources. Internally you can achieve this by making use of the UNIX "nice" command. To take advantage of this feature, set the environment variable SKY_NICE to a numeric value reflecting the number of "nice" categories you wish to reduce SkyMobile processes by (refer to the main page on the "nice" command for further information). Changing the SKY_NICE environment variable limits the resources that all services can consume that are wholly contained within the SkyMobile bundle (for example, GCP, VTI).

64-bit Compatibility Issues

You can build the Sky JNI libraries to run in either 32 or 64 bit mode, but you must take extreme care to ensure that the appropriate memory model is consistently observed throughout the build and run process. This includes:

- Ensuring that the version of the SAP RFC SDK library used is appropriate to the target memory model (either 32 or 64 bit).
- Ensuring that the Sky JNI libraries are compiled with compile and link flags appropriate to the target memory model (again, either 32 or 64 bit).
- Ensuring that the Java Virtual Machine in use is capable of running in the correct mode (either 32 or 64 bit), and that any run-time command line options to enable the correct memory model are used.

Failure to observe these steps (for example, using a 64-bit version of the SAP RFC SDK whilst trying to build a 32-bit version of the Sky JNI libraries) can cause a variety of serious difficulties whilst compiling and/or running the Sky software. The precise results varies depending on the platform in use and the nature of the incompatibility.

3.9.11.4 Monitoring Kony for SAP in UNIX

> *Important:* This page contains macros or features from a plugin that require a valid license. You need to contact your administrator.

The Sky Application Server applications run continuously as Java JVMs. Typically each instance of a JVM runs as a separate UNIX process. (In some UNIX environments this may not be the case). Type the UNIX top command to view the current active processes:

UNIX Processes

```
System: Skynet2 Fri Aug 10 16:53:33 2001
Load averages: 1.41, 1.67, 1.67
254 processes: 237 sleeping, 13 running, 4 zombies
Cpu states: CPU LOAD USER NICE SYS IDLE BLOCK SWAIT INTR SSYS
0 1.39 27.4% 0.0% 10.9% 61.6% 0.0% 0.0% 0.0% 0.0%
1 1.57 5.4% 0.0% 3.4% 91.3% 0.0% 0.0% 0.0% 0.0%
2 1.23 39.6% 0.2% 22.3% 38.0% 0.0% 0.0% 0.0% 0.0%
3 1.46 46.3% 0.4% 15.1% 38.2% 0.0% 0.0% 0.0% 0.0%
— ---- ----- ----- ----- ----- ----- ----- ----- -----
avg 1.41 29.6% 0.0% 12.9% 57.5% 0.0% 0.0% 0.0% 0.0%

Memory: 915240K (488196K) real, 1111880K (657740K) virtual, 201904K
free Page# 1/29

CPU TTY PID USERNAME PRI NI SIZE RES STATE TIME %WCPU %CPU COMMAND
0 ? 23576 orasky 154 20 22720K 2428K sleep 21:04 35.83 35.77 oracleSky
3 ? 23563 skyadm 236 20 166M 152M run 13:04 29.00 28.95 dw.sapSky_D
3 ? 2434 root 154 20 64968K 5424K sleep 43:11 12.20 12.18 dm_fc_scsi_
1 ? 27793 orasky 154 20 22592K 1340K sleep 3:35 7.58 7.57 oracleSky
3 ? 26430 orasky 154 20 22720K 2428K sleep 1:31 6.45 6.44 oracleSky
3 ? 17179 orasky 154 20 28864K 5500K sleep 1:54 6.02 6.01 oracleSky
```

Unknown macro: 'highlight'

3 ? 18681 skyadm 155 20 37216K 23336K sleep 3:33 3.99 3.98 dw.sapSky

The Java Virtual Machines show up as either "java" or "jre" (if executing Java runtime environment). It is not possible from this display to differentiate between the Sky Application Servers. Type `ps -ef | grep java` (or `jre`) to get the Java process details and `ps ef | grep {product id}` to associate the startup shell script with the corresponding Java process.

Example

```
$ ps -ef|grep java
skyadm 14459 14958 11 Aug 7 ? 260:45 /opt/java1.2/jre/bin/../bin/PA-
skytech 12828 11476 1 16:55:45 pts/tc 0:00 grep java

$ ps -ef|grep vti
skytech 12937 11476 0 16:56:17 pts/tc 0:00 grep vti
skyadm 14958 1 0 Aug 7 ? 0:00 ./vti.sh ./vti.sh
```

Use the parent/child process ids to match which java JVM processes are spawned by VTI. For example, Java process 14459 is spawned by process 14958 which is `./vti.sh` that is the VTI startup script.

3.9.11.5 General Systems Management

This section covers some of the main systems management activities typically taking place in a UNIX environment. For more detailed information, refer to System management.

Starting a SkyMobile Application Server

Using GCP

```
gcp.sh {product id} start
For example: gcp.sh vti start
```

Directly

```
nohup <<SKY HOME>>/{product id}/{product id}.sh &
```

Checking the Status of a SkyMobile Application Server

All Sky Application Servers server accept status commands through their command ports, and respond with their version and the date they were started. There are various methods available to issue status commands. The Web browser method provides a detailed report of the Sky Application Server status and is the recommended approach. It is suggested that you "bookmark" each Sky Application Server instance in a Web browser for ease of use.

Using GCP

gcp.sh {product id} status
For example: gcp.sh vti status

Using a Telnet Session

Connect to IP address/localhost Port 50?0 with local echo on.

- Type status and hit return.
- A status message is returned and the connection is closed.

> ***Note:*** The port number is dependent on the command port configured for the Application Server.

- Using a Web Browser
  Connect to <http://IP address:501> and navigate from there.

Stopping a Sky Secure Container

The Sky Secure Container / Access Gateway accepts stop commands through its command port and responds with a stopping message. There are various methods available to issue stop commands. Find the best method to suits you.

Using GCP:

gcp.sh {Product id} stop
For example: gcp.sh vti stop

Using a Telnet session

Connect to IP/LocalHost Port 50?0 (with local echo on).

- Type stop and hit return.
- A stopping message is issued and the Container shuts down.

> ***Note:*** The port number is dependent on the command port configured for the Application Server.

3.9.11.6 UNIX Global Command Processor

The global command processor enables commands to be directed to SkyMobile Secure Container and Gateway processes on the same Unix machine. The primary advantage of GCP is to abstract the system management processes from having to know the exact directory structure or port configuration of the servers and gateways. It is also useful to enable commands to be issued from the command line. GCP is delivered as a standard component of a Unix installation. Its configuration file (`gcp.cfg`) is located in the `GCP` directory.

Example configuration file

```
[General]

[Network]
UseNativeSockets =

[VTI]
StartCommand = "nohup ./vti.sh >> vti.nohup.out &"
WorkingDirectory = ../vti
CommandPort = 5070
```

Normally, the sample configuration file (`gcp.sample.cfg`) suffices, but make need to be changed to reflect command port changes or to support multiple instances of servers or gateways.

3.9.12 Configuration Options

This section describes the general configuration options that are common for all Secure Container platforms. The `skymobile` configuration file contains: global, client, server and general configurations in the following format: {type}.{section}.{option} = {value}. These entries are typically defined in the MEAP Server Profiler and then pushed out to the Secure Container as part of the provisioning process. In the case of a Access Gateway, you must implement the configuration file manually.

Example

SERVER.GENERAL.SERVERGROUP = my server group
SERVER.GENERAL.SERVERID = my server

The following documentation is broken down into the following types:

- Global Configuration
- Client Configuration
- Server Configuration

3.9.12.1 Global Configuration

The following section details all the sections and associated options for the Global configuration type.

- GLOBAL.CONFIGURATION
- GLOBAL.STARTUP

GLOBAL.CONFIGURATION

This configuration section controls options around the configuration behaviour.

Option	Description	Default
EDIT	A true/false flag that explicitly specifies whether the configuration is editable through the inbuilt client configuration viewers in APC, IPC and BPC. This has no effect on WPC.	True
ENCRYPT	A true/false flag that explicitly specifies whether the configuration is stored in an encrypted format.	False
VIEW	A true/false flag that explicitly specifies whether the configuration is editable through the inbuilt client configuration viewers in APC, IPC and BPC. This has no effect on WPC	True

Go back to the top

GLOBAL.STARTUP

This configuration section controls options around the startup behaviour.

Option	Description	Default
PROVISIONCHECK	A true/false flag that explicitly specifies whether the container does a provision check call at startup.	False

Go back to the top

3.9.12.2 Client Configuration

The following section details all the sections and associated options for the Client configuration type.

- CLIENT.ABOUTBOX
- CLIENT.CACHE
- CLIENT.GENERAL
- CLIENT.IDENTITYCHECK

- CLIENT.INPUT
- CLIENT.INPUTAID-XXX
- CLIENT.LOG
- CLIENT.PRESENTATION
- CLIENT.SERVER
- CLIENT.SERVERBACKUP
- CLIENT.XAI

CLIENT.ABOUTBOX

This configuration section controls the appearance and presentation of the about box (that is invoked from the "About" menu option). They override the general-purpose settings in the Presentation section.

Option	Description	Default
BUTTONPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each button.	0
BUTTONPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average button width.	0
BUTTONSPACING	The number of abuttons i.e. dditional logical pixels to space between spread them out.	0
FONTNAME	The name of the base font typeface that WPC should use when rendering the about box.	Arial
FONTSIZE	The point size of the base font typeface that WPC should use when rendering the about box.	9

Option	Description	Default
FONTWEIGHT	The "weight" of the base font typeface that WPC should use when rendering the about box (expressed as a number from 1 - 1000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a superthin font typeface, while a value of 1000 indicates an ultrabold font typeface.	400
WINDOWHEIGHT	The height of the "about" window, measured in logical pixels.	400
WINDOWWIDTH	The width of the "about" window, measured in logical pixels.	400

Go back to the top

CLIENT.CACHE

This configuration section controls the cache (that is primarily used to store images locally so that you need not continually fetch them from the server).

Option	Description	Default
CACHEDIRECTORY	The name of the cache directory. This directory is used to permanently store graphics and other data objects referenced by applications.	The current directory.
CACHESIZE	The maximum size allocated to the cache in bytes, after which the last referenced is automatically removed to make space available.	Unlimited

Option	Description	Default
USESERVERCACHE	A true/false flag that explicitly specifies whether or not the client should share the server's cached binary files (instead of maintaining its own cache). Normally the client identifies scenarios, when this is possible and default the setting appropriately. You need to set it normally only in unusual scenarios.	Is ascertained according to the situation.

Go back to the top

CLIENT.GENERAL

This configuration section contains general-purpose configuration settings that pertain to the client operation as a whole.

Option	Description	Default
ALLKEYS	A true/false flag that is only used when running under Windows Mobile. When set to true, this flag allows SkyMobile to intercept ALL keystrokes, including those normally reserved for use by the device as "soft" keys (for example, F1 and F2 are sometimes used to bring up the left and right device menus). Configuring this flag to *false* restores the default device behaviour that means that SkyMobile does not intercept these keys and cannot use them to trigger events within an application.	False
ALLOWMULTIPLEINSTANCES	Allow multiple instances of the presentation client to be started on the same computer.	False

Option	Description	Default
APPLICATIONNUMBER	The application number that the client should request from the application server. If left blank or not specified, the default application and version numbers from the application server port is used instead.	As noted.
CLIENTSESSIONTIMEOUT	This is the maximum time interval (in seconds) that the presentation client waits in an idle state before terminating the current session and returning to the initial screen of the application (that is typically the login screen). User activity such as mouse movement, mouse clicks, keystrokes and the like all prevents this from occurring. This feature is useful for ensuring that an application is not left "signed in" for extended periods of time. A zero or negative value here indicates that no timeout is applicable.	0 (no timeout)
CLIENTSESSIONTIMEOUTACTION	This is the action to undertake in the event that a client session timeout occurs. The following values are permitted: Logout - Log out of the application and re-initialize the user session (return to the first screen). Shutdown - Shut the client down completely and exit the application.	Logout
DEVICENUMBER	The device number that the client should use to identify itself to the application server. If left blank or not specified, the default device number from the application server port is used instead.	As noted.

Option	Description	Default
HOSTSTATUSMESSAGES	Indicates whether or not to display pop-up messages relating to changes in the host connectivity status. The following values are permitted: None - Host connectivity status messages never appear. Initial - Only the initial host connectivity status message appears(on entry into the application). DisableInitial - The initial host connectivity status message does not appear, but all such subsequent messages appear. All - All host connectivity status messages appear.	All
KEYBOARDONLY	A true/false flag that indicates whether the presentation client is running on a device with no mouse or touchscreen (for example, Windows Smartphone).	False
LOGOFFMESSAGE	A true/false flag that indicates whether the popup message to advise the user of a logoff should be disabled.	False
SOUNDS	A true/false flag that indicates whether to play the sounds that the server sends.	True
UNICODEENABLED	A true/false flag that switches the presentation client between Unicode and non-Unicode modes. If enabled, the client generates all outputs (for example, log files) in Unicode format. You should enable the flag if you desire the foreign language support.	False

Option	Description	Default
VERSIONNUMBER	The version number that the client should request from the application server. If left blank or not specified, the default application and version numbers from the application server port is used instead.	As noted.

Go back to the top

CLIENT.IDENTITYCHECK

This configuration section is used to implement SkyMobile Identity Management. It defines the Access Gateway port to connect to, encryption options (optional) and the identity service type to use. You may use other options to influence the user pop-up display that the identity management processing generated.

Option	Description	Default
LABEL:FOOTERTEXT	A text value to display at the bottom of the identity check screen, below the data fields and the submit button. Intended to contain instructions, warnings and/or legal disclaimers. Overrides any value that the Access Gateway sends.	None (blank)
LABEL:SCREENTITLE	A text value that, if present, replaces the standard "Identity Check" screen title. Overrides any value that the Access Gateway sends.	Identity Check
LABEL:SUBMITBUTTON	A text value that, if present, replaces the standard text on the identity check "Submit" button. Overrides any value that the Access Gateway sends.	Submit
PORT	The port number of the Access Gateway to connect to.	None
SERVER	The IP address of the Access Gateway to connect to.	None

Option	Description	Default
SERVICE	The name of the identity management service to use. If left blank, the default configured against the gateway port is used.	None
TRACE	Enable popup tracing of identity requests and responses. (Only available if CLIENT.SERVER.XMLDEBUG is also enabled)	False
RESPONSETIMEOUT	Time (in seconds) to wait for a response from the identity managment server.	10

Go back to the top

CLIENT.INPUT

This configuration section controls how character input is treated.

Option	Description	Default
CHARACTERSUBSTITUTION:X	This directive indicates that you should always substitute a specific incoming character with one or more replacement characters. The "x" represents the incoming character to be substituted. The value assigned to the configuration option represents the value with which you should replace the incoming character whenever it is received. In both values, you can represent nonprintable characters by a backslash followed by a 3-digit decimal notation for the character's value. For example, the "Escape" character is represented as "\027".	No default.

Go back to the top

CLIENT.INPUTAID-XXX

These configuration sections allow control over the appearance and presentation of specific input aids. They override the general-purpose settings in the [Presentation] section. Valid entries for the "xxx" portion of the section name are "AlphanumericKeypad", "Calculator", "Duration", "NumericKeypad", "QwertyKeypad", "RapidTextEntry", and "Time".

Option	Description	Default
ALLOWDUPLICATES	(Rapid text input aid only) A true/false flag to control whether or not duplicate values are permitted in the rapid text input aid.	False
ALLOWREMOVAL	(Rapid text input aid only) A true/false flag to control whether or not values are allowed to be removed in the rapid text input aid.	True
BUTTONPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each button.	0
BUTTONPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average button width.	0
BUTTONSPACING	The number of additional logical pixels to space between buttons that is spread them out.	0
DISPLAYSTATE	(QWERTY keypad input aid only) Sets the initial state of the QWERTY keypad when it is invoked. Two values are possible: "Shift" or "Symbol", each of which causes the appropriate set of keys to display. Any other value causes the normal initial keyset to be shown.	No value (normal initial keyset applies).
FONTNAME	The name of the base font typeface that WPC should use when rendering the relevant input aid.	Arial

Option	Description	Default
FONTSIZE	The point size of the base font typeface that WPC should use when rendering the relevant input aid.	9
FONTWEIGHT	The "weight" of the base font typeface that WPC should use when rendering the relevant input aid (expressed as a number from 11000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a superthin font typeface, while a value of 1000 indicates an ultrabold font typeface.	400
INPUTDELIMITER	(Rapid text input aid only) The delimiter character(s) that signal termination of an input value.	\r\n (carriage return followed by a line feed)
OUTPUTDELIMITER	(Rapid text input aid only) The delimiter character(s) to display on output of a complete value list in the field receiving input from the rapid text input aid.	#
SOUNDFAILURE	(Rapid text input aid only) The name of a sound file to play when an input error occurs in the rapid text input aid. The sound to be played must be in .wav format.	- (no sound)
SOUNDSUCCESS	(Rapid text input aid only) The name of a sound file to play when a value is successfully entered in the rapid text input aid. The sound to be played must be in .wav format.	- (no sound)

Option	Description	Default
WINDOWHEIGHT	(Rapid text input aid only) Configures the height of the window in logical pixels.	200
WINDOWWIDTH	(Rapid text input aid only) Configures the width of the window in logical pixels.	200

Go back to the top

CLIENT.LOG

This configuration section controls the logging and tracing capabilities.

Option	Description	Default
Option	Description	
ACTIVE	A true/false flag that controls whether or not logging should take place.	False
DIRECTORY	The directory into which WPC writes its log files. If not configured, this parameter defaults to "." (the current working directory).	As noted.
EXCLUDES	Indicates the types of log messages that should be excluded from the log file. There are 5 types of log messages - Information (I), Warning (W), Trace (T), Error (E) and Abort (A). The log message types to exclude should all be concatenated together. Thus, to exclude all Information and Trace messages from the log file, the appropriate setting would be "IT".	- (no types excluded)

Option	Description	Default
MAXFILES	This setting restricts the maximum number of log files the Application Server keeps under D, W, and P strategies. If left blank or not set then no maximum is enforced. 0 indicates no maximum.	0 (no maximum)
PERFORMANCETRACKING	A true/false flag that controls whether or not performance tracing messages are written to the WPC log.	False
SHOWMILLISECONDS	A true/false flag that controls whether milliseconds are shown in log messages. This feature is useful for measuring sub-second timings and events.	False
STRATEGY	The logging strategy to be adopted. Must be one of the following: D - A new log file is to be created daily. W - A new log file is to be created weekly. P - A new log file is to be created for each process instance O - Overwrite the log file each time the server is restarted.	O

Go back to the top

CLIENT.PRESENTATION

This configuration section controls the overall appearance and presentation of the client.

Option	Description	Default
AUTOSUGGEST	(Windows only) A true/false flag that indicates whether or not an auto suggest popup is available to text input fields. The auto suggestions list grows each time a value is entered into any of these text input fields.	False
CHARPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each line.	0
CHARPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average character width.	0
DEFAULTBACKBUTTON	(Windows and Blackberry only) A true/false flag that indicates whether a back button is displayed on the screen by default.	False
DISABLEABOUTMENU	A true/false flag that indicates whether to display the "About" menu option (remove from the menu).	False
DISABLEMINIMIZEMENU	A true/false flag that indicates whether the you should disable the "Minimize" menu option (remove from the menu).	True
DISABLESELFTESTMENU	A true/false flag that indicates whether to disable the "Self Test" (removefrom the menu).	False

Option	Description	Default
DISABLESHUTDOWNMENU	A true/false flag that indicates whether the "Shutdown" menu option should be disabled (removed from the menu).	False
DISPLAYIMAGESIZES	A true/false flag to control whether the available and final sizes of images are displayed. This can be useful when optimizing image sizes for use within an application.	False
DISPLAYLAYOUTGRID	A true/false flag that indicates whether or not a layout grid is displayed. This is an application design aid that helps to highlight some of the calculations made by the layout process. It can be useful during screen design.	False
FIXEDSIZECOLUMNWIDTH	Width in pixels to be used by each column. Alternatively, a percentage value may be specified by appending the '%' sign. Only used if LAYOUTHORIZONTAL is specified as FIXED SIZE.	
FIXEDSIZELINEHEIGHT	Height in pixels to be used by each line. Alternatively, a percentage value may be specified by appending the '%' sign. Only used if LAYOUTVERTICAL is specified as FIXED SIZE.	

Option	Description	Default
FONTNAME	The name of the base font typeface to use when rendering screens.	Arial
FONTSIZE	The point size of the base font typeface to use when rendering screens.	9
FONTWEIGHT	(Windows only) The "weight" of the base font typeface to use when rendering screens (expressed as a number from 1-1000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a super-thin font typeface, while a value of 1000 indicates an ultra-bold font typeface.	400
HIGHRESOLUTIONGRAPHICS	(Windows only) A true/false flag that indicates whether high-resolution graphics mode should be enabled. This effectively doubles both the vertical and horizontal resolution available for graphics, potentially resulting in a much sharper image. However, existing graphics will typically need to be rescaled (doubled in both height and width) in order to be properly sized when this mode is enabled.	False

Option	Description	Default
HORIZONTALSCROLLSPEED	(Windows only) The horizontal scroll speed of the main window (the higher the number, the faster the scrolling).	1
LEFTMARGIN	The size of the left margin in the window, in logical pixels. This is the gap allowed between the leftmost window border and the extreme lefthand side of the user screen elements.	No default value.
LAYOUTHORIZONTAL	The layout algorithm to use for the vertical axis. The following values are possible: BEST FIT FIXED SIZE LEGACY	Varies by platform.
LAYOUTVERTICAL	The layout algorithm to use for the vertical axis. The following values are possible: BEST FIT FIXED SIZE LEGACY	Varies by platform.
LOGIMAGESIZES	Similar to the "DisplayImageSizes" option, this is a true/false flag that controls whether or not available and final sizes of images are written to the client log.	False

Option	Description	Default
MAXIMISEONSTARTUP	(Windows desktop only) A true/false flag that indicates whether the window should be maximised on startup. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices.	False
MAXIMUMZOOMIN	(iPhone only) The largest scale factor permitted during a "zoom in" operation on the screen. Setting this value to 1 effectively means that no zooming in is permitted. By default it is set to 2, meaning that the user can zoom in until the screen contents are 200% of their original size.	2
MAXIMUMZOOMOUT	(iPhone only) The largest scale factor permitted during a "zoom out" operation on the screen. Setting this value to 1 effectively means that no zooming out is permitted. By default it is set to 2, meaning that the user can zoom out until the screen contents are 50% of their original size.	2
MESSAGEBOXNOFOCUS	(Windows only) A true/false flag that indicates whether or not the "enter" (return) key should be disabled when a pop-up message box appears. This helps to avoid inadvertent dismissal of errors during activities such as bar code scanning.	False

Option	Description	Default
SELECTTEXTONFOCUSCHANGE	(Windows only) A true/false flag that indicates whether or not text in an input field should be selected when focus is changed.	True
SHOWFUNCTIONKEYSINMENU	A true/false flag that indicates whether the name of each function key shown in the menu is shown (for example, "Refresh F5" rather than simply "Refresh").	False
SHOWHORIZONTALSCROLLBAR	(Windows only) A true/false flag that indicates whether a horizontal scroll bar should be displayed as part of the window. The horizontal scroll bar only becomes active if the contents of the screen are wider than the available display area.	False
SHOWMENU	A true/false flag that indicates whether the menu is displayed.	True
SHOWVERTICALSCROLLBAR	(Windows only) A true/false flag that indicates whether a vertical scroll bar should be displayed as part of the window. The vertical scroll bar only becomes active if the contents of the screen are taller than the available display area.	False

Option	Description	Default
SHOWWINDOWCONTROLS	A true/false flag that indicates whether window controls (minimise, maximise and close buttons) should be made available on the WPC window.	True
STATUSBARICONSET	(Windows only) The icon set to use for display on the status bar. There are currently two icon sets available: Basic - Host connectivity is shown by a simple tick/cross. Animated - Host connectivity icons resemble terminal screens and animate when communication with the server is taking place.	Basic
TABLEPAGECONTROLPADDINGWIDTH	This is the number of additional logical pixels by which to increase the size of the page links in table page controls. It increases the size of the "clickable" area beyond the bounds of the link text itself.	6
TABLEPAGECONTROLSPACING	This is the number of additional logical pixels by which to increase the separation between page links in table page controls. It has no effect on the size of the "clickable" area of the link itself.	0

Option	Description	Default
TOPMARGIN	The size of the top margin in the window, in logical pixels. This is the gap allowed between the topmost window (or menu) border and the extreme top of the user screen elements.	No default value.
USESYSTEMFONT	(iPhone only) A true/false flag that indicates whether the default system font should be used. When true, this overrides any font name and font size specified in the configuration.	True
VERTICALSCROLLSPEED	(Windows only) The vertical scroll speed of the main window (the higher the number, the faster the scrolling).	1
WINDOWHEIGHT	(Windows desktop only) The height of the window in logical pixels. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices.	No default value.

Option	Description	Default
WINDOWHORIZONTALALIGNMENT	(Windows desktop only) The horizontal alignment of the window on the screen. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices. May assume any of the following values: Left - The far left-hand side of the screen. Centre - The centre of the screen. Right - The far right-hand side of the screen.	Centre
WINDOWVERTICALALIGNMENT	(Windows desktop only) The vertical alignment of the window on the screen. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices. May assume any of the following values. Top - The top of the screen. Centre - The centre of the screen. Bottom - The bottom of the screen.	Centre
WINDOWWIDTH	(Windows desktop only) The width of the window in logical pixels. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices.	No default value.

Go back to the top

CLIENT.SERVER

This configuration section controls communication between the Client and the Server.

Option	Description	Default
COMPRESSIONALGORITHM	The compression algorithm to use when compressing data sent to the server. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended).	None
ENCRYPTION	The encryption algorithm to use when encrypting data sent to the server. The following encryption algorithms are available: BLOWFISH - The "Blowfish" encryption algorithm (recommended). RSA - The "RSA" encryption algorithm.	None
ENCRYPTIONKEY	The encryption key to use when encrypting data. If no encryption key is specified, the server looks for one in an encryption file instead.	No default value.
ENCRYPTIONKEYFILE	The file in which the encryption key is stored.	No default value.
ENCRYPTIONKEYHEXENCODED	A true/false flag that indicates whether the encryption key is hex-encoded.	False
LAUNCHSERVER	A true/false flag that indicates whether the client should attempt to launch and manage a server on the host it is running upon.	False

Option	Description	Default
LAUNCHIMAGE	The name of the executable file to invoke to launch the server.	No default value.
LAUNCHARGUMENTS	The arguments to pass to the executable file when the server is launched.	No default value.
LAUNCHDIRECTORY	The working directory from which you should launch the server.	No default value.
LAUNCHWINDOW	A true/false flag that indicates whether you should launch the server in a visible window. When this flag is set to false, the server process launches in a process that is not visible to the user.	False
LAUNCHTIMEOUT	The maximum period of time (in seconds) that the client should wait for the server to launch before giving up. For the purpose of this timeout, the server is considered to have successfully launched if it has managed to bind successfully to the port and can therefore offer the client a connection on which to engage in further dialogue about the server's readiness to launch the desired application.	10 (seconds)

Option	Description	Default
RESPONSETIMEOUT	The maximum period of time (in seconds) that the client will await a response from the server before timing out. A value of 0 indicates no timeout is applicable, in which case the client will wait indefinitely for a response from the server.	30 (seconds)
SERVERCOMMANDPORT	The port number of the server's command port.	No default value.
SERVERHOSTNAME	The name of the host on which the server is running, in either human readable or raw IP address format.	No default value.
SERVERPORT	The port number on which the server is listening for incoming client connections.	No default value.
SERVERSHUTDOWNTIMEOUT	The maximum period of time (in seconds) that the client will wait when it shuts down and attempts to terminate a server that it has launched.	10 (seconds)
XMLDEBUG	A true/false flag that indicates whether the client should display all incoming and outgoing XML in popup message boxes. This can be useful for debugging purposes.	False
XMLDEBUGTOLOG	A true/false flag that indicates whether the client should log all incoming and outgoing XML in the client log file. This can be useful for debugging purposes.	False

Go back to the top

CLIENT.SERVERBACKUP

This configuration section allows the Client to initiate and monitor backups against the server.

Option	Description	Default
CONFIRMBACKUP	A true/false flag that indicates whether the user needs to confirm any backup that the WPC initiates on shutdown.	False
INCREMENTALSBEFOREFULL	If WPC initiates an incremental backup on shutdown, this is the number of times that an incremental backup takes place before a full backup occurs instead. For example, on every 10th occasion, you can configure WPC to force a full backup rather than an incremental one. A zero or blank value indicates that WPC should never force a full backup.	0
LAUNCHBACKUPONSHUTDOWN	A true/false flag that indicates whether WPC should initiate a backup against the server when it shuts down.	False
USEINCREMENTALBACKUPS	A true/false flag that indicates whether WPC should initiate an incremental backup rather than a full backup on shutdown.	False

Go back to the top

CLIENT.XAI

This configuration section controls behaviour in connection with the External Application Invocation (XAI) interface.

Option	Description	Default
XMLDEBUGTOLOG	A true/false flag that indicates whether WPC should write any XAI XML messages received to its log.	False
XMLDEBUGTOSCREEN	A true/false flag that indicates whether WPC should display any XAI XML messages received to the screen through a pop-up message.	False

Go back to the top

3.9.12.3 Server Configuration

The following section details all the sections and associated options for the Server configuration type.

- SERVER.GENERAL
- SERVER.APPLICATIONCACHE
- SERVER.BACKUPRESTORE
- SERVER.BINARYFILES
- SERVER.DIALUPCONNECTION_XXX
- SERVER.HEARTBEAT
- SERVER.HOSTFUNCTIONAILTYSUPPORT
- SERVER.HOSTFUNCTIONNAMES
- SERVER.HOSTINTERFACE_RFC
- SERVER.HOSTINTERFACE_GATEWAY
- SERVER.IDENTITYPORT
- SERVER.IDENTITYSERVICE:XXX

- SERVER.JAVA
- SERVER.LOCALDATABASE
- SERVER.LOG
- SERVER.MONITOR
- SERVER.NETWORK
- SERVER.POLLDIRECTORY_XXX
- SERVER.PRINTTEMPLATE_XXX
- SERVER.RELAYCONNECTIONS
- SERVER.SERVEREXITS
- SERVER.STORAGE
- SERVER.COMMANDPORT_XXX
- SERVER.HTTPCONTROLPORT_XXX
- SERVER.GUIAPPLICATIONPORT_XXX
- SERVER.HTTPAPPLICATIONPORT_XXX
- SERVER.TERMINALAPPLICATIONPORT_XXX
- SERVER.XMLGATEWAYPORT
- SERVER.RELAYPORT_XXX
- SERVER.CUSTOMPORT_XXX
- SERVER.CUSTOMSERIALDEVICE_XXX
- SERVER.TERMINALTYPE_XXX

SERVER.GENERAL

This section contains parameters of a general nature that affect the operation as a whole.

Option	Description	Default
SERVERID	A unique identifier to distinguish this particular instance of Application Server from any others that may be running on the same machine and/or against the same SAP instance. You can use this identifier to perform restricted data downloads. If left blank or not set, then the server ID is set to the name of the host machine on which the server runs.	As noted.
SERVERGROUP	This is an arbitrary identifier that is used to group together Application Servers that have common behaviour (for example, all those in a certain geographic location) so that SAP-side processing can threat them in a similar fashion.	- (no group)
RESTRICTTOHOST	Name of the host on which the Application Server must run. If a different host name is detected, the server shuts down immediately.	- (any host)
RESTRICTTOUSER	The user ID that you must use to start the server. The server execution may require specific authorizations, and thus a special user may need to start specified, and this user is not the current user, the server shuts down immediately.	- (any user)
LICENSEFILE	The name of the license file.	license.dat

Option	Description	Default
AUTORESTART	A true/false flag that, when enabled, causes the server to automatically restart itself if it is not shut down properly (for example, in the event of the process crashing or being killed).	False
DATEFORMAT	The format in which dates are to be entered and displayed.	dd/mm/yyyy
TIMEFORMAT	The format in which you need to enter times and display.	hh:mm:ss
TIMEZONE	The time zone in which the server runs (overrides the default setting that the Java Virtual Machine supplies.)	Server default
UNICODEENABLED	A true/false flag that switches between Unicode and non-Unicode modes. If enabled, the server communicates with SAP in Unicode mode and also generates all outputs (for example, log files) in Unicode format. This flag should be enabled if foreign language support is desired.	False
SHUTDOWNTIMEOUT	During shutdown, this is the maximum time interval (in seconds) that the server waits for outstanding tasks to complete before forcibly terminating them.	30
ABORTIFOUTOFMEMORY	A true/false flag that indicates whether the Application Server should terminate if it runs out of memory. It is generally advisable to enable this option, as running out of available memory can potentially cause unpredictable application behaviour.	True

Option	Description	Default
CLIENTSCROLLOVERRIDE	A true/false flag that, when enabled, causes all tables to be converted to client scrollable tables.	False
DUMPCONFIGTOLOG	A true/false flag that, when disabled, prevents the configuration being written to the log on startup.	True

Go back to the top

SERVER.APPLICATIONCACHE

This section contains information controlling the behaviour of the application cache.

Option	Description	Default
APPLICATIONREFRESHINTERVAL	The period of idle time (in seconds) between checks to ensure that the server currently has the latest version of the application data. A value of 0 (zero) means that applications are only checked for changes when the server first starts up.	30

Option	Description	Default
APPLICATIONREFRESHLOCK	Controls locking of applications whilst they are being refreshed. The following values are permitted: None - Users are permitted to proceed on initial entry as soon as the initial application is loaded (without waiting for updates/changes to download). Users may continue to use the application while it gets updated in the background. Current - Users are not permitted to proceed on initial entry until available updates/changes to their first application are applied. Users are temporarily locked out of an application if a background update to it takes place whilst they are using it. All - Users are not permitted to proceed on initial entry until available updates/changes are applied to all their applications. Users are temporarily locked out if a background update takes place to any application.	None
REFRESHBYFUNCTION	A true/false flag that controls whether the server downloads applications all at once, or one function at a time. Refreshing applications by function is somewhat slower but uses far less memory, making it an appropriate choice for resource constrained devices.	False

Option	Description	Default
FUNCTIONCACHESIZE	The maximum number of functions that the server will keep in memory at any given point in time. Functions beyond this number are flushed on a least recently used basis, and need to be reloaded the next time they are needed. If not specified or left as zero, no function cache limit is imposed.	0 (no limit)
MAINTAINHOSTCONNECTIVITY	Whether or not to keep the connection open once the application cache has been updated.	True
PRELOADERDISABLED	Indicates whether the function preloader should be disabled. The function preloader uses statistical information to attempt to predict which functions users are likely to require, and loads them into memory in anticipation. It improves user response time at the cost of a small performance overhead on the application server. The preloader should not need to be disabled in most situations.	False

Option	Description	Default
PRELOADERPROBABILITYTHRESHOLD	This is an integer value from 0-100 that controls the sensitivity of the function preloader (when it is in use). If, based on historical statistics, the likelihood that a user will request a function from their current point within the application exceeds this percentage, the preloader will load the function, assuming there is room in the function cache to do so. The lower the value, the more sensitive the function preloader becomes, and vice versa. For example, if the probability threshold were set to 10, then the preloader would attempt to load any function with a 10% or higher chance of being requested next, making it considerably more sensitive to user activity. If the probability were set to 50, then the preloader would only attempt to load functions with a 50% or better chance of being requested next, making it somewhat less sensitive.	30

Go back to the top

SERVER.BACKUPRESTORE

This section contains information controlling the Application Server backup and restore facility.

Option	Description	Default
BACKUPDIRECTORY	The name of the directory in which the server should store its backup files. If left blank or not specified, the backup/restore facility is disabled.	No default value.
FULLBACKUPITEMS	A semi-colon separated list of files and/or directory pathnames that should be written to the backup file when a full backup takes place.	- (no files)
INCREMENTALBACKUPITEMS	A semi-colon separated list of files and/or directory pathnames that should be written to the backup file when an incremental backup takes place. The files selected must be a subset of the items selected in the FullBackupItems list.	Same as for full backup.
AUTOFULLBACKUPTIMES	A comma-separated list of times at which the server should automatically trigger a full backup (in the format HH:MM:SS). Obviously it is undesirable to have a backup running while data is altered, so if this feature is used it should typically be set to run at time(s) during which the server is completely idle.	- (never)

Option	Description	Default
AutoIncrementalBackupInterval	If specified, this value denotes a period of time (in seconds) after which the server automatically performs an incremental backup of any files present in the incremental backup list that are changed. Incremental backups continue to take place automatically while the server is "live". If set to zero or not specified, the server does not automatically perform incremental backups.	0 (disables automatic incremental backups)
FullBackupRetentionCount	Specifies the maximum number of historical full backups to retain on disk. All incremental backup information associated with each full backup is also retained.	1
UseCompression	A true/false flag that indicates whether the backup file should be compressed as it is written. This has the effect of slowing the backup process somewhat, but the resultant backup file is usually considerably smaller in size.	False
CompressionAlgorithm	The compression algorithm to use if compression is to take place. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.

Go back to the top

SERVER.BINARYFILES

This section contains information pertaining to binary files (such as images) the server uses.

Option	Description	Default
DownloadDirectory	The directory in which the server should store binary files downloaded from the host. If not set, this parameter defaults to "." (the current working directory).	As noted.
UploadDirectory	The directory in which the server should look for binary files to be uploaded to the host. If not set, this parameter defaults to "." (the current working directory).	As noted.
PackageRefreshInterval	The time interval (in seconds) between checks to see whether there are any packages to be downloaded from the host.	900 (seconds)
DynamicBinaryRefreshInterval	The time interval (in seconds) between synchronization checks of dynamic binary files stored in the VTI_BINARY_FILE table.	900 (seconds)
HexModeEncoding	This parameter specifies whether to transfer binary files in raw hex format, or in raw binary format. Raw binary (false) is faster and takes less space, but can encounter code page conversion problems between platforms using different data storage types, for example, EBCDIC host to ASCI client.	True
MaintainHostConnectivity	Whether or not to keep the connection open once the application cache is updated.	True

Option	Description	Default
MemoryRetentionPeriod	The period of time (in seconds) for which the server retains the contents of any binary files in memory after they are accessed. If the specified time interval passes without the binary file is accessed, it is flushed from memory to help conserve memory usage. Any subsequent access to the same binary file forces the server to re-read it from disk. A setting of zero (0) indicates you should immediately flush the binary files from memory after they are read. A negative setting indicates that you should retain all binary files in memory indefinitely.	-1
ErrorRetryInterval	If the server experiences an error when it is downloading or refreshing a binary file in the background, this is the number of seconds that it waits before it tries again.	300 (seconds)
DefaultBinaryGroup	The default binary group to which newly created binary files should be assigned (e.g. images created from a drawing object field), assuming no binary group is specified.	None (no group)
UpdateHostMd5Hashes	A true/false flag that controls whether or not the server will update the SAP host system with an MD5 hash for any binary files that it receives without one.	True
FileLocation:XXX	Specifies a file location (directory) into which downloaded binary files can be placed. The XXX should match the file location code specified in the SAP workbench. The value assigned to the configuration parameter should be the pathname to the directory to be targeted.	No default value

Go back to the top

SERVER.DIALUPCONNECTION_XXX

These section(s) contain information pertaining the dial-up connections that the server can make use of to connect to other networks. This includes the use of Virtual Private Networks (VPNs). You can set the server to automatically connect and disconnect to such networks at particular times of day. User-written exits can also make use of dial-up connections defined in the configuration file.

> *Important:* This feature is only available on the Windows and Windows CE platforms.

Option	Description	Default
DialupSettingsName	The name of the Windows dial-up settings that you should use to dial the desired network.	No default value.
DialupUsername	The username to use when dialing the remote network. If left blank or not specified, this defaults to the username stored with the dialup entry.	As noted.
DialupPassword	The password to use when dialing the remote network. If left blank or not specified, this defaults to the password stored with the dialup entry.	As noted.

Option	Description	Default
AutoConnectionTimes	A comma-separated list of time ranges during which the server should attempt to ensure that a connection to the desired network is automatically established. Times must match the pattern HH:MM:SS, and be in 24-hour format. A valid time range, for example, may be 17:00:00-18:00:00, indicating an hour-long window between 3pm and 4pm. To ensure continuous connection, enter 00:00:00-23:59:59. If left blank or not specified, this parameter indicates that the server should not automatically dial the target network. You can establish a connection to the network still in such a situation through the use of user-written exits code.	- (no automatic connection)
AutoConnectionTimeout	The maximum period of time (in seconds) that the server should wait whilst attempting to establish a connection, before assuming that you cannot make the connection and temporarily give up.	30 (seconds)
ReconnectionInterval	The time interval (in seconds) that the server waits after an unsuccessful attempt to connect to the network, before trying again.	120 (seconds)

Go back to the top

SERVER.HEARTBEAT

This section contains information about the way in which the server regularly reports its status to the central SAP host.

Option	Description	Default
HeartbeatInterval	The period of idle time (in seconds) between SAP status notifications when not connected.	15

Option	Description	Default
HeartbeatConnectedInterval	The period of idle time (in seconds) between SAP status notifications when connected. If left blank or not specified, this defaults to the same value as the HeartbeatInterval.	As noted.
HeartbeatHostName	An override for the name of the host reported to SAP during the heartbeat status notification. If left blank or not set then the name of the host machine on which the server is runs is used (in the vast majority of cases this is the most appropriate setting).	As noted.
HeartbeatHostAddress	An override for the address of the host reported to SAP during the heartbeat status notification. It must be in the standard IP address format xxx.xxx.xxx.xxx. If left blank or not set then the IP address of the host machine on which the server runs is used (in the vast majority of cases this is the most appropriate setting).	As noted.
HeartbeatLossSeverity	Allows control over the severity of the log messages written to the log when heartbeat is lost. Possible values are as follows: I - Log messages for heartbeat loss are treated as information (INFO) messages. W - Log messages for heartbeat loss are treated as warning (WARN) messages. E - Log messages for heartbeat loss are treated as error (ERROR) messages.	E

Go back to the top

SERVER.HOSTFUNCTIONALITYSUPPORT

This section is used to tell the server what types of functionality the back-end host supports. It permits certain areas of functionality to be disabled, for example, if the host does not support. You can make the server to run in standalone mode if all HostFunctionalitySupport options are disabled.

Option	Description	Default
FunctionalitySupport:Application	Indicates whether the host supports download and refresh of applications.	true
FunctionalitySupport:BinaryFiles	Indicates whether the host supports download of binary files.	true
FunctionalitySupport:Heartbeat	Indicates whether the host supports heartbeat messages.	true
FunctionalitySupport:LocalDatabase	Indicates whether the host supports download, refresh and upload of local database tables.	true
FunctionalitySupport:Packages	Indicates whether the host supports download of packages.	true
FunctionalitySupport:SessionManager	Indicates whether the host supports session manager functionality.	true

Go back to the top

SERVER.HOSTFUNCTIONNAMES

This section controls the functions that the server invokes against the back-end host(s). You can override all of the following configuration items at the host interface level if they are placed inside the configuration file section referring to the host interface in question. You should not change these defaults unless the Sky support instructs.

Option	Description	Default
Option	Description	

Option	Description	Default
HeartbeatFunctionName	The name of the function invoked by the heartbeat (defaults to /SKY/VTI_HEARTBEAT).	
LdbDownloadFunctionName	The name of the function invoked by a local database download (defaults to /SKY/VTI_DOWNLOAD_LOCAL_DATA).	
LdbRefreshFunctionName	The name of the function invoked by a local database refresh (defaults to /SKY/VTI_REFRESH_LOCAL_DATA).	
LdbUploadFunctionName	The name of the function invoked by a local database upload (defaults to /SKY/VTI_UPLOAD_LOCAL_DATA).	
RelayConnectionsFunctionName	The name of the function used to notify the host of the relay connections currently available on the server's relay ports (defaults to /SKY/JAVA_RELAY_CONNECTIONS).	
SelectBinaryFunctionName	The name of the function used to retrieve the data associated with a binary file (defaults to /SKY/VTI_SELECT_BINARY_DATA).	
SelectDefinitionsFunctionName	The name of the function used to retrieve an application definition (defaults to /SKY/VTI_SELECT_DEFINITIONS).	As noted.
SelectFunctionDataFunctionName	The name of the function used to retrieve the definition of a single function (defaults to /SKY/VTI_SELECT_FUNCTION_DATA).	As noted.

Option	Description	Default
Select PackageFunctionName	The name of the function used to retrieve details about a package (defaults to /SKY/VTI_SELECT_PACKAGE).	As noted.
SelectPackagesFunctionName	The name of the function used to retrieve information about the packages relevant to the server instance (defaults to /SKY/VTI_SELECT_PACKAGES).	As noted.
SelectTimestampsFunctionName	The name of the function used to retrieve information about the applications relevant to the server instance (defaults to /SKY/VTI_SELECT_TIMESTAMPS).	As noted.
SessionManagerFunctionName	The name of the function used to process session manager calls (defaults to /SKY/VTI_SESSION_MANAGER).	As noted.
UpdateMd5HashFunctionName	The name of the function used to send the host an MD5 hash for a binary file that doesn't currently have one (defaults to /SKY/VTI_UPDATE_MD5_HASH).	As noted.
UpdatePackageStatusFunctionName	The name of the function used to notify a change in the current status of a package deployment (defaults to /SKY/VTI_UPDATE_PACKAGE_STATUS).	As noted.
UploadBinaryDataFunctionName	The name of the function used to upload data of a binary file (defaults to /SKY/VTI_UPLOAD_BINARY_DATA).	As noted.

Go back to the top

SERVER.HOSTINTERFACE_RFC

These section(s) contains the information that the server requires to establish a direct RFC connection to one or more SAP instance(s).

Option	Description	Default
HostInterfaceType	Must set to "RfcClient" for direct RFC connections.	RfcClient
SapClient	Numeric SAP client identifier.	No default value.
SapUser	The SAP user id to use. Should be a CPIC user.	No default value.
SapPassword	The SAP password to use.	No default value.
SapLanguage	The SAP language identifier, for example, E (English).	E
SapTraceLevel	The SAP RFC trace level (should be left as 0 in normal operation).	0
SapUseLB	A true/false flag to instruct the server whether to use the SAP load balancing feature.	No default value.
SapLBHost	Host for load balancing.	No default value.

Option	Description	Default
SapLBSystem	SAP system identifier for load balancing.	No default value.
SapLBGroup	The SAP load balancing group.	No default value.
SapHost	A specific SAP host to connect to (alternative to load balancing).	No default value.
SapSystem	A specific SAP system identifier to connect to (alternative to load balancing).	No default value.
SapConnectionTimeout	The number of seconds to wait before terminating a attempt to connect to a SAP system.	0 (no timeout)
JniLogging	A true/false flag that controls whether JNI tracing information is written to the log. This trace is primarily intended for Sky support.	false (no JNI tracing)
UseTransferFields	Indicates whether information on local database table fields should be sent during table downloads, refreshes, and uploads. For SAP systems this is not needed and should be set to "false".	False

Option	Description	Default
MaximumOptimizationLevel	This is a numeric value that limits optimization behaviour during session manager processing. The highest optimization currently supported is 1. A value of 0 completely turns off session manager optimization. In most scenarios there is no need to set this value.	1
MinimumNumberHandlers	The minimum number of request handlers to keep on hand to process incoming Access Gateway requests. ***Note:*** This value only pertains to incoming requests made through an Access Gateway port. Unless the VTI server also acts as an Access Gateway, there is no need to set this value.	0
MaximumNumberHandlers	The maximum number of request handlers to keep on hand to process incoming Access Gateway requests. ***Note:*** This value only pertains to incoming requests made through a connection on an Access Gateway port. Unless the VTI server also acts as an Access Gateway, there is no need to set this value.	0 (no limit)

Go back to the top

SERVER.HOSTINTERFACE_GATEWAY

These section(s) contains the information that the Application Server requires to establish a connection to a back-end host system through a gateway.

Option	Description	Default
HostInterfaceType	You must set to "XmlClient" for Access Gateway connections.	XmlClient
XmlGatewayHosts	A comma-separated list of the host(s) on which Access Gateway(s) run(s). Each value can take the form of either a host name or an IP address.	No default value.
XmlGatewayHostSelection	The manner in which an Access Gateway should be selected from the list of those available. The following values are permitted: Random - You should select the Access Gateway at random. Sequential - Access Gateway entries that appear earlier in the list have preference over those that appear later.	Sequential
XmlGatewayPort	The port number on which the Access Gateway (s) are listening for incoming XML. ***Note:*** When multiple Access Gateways are in use, you must configure them all to use the same port.	No default value.
XmlGatewayResponseTimeout	The maximum period of time (in seconds) for which the server waits for an initial response from the Access Gateway before timing the transaction out.	30 (seconds)
XmlGatewayStallTimeout	The maximum period of time (in seconds) for which the server allows an incoming response to stall (that is send no further data) before timing the transaction out.	30 (seconds)

Option	Description	Default
UseCompression	A true/false flag that indicates whether the XML data sent to the gateway should be compressed. This greatly reduces bandwidth consumption.	False
CompressionAlgorithm	The compression algorithm to use when compressing data sent to the Access Gateway. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.
UseEncryption	A true/false flag that indicates whether the XML data sent to the gateway should be encrypted. Obviously, this is a more secure option, particularly over a public network.	false
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to the Access Gateway. The following encryption algorithms are available: AES - The AES (Rijndael) encryption scheme. BLOWFISH - The "blowfish" encryption scheme.	No default value.
EncryptionKey	The encryption key to use when encrypting data. If no encryption key is specified, the server looks for one in an encryption key file instead.	No default value.
EncryptionKeyFile	The file in which the encryption key is stored.	No default value.

Option	Description	Default
EncryptionKeyHexEncoded	A true/false flag that indicates whether the encryption key is hex-encoded.	false
EncryptionKeyStrength	If handshaking encryption is in use, this parameter specifies the length of the randomly generated session key to be used following the initial handshake (in number of bits).	No default value.
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the Access Gateway. If left blank, no initial handshaking is performed. The following encryption algorithms are available: RSA - The "RSA" encryption algorithm (recommended).	No default value.
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the Access Gateway. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	No default value.
HandshakeEncryptionKeyHex-Encoded	A true/false flag that indicates whether the handshaking encryption key is hexencoded.	False
XmlDebug	A true/false flag that indicates whether all XML data sent to, and received from, the Access Gateway should be logged. This facility can be useful for debugging in some situations, but it generates large log files and should generally be left off.	False

Option	Description	Default
UseCondensedXml	A true/false flag that indicates whether or not XML should be sent in "condensed mode". Barring problem scenarios, this option should normally be enabled. Turning it off has a performance impact.	True
UseTransferFields	Indicates whether information on local database table fields should be sent during table downloads, refreshes, and uploads. For SAP systems this is not needed and should be set to "false".	False
MinimumNumberHandlers	The minimum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0
MaximumNumberHandlers	The maximum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via a connection on an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0 (no limit)

Go back to the top

SERVER.IDENTITYPORT

This section describes the options available to implement an identity management port. Typically, identity requests are directed through a normal gateway port, but this configuration gives you the option to expose a specific identity port.

Option	Description	Default
Port	The port number to listen on for incoming messages.	None
DefaultIdentityService	The name of an implemented identity management service to use as the default if the calling Secure Container does not specify one.	None
UseCompression	A true/false flag that indicates whether the XML data sent to connected clients on the port should be compressed.	True
CompressionAlgorithm	The compression algorithm to use when compressing data sent to connected clients on the port. The following compression algorithms are available: GZIP - The GZIP compression algorithm. ZLIB - The ZLIB compression algorithm.	GZIP
UseEncryption	A true/false flag that indicates whether the XML data sent to connected clients on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	False

Option	Description	Default
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to connected clients on the port. The following encryption algorithms are available: AES - The AES encryption algorithm. BLOWFISH - The Blowfish encryption algorithm. DES - The DES encryption algorithm. 3DES - The triple-DES encryption algorithm.	None
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the remote partner. If left blank, no initial handshaking is performed. The following encryption algorithms are available: RSA - The RSA encryption algorithm.	None
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the remote partner. If no encryption key is specified, the server looks for one in an encryption key file instead.	None
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	None
HandshakeEncryptionKeyHexEncoded	A true/false flag that indicates whether the handshaking encryption key is hex encoded.	False

Option	Description	Default
DefaultIdentityService	The name of a implemented identity management service to use as the default if the calling Secure Container does not specify one.	None

Go back to the top

SERVER.IDENTITYSERVICE:XXX

The identity service section is used to implement SkyMobile identity management services. The ones configured here are the inbuilt Sky services. Customers can also implement their own service modules, in which case they implement their own custom sections.

SERVER.IDENTITYSERVICE:LDAP

Option	Description	Default
SERVICETYPE	LDAP	
USEDOMAIN	A flag to indicate whether to capture and/or use a domain as part of the login process.	False
DOMAIN	If USEDOMAIN is true, any value entered for this parameter is always picked up as the user's domain. The user is not prompted to enter their domain.	None
DEFAULTDOMAIN	If USEDOMAIN is true and DOMAIN is not specified, any value entered for this parameter is the default value for the domain field when the user is prompted for the domain.	None

Option	Description	Default
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.	0
SERVER	The name or address of the LDAP server.	None
PORT	The port on which to connect to the LDAP server.	389 if SSL is not enabled 443 if SSL is enabled
BINDCONTEXT	The initial bind context to use when attaching to the LDAP directory.	"/"
TIMEOUT	The maximum number of seconds for which to wait for a response from the LDAP server before giving a timeout error.	10
USESSL	A flag to indicate whether or not to use SSL when communicating with the LDAP server.	False
SSLKEYSTORE	For use only when SSL is enabled, this is the pathname to the SSL key store file.	None
SSLKEYSTOREPASSWORD	For use only when SSL is enabled, this is the password to the SSL key store file.	None
SSLTRUSTSTORE	For use only when SSL is enabled, this is the pathname to the SSL trust store file.	None

Option	Description	Default
SSLTRUSTSTOREPASSWORD	For use only when SSL is enabled, this is the password to the SSL trust store file.	None
SSLDEBUG	For use only when SSL is enabled, this is a flag to indicate whether you should enable SSL debugging.	False
REGISTERSESSIONS	To enable advanced session security (only required when enabled in YVTI in the MEAP)	False
HOSTINTERFACE	The name of the host interface to use when communicating with SAP. (Only required if REGISTERSESSIONS is enabled)	None

SERVER.IDENTITYSERVICE:SAP

Option	Description	Default	
SERVICETYPE	SAP		
HOSTINTERFACE	The name of the host interface to use when communicating with SAP.	None	
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.		0

SERVER.IDENTITYSERVICE:SKY

Option	Description	Default
SERVICETYPE	SkyMobile	

Option	Description	Default
HOSTINTERFACE	The name of the host interface to use when communicating with SAP.	15
LOGFAILURES	Log unsuccessful identity check attempts	False
LOGSUCCESSES	Log successful identity checks	False
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.	0
SENDPASSWORDTOHOST	Send the identity password to the host to be used by the server by user switching	False

Go back to the top

SERVER.JAVA

This section controls invocation of the Java Virtual Machine (JVM).

Option	Description	Default
JAVACOMMAND	The path to the executable used to launch the Java Virtual Machine (for example, `java.exe`). Under Windows, this should be in short file name (MS-DOS 8.3) format, for maximum compatibility. If not specified, the value of the environment variable SKY_JAVA_CMD is used instead.	As noted.
JAVAARGUMENTS	Any additional arguments to be passed to the Java command line. Common examples are arguments to increase the maximum memory ceiling (-Xmx128m) or run with a reduced signal set (-Xrs).	- (no arguments)

Go back to the top

SERVER.LOCALDATABASE

This section contains information pertaining to the locally cached database tables.

Option	Description	Default
MINIMUMNUMBERUPDATETHREADS	The minimum number of threads available for use in keeping the local database tables up to date.	1
MAXIMUMNUMBERUPDATETHREADS	The maximum number of threads available for use in keeping the local database tables up to date.	3
KEEPHOSTCONNECTIONS	A true/false flag that controls whether the local database processing threads attempt to maintain a continuous connection to the host.	True
ERRORRETRYINTERVAL	If the server experiences an error when it updates a table in the background, this is the number of seconds that it waits before it tries again.	3600 (seconds)
TRANSFERBUFFERSIZE	The maximum number of bytes that may be transferred to the Application Server from the host. This setting prevents large transfers from overwhelming devices with a low amount of memory or processing power. Instead, the transfer is broken into multiple 'chunks' that are transferred in multiple calls.	0 (no restriction)

Go back to the top

SERVER.LOG

This section contains information about the logging and tracing capabilities.

Option	Description	Default
DIRECTORY	The directory into which the server writes its log files. If not configured, this parameter defaults to "." (the current working directory).	As noted.
STRATEGY	The logging strategy to be adopted. Must be one of the following: D - A new log file is to be created daily. W - A new log file is to be created weekly. P - A new log file is to be created for each process instance O - Overwrite the log file each time the server is restarted.	O
EXCLUDES	Indicates the types of log messages that you should exclude from the log file. There are six types of log messages - Information (I), Warning (W), Performance Track (P), Trace (T), Error (E) and Abort(A). The log message types to exclude should all be concatenated together. Thus, to exclude all Information and Trace messages from the log file, the appropriate setting would be "IT".	- (no types excluded)

Option	Description	Default
MAXFILES	This setting restricts the maximum number of log files the Application Server keeps under D, W, and P strategies. If left blank or not set then no maximum is enforced. 0 indicates no maximum.	0 (no maximum)
SHOWMILLISECONDS	A true/false flag that controls whether milliseconds are shown in log messages. This feature is useful for measuring sub-second timings and events.	False
PERFORMANCETRACKING	A true/false flag that controls whether or not performance tracking messages are written to the VTI log.	False

Option	Description	Default
MESSAGESEVERITYOVERRIDE:XXX	This configuration directive permits specific override of the behaviour of individual log file messages. There can be multiple instances of this directive within the [Log] section. By default, XXX is assumed to be the name of a log message resource identifier (as listed in the `vti.sample.properties` file) - for example, VTI_MSGFMT_VTI_STARTED. However, if the value specified in place of the XXX is enclosed in double quotes, then it is instead assumed to be a text pattern to match against the log message text (for example, "Cannot read directory *". The value assigned to the directive should be one of the available types of log message (I, W, P, T, E, or A). This causes all log messages that match either the specified resource identifier or the specified text pattern, to be treated as though they had the severity specified. Thus, you can reduce the individual error messages in severity to warnings, for example, or just information messages. Additionally, you can specify a value of "X" that causes matching messages to be completely ignored (excluded). You must use this feature with caution as you are generally not recommended to ignore or downgrade error messages.	N/A

Go back to the top

SERVER.MONITOR

This section contains setup information for internal monitoring capability.

Option	Description	Default
USEMONITOR	A true/false flag that enables or disables the internal monitoring capability.	False
MONITORINTERVAL	The time interval (in seconds) between monitoring status checks.	60 (seconds)
GARBAGECOLLECT	A true/false flag that indicates whether Java garbage collection (that expends resources in an attempt to free up memory) should be performed as part of each monitoring status check. If set to true, garbage collection takes place each time a monitoring status check occurs. If set to false, the Java Virtual Machine manages garbage collection.	False
LOGMEMORYUSAGE	A true/false flag that indicates whether current memory usage information should be written to the log as part of each monitoring status check. If the both this option and the GarbageCollect option are set to true, memory usage information is logged both before and after the garbage collection takes place.	False
LOGTHREADCOUNT	A true/false flag that indicates whether the number of active threads should be written to the log as part of each monitoring status check.	False
LOGTHREADNAMES	A true/false flag that indicates whether the names and states of active threads should be written to the log as part of each monitoring status check.	False

Option	Description	Default
CLEARRFCTRACEFILES	A true/false flag that indicates whether to automatically delete logging/trace files that the SAP RFC library created.	False
RFCTRACEEXPIRYDAYS	The maximum number of days for which to retain SAP RFC trace files when automatic deletion is enabled.	7 (days)

Go back to the top

SERVER.NETWORK

This section contains information on the servers interaction with the network.

Option	Description	Default
AcceptTimeout	The maximum period of time (in milliseconds) for which the thread associated with each port blocks while waiting for new incoming connections. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)

Option	Description	Default
BindTimeout	The maximum period of time (in seconds) for which the server waits while attempting to bind to its listening ports. If another process is listening on its designated ports, or if the bind process fails, then the server is not be able to establish the listening port. Once the timeout occurs, it closes down with an appropriate error message. If this parameter is set to zero or not specified, then the server waits indefinitely to bind to its listening ports in the hope that they eventually become available.	0 (no timeout)
CheckForHostAddressChanges	A true/false flag that indicates whether the server should monitor the available network interfaces for changes to its IP address and/or host name. In situations where the network interface configuration is not subject to change, this can represent an unwanted processing overhead.	True
HostNameResolution	A true/false flag that indicates whether host name resolution is desired. Resolution of the host name can sometimes be a considerable overhead on performance. For this reason it is disabled by default. However, you should enable in situations where performance is not an issue, and display of the hostname rather than the IP address is felt to be of benefit.	False

Option	Description	Default
NetworkErrorSampleSize	The sample size for network error monitoring. The server maintains a moving window of the success or failure of the last n network calls, where n is the value of this parameter. If not set or left as zero, any network error is treated as fatal for the port or connection on which it occurs. For most well-tuned networks this should be the chosen setting.	0 (no error sampling)
NetworkErrorThreshold	The error threshold for network error monitoring. Must be less than or equal to the NetworkErrorSampleSize parameter. When this number of network errors occurs within the sample, the server will conclude that the connection or listening port has unrecoverable network problems and will close it down. It will then attempt to re-establish the connection or port in the hope of obtaining a more stable socket.	0 (no error sampling)
PreferredIpAddresses	A comma-delimited list of addresses and/or address ranges that the server will prefer to assume in a multihomed network scenario (in order of preference). These addresses must be in raw IP address form (xxx.xxx.xxx.xxx). An asterisk (*) can be used as a wildcard to substitute for any single value within the address. It is also possible to specify a valid range for a given octet by specifying the upper and lower boundaries, separated by a hyphen (for example, 192.168.2.100-200).	- (no preferred addresses)

Option	Description	Default
ReadTimeout	The maximum period of time (in milliseconds) for which the threads associated with connections on this port will block while waiting for input. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)
UseNativeSockets	A true/false flag that indicates whether native socket calls should be used instead of the capabilities provided by the Java virtual machine. This can sometimes provide additional information when errors occur, and will typically improve performance.	false
ValidIpAddresses	A comma-delimited list of addresses from which incoming connections will be accepted. These addresses can either be in humanreadable form, or in IP address form (xxx.xxx.xxx.xxx). An asterisk (*) can be used as a wildcard to substitute for any single value within the address. It is also possible to specify a valid range for a given octet by specifying the upper and lower boundaries, separated by a hyphen (for example, 192.168.2.100-200). If this parameter is specified, then any incoming connection from a host not listed will be rejected. If this parameter is left blank or not specified, then by default all incoming connections are considered valid regardless of the host from which they originate. Note that this parameter can be overridden for individual ports.	- (all addresses are valid)

Option	Description	Default
InvalidIpAddresses	A comma-delimited list of addresses from which incoming connections will not be accepted. This format of this list is identical to that described for the valid IP address list. Note that the invalid IP address list has a higher order of precedence than the valid IP address list, so a host which matches both lists is considered invalid. Note that this parameter can be overridden for individual ports.	- (no addresses are invalid)

Go back to the top

SERVER.POLLDIRECTORY_XXX

These section(s) contain information pertaining to one or more poll directories that the Application Server should monitor for incoming files. When an incoming file is detected, a user written Java exit automatically invokes to take any action that may be appropriate.

Option	Description	Default
IncomingDirectory	The pathname to a directory in which an external process places incoming files.	No default value.
ArchiveDirectory	The pathname to a directory in which files the server places the files once they are successfully processed. If left blank or not present, then any successfully processed files are deleted rather than being transferred to an archive directory.	As noted.

Option	Description	Default
ErrorDirectory	The pathname to a directory in which the server places the files once they are unsuccessfully processed (that is found to have errors). If left blank or not present, then any files that you cannot process are deleted rather than being transferred to an error directory.	As noted.
IncomingFilePattern	A filename pattern that is used to check whether files in the incoming directory are suitable for processing. Wildcard characters are permitted: an asterisk (*) matches any number of characters, and a question mark (?) matches any single character. If left blank or not present, then any file in the incoming directory is considered to be suitable for processing.	As noted.
PollFrequency	The time interval (in seconds) between polls of the incoming file directory.	No default value.
PollFrequencyMilliseconds	The time interval (in milliseconds) between polls of the incoming file directory. This interval is added to the number of seconds specified in the "PollFrequency" item.	0
LogIncomingFiles	A true/false flag that indicates whether log messages should be written for each incoming file that is detected.	false

Option	Description	Default
StopPollingOnError	A true/false flag that indicates the normal behaviour for the poll directory when an error occurs. If set to true, all polling will stop when an error is encountered, and polling must be manually restarted from the web status page. If set to false, polling will continue to take place after an error is encountered.	false
ShellCommand	The command that should be executed to invoke a system shell. Normally the server is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh --s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by the server, try changing this parameter to "cmd.exe /c".	As noted.
SystemCommand	The command to be executed whenever an incoming file is detected. If not specified or left blank, it is assumed that no system command needs to be executed.	As noted.

Option	Description	Default
ProcessCommandViaInputStream	A true/false flag to control whether or not the system command should be read from an input stream. In most cases this flag should not need to be explicitly set as it defaults to behaviour that is normally appropriate for the target OS.	As noted.
PollDirectoryExit	The fully qualified name of the userwritten Java exit to be invoked when an incoming file is detected. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.

Go back to the top

SERVER.PRINTTEMPLATE_XXX

These section(s) contain information pertaining to one or more print templates that the Application Server then makes available to user written Java exits.

Option	Description	Default
TemplateFileName	The pathname of the text file containing the raw print template information.	No default value.

Option	Description	Default
IgnoreUnknownVariables	A true/false flag that specifies whether or not any unknown substitution variables found in the template file should be ignored. Normally this gives rise to an error. However, if this option is enabled, processing silently continues and the unknown substitution variable is left unchanged. This flag can be useful in cases where the print template is a postscript file containing characters that resemble substitution variables.	False
CreateWorkFiles	A true/false flag that indicates whether to create a work file containing the output from the print template when it is invoked.	True
WorkDirectory	The pathname to the directory into which work files should be written.	No default value.
WorkFileExtension	The file extension to use when creating work files.	.txt
DeleteWorkFiles	A true/false flag that indicates whether work files should be immediately deleted following processing.	True
WorkFileExpiryDays	The maximum number of days for which to retain work files (where they are not deleted immediately after being processed).	0
ExecutePrintCommand	A true/false flag that indicates whether a print command should be executed when the print template is invoked.	True

Option	Description	Default
ShellCommand	The command that should be executed to invoke a system shell. Normally the server is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh --s If you are running under Windows 2000, XP or 2003, and find that the print command you are executing works correctly from the command line, but does not work when invoked by the server, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
PrintCommand	The system command to be executed in order to print the contents of the work file. This system command may contain substitution variables that will be replaced with a corresponding runtime value. A list of the available substitution variables is as follows: &century& - The current four-digit century. &day& - The current two-digit day of the month. &filedir& - The pathname of the work file directory. &filename& - The name of the work file. &filepath& - The full pathname of the work file. &hour& - The current two-digit hour. &minute& - The current two-digit minute. &month& - The current two-digit month. &second& - The current two-digit second. &year& - The current two-digit year.	No default value.
ProcessCommandViaInputStream	A true/false flag to control whether or not the system command should be read from an input stream. In most cases this flag should not need to be explicitly set as it defaults to behaviour that is normally appropriate for the target OS.	As noted.
PrintCommandTimeout	The maximum period of time (in seconds) for which to wait for a print command to finish executing. If the print command takes longer than this period of time to complete, it is terminated by the server. A timeout value of 0 means that no timeout is applicable.	10 (no timeout)

Option	Description	Default
LogPrintCommands	A true/false flag that indicates whether information messages should be written to the log each time a print command is executed.	True
SubstitutionVariable:XXX	Allows print template substitution variables to be specified in the configuration file. XXX should be the name of the substitution variable in question. For example, it would be possible to set a configuration option as follows: SubstitutionVariable:LARGEFONT = \027w This would cause any instances of &LARGEFONT& in the print template to be replaced with the specified character sequence (an escape [ASCII 27] followed by a small 'w').	No default value.

Go back to the top

SERVER.RELAYCONNECTIONS

This section controls whether (and how) the Application Server opens connections to a relay port running on a gateway. A relay connection allows remote manipulation of the server through an interface similar to that the server web status page offers.

Option	Description	Default
RELAYHOST	The name or address of the host on which the relay port is available (that is the gateway machine). If left blank or not specified, relay connections are not permitted.	No default value.

Option	Description	Default
RELAYPORT	The port number of the relay port. If left blank or not specified, relay connections are not permitted.	No default value.
ALLOWCONFIGCHANGESFROMWEBPAGE	A true/false flag that controls whether you can modify the configuration file through the relay web interface.	False
ALLOWFILEENQUIRYFROMWEBPAGE	A true/false flag that controls whether you use the remote file management to view files through the relay web interface.	False
ALLOWFILEUPDATESFROMWEBPAGE	A true/false flag that controls whether you use the remote file management to view files through the relay web interface.	False
ALLOWLDBENQUIRYFROMWEBPAGE	A true/false flag that controls whether you can view the local database tables through the relay web interface.	true
ALLOWLDBUPDATESFROMWEBPAGE	A true/false flag that controls whether you can update local database tables through the relay web interface.	
AUTHORISATIONUSERNAME	The user ID to be used to secure the relay web interface. If not present or left blank, no logon security is enforced.	- (no logon security)

Option	Description	Default
AUTHORISATIONPASSWORD	The password to be used to secure the relay web interface.	- (no password)
HTMLBUTTONLOCATION	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next/previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward.	Bottom
HTMLPAGESIZE	The maximum number of lines to display on the web pages.	15
SHOWLDBROWCOUNTS	A true/false flag that indicates whether, by default, row counts are listed against local database tables listed in the relay web interface.	False
SUPPRESSLOGO	A true/false flag that allows suppression of the Sky Technologies logo on the relay web pages (to conserve bandwidth and improve performance).	False

Option	Description	Default
USEHTMLCOMPRESSION	A true/false flag that indicates whether HTML compression should be used for web pages sent to browsers that support it.	True
RECONNECTSECONDS	The time interval (in seconds) between attempts to reconnect if the relay connection is severed.	30

Go back to the top

SERVER.SERVEREXITS

This section contains information on user-written Java exits that the server should invoke. Server exits are not associated with any individual user session and effectively allow custom-written code to be executed in the background by the server.

Option	Description	Default
<SERVER EXIT CLASS NAME>	The fully qualified class name for each server exit class should be placed as a parameter in the "ServerExits" section to the left of the equals sign. The class in question must be available to the JVM in which the server runs (that is the class, or the jar/zip file containing the class, should be placed inside the `classes` directory). The value assigned to each server exit parameter is a true/false flag that tells the server whether or not the server exit is currently enabled.	No default value.

Go back to the top

SERVER.STORAGE

This section contains information on where and how the Application Server should cache information that needs to be persistently stored (including both the application cache and the local database).

Option	Description	Default
StorageType	The storage strategy to be adopted. Must be one of the following: File - Persistent data should be stored as files within a local file system (this is the default behaviour). JDBC - Persistent data should be stored in a relational database. Communication with the database takes place through a JDBC driver. Database - Same as JDBC. SkyDB - Persistent data should be stored in the embedded SkyDB database. BlackBerry - Internal blackberry persistence storage.	File BlackBerry defaults to BlackBerry (internal persistance engine).
StorageRootDirectory	For use when the storage type is "File". This parameter tells the server which directory is set aside for its use in storing persistent data. Defaults to "." (The current working directory for the application or web server into which the server is loaded).	As noted.

Option	Description	Default
StorageDatabaseDriver	For use when the storage type is "JDBC". Should be set to the fully qualified name of the JDBC driver to be used in communicating with the storage database. The class in question must be available to the JVM in which the server runs (that is the class, or the jar/zip file containing the class, should be placed inside the `classes` directory).	No default value.
StorageDatabaseURL	For use when the storage type is "JDBC". The precise format of this URL varies depending on the JDBC driver in use. Consult the documentation on your JDBC driver for further information.	No default value.
StorageDatabaseUser	For use when the storage type is "JDBC". The database user to use when establishing a connection to the storage database.	No default value.
StorageDatabasePassword	For use when the storage type is "JDBC". The password to use when establishing a connection to the storage database.	No default value.

Option	Description	Default
StorageNationalCharacter-SupportType	For use when the storage type is "JDBC". Describes the type of support for "National" characters (that is storage of nonEnglish or Unicode data). Three options are available: None - No such support is offered by the database. If runs in Unicode mode against a database of this type, all data is encoded/decoded on the way to/from the database. Standard - Indicates that data types such as NCHAR, NVARCHAR are fully supported by the database. Inbuilt - Indicates that Unicode data is handled transparently by the database with no special SQL syntax required.	Standard
StorageReservedWordProtection	For use when the storage type is "JDBC". This true/false flag helps circumvents problems with column names that may happen to be reserved words for the database currently in use. When enabled, all database column names are automatically prefixed with the specified prefix (see the next item).	True

Option	Description	Default
StorageReservedWordProtection-Prefix	For use when the storage type is "JDBC". This specifies the prefix to use in front of column names when reserved word protection is in use.	"_" (underscore)
StorageSqlLogging	For use when the storage type is "JDBC" or "SkyDB". This true/false flag indicates whether or not SQL sent to the storage database should be written to the log. This can be useful for debugging in some situations, but generates large log files and should generally be left off.	False
StorageCheckColumnSizes	For use when the storage type is "JDBC". This true/false flag enables or disables checking of column sizes at startup. For databases with dynamic column sizing (for example, H2), this check is not necessary and can in fact give unnecessary warnings.	True
StorageDatabaseInitialSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that is executed when the server first starts (immediately after establishing its first connection to the database).	- (no custom SQL)

Option	Description	Default
StorageDatabaseFinalSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that is executed when the server shuts down, just before it disconnects from the database for the last time.	- (no custom SQL)
StorageSessionInitialSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that is executed each time a new connection is made to the database (before anything else occurs on the connection).	- (no custom SQL)
StorageSessionFinalSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that is executed just before a connection to the database is closed (after everything else has completed on the connection).	- (no custom SQL)

Option	Description	Default
StorageNullValueChecking	For use when the storage type is "JDBC". A true/false flag to indicate whether the database stores empty strings as nulls, hence necessitating null value checks whenever data is fetched from the database. Oracle is one example of a database that demonstrates this behaviour. ***Note:*** You must set this flag to "true" if Oracle is in use.	False (no null value checking)

Option	Description	Default
StorageDatabaseFileName	For use when the storage type is "SkyDB". This is the name of the database file. ***Note:*** For blackberry devices, this configuration item does not specify the direct path to the database file. Instead it is expecting one of two values for this parameter. InternalFileSystem, or ExternalFileSystem. If InternalFileSystem is specified, the application will look for any root file system based on the blackberry internal 'store' memory, and secondly for any root file system based on the blackberry internal 'system' memory. If ExternalFileSystem is specified, the application will look for any root file system based on an 'sdcard', secondly it will look for any root file system based on a 'cfcard', and thirdly it will look for any root file system based on a 'memorystick'. Refer to BlackBerry Storage Options for more information.	No default value. BlackBerry defaults to InternalFileSystem.

Option	Description	Default
StoragePageCacheSize	For use when the storage type is "SkyDB". This is the maximum number of pages that will be cached in memory simultaneously. Each page is 4096 bytes in size, and also carries with it an in-memory representation of the items that it contains. Overall, a memory consumption of something less than 10 KB per cached page can typically be assumed. A zero or negative value turns page caching off, but this is not recommended for performance reasons.	1250
StorageReadLocks	For use when the storage type is "SkyDB". This true/false flag controls whether or not SkyDB takes out locks on a table during read operations (e.g. selection of records).	false
StorageInternalLogging	For use when the storage type is "SkyDB". This true/false flag controls whether or not SkyDB writes information about its internal operations to the system log.	false

Option	Description	Default
StorageConsistencyCheck	For use when the storage type is "SkyDB". This true/false flag controls whether or not the database should be checked for consistency on startup. This slightly delays the startup process, but will report any problems encountered in the log.	false
StorageCompressOnShutdown	For use when the storage type is "SkyDB". This true/false flag indicates whether or not the database should be compressed during the shutdown process. This makes the database file smaller, but slightly delays the shutdown process.	true
StorageIndexDefragOnShutdown	For use when the storage type is "SkyDB". This true/false flag controls whether or not index defragmentation is performed during the shutdown process. This decreases the overall size of the database and also improves index performance, but slightly delays the shutdown process.	true

Option	Description	Default
StorageInternalizeStrings	For use when the storage type is "SkyDB". This true/false flag indicates whether or not string data should be "internalised". This consolidates shared strings and results in much lower memory footprint. It has a slight performance impact, which is typically offset by the much lower memory consumption. ***Note:*** Do not enable this feature when using the CrEme JVM, as too much string data results in internal overflows within the JVM that cause it to "crash" with the error "*** panic[21]: 16-bit string hash table overflow"	false
StorageIndexRebuild	A true/false flag that tells VTI whether to rebuild all indices (both primary and secondary) on all tables at startup. This flag is primarily intended for use during upgrades from a previous release. It should not be left enabled as a matter of course, as this has a performance impact.	false

Option	Description	Default
StorageFullEncryption	A true/false flag that controls whether the database contents get encrypted. If enabled the EncryptionAlgorithm, EncryptionKey/EncryptionKeyFile options become relevant. Currently only supported by SkyDB.	false
EncryptionAlgorithm	This option is currently only supported by SkyDB. It specifies the (symmetric) encryption algorithm to be used to encrypt secured data. The following values are permitted: AES - The AES (Rijndael) encryption scheme. BLOWFISH - The "blowfish" encryption scheme.	BLOWFISH
EncryptionKey	This option is currently only supported by SkyDB. It specifies the encryption key to be used to encrypt secured data. If no encryption key is specified, the server will look for one in an encryption key file instead.	As noted.
EncryptionKeyFile	This option is currently only supported by SkyDB. It specifies the name of the file containing the encryption key. If no encryption key is present here either, one will be internally generated by the server.	As noted.

Option	Description	Default
EncryptionKeyHexEncoded	This option is currently only supported by SkyDB. It is a true/false flag that specifies whether or not the encryption key is hexencoded.	false
StorageHandlerAllocationTimeout	The maximum period of time (in seconds) that the server will wait for access to its persistent storage cache during any given operation. If a storage handler cannot be allocated within this time period then the operation in question will fail.	10 (seconds)
StorageHandlerInactivityTimeout	The maximum period of time (in seconds) that the server will keep hold of an idle storage handler. If any given storage handler remains unused for this length of time, any associated database connection will be closed, and the storage handler is discarded.	60 (seconds)
MinimumNumberStorageHandlers	The minimum number of storage handlers that the server will keep on hand for use by processes needing to access the persistent storage cache. This parameter defaults to 0 for the "File" storage strategy, and 3 for the "Database" storage strategy.	As noted

Option	Description	Default
MaximumNumberStorageHandlers	The maximum number of storage handlers that the server will keep on hand for use by processes needing to access the persistent storage cache. If set to 0, no maximum is enforced. This parameter defaults to 0 for the "File" storage strategy, and 10 for the "Database" storage strategy.	As noted

Go back to the top

SERVER.COMMANDPORT_XXX

These section(s) contain information on the command port(s) that are used to accept incoming commands such as stopping, restarting, reporting status, and so on. Normally only one command port is needed, but you can set up more than one this is desired. The xxx should be substituted with a unique meaningful name for the command port.

Option	Description	Default
Port	The port number to listen on for internal commands.	No default value (normally configured to 5070).

Go back to the top

SERVER.HTTPCONTROLPORT_XXX

These section(s) contain parameters for the HTTP web status interface. Normally only one HTTP control port is needed, but you can set up more than one if this is desired. The xxx should be substituted with a unique meaningful name for the HTTP control port.

Option	Description	Default
Port	The port number to listen on for Web HTTP commands.	No default value (normally set to 5071).
AllowConfigChangesFromWebPage	A true/false flag that controls whether you can modify the configuration file through the administration web interface.	False
AllowFileEnquiryFromWebPage	A true/false flag that controls whether you can use a remote file management to view files through the administration web interface.	False
AllowFileUpdatesFromWebPage	A true/false flag that controls whether you can use remote file management update files through the administration web interface.	False
AllowLdbEnquiryFromWebPage	A true/false flag that controls whether you can view the local database tables through the administration web interface.	True
AllowLdbUpdatesFromWebPage	A true/false flag that controls whether you can update the local database tables through the administration web interface.	False
AuthorisationUsername	The user ID to be used to secure the web status page. If not present or left blank, no logon security is enforced when accessing the web status page.	- (no logon security)

Option	Description	Default
AuthorisationPassword	The password to be used to secure the web status page.	- (no password)
HtmlButtonLocation	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next/previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward.	Bottom
HtmlPageSize	The maximum number of lines to display on the web pages.	15
ShowLdbRowCounts	A true/false flag that indicates whether, by default, row counts are listed against local database tables listed in the administration web interface.	False
SuppressLogo	A true/false flag that allows suppression of the Sky Technologies logo on the web status pages (to conserve bandwidth and improve performance).	False
UseHtmlCompression	A true/false flag that indicates whether you should send HTML compression to browsers that support it.	True

Go back to the top

SERVER.GUIAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow GUI clients to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the GUI application port.

Option	Description	Default
PORT	The port number to listen on for incoming BPC/WPC application requests.	No default value (normally set to 5074).
APPLICATION	The identifying number of the application to be provided to BPC/WPC clients connecting on this port.	000
VERSION	The identifying number of the application version to be provided to BPC/WPC clients connecting on this port.	000

Option	Description	Default
DEVICE	The identifying number to be assigned to BPC/WPC clients connecting on this port.	000
HOSTINTERFACE	The host interface definition to use.	No default value.
CLIENTCANREQUESTAPPLICATION	A true/false flag that indicates whether a BPC/WPC client is permitted to override the default application and/or version number with the application and/or version numbers specified in the WPC configuration file.	true

Option	Description	Default
SESSIONTIMEOUT	The maximum period of time (in minutes) for which the server retains the context of an idle (inactive) session before flushing it from memory. Because BPC/WPC clients are not continuously connected, no notification is sent to the user when this occurs. If a user is inactive for greater than the specified timeout period, the next time they attempt to connect to the server through BPC/WPC, a new session is established and they are sent to the application logon screen. If this value is set to zero or a negative number, then no session timeout is enforced. This is not	60 (minutes)

Option	Description	Default
USECOMPRESSION	A true/false flag that indicates whether the XML data sent to BPC/WPC sessions on the port should be compressed. This greatly reduces bandwidth consumption.	False
COMPRESSIONALGORITHM	The compression algorithm to use when compressing data sent to BPC/WPC sessions on the port. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.

Option	Description	Default
USEENCRYPTION	A true/false flag that indicates whether the XML data sent to BPC/WPC sessions on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	False
ENCRYPTIONALGORITHM	The encryption algorithm to use when encrypting data sent to BPC/WPC sessions on the port. The following encryption algorithms are available: BLOWFISH - The "blowfish" encryption scheme.	No default value.

Option	Description	Default
ENCRYPTIONKEY	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
ENCRYPTIONKEYFILE	The file in which the encryption key is stored.	No default value.
ENCRYPTIONKEYHEXENCODED	A true/false flag that indicates whether the encryption key is hex-encoded.	false

Go back to the top

SERVER.HTTPAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow browsers to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the HTTP application port.

Option	Description	Default
PORT	The port number to listen on for incoming HTTP application requests.	No default value (normally set to 5073).
APPLICATION	The identifying number of the application to be provided to browsers connecting on this port.	000
VERSION	The identifying number of the application version to be provided to browsers connecting on this port.	000
DEVICE	The identifying number to be assigned to browsers connecting on this port.	000
HOSTINTERFACE	The host interface definition to use.	No default value.
CLIENTCANREQUESTAPPLICATION	A true/false flag that indicates whether a browser is permitted to override the default application and/or version number using query string parameters (the relevant query string parameters are "appNum" and "verNum").	true

Option	Description	Default
SESSIONTIMEOUT	The maximum period of time (in minutes) for which the server retains the context of an idle (inactive) session before flushing it from memory. Because a browser is not continuously connected, no notification is sent to the user when this occurs. If a user is inactive for greater than the specified timeout period, the next time they attempt to connect to the server through their browser, a new session is established and they are sent to the application logon screen. If this value is set to zero or a negative number, then no session timeout is enforced. This is not recommended since there is then no way for the server to remove idle sessions, resulting in increasing memory consumption over time.	60 (minutes)

Go back to the top

SERVER.TERMINALAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow terminal based devices to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the terminal application port.

Option	Description	Default
Port	The port number to listen on for incoming terminal connections.	No default value (normally set to 5072).
Application	The identifying number of the application to be provided to devices connecting on this port. If terminal enquiry is in use then this is the default application number that is used if the terminal enquiry process cannot determine the application number.	000
Version	The identifying number of the application version to be provided to devices connecting on this port. If terminal enquiry is in use then this is the default version number that is used if the terminal enquiry process cannot determine the version number.	000
Device	The identifying number to be assigned to devices connecting on this port. If terminal enquiry is in use then this is the default device number that is used if the terminal enquiry process cannot determine the device number.	000
ErrorBeeps	The number of "beeps" to be sent to a terminal device when an application error occurs. This allows errors to be made audibly distinguishable from beeps the device may issue for other reasons (for example, when scanning a bar code).	1

Option	Description	Default
ExitDelay	The number of seconds for which the server should pause before closing down a connection. This gives the user a chance to read any error messages displayed in the event that the terminal device clears the screen (or scrolls off any information displayed) while resetting.	2
HostInterface	The host interface definition to use.	No default value.
TerminalEnquiry	A true/false flag that indicates whether or not terminal enquiry is to be used for this port. Terminal enquiry allows the server to query any device connecting to it to establish the application number, version number, and/or device number to be used. The answerback string for each device making use of this feature should be set as follows: VTI[A=xxx,V=yyy,D=zzz] where xxx, yyy and zzz are (respectively) the application number, version number, and device number to be used. If the "ClientCanRequestApplication" parameter is set on, then any settings acquired through terminal enquiry override the default values set for the port. Where terminal enquiry is not used (that is this flag is set to "false") then the default values for the port are always used.	True

Option	Description	Default
TerminalEnquiryTimeout	The maximum time period (in seconds) for which the server should wait for a response from a connecting terminal device during terminal enquiry. If the timeout expires, the server assumes that no further identifying information is forthcoming from the device and then continues processing. This typically occurs when the format of the answerback string is incorrect. In the normal course of events the terminating right square bracket character "]" tells the server that the answerback string is complete and that it can proceed.	10 (seconds)
TerminalType	The terminal type definition to use. This option enables custom and other terminal types to be implemented and supported. The server supports the following inbuilt terminal types: VT220 - VT220 emulation using 7-bit ASCII (same as VT220-7Bit) VT220-7Bit - VT220 emulation using 7-bit ASCII VT220-8Bit - VT220 emulation using 8-bit ASCII You can set up additional variant terminal types in the configuration file through the TerminalType section (described hereafter).	VT220
ClientCanRequestApplication	A true/false flag that indicates whether a client connection can override the default application and/or version number using the terminal enquiry sequence.	True

Option	Description	Default
ClearFieldValueOnReEntry	A true/false flag that tells the server whether or not to clear any existing value in a screen field when a new value is entered.	False
ColorEnabled	Whether the connected devices support VT220 colours.	Ttrue
SessionTimeout	The maximum period of time (in minutes) for which the server permits an idle (inactive) connection to remain connected before disconnecting it. If the value is set to zero or a negative number, then no session timeout is enforced.	60 (minutes)
TraceInput	A true/false flag that controls whether the server should write detailed information about incoming characters to the log. This is typically used for debugging device keystroke behaviour (for example, function key mappings), and should generally be left disabled.	False

Go back to the top

SERVER.XMLGATEWAYPORT

These section(s) contain parameters for port(s) on which the Application Server is to make an Access Gateway available. This allows SkyMobile to connect to a back-end host by using the Access Gateway port as an intermediary. For example, other instances of Application Server could connect to a back-end SAP instance through an XML port running on a central server.

Option	Description	Default
Port	The port number to listen on for incoming messages.	No default value (normally set to 5075).
HostInterface	The host interface definition into which the port is feed.	No default value.
DefaultIdentityService	The name of a implemented identity management service to use as the default if the calling Secure Container sends an identity management request and does not specify one.	None
MaximumConnectedServers	The maximum number of separate servers permitted to connect through this gateway port. This is normally used to control load balancing. A value of 0 indicates that there is no maximum applicable.	0 (no limit)
MaximumMessageSize	The maximum permissible message size allowable on the port (in bytes). Messages greater than this size cause the incoming connection that is issued them to be immediately closed. This parameter is primarily for use when the port is publicly exposed, and helps to protect the server from uninvited incoming traffic that isn't in the correct format for processing. A value of 0 indicates that there is no maximum message size.	0 (no limit)

Option	Description	Default
UseCompression	A true/false flag that indicates whether XML data sent to connected clients should be compressed. This greatly reduces bandwidth consumption.	False
CompressionAlgorithm	The compression algorithm to use when compressing data sent to connected clients on the port. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value
Use Encryption	A true/false flag that indicates whether the XML data sent to connected clients on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	False
AllowUnencryptedProvisioning	A true/false flag that specifies whether an unencrypted provisioning request may still be accepted by this port even if encryption is enabled.	False
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to connected clients on the port. The following encryption algorithms are available: BLOWFISH - The "blowfish" encryption scheme (recommended). RSA - The "RSA" encryption algorithm.	No default value.
EncryptionKey	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.

Option	Description	Default
EncryptionKeyFile	The file in which the encryption key is stored.	No default value.
EncryptionKeyHexEncoded	A true/false flag that indicates whether the encryption key is hex-encoded.	False
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the remote partner. If left blank, no initial handshaking is performed. The following encryption algorithms are available: BLOWFISH - The "Blowfish" encryption algorithm. RSA - The "RSA" encryption algorithm (recommended).	No default value.
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the remote partner. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	No default value.
HandshakeEncryptionKeyHex-Encoded	A true/false flag that indicates whether the handshaking encryption key is hexencoded.	false
PingCheckIdleSeconds	The period of time (in seconds) for which a connected client can remain idle before a "ping check" is performed to establish whether the connection is still "live".	600 (10 minutes)

Option	Description	Default
PingCheckTimeout	The period of time (in milliseconds) for which the server will wait for a response after performing a "ping check'. Once this time interval has passed, the connection is no longer considered "live" and is closed.	60000 (1 minute)
UseHttp	A true/false flag that indicates whether or not incoming communications will use the HTTP and/or HTTPS protocols. If enabled, the gateway port expects incoming SkyMobile communications to be bundled as HTTP/HTTPS traffic (potentially coming in via an HTTP/HTTPS proxy). Otherwise, normal SkyMobile communications are expected.	false
XmlDebug	A true/false flag that controls whether the server writes information about incoming and outgoing XML received on the port, into its log. This option has a substantial performance impact and generates very large log files, so it should generally be left disabled.	false

Go back to the top

SERVER.RELAYPORT_XXX

These section(s) contain parameters for port(s) on which the Application Server can accept incoming relay connections from other servers.

Option	Description	Default
PORT	The port number to listen on for incoming messages.	No default value (normally set to 5076).
IDLETIMEOUT	The maximum time interval (in seconds) that the server allows a relay connection to remain unresponsive before concluding that is no longer working correctly and closes it.	3600 (5 minutes)
PINGTIMEOUT	The maximum time interval (in seconds) that the server allows a relay connection to remain idle before "pinging" it to make sure that it still works correctly.	60 (1 minute)

Go back to the #top

SERVER.CUSTOMPORT_XXX

These section(s) contain parameters for port(s) with specially customized or unique behaviour. Custom ports accept incoming messages through connections made on the port to which they are listening. Each time such a message arrives, the server invokes a user developed exit that is responsible for processing the message. This allows for extreme flexibility in the way in which you can make the server to interact with devices and applications conversing over a TCP/IP connection.

Option	Description	Default
Port	The port number to listen on for incoming messages.	No default value (normally set to 5077).

Option	Description	Default
InputMessageType	The expected format for incoming messages. This tells the server when an incoming message is complete and thus ready for processing. This parameter must be one of the following: D - Delimited input messages. Incoming messages are terminated with a specified delimiter. This is the default input mode. F - Fixed length messages. Incoming messages are all the same length. O - The message length is embedded in incoming messages at a specified offset position from the start of the message. In other words, the message contains a numeric field at a known position that contains the length of the message itself. U - Unformatted input. Any input received is processed immediately upon you receive. This generally results in a message that is sent for each incoming character, but is dependent on network behaviour.	D (delimited input)
InputDelimiters	Only used for input message type "D". Specifies the character sequence(s) that are to act as delimiters for incoming messages. You can specify more than one value by separating the various delimiters with commas. You can specify non printable characters by entering a backslash followed by the 3-digit ASCII value of the character. Thus, for example, you can enter a new line (ASCII value 10) as \010.	\010 (new line)
InputIgnored	Only used for input message type "D". Specifies any character sequence(s) that you should ignore and remove from incoming messages. You can specify more than one value by separating the strings to be ignored with commas. You can specify non-printable characters by entering a backslash followed by the decimal ASCII value of the character.	\013 (carriage return)

Option	Description	Default
InputStripDelim	Only used for input message type "D". Specifies whether or not delimiters should be stripped from the message prior to it being processed.	true
InputFixedLength	Only used for input message type "F". Specifies the length of incoming fixed length messages.	No default value.
InputOffsetStart	Only used for input message type "O". Specifies the offset from the start of the message at which to find the message length within the message. For example, if the message length were to be contained in characters 4-8 of the message, the offset would be 3, since character position 4 is 3 characters in from character position 1 (the start of the message).	No default value.
InputOffsetLength	Only used for input message type "O". Specifies the length of the message length field within the message. For example, if the message length is to be contained in characters 4-8 of the message, the length would be 5 (since it takes up character positions 4, 5, 6, 7, and 8, for a total of five characters).	No default value.
CustomPortExit	The fully qualified class name for the Java custom port exit class to invoke when a message is received. The class in question must be available to the JVM in which the server runs (that is, you should replace the class, or the jar/zip file containing the class inside the `classes` directory).	No default value.

Go back to the top

SERVER.CUSTOMSERIALDEVICE_XXX

These section(s) contain parameters for serially connected devices that you need to integrate with the server. Custom serial devices accept incoming messages on a specified serial port. Each time such a message arrives, the server invokes a user-developed exit that is responsible for processing the message.

Option	Description	Default
CommPort	The name of the communications port on which to listen for incoming messages (for example COM1).	No default value.
Speed	The baud rate at which communication with the serial device should take place. If left blank or not specified, the current baud rate of the device is used.	As noted.
DataBits	The number of data bits used for communication with the serial device. You must have valid entries in the range 5-8. If left blank or not specified, the current number of data bits for the device is used.	As noted.
StopBits	The number of stop bits used for communication with the serial device. Valid entries are 1, 1.5, or 2. If left blank or not specified, the current number of stop bits for the device is used.	As noted.
Parity	The parity settings to use during communication with the serial device. Valid entries are None, Odd, Even, Mark, or Space. If left blank or not specified, the current parity setting for the device is used.	As noted.

Option	Description	Default
FlowControl	The flow control (handshaking) settings to use during communication with the serial device. Valid entries are as follows: None - No flow control. XonXoff - XON/XOFF handshaking is used. If left blank or not specified, the current flow control setting for the device is used.	As noted.
InitialConnect	A true/false flag that controls whether or not the server attempts to automatically establish and maintain a connection to the serial device. This may be undesirable in some situations, for example if communications with the device in question are initiated upon demand and controlled by a user exit.	True
DisconnectDelay	The time interval (in milliseconds) that the server pauses after disconnecting from the device. Some serially connected devices experience problems if sufficient time is not allowed to permit the device to completely shut down before it is used again. This pause is intended to ensure that such problems do not arise, but you can reduce or remove in the interests of efficiency where it is not needed (or conversely, increased in cases where the device in question requires a long time to shut down properly).	500 (milliseconds)

Option	Description	Default
ReconnectInterval	The time interval (in seconds) to pause after an unsuccessful attempt to connect to the serial device, before trying again. This parameter is only used when the server is responsible for maintaining the connection to the device (that . InitialConnect is set to "true"). A negative or zero value indicates that no attempt to reconnect should take place.	0 (do not reconnect)
UseNativeComm	A true/false flag that controls whether the native serial communications library is used instead of the Java Comm Extension.	false
InputMessageType	The expected format for incoming messages. This tells the server when an incoming message is complete and thus ready for processing. This parameter must be one of the following: D - Delimited input messages. Incoming messages will be terminated with a specified delimiter. This is the default input mode. F - Fixed length messages. Incoming messages are all the same length. O - The message length is embedded in incoming messages at a specified offset position from the start of the message. In other words, the message contains a numeric field at a known position which contains the length of the message itself. U - Unformatted input. Any input received is processed immediately upon being received. This generally results in a message being sent for each incoming character, but is dependent on network behaviour.	D (delimited input)

Option	Description	Default
InputDelimiters	Only used for input message type "D". Specifies the character sequence(s) that are to act as delimiters for incoming messages. More than one value can be specified by separating the various delimiters with commas. Nonprintable characters can be specified by entering a backslash followed by the 3-digit ASCII value of the character. Thus, for example, a new line (ASCII value 10) would be entered as \010.	\010 (new line)
InputIgnored	Only used for input message type "D". Specifies any character sequence(s) that are to be ignored and removed from incoming messages. More than one value can be specified by separating the strings to be ignored with commas. Non-printable characters can be specified by entering a backslash followed by the decimal ASCII value of the character.	\013 (carriage return)
InputStripDelim	Only used for input message type "D". Specifies whether or not delimiters should be stripped from the message prior to it being processed.	true
InputFixedLength	Only used for input message type "F". Specifies the length of incoming fixed length messages.	No default value.
InputOffsetStart	Only used for input message type "O". Specifies the offset from the start of the message at which to find the message length within the message. For example, if the message length were to be contained in characters 4-8 of the message, the offset would be 3, since character position 4 is 3 characters in from character position 1 (the start of the message).	No default value.

Option	Description	Default
InputOffsetLength	Only used for input message type "O". Specifies the length of the message length field within the message. For example, if the message length were to be contained in characters 4-8 of the message, the length would be 5 (since it takes up character positions 4, 5, 6, 7, and 8, for a total of 5 characters).	No default value.
CustomSerialDeviceExit	The fully qualified class name for the Java custom serial device exit class to be invoked when a message is received. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.

Go back to the top

SERVER.TERMINALTYPE_XXX

This Application Server configuration section contains definitions for additional terminal emulation modes (other than the inbuilt ones: VT220 and VT220-8bit) to be supported by SkyMobile. The xxx should be substituted with a unique meaningful name for the emulation mode. This name can then be assigned to the TerminalType parameter in a TerminalApplicationPort section.

Option	Description	Default
DisplaySequence:BEEP	The character sequence to display to cause the terminal to beep. As with all other sequences in this section, you may present non-printable characters by a backslash ("\") followed by a three-digit decimal ASCII character value. For example, the terminal beep character is normally ASCII character 007 in decimal. You can represent this in the configuration file as \007.	As per base terminal type if specified, otherwise no default.
DisplaySequence:BOLD	The character sequence to display to cause the terminal device to turn bold mode on.	As per base terminal type if specified, otherwise no default.
DisplaySequence:BLINK	The character sequence to display to cause the terminal device to turn blinking mode on.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:CLEAR	The character sequence to display to cause the terminal device to clear the screen.	As per base terminal type if specified, otherwise no default.
DisplaySequence:SETCURSOR	The character sequence to display to cause the terminal device to set the cursor position to specified X and Y co-ordinates. This setting interacts with the SetCursorXOffset, SetCursorYOffset, SetCursorXToken, and SetCursorYToken parameters (see below).	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOBOLD	The character sequence to display to cause the terminal device to turn bold mode off.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:NOBLINK	The character sequence to display to cause the terminal device to turn blinking mode off.	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOREVERSE	The character sequence to display to cause the terminal device to turn reverse (inverse video) mode off.	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOUNDERLINE	The character sequence to display to cause the terminal device to turn underline mode off.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:REVERSE	The character sequence to display to cause the terminal device to turn reverse (inverse video) mode on.	As per base terminal type if specified, otherwise no default.
DisplaySequence:TERMENQ	The character sequence to display to initiate terminal enquiry (that is to trigger the terminal device to response with its answerback string).	As per base terminal type if specified, otherwise no default.
DisplaySequence:UNDERLINE	The character sequence to display to cause the terminal device to turn underline mode on.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
FunctionKeySequence:xxx	The character sequence that the terminal device generates to signify that function key xxx is activated.	As per base terminal type if specified, otherwise no default.
KeySequence:BACKSPACE	The character sequence that the terminal device generates to signify that a destructive backspace is requested.	As per base terminal type if specified, otherwise no default.
KeySequence:BACKTAB	The character sequence that the terminal device generates to signify that a backtab (tab backwards) is requested.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:INSERT	The character sequence that that the terminal device generates to signify a request to toggle between insert and replace modes.	As per base terminal type if specified, otherwise no default.
KeySequence:DELETE	The character sequence that the terminal device generates to signify a request to delete the character over which the cursor is currently positioned.	As per base terminal type if specified, otherwise no default.
KeySequence:UP	The character sequence that the terminal device generates to signify that the up arrow key is pressed.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:DOWN	The character sequence that the terminal device generates to signify that the down arrow key is pressed.	As per base terminal type if specified, otherwise no default.
KeySequence:RIGHT	The character sequence that the terminal device generates to signify that the right arrow key is pressed.	As per base terminal type if specified, otherwise no default.
KeySequence:LEFT	The character sequence that the terminal device generates to signify that the left arrow key has is pressed.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:TAB	The character sequence that the terminal device generates to signify that a tab (forwards) is requested.	As per base terminal type if specified, otherwise no default.
SetCursorXOffset	The X offset to use when directing the cursor to a particular position on the screen. By default, the first column on the screen is considered to be column 0. If the cursor addressing system for the emulation mode treats the first column to be column 1, a value of 1 is required for the X offset.	As per base terminal type if specified, otherwise 0.
SetCursorYOffset	The Y offset to use when directing the cursor to a particular position on the screen. By default, the first line on the screen is considered to be line 0. If the cursor addressing system for the emulation mode treats the first line to be line 1, a value of 1 is required for the Y offset.	As per base terminal type if specified, otherwise 0.
SetCursorXToken	The token used to represent the position of the X co-ordinate value inside the DisplaySequence:SETCURSOR string.	<X>
SetCursorYToken	The token used to represent the position of the Y co-ordinate value inside the DisplaySequence:SETCURSOR string.	<Y>

Go back to the top

3.9.12.4 Example Configurations

About

A number of different example configurations are provided here to show typical implementations. Select from the list below:

Example Configurations

- Gateway SAP RFC Connection
- Container Connection to Gateway
- Gateway Port Configuration
- Windows Mobile Secure Container Configuration

Gateway SAP RFC Connection

The following examples show the configuration to connect to a SAP host through a direct RFC connection and also through SAP Logon Load Balancing. Refer to the config section for more information.

Specific Host RFC Connection

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAPRFCCLIENT.JNILOGGING = false
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MINIMUMNUMBERHANDLERS = 5
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPHOST = nnn.nnnn.nnn.nnn (or
name)
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBGROUP =
```

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBHOST =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPPASSWORD = password
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPSYSTEM = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSELB = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSER = CPIC_USER
```

SAP Logon Load Balancing RFC Connection

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAPRFCCLIENT.JNILOGGING = false
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MINIMUMNUMBERHANDLERS = 5
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPHOST =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBGROUP = SAP Logon Group Name
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBHOST = nnn.nnnn.nnn.nnn (or
name)
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBSYSTEM = SAP SID
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPPASSWORD = password
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPSYSTEM =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSELB = 1
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSER = CPIC_USER
```

Go back to the top

Container Connection to Gateway

This example configuration is for a Secure Container connecting to a Access Gateway port without using encryption. See the config section for more information on the options. If multiple Access Gateways are used in a fail over configuration, then you need to list all the gateway TCP/IP addresses against the XMLGATEWAYHOSTS option. See here for more details.

Example Gateway Host Interface Configuration

```
SERVER.HOSTINTERFACE:XMLCLIENT.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONALGORITHM =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEY =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYFILE =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYHEXENCODED =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYSTRENGTH =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONALGORITHM =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEY =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEYFILE =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEYHEXENCODED =
SERVER.HOSTINTERFACE:XMLCLIENT.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:XMLCLIENT.MAXIMUMNUMBERHANDLERS = 20
SERVER.HOSTINTERFACE:XMLCLIENT.MAXIMUMOPTIMIZATIONLEVEL =
SERVER.HOSTINTERFACE:XMLCLIENT.MINIMUMNUMBERHANDLERS = 0
SERVER.HOSTINTERFACE:XMLCLIENT.USECOMPRESSION = true
SERVER.HOSTINTERFACE:XMLCLIENT.USECONDENSEDXML =
SERVER.HOSTINTERFACE:XMLCLIENT.USEENCRYPTION = false
SERVER.HOSTINTERFACE:XMLCLIENT.USETRANSFERFIELDS =
SERVER.HOSTINTERFACE:XMLCLIENT.XMLDEBUG = false
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYHOSTS = nnn.nnn.nnn.nnn (or
name)
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYHOSTSELECTION = Sequential
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYPORT = nnnnnn
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYRESPONSETIMEOUT = 30
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYSTALLTIMEOUT = 30
```

Go back to the top

Gateway Port Configuration

This example configuration is for an Access Gateway port. Refer to the config section for more information.

Example Access Gateway Port Configuration

```
SERVER.XMLGATEWAYPORT:PORTnnnnnn.COMPRESSIONALGORITHM = GZIP
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEYFILE =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEYHEXENCODED =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.HOSTINTERFACE = xxxxxxxxxx
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PINGCHECKIDLESECONDS =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PINGCHECKTIMEOUT =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PORT = nnnnnn
SERVER.XMLGATEWAYPORT:PORTnnnnnn.USECOMPRESSION = true
SERVER.XMLGATEWAYPORT:PORTnnnnnn.USEENCRYPTION = false
SERVER.XMLGATEWAYPORT:PORTnnnnnn.XMLDEBUG = false
```

Go back to the top

Windows Mobile Secure Container Configuration

This example configuration is for a secure container that runs on Windows Mobile. Refer to the config section for more information.

Server configuration

```
CLIENT.GENERAL.ALLOWMULTIPLEINSTANCES = false
CLIENT.GENERAL.DISABLELOGOFFMESSAGE = false
CLIENT.GENERAL.DISABLESOUND = false
CLIENT.GENERAL.HOSTSTATUSMESSAGES = All
CLIENT.GENERAL.KEYBOARDONLY = false
CLIENT.GENERAL.UNICODEENABLED = true
CLIENT.LOG.LOGACTIVE = true
CLIENT.LOG.LOGFILEDIRECTORY = ./log
CLIENT.LOG.MAXFILES = 7
CLIENT.LOG.PERFORMANCETRACKING = false
CLIENT.LOG.SHOWMILLISECONDS = false
CLIENT.LOG.STRATEGY = Daily
CLIENT.PRESENTATION.AUTOSUGGEST = false
CLIENT.PRESENTATION.CHARPADDINGHEIGHT = 2
CLIENT.PRESENTATION.CHARPADDINGWIDTH = 1
CLIENT.PRESENTATION.DISABLEABOUTMENU = false
CLIENT.PRESENTATION.DISABLEMINIMIZEMENU = true
CLIENT.PRESENTATION.DISABLESELFTESTMENU = false
CLIENT.PRESENTATION.DISABLESHUTDOWNMENU = false
CLIENT.PRESENTATION.DISPLAYIMAGESIZES = false
CLIENT.PRESENTATION.DISPLAYLAYOUTGRID = false
CLIENT.PRESENTATION.FIXEDSIZELINEHEIGHT = 25
CLIENT.PRESENTATION.FONTNAME = Arial
CLIENT.PRESENTATION.FONTSIZE = 9
CLIENT.PRESENTATION.HIGHRESOLUTIONGRAPHICS = true
CLIENT.PRESENTATION.LAYOUTHORIZONTAL =
CLIENT.PRESENTATION.LAYOUTVERTICAL =
CLIENT.PRESENTATION.LEFTMARGIN = 10
CLIENT.PRESENTATION.LOGIMAGESIZES = false
CLIENT.PRESENTATION.MAXIMISEONSTARTUP = false
CLIENT.PRESENTATION.MESSAGEBOXNOFOCUS = false
CLIENT.PRESENTATION.SELECTTEXTONFOCUSCHANGE = true
```

```
CLIENT.PRESENTATION.SHOWFUNCTIONKEYSINMENU = false
CLIENT.PRESENTATION.SHOWHORIZONTALSCROLLBAR = true
CLIENT.PRESENTATION.SHOWMENU = true
CLIENT.PRESENTATION.SHOWVERTICALSCROLLBAR = false
CLIENT.PRESENTATION.SHOWWINDOWCONTROLS = true
CLIENT.PRESENTATION.STATUSBARERRORCOLOR = red
CLIENT.PRESENTATION.STATUSBARICONSET = Basic
CLIENT.PRESENTATION.STATUSBARINFOCOLOR = green
CLIENT.PRESENTATION.TOPMARGIN = 10
CLIENT.PRESENTATION.WINDOWHEIGHT = 500
CLIENT.PRESENTATION.WINDOWHORIZONTALALIGNMENT = Centre
CLIENT.PRESENTATION.WINDOWVERTICALALIGNMENT = Centre
CLIENT.PRESENTATION.WINDOWWIDTH = 500
CLIENT.SERVER.COMPRESSIONALGORITHM = GZIP
CLIENT.SERVER.LAUNCHSERVER = true
CLIENT.SERVER.SERVERCOMMANDPORT = 5070
CLIENT.SERVER.SERVERHOSTNAME = localhost
CLIENT.SERVER.SERVERPORT = 5074
CLIENT.SERVER.SERVERRESPONSETIMEOUT = 30
CLIENT.SERVER.USEENCRYPTION = false
CLIENT.SERVER.XMLDEBUGTOSCREEN = false
GLOBAL.CONFIGURATION.EDIT = TRUE
```

Go back to the top

3.9.12.5 Connecting to SAP

The Windows Secure Container may connect to one or more 'back end' systems (for example, SAP RFC), or connect to one or more gateway ports. In either case, you must configure one or more HostInterface sections in the configuration. Refer to the Configuration options for a detailed description of these sections and their options. Once the host connection sections are declared, the application port configurations refer to the SAP connection to use. Refer to the Example SAP RFC connection and Example Client Gateway connection for more information.

By default, SkyMobile connects to SAP using a special CPIC (Interface) user specified in its configuration file `skymobile.cfg`. This user cannot log onto SAP directly and is only used to process business transactions. If you wish mobile users to log on directly to SAP using their allocated SAP user id and password, then you have to implement a logon screen that captures the user and password. The logon screen then executes a ABAP verify screen exit that performs a special SAP function call and if all is ok, performs a logon context switch, that is instructs SkyMobile to re-logon to SAP using the supplied user and password. A sample ABAP exit (/SKY/YVTIVLOG) is provided that contains all the code necessary. This option is only valid for real-time (online/synchronous) connections with SAP. Offline connections (asynchronous) always use the default SAP user, as specified in the Application Server configuration file.

> ***Note:*** You are advised not to use direct user connections with SAP as this can consume system resources and impact performance. It is far more efficient for all container connections to utilize a shared RFC connection pool and pass through the user credentials as part of the SkyMobile transmission protocol. You can use this userid as meta-data in SAP transaction processing.

3.9.12.6 Tracking Application Servers Users and Devices

With mobile applications, it is important to be able to identify which devices are connected and their current status. SkyMobile provides facilities to identify specific Application Servers and client connections.

Server ID and Group

f no server id is specified, it defaults to the current computer host name or is automatically provisioned. The Server ID is especially useful for managing selective downloads and refreshes data and application definitions; if the Server ID is xyz, then download a specific customer, product profile. It is also necessary to restrict application and local database definitions.

In addition, you can specify an Application Server group in the configuration file. This also may be up to 15 characters. The server group is useful to identify servers for a specific application, and geographic group. You may use in the workbench to define configuration at a group level and effectively filter selection lists.

skymobile.cfg

SERVER.GENERAL.SERVERGROUP = Mobile_Sales
SERVER.GENERAL.SERVERID = PRODUCTION1

The Device Number

You may allocate a three digit device id to each client to help identify specific connections. You may configure the device id for Windows presentation (WPC) and VT220 text terminal modes. For WPC, this is specified in the `skymobile.cfg` configuration file. For VT220 text terminals, you may pass the device id through an answer-back string defined in the telnet client, for example, VTI[d=nnn) where 'd' indicates device and 'nnn' the unique three digit number. You may specify the device number through a screen procedure or programmatically through the ABAP or Java APIs. Once specified, the device number is linked to all requests.

Sessions

Every connection to SkyMobile is allocated a unique session id. This is used internally to identify transactions and link them to specific users and devices. With a SAP back end, there are two session ids, a local session id and a SAP session ID. The SAP session manager displays all connected sessions and their attributes. When an application is logged off, the associated session is de-activated.

User Identification

You can achieve user identification through populating the user id variable that is stored and passed with all requests. Along with the user id, a password, and work area/group may also be populated and used for security checks, transaction identification, data selection. These variables may be populated from procedures and/or through the ABAP/Java programming APIs.

3.9.12.7 Global Configuration

The following section details all the sections and associated options for the Global configuration type.

- GLOBAL.CONFIGURATION
- GLOBAL.STARTUP

GLOBAL.CONFIGURATION

This configuration section controls options around the configuration behaviour.

Option	Description	Default
EDIT	A true/false flag that explicitly specifies whether the configuration is editable via the inbuilt client configuration viewers in APC, IPC and BPC. This has no effect on WPC.	True
ENCRYPT	A true/false flag that explicitly specifies whether the configuration is stored in an encrypted format.	False
VIEW	A true/false flag that explicitly specifies whether the configuration is editable through the inbuilt client configuration viewers in APC, IPC and BPC. This has no effect on WPC.	True

Go back to the top

GLOBAL.STARTUP

This configuration section controls options around the startup behaviour.

Option	Description	Default
PROVISIONCHECK	A true/false flag that explicitly specifies whether the container does a provision check call at startup.	False

Go back to the top

3.9.12.8 Client Configuration

The following section details all the sections and associated options for the Client configuration type:

- CLIENT.ABOUTBOX
- CLIENT.CACHE
- CLIENT.GENERAL
- CLIENT.IDENTITYCHECK
- CLIENT.INPUT
- CLIENT.INPUTAID-XXX
- CLIENT.LOG
- CLIENT.PRESENTATION
- CLIENT.SERVER
- CLIENT.SERVERBACKUP
- CLIENT.XAI

CLIENT.ABOUTBOX

This configuration section controls the appearance and presentation of the about box (that is invoked from the **About** menu option). They override the general-purpose configurations in the Presentation section.

Option	Description	Default
BUTTONPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each button.	0
BUTTONPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average button width.	0

Option	Description	Default
BUTTONSPACING	The number of additional logical pixels to space between buttons; spread them out.	0
FONTNAME	The name of the base font typeface that WPC should use when rendering the about box.	Arial
FONTSIZE	The point size of the base font typeface that WPC should use when rendering the about box.	9
FONTWEIGHT	The "weight" of the base font typeface that WPC should use when rendering the about box (expressed as a number from 1 - 1000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a superthin font typeface, while a value of 1000 indicates an ultrabold font typeface.	400
WINDOWHEIGHT	The height of the "about" window, measured in logical pixels.	300
WINDOWWIDTH	The width of the "about" window, measured in logical pixels.	400

CLIENT.CACHE

This configuration section controls the cache (which is primarily used to store images locally so that they don't need to be continually fetched from the server).

Option	Description	Default
CACHEDIRECTORY	The name of the cache directory. This directory is used to permanently store graphics and other data objects referenced by applications.	The current directory.

Option	Description	Default
CACHESIZE	The maximum size allocated to the cache in bytes, after which the last referenced is automatically removed to make space available.	Unlimited
USESERVERCACHE	A true/false flag that explicitly specifies whether or not the client should share the server's cached binary files (instead of maintaining its own cache). Normally the client identifies scenarios when this is possible and default the setting appropriately. You need to normally configure it only in unusual scenarios.	Will be ascertained according to the situation.

CLIENT.GENERAL

This configuration section contains general-purpose configurations that pertain to the client operation as a whole.

Option	Description	Default
ALLKEYS	A true/false flag that is only used when running under Windows Mobile. When configured true, this flag allows SkyMobile to intercept ALL keystrokes, including those normally reserved for use by the device as "soft" keys (for example, F1 and F2 are sometimes used to bring up the left and right device menus). Configuring this flag to false restores the default device behaviour that means that SkyMobile does not intercept these keys and cannot use them to trigger events within an application.	False
ALLOWMULTIPLEINSTANCES	Allow multiple instances of the presentation client to be started on the same computer.	False

Option	Description	Default
APPLICATIONNUMBER	The application number that the client should request from the application server. If left blank or not specified, the default application and version numbers from the application server port is used instead.	As noted.
CLIENTSESSIONTIMEOUT	This is the maximum time interval (in seconds) that the presentation client will wait in an idle state before terminating the current session and returning to the initial screen of the application (which is typically the login screen). User activity such as mouse movement, mouse clicks, keystrokes and the like will all prevent this from occurring. This feature is useful for ensuring that an application is not left "signed in" for extended periods of time. A zero or negative value here indicates that no timeout is applicable.	0 (no timeout)
CLIENTSESSIONTIMEOUTACTION	This is the action to undertake in the event that a client session timeout occurs. The following values are permitted: Logout - Log out of the application and re-initialize the user session (return to the first screen). Shutdown - Shut the client down completely and exit the application.	Logout
DEVICENUMBER	The device number that the client should use to identify itself to the application server. If left blank or not specified, the default device number from the application server's port is used instead.	As noted.

Option	Description	Default
HOSTSTATUSMESSAGES	Indicates whether or not to display pop-up messages relating to changes in the host connectivity status. The following values are permitted: None - Host connectivity status messages are never displayed. Initial - Only the initial host connectivity status message is displayed (on entry into the application). DisableInitial - The initial host connectivity status message is not displayed, but all such subsequent messages are displayed. All - All host connectivity status messages are displayed.	All
KEYBOARDONLY	A true/false flag that indicates whether the presentation client is running on a device with no mouse or touchscreen (for example, Windows Smartphone).	False
LOGOFFMESSAGE	A true/false flag that indicates whether the popup message to advise the user of a logoff should be disabled.	False
SOUNDS	A true/false flag that indicates whether sounds sent by the server should be played.	True

Option	Description	Default
UNICODEENABLED	A true/false flag that switches the presentation client between Unicode and non-Unicode modes. If enabled, the client generates all outputs (for example, log files) in Unicode format. This flag should be enabled if foreign language support is desired.	False
VERSIONNUMBER	The version number that the client should request from the application server. If left blank or not specified, the default application and version numbers from the application server's port is used instead	As noted.

CLIENT.IDENTITYCHECK

This configuration section is used to implement Identity Management. It defines the Access Gateway port to connect to, encryption options (optional) and the identity service type to use. Other options may be used to influence the user pop-up display that is generated by identity management processing.

Option	Description	Default
LABEL:FOOTERTEXT	A text value to display at the bottom of the identity check screen, below the data fields and the submit button. Intended to contain instructions, warnings and/or legal disclaimers. Overrides any value sent by the Access Gateway.	None (blank)
LABEL:SCREENTITLE	A text value that, if present, replaces the standard "Identity Check" screen title. Overrides any value sent by the Access Gateway.	Identity Check
LABEL:SUBMITBUTTON	A text value that, if present, replaces the standard text on the identity check "Submit" button. Overrides any value sent by the Access Gateway.	Submit

Option	Description	Default
ALLOWAUTOLOGON	Allow the clients to automatically log on using the last cached credentials.	False
DEFAULT:XXXX	Allow the defaulting of an identity parameter XXXX to be disabled by configuring this value to *false*.	True
PORT	The port number of the Access Gateway to connect to.	None
SERVER	The IP address of the Access Gateway to connect to.	None
SERVICE	The name of the identity management service to use. If left blank, the default configured against the gateway port is used.	None
TRACE	Enable popup tracing of identity requests and responses. (Only available if CLIENT.SERVER.XMLDEBUG is also enabled)	False
RESPONSETIMEOUT	Time (in seconds) to wait for a response from the identity management server.	10

CLIENT.INPUT

This configuration section controls how character input is treated.

Option	Description	Default
CHARACTERSUBSTITUTION:X	This directive indicates that a specific incoming character should always be substituted with one or more replacement characters. The "x" represents the incoming character to be substituted. The value assigned to the configuration option represents the value with which the incoming character should be replaced whenever it is received. In both values, nonprintable characters can be represented by a backslash followed by a 3-digit decimal notation for the character's value. For example, the "Escape" character is represented as "\027".	No default.

CLIENT.INPUTAID-XXX

These configuration sections allow control over the appearance and presentation of specific input aids. They override the general-purpose configurations in the [Presentation] section. Valid entries for the "xxx" portion of the section name are "AlphanumericKeypad", "Calculator", "Duration", "NumericKeypad", "QwertyKeypad", "RapidTextEntry", and "Time".

Option	Description	Default
ALLOWDUPLICATES	(Rapid text input aid only) A true/false flag to control whether or not duplicate values are permitted in the rapid text input aid.	False
ALLOWREMOVAL	(Rapid text input aid only) A true/false flag to control whether or not values are allowed to be removed in the rapid text input aid.	True
BUTTONPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each button.	0
BUTTONPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average button width.	0

Option	Description	Default
BUTTONSPACING	The number of additional logical pixels to space between buttons i.e. spread them out.	0
DISPLAYSTATE	(QWERTY keypad input aid only) Configures the initial state of the QWERTY keypad when it is invoked. Two values are possible: "Shift" or "Symbol", each of which causes the appropriate set of keys to appear. Any other value causes the normal initial keyconfigured shown.	No value (normal initial keyset applies).
FONTNAME	The name of the base font typeface that WPC should use when rendering the relevant input aid.	Arial
FONTSIZE	The point size of the base font typeface that WPC should use when rendering the relevant input aid.	9
FONTWEIGHT	The "weight" of the base font typeface that WPC should use when rendering the relevant input aid (expressed as a number from 11000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a superthin font typeface, while a value of 1000 indicates an ultrabold font typeface.	400
INPUTDELIMITER	(Rapid text input aid only) The delimiter character(s) that signal termination of an input value.	\r\n (carriage return followed by a line feed)

Option	Description	Default
OUTPUTDELIMITER	(Rapid text input aid only) The delimiter character(s) to display on output of a complete value list in the field receiving input from the rapid text input aid.	#
SOUNDFAILURE	(Rapid text input aid only) The name of a sound file to play when an input error occurs in the rapid text input aid. The sound to be played must be in .wav format.	- (no sound)
SOUNDSUCCESS	(Rapid text input aid only) The name of a sound file to play when a value is successfully entered in the rapid text input aid. The sound to be played must be in .wav format.	- (no sound)
WINDOWHEIGHT	(Rapid text input aid only) Configures the height of the window in logical pixels.	200
WINDOWWIDTH	(Rapid text input aid only) Configures the width of the window in logical pixels.	200

CLIENT.LOG

This configuration section controls the logging and tracing capabilities.

Option	Description	Default
ACTIVE	A true/false flag that controls whether or not logging should take place.	False
DIRECTORY	The directory into which WPC writes its log files. If not configured, this parameter defaults to "." (the current working directory).	As noted.

Option	Description	Default
EXCLUDES	Indicates the types of log messages that should be excluded from the log file. There are five types of log messages - Information (I), Warning (W), Trace (T), Error (E) and Abort (A). The log message types to exclude should all be concatenated together. Thus, to exclude all Information and Trace messages from the log file, the appropriate configuration would be "IT".	- (no types excluded)
MAXFILES	This configuration restricts the maximum number of log files the Application Server keeps under D, W, and P strategies. If left blank or not configured, then no maximum is enforced. 0 indicates no maximum.	0 (no maximum)
PERFORMANCETRACKING	A true/false flag that controls whether or not performance tracing messages are written to the WPC log.	False
SHOWMILLISECONDS	A true/false flag that controls whether milliseconds are shown in log messages. This feature is useful for measuring sub-second timings and events.	False
STRATEGY	The logging strategy to be adopted. Must be one of the following: D - A new log file is to be created daily. W - A new log file is to be created weekly. P - A new log file is to be created for each process instance O - Overwrite the log file each time the server is restarted.	O

CLIENT.PRESENTATION

This configuration section controls the overall appearance and presentation of the client.

Option	Description	Default
AUTOSUGGEST	(Windows only) A true/false flag that indicates whether or not an auto suggest popup is available to text input fields. The auto suggestions list grows each time a value is entered into any of these text input fields.	False
CHARPADDINGHEIGHT	The number of additional logical pixels to add to (or subtract from) the calculated height of each line.	0
CHARPADDINGWIDTH	The number of additional logical pixels to add to (or subtract from) the calculated average character width.	0
DEFAULTBACKBUTTON	(Windows and Blackberry only) A true/false flag that indicates whether a back button is displayed on the screen by default.	False
DISABLEABOUTMENU	A true/false flag that indicates whether the "About" menu option should be disabled (removed from the menu).	False
DISABLEMINIMIZEMENU	A true/false flag that indicates whether the "Minimize" menu option should be disabled (removed from the menu).	True
DISABLESELFTESTMENU	A true/false flag that indicates whether the "Self Test" menu option should be disabled (removed from the menu).	False

Option	Description	Default
DISABLESHUTDOWNMENU	A true/false flag that indicates whether the "Shutdown" menu option should be disabled (removed from the menu).	False
DISPLAYIMAGESIZES	A true/false flag to control whether the available and final sizes of images are displayed. This can be useful when optimizing image sizes for use within an application.	False
DISPLAYLAYOUTGRID	A true/false flag that indicates whether or not a layout grid is displayed. This is an application design aid that helps to highlight some of the calculations made by the layout process. It can be useful during screen design.	False
FIXEDSIZECOLUMNWIDTH	Width in pixels to be used by each column. Alternatively, a percentage value may be specified by appending the '%' sign. Only used if LAYOUTHORIZONTAL is specified as FIXED SIZE.	
FIXEDSIZELINEHEIGHT	Height in pixels to be used by each line. Alternatively, a percentage value may be specified by appending the '%' sign. Only used if LAYOUTVERTICAL is specified as FIXED SIZE.	

Option	Description	Default
FONTNAME	The name of the base font typeface to use when rendering screens.	Arial
FONTSIZE	The point size of the base font typeface to use when rendering screens.	9
FONTWEIGHT	(Windows only) The "weight" of the base font typeface to use when rendering screens (expressed as a number from 1-1000). The higher the number, the "heavier" the font becomes. Normal text has a weight of 400, and bold text has a weight of 700. A value of 1 indicates a super-thin font typeface, while a value of 1000 indicates an ultra-bold font typeface.	400
HIGHRESOLUTIONGRAPHICS	(Windows only) A true/false flag that indicates whether high-resolution graphics mode should be enabled. This effectively doubles both the vertical and horizontal resolution available for graphics, potentially resulting in a much sharper image. However, existing graphics will typically need to be rescaled (doubled in both height and width) in order to be properly sized when this mode is enabled.	False

Option	Description	Default
HORIZONTALSCROLLSPEED	(Windows only) The horizontal scroll speed of the main window (the higher the number, the faster the scrolling).	1
LEFTMARGIN	The size of the left margin in the window, in logical pixels. This is the gap allowed between the leftmost window border and the extreme lefthand side of the user screen elements.	No default value.
LAYOUTHORIZONTAL	The layout algorithm to use for the vertical axis. The following values are possible: BEST FIT FIXED SIZE LEGACY	Varies by platform.
LAYOUTVERTICAL	The layout algorithm to use for the vertical axis. The following values are possible: BEST FIT FIXED SIZE LEGACY	Varies by platform.
LOGIMAGESIZES	Similar to the "DisplayImageSizes" option, this is a true/false flag that controls whether or not available and final sizes of images are written to the client log.	False

Option	Description	Default
MAXIMISEONSTARTUP	(Windows desktop only) A true/false flag that indicates whether the window should be maximized on startup. Has no effect under Windows Mobile/CE, as the window is always maximized on such devices.	False
MAXIMUMZOOMIN	(iPhone only) The largest scale factor permitted during a "zoom in" operation on the screen. Configuring this value to 1 effectively means that no zooming in is permitted. By default it is configured to 2, meaning that the user can zoom in until the screen contents are 200% of their original size.	2
MAXIMUMZOOMOUT	(iPhone only) The largest scale factor permitted during a "zoom out" operation on the screen. Configuring this value to 1 effectively means that no zooming out is permitted. By default it is configured 2, meaning that the user can zoom out until the screen contents are 50% of their original size.	2

Option	Description	Default
MESSAGEBOXNOFOCUS	(Windows only) A true/false flag that indicates whether or not the "enter" (return) key should be disabled when a pop-up message box appears. This helps to avoid inadvertent dismissal of errors during activities such as bar code scanning.	False
RECYCLEROWS	(Android only) A true/false flag to specify whether table rows may be recycled(reused) when displaying tables. Increases performance for large tables.	True
SELECTTEXTONFOCUSCHANGE	(Windows only) A true/false flag that indicates whether or not text in an input field should be selected when focus is changed.	True
SHOWFUNCTIONKEYSINMENU	A true/false flag that indicates whether the name of each function key shown in the menu is shown (for example, "Refresh F5" rather than simply "Refresh").	False
SHOWHORIZONTALSCROLLBAR	(Windows only) A true/false flag that indicates whether a horizontal scroll bar should be displayed as part of the window. The horizontal scroll bar only becomes active if the contents of the screen are wider than the available display area.	False

Option	Description	Default
SHOWMENU	A true/false flag that indicates whether the menu is displayed.	True
SHOWVERTICALSCROLLBAR	(Windows only) A true/false flag that indicates whether a vertical scroll bar should be displayed as part of the window. The vertical scroll bar only becomes active if the contents of the screen are taller than the available display area.	False
SHOWWINDOWCONTROLS	A true/false flag that indicates whether window controls (minimize, maximize and close buttons) should be made available on the WPC window.	True
STATUSBARICONSET	(Windows only) The icon configured use for display on the status bar. There are currently two icon sets available: Basic - Host connectivity is shown by a simple tick/cross. Animated - Host connectivity icons resemble terminal screens and animate when communication with the server is taking place.	Basic
TABLEPAGECONTROLPADDINGWIDTH	This is the number of additional logical pixels by which to increase the size of the page links in table page controls. It increases the size of the "clickable" area beyond the bounds of the link text itself.	6

Option	Description	Default
TABLEPAGECONTROLSPACING	This is the number of additional logical pixels by which to increase the separation between page links in table page controls. It has no effect on the size of the "clickable" area of the link itself.	0
TOPMARGIN	The size of the top margin in the window, in logical pixels. This is the gap allowed between the topmost window (or menu) border and the extreme top of the user screen elements.	No default value.
USESYSTEMFONT	(iPhone only) A true/false flag that indicates whether the default system font should be used. When true, this overrides any font name and font size specified in the configuration.	True
VERTICALSCROLLSPEED	(Windows only) The vertical scroll speed of the main window (the higher the number, the faster the scrolling).	1
WINDOWHEIGHT	(Windows desktop only) The height of the window in logical pixels. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices.	No default value.

Option	Description	Default
WINDOWHORIZONTALALIGNMENT	(Windows desktop only) The horizontal alignment of the window on the screen. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices. May assume any of the following values: Left - The far left-hand side of the screen. Centre - The centre of the screen. Right - The far right-hand side of the screen.	Centre
WINDOWVERTICALALIGNMENT	(Windows desktop only) The vertical alignment of the window on the screen. Has no effect under Windows Mobile/CE, as the window is always maximised on such devices. May assume any of the following values. Top - The top of the screen. Centre - The centre of the screen. Bottom - The bottom of the screen.	Centre
WINDOWWIDTH	(Windows desktop only) The width of the window in logical pixels. Has no effect under Windows Mobile/CE, as the window is always maximized on such devices.	No default value.

CLIENT.SERVER

This configuration section controls communications between the Client and the Server.

Option	Description	Default
COMPRESSIONALGORITHM	The compression algorithm to use when compressing data sent to the server. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended).	None
ENCRYPTION	The encryption algorithm to use when encrypting data sent to the server. The following encryption algorithms are available: BLOWFISH - The "Blowfish" encryption algorithm (recommended). RSA - The "RSA" encryption algorithm.	None
ENCRYPTIONKEY	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption file instead.	No default value.
ENCRYPTIONKEYFILE	The file in which the encryption key is stored.	No default value.
ENCRYPTIONKEYHEXENCODED	A true/false flag that indicates whether the encryption key is hex-encoded.	False
LAUNCHSERVER	A true/false flag that indicates whether the client should attempt to launch and manage a server on the host it is running upon.	False

Option	Description	Default
LAUNCHIMAGE	The name of the executable file to invoke to launch the server.	No default value.
LAUNCHARGUMENTS	The arguments to pass to the executable file when the server is being launched.	No default value.
LAUNCHDIRECTORY	The working directory from which the server should be launched.	No default value.
LAUNCHWINDOW	A true/false flag that indicates whether the server should be launched in a visible window. When this flag is configured false, the server process will be launched in a process that is not visible to the user.	False
LAUNCHTIMEOUT	The maximum period of time (in seconds) that the client should wait for the server to launch before giving up. For the purpose of this timeout, the server is considered to have successfully launched if it has managed to bind successfully to the port and can therefore offer the client a connection on which to engage in further dialogue about the server's readiness to launch the desired application.	10 (seconds)

Option	Description	Default
RESPONSETIMEOUT	The maximum period of time (in seconds) that the client will await a response from the server before timing out. A value of 0 indicates no timeout is applicable, in which case the client will wait indefinitely for a response from the server.	30 (seconds)
SERVERCOMMANDPORT	The port number of the server's command port.	No default value.
SERVERHOSTNAME	The name of the host on which the server is running, in either human readable or raw IP address format.	No default value.
SERVERPORT	The port number on which the server is listening for incoming client connections.	No default value.
SERVERSHUTDOWNTIMEOUT	The maximum period of time (in seconds) that the client will wait when it shuts down and attempts to terminate a server that it has launched.	10 (seconds)
XMLDEBUG	A true/false flag that indicates whether the client should display all incoming and outgoing XML in popup message boxes. This can be useful for debugging purposes.	False
XMLDEBUGTOLOG	A true/false flag that indicates whether the client should log all incoming and outgoing XML in the client log file. This can be useful for debugging purposes.	False

CLIENT.SERVERBACKUP

This configuration section allows the Client to initiate and monitor backups against the server.

Option	Description	Default
CONFIRMBACKUP	A true/false flag that indicates whether the user needs to confirm any backup initiated by WPC on shutdown.	False
INCREMENTALSBEFOREFULL	If WPC is initiating an incremental backup on shutdown, this is the number of times that an incremental backup will take place before a full backup occurs instead. For example, on every tenth occasion, WPC could be configured to force a full backup rather than an incremental one. A zero or blank value indicates that WPC should never force a full backup.	0
LAUNCHBACKUPONSHUTDOWN	A true/false flag that indicates whether WPC should initiate a backup against the server when it shuts down.	False
USEINCREMENTALBACKUPS	A true/false flag that indicates whether WPC should initiate an incremental backup rather than a full backup on shutdown.	False

CLIENT.XAI

This configuration section controls behaviour in connection with the External Application Invocation (XAI) interface.

Option	Description	Default
XMLDEBUGTOLOG	A true/false flag that indicates whether WPC should write any XAI XML messages received to its log.	False

Option	Description	Default
XMLDEBUGTOSCREEN	A true/false flag that indicates whether WPC should display any XAI XML messages received to the screen through a pop-up message.	False

3.9.12.9 Server Configuration

The following section details all the sections and associated options for the Server configuration type.

SERVER.GENERAL

This section contains parameters of a general nature that affect the operation as a whole.

Option	Description	Default
SERVERID	A unique identifier to distinguish this particular instance of Application Server from any others that may run on the same machine and/or against the same SAP instance. You can use this identifier to perform restricted data downloads. If left blank or not configured, then the server ID is configured to the name of the host machine on which the server runs.	As noted.
SERVERGROUP	This is an arbitrary identifier that is used to group together Application Servers that have common behavior (for example, all those in a certain geographic location) so that you can treat in a similar fashion by SAP-side processing.	- (no group)
RESTRICTION	Name of the host on which the Application Server must run. If a different host name is detected, the server shuts down immediately.	- (any host)
RESTRICTTOUSER	The user ID that you must use to start the server. The server execution may require specific authorizations, and thus a special user may need to start. If specified, and this user is not the current user, the server shuts down immediately.	- (any user)

Option	Description	Default
LICENSEFILE	The name of the license file.	license.dat
AUTORESTART	A true/false flag that, when you enable causes the server to automatically restart itself if it is not shut down properly (for example, in the event of the process crashing or being killed).	False
DATEFORMAT	The format in which you need to enter and display the dates.	dd/mm/yyyy
TIMEFORMAT	The format in which you need to enter and display times.	hh:mm:ss
TIMEZONE	The time zone in which the server runs (overrides the default setting supplied by the Java Virtual Machine).	Server default
UNICODEENABLED	A true/false flag that switches between Unicode and non-Unicode modes. If you enable, the server communicates with SAP in Unicode mode and also generates all outputs (for example, log files) in Unicode format. Enable this flag if you desire for foreign language support.	False
SHUTDOWNTIMEOUT	During shutdown, this is the maximum time interval (in seconds) that the server waits for outstanding tasks to complete before forcibly terminating them.	30

Option	Description	Default
ABORTIFOUTOFMEMORY	A true/false flag that indicates whether the Application Server should terminate if it runs out of memory. It is generally advisable to enable this option, as running out of available memory can potentially cause unpredictable application behavior.	True
CLIENTSCROLLOVERRIDE	A true/false flag that, when you enable causes all tables to convert to client scrollable tables.	False
DUMPCONFIGTOLOG	A true/false flag that, when you disable prevents the configuration from written to the log on startup.	True

SERVER.APPLICATIONCACHE

This section contains information controlling the behavior of the application cache.

Option	Description	Default
APPLICATIONREFRESHINTERVAL	The period of idle time (in seconds) between checks to ensure that the server currently has the latest version of the application data. A value of 0 (zero) means that applications are only checked for changes when the server first starts up.	30

Option	Description	Default
APPLICATIONREFRESHLOCK	Controls locking of applications while they are refreshed. The following values are permitted: None - Users are permitted to proceed on initial entry as soon as the initial application is loaded (without waiting for updates/changes to download). Users may continue to use the application while it is updated in the background. Current - Users are not permitted to proceed on initial entry until available updates/changes to their first application is applied. Users are temporarily locked out of an application if a background update to it takes place while they use it. All - Users are not permitted to proceed on initial entry until available updates/changes that are applied to all their applications. Users are temporarily locked out if a background update takes place to any application.	None
REFRESHBYFUNCTION	A true/false flag that controls whether the server downloads applications all at once, or one function at a time. Refreshing applications by function is somewhat slower but uses far less memory, making it an appropriate choice for resource constrained devices.	False

Option	Description	Default
FUNCTIONCACHESIZE	The maximum number of functions that the server keeps in memory at any given point in time. Functions beyond this number are flushed on a least recently used basis, and need to reload the next time they are needed. If not specified or left as zero, no function cache limit is imposed.	0 (no limit)
MAINTAINHOSTCONNECTIVITY	Whether or not to keep the connection open once the application cache is updated.	True
PRELOADERDISABLED	Indicates whether you need to disable the function preloader. The function preloader uses statistical information to attempt to predict which functions users are likely to require, and loads them into memory in anticipation. It improves user response time at the cost of a small performance overhead on the application server. You need to disable the preloader in most of the situations.	False

Option	Description	Default
PRELOADERPROBABILITYTHRESHOLD	This is an integer value from 0-100 that controls the sensitivity of the function preloader (when it is in use). If, based on historical statistics, the likelihood that a user will request a function from their current point within the application exceeds this percentage, the preloader will load the function, assuming there is room in the function cache to do so. The lower the value, the more sensitive the function preloader becomes, and vice versa. For example, if the probability threshold were set to 10, then the preloader would attempt to load any function with a 10% or higher chance of being requested next, making it considerably more sensitive to user activity. If the probability were set to 50, then the preloader would only attempt to load functions with a 50% or better chance of being requested next, making it somewhat less sensitive.	30

SERVER.BACKUPRESTORE

This section contains information controlling the Application Server backup and restore facility.

Option	Description	Default
BACKUPDIRECTORY	The name of the directory in which the server should store its backup files. If left blank or not specified, the backup/restore facility is disabled.	No default value.
FULLBACKUPITEMS	A semi-colon separated list of files and/or directory pathnames that should be written to the backup file when a full backup takes place.	- (no files)
INCREMENTALBACKUPITEMS	A semi-colon separated list of files and/or directory pathnames that should be written to the backup file when an incremental backup takes place. The files selected must be a subset of the items selected in the FullBackupItems list.	Same as for full backup.

Option	Description	Default
AUTOFULLBACKUPTIMES	A comma-separated list of times at which the server should automatically trigger a full backup (in the format HH:MM:SS). Obviously it is undesirable to have a backup running while data is being altered, so if this feature is used it should typically be set to run at time(s) during which the server will be completely idle.	- (never)

Option	Description	Default
AutoIncrementalBackupInterval	If specified, this value denotes a period of time (in seconds) after which the server will automatically perform an incremental backup of any files present in the incremental backup list that have changed. Incremental backups will continue to take place automatically while the server is "live". If set to zero or not specified, the server will not automatically perform incremental backups.	0 (disables automatic incremental backups)
FullBackupRetentionCount	Specifies the maximum number of historical full backups to retain on disk. All incremental backup information associated with each full backup is also retained.	1

Option	Description	Default
UseCompression	A true/false flag that indicates whether the backup file should be compressed as it is written. This has the effect of slowing the backup process somewhat, but the resultant backup file is usually considerably smaller in size.	False
CompressionAlgorithm	The compression algorithm to use if compression is to take place. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.

SERVER.BINARYFILES

This section contains information pertaining to binary files (such as images) used by the server.

Option	Description	Default
DownloadDirectory	The directory in which the server should store binary files downloaded from the host. If not set, this parameter defaults to "." (the current working directory).	As noted.
UploadDirectory	The directory in which the server should look for binary files to be uploaded to the host. If not set, this parameter defaults to "." (the current working directory).	As noted.
PackageRefreshInterval	The time interval (in seconds) between checks to see whether there are any packages to be downloaded from the host.	900 (seconds)
DynamicBinaryRefreshInterval	The time interval (in seconds) between synchronisation checks of dynamic binary files stored in the VTI_BINARY_FILE table.	900 (seconds)
HexModeEncoding	This parameter specifies whether to transfer binary files in raw hex format, or in raw binary format. Raw binary (false) is faster and takes less space, but can encounter code page conversion problems between platforms using different data storage types for example, EBCDIC host to ASCI client.	true
MaintainHostConnectivity	Whether or not to keep the connection open once the application cache has been updated.	True

Option	Description	Default
MemoryRetentionPeriod	The period of time (in seconds) for which the server will retain the contents of any binary files in memory after they have been accessed. If the specified time interval passes without the binary file being accessed, it will be flushed from memory to help conserve memory usage. Any subsequent access to the same binary file will force the server to re-read it from disk. A setting of zero (0) indicates that binary files should always be immediately flushed from memory after being read. A negative setting indicates that all binary files should be retained in memory indefinitely.	-1
ErrorRetryInterval	If the server experiences an error when it is downloading or refreshing a binary file in the background, this is the number of seconds that it waits before it tries again.	300 (seconds)
DefaultBinaryGroup	The default binary group to which newly created binary files should be assigned (for example, images created from a drawing object field), assuming no binary group is specified.	None (no group)
UpdateHostMd5Hashes	A true/false flag that controls whether or not the server will update the SAP host system with an MD5 hash for any binary files that it receives without one.	true

Option	Description	Default
FileLocation:XXX	Specifies a file location (directory) into which downloaded binary files can be placed. The XXX should match the file location code specified in the SAP workbench. The value assigned to the configuration parameter should be the pathname to the directory to be targeted.	No default value

SERVER.DIALUPCONNECTION_XXX

These section(s) contain information pertaining the dial-up connections that the server can make use of to connect to other networks. This includes the use of Virtual Private Networks (VPNs). The server can be set to automatically connect and disconnect to such networks at particular times of day. User-written exits can also make use of dial-up connections defined in the configuration file.

> ***Note:*** This feature is only available on the Windows and Windows CE platforms.

Option	Description	Default
DialupSettingsName	The name of the Windows dial-up settings to be used to dial the desired network.	No default value.
DialupUsername	The username to be used when dialling the remote network. If left blank or not specified, this defaults to the username stored with the dialup entry.	As noted.
DialupPassword	The password to be used when dialling the remote network. If left blank or not specified, this defaults to the password stored with the dialup entry.	As noted.

Option	Description	Default
AutoConnectionTimes	A comma-separated list of time ranges during which the server should attempt to ensure that a connection to the desired network is automatically established. Times must match the pattern HH:MM:SS, and be in 24-hour format. A valid time range, for example, might be 17:00:00-18:00:00, indicating an hour-long window between 3pm and 4pm. To ensure continuous connection, enter 00:00:00-23:59:59. If left blank or not specified, this parameter indicates that the server should not automatically dial the target network. Connection to the network can still be established in such a situation via the use of user-written exits code.	- (no automatic connection)
AutoConnectionTimeout	The maximum period of time (in seconds) that the server should wait whilst attempting to establish a connection, before assuming that the connection cannot currently be made and temporarily giving up.	30 (seconds)
ReconnectionInterval	The time interval (in seconds) that the server will wait after an unsuccessful attempt to connect to the network, before trying again.	120 (seconds)

SERVER.HEARTBEAT

This section contains information about the way in which the server regularly reports its status to the central SAP host.

Option	Description	Default
HeartbeatInterval	The period of idle time (in seconds) between SAP status notifications when not connected.	15
HeartbeatConnectedInterval	The period of idle time (in seconds) between SAP status notifications when connected. If left blank or not specified, this defaults to the same value as the HeartbeatInterval.	As noted.
HeartbeatHostName	An override for the name of the host reported to SAP during the heartbeat status notification. If left blank or not set then the name of the host machine on which the server is running is used (in the vast majority of cases this will be the most appropriate setting).	As noted.
HeartbeatHostAddress	An override for the address of the host reported to SAP during the heartbeat status notification. Must be in the standard IP address format xxx.xxx.xxx.xxx. If left blank or not set then the IP address of the host machine on which the server is running is used (in the vast majority of cases this will be the most appropriate setting).	As noted.
HeartbeatLossSeverity	Allows control over the severity of the log messages written to the log when heartbeat is lost. Possible values are as follows: I - Log messages for heartbeat loss are treated as information (INFO) messages. W - Log messages for heartbeat loss are treated as warning (WARN) messages. E - Log messages for heartbeat loss are treated as error (ERROR) messages.	E

SERVER.HOSTFUNCTIONAILTYSUPPORT

This section is used to tell the server what types of functionality are supported by the back-end host. It permits certain areas of functionality to be disabled, for example, if they are not supported by the host. The server can be

made to run in standalone mode if all HostFunctionalitySupport options are disabled.

Option	Description	Default
FunctionalitySupport:Application	Indicates whether the host supports download and refresh of applications.	true
FunctionalitySupport:BinaryFiles	Indicates whether the host supports download of binary files.	true
FunctionalitySupport:Heartbeat	Indicates whether the host supports heartbeat messages.	true
FunctionalitySupport:LocalDatabase	Indicates whether the host supports download, refresh and upload of local database tables.	true
FunctionalitySupport:Packages	Indicates whether the host supports download of packages.	true
FunctionalitySupport:SessionManager	Indicates whether the host supports session manager functionality.	true

SERVER.HOSTFUNCTIONNAMES

This section controls which functions are invoked by the server against the back-end host(s). All of the following configuration items can also be overridden at the host interface level if they are placed inside the configuration file section referring to the host interface in question. These defaults should not be changed unless instructed by Sky support.

Option	Description	Default
HeartbeatFunctionName	The name of the function invoked by the heartbeat (defaults to /SKY/VTI_HEARTBEAT).	As noted.

Option	Description	Default
LdbDownloadFunctionName	The name of the function invoked by a local database download (defaults to /SKY/VTI_DOWNLOAD_LOCAL_DATA).	As noted.
LdbRefreshFunctionName	The name of the function invoked by a local database refresh (defaults to /SKY/VTI_REFRESH_LOCAL_DATA).	As noted.
LdbUploadFunctionName	The name of the function invoked by a local database upload (defaults to /SKY/VTI_UPLOAD_LOCAL_DATA).	As noted.
RelayConnectionsFunctionName	The name of the function used to notify the host of the relay connections currently available on the server's relay ports (defaults to /SKY/JAVA_RELAY_CONNECTIONS).	As noted.
SelectBinaryFunctionName	The name of the function used to retrieve the data associated with a binary file (defaults to /SKY/VTI_SELECT_BINARY_DATA).	As noted.
SelectDefinitionsFunctionName	The name of the function used to retrieve an application definition (defaults to /SKY/VTI_SELECT_DEFINITIONS).	As noted.
SelectFunctionDataFunctionName	The name of the function used to retrieve the definition of a single function (defaults to /SKY/VTI_SELECT_FUNCTION_DATA).	As noted.
Select PackageFunctionName	The name of the function used to retrieve details about a package (defaults to /SKY/VTI_SELECT_PACKAGE).	As noted.

Option	Description	Default
SelectPackagesFunctionName	The name of the function used to retrieve information about the packages relevant to the server instance (defaults to /SKY/VTI_SELECT_PACKAGES).	As noted.
SelectTimestampsFunctionName	The name of the function used to retrieve information about the applications relevant to the server instance (defaults to /SKY/VTI_SELECT_TIMESTAMPS).	As noted.
SessionManagerFunctionName	The name of the function used to process session manager calls (defaults to /SKY/VTI_SESSION_MANAGER).	As noted.
UpdateMd5HashFunctionName	The name of the function used to send the host an MD5 hash for a binary file that does not currently have one (defaults to /SKY/VTI_UPDATE_MD5_HASH).	As noted.
UpdatePackageStatusFunctionName	The name of the function used to notify a change in the current status of deployment of a package (defaults to /SKY/VTI_UPDATE_PACKAGE_STATUS).	As noted.
UploadBinaryDataFunctionName	The name of the function used to upload the data of a binary file (defaults to /SKY/VTI_UPLOAD_BINARY_DATA).	As noted.

SERVER.HOSTINTERFACE_RFC

These section(s) contains the information required by the server to establish a direct RFC connection to one or more SAP instance(s).

Option	Description	Default
HostInterfaceType	Must be set to "RfcClient" for direct RFC connections.	RfcClient
SapClient	Numeric SAP client identifier.	No default value.
SapUser	The SAP user id to use. Should be a CPIC user.	No default value.
SapPassword	The SAP password to use.	No default value.
SapLanguage	The SAP language identifier for example, E (English).	E
SapTraceLevel	The SAP RFC trace level (should be left as 0 in normal operation).	0
SapUseLB	A true/false flag to instruct the server whether to use the SAP load balancing feature.	No default value.
SapLBHost	Host for load balancing.	No default value.
SapLBSystem	SAP system identifier for load balancing.	No default value.

Option	Description	Default
SapLBGroup	The SAP load balancing group.	No default value.
SapHost	A specific SAP host to connect to (alternative to load balancing).	No default value.
SapSystem	A specific SAP system identifier to connect to (alternative to load balancing).	No default value.
SapConnectionTimeout	The number of seconds to wait before terminating a attempt to connect to a SAP system.	0 (no timeout)
JniLogging	A true/false flag that controls whether JNI tracing information is written to the log. This trace is primarily intended for Sky support.	false (no JNI tracing)
UseTransferFields	Indicates whether information on local database table fields should be sent during table downloads, refreshes, and uploads. For SAP systems this is not needed and should be set to "false".	False
MaximumOptimizationLevel	This is a numeric value that limits optimization behavior during session manager processing. The highest optimization currently supported is 1. A value of 0 will completely turn off session manager optimization. In most scenarios there should be no need to set this value.	1

Option	Description	Default
MinimumNumberHandlers	The minimum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0
MaximumNumberHandlers	The maximum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via a connection on an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0 (no limit)

SERVER.HOSTINTERFACE_GATEWAY

These section(s) contains the information required by the Application Server to establish a connection to a back-end host system via a gateway.

Option	Description	Default
HostInterfaceType	Must be set to "XmlClient" for Access Gateway connections.	XmlClient
XmlGatewayHosts	A comma-separated list of the host(s) on which Access Gateway(s) are running. Each value can take the form of either a host name or an IP address.	No default value.

Option	Description	Default
XmlGatewayHostSelection	The manner in which an Access Gateway should be selected from the list of those available. The following values are permitted: Random - The Access Gateway should be selected at random. Sequential - Access Gateway entries that appear earlier in the list have preference over those which appear later.	Sequential
XmlGatewayPort	The port number on which the Access Gateway (s) are listening for incoming XML. Please note that when multiple Access Gateways are in use, they must all be configured to use the same port.	No default value.
XmlGatewayResponseTimeout	The maximum period of time (in seconds) for which the server will wait for an initial response from the Access Gateway before timing the transaction out.	30 (seconds)
XmlGatewayStallTimeout	The maximum period of time (in seconds) for which the server will allow an incoming response to stall (i.e. send no further data) before timing the transaction out.	30 (seconds)
UseCompression	A true/false flag that indicates whether the XML data sent to the gateway should be compressed. This greatly reduces bandwidth consumption.	false

Option	Description	Default
CompressionAlgorithm	The compression algorithm to use when compressing data sent to the Access Gateway. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.
UseEncryption	A true/false flag that indicates whether the XML data sent to the gateway should be encrypted. Obviously, this is a more secure option, particularly over a public network.	false
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to the Access Gateway. The following encryption algorithms are available: AES - The AES (Rijndael) encryption scheme. BLOWFISH - The "blowfish" encryption scheme.	No default value.
EncryptionKey	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
EncryptionKeyFile	The file in which the encryption key is stored.	No default value.
EncryptionKeyHexEncoded	A true/false flag that indicates whether the encryption key is hex-encoded.	false

Option	Description	Default
EncryptionKeyStrength	If handshaking encryption is in use, this parameter specifies the length of the randomly generated session key to be used following the initial handshake (in number of bits).	No default value.
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the Access Gateway. If left blank, no initial handshaking is performed. The following encryption algorithms are available: RSA - The "RSA" encryption algorithm (recommended).	No default value.
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the Access Gateway. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	No default value.
HandshakeEncryptionKeyHex-Encoded	A true/false flag that indicates whether the handshaking encryption key is hexencoded.	False
XmlDebug	A true/false flag that indicates whether all XML data sent to, and received from, the Access Gateway should be logged. This facility can be useful for debugging in some situations, but it generates large log files and should generally be left off.	False

Option	Description	Default
UseCondensedXml	A true/false flag that indicates whether or not XML should be sent in "condensed mode". Barring problem scenarios, this option should normally be enabled. Turning it off has a performance impact.	True
UseTransferFields	Indicates whether information on local database table fields should be sent during table downloads, refreshes, and uploads. For SAP systems this is not needed and should be set to "false".	False
MinimumNumberHandlers	The minimum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0
MaximumNumberHandlers	The maximum number of request handlers to keep on hand to process incoming Access Gateway requests. Please note that this value ONLY pertains to incoming requests made via a connection on an Access Gateway port. Unless the VTI server is also acting as an Access Gateway, there is no need to set this value.	0 (no limit)

SERVER.HTTPSERVICESPORT_XXX

This section contains parameters for port (s) to allow HTTP access to services defined through the Integration Object workbench in SAP. The xxx should be substituted with a unique meaningful name for the HTTP services port.

Option	Description	Default
PORT	The port number to listen on for incoming HTTP service requests. No default value.	No default value.
HOSTINTERFACE	The host interface definition to use. No default value.	No default value.
DEFAULTIDENTITYSERVICE	The name of an implemented identity management service to use as the default if the calling Secure Container sends an identity management request and does not specify one.	None
XMLDEBUG	A true / false flag that controls whether the server writes information about incoming and outgoing XML received on the port, into its log. This option has a substantial performance impact and generates very large log files, so it should generally be left disabled.	false
ALLOWHEADERSINBODY	A true / false flag that controls whether the server replicates the HTTP headers processing using the message body. If enabled, the HTTP services port looks for a headers element in the message body on inbound messages and treats any attribute of that element as if it were an HTTP header. On outbound processing, it inserts a headers element into the response body and adds an attribute for each response header.	false

SERVER.IDENTITYPORT

This section describes the options available to implement a identity management port. Typically, identity requests are directed through a normal gateway port, but this configuration gives you the option to expose a specific identity port.

Option	Description	Default
Port	The port number to listen on for incoming messages.	None
DefaultIdentityService	The name of an implemented identity management service to use as the default if the calling Secure Container does not specify one.	None
UseCompression	A true/false flag that indicates whether the XML data sent to connected clients on the port should be compressed.	True
CompressionAlgorithm	The compression algorithm to use when compressing data sent to connected clients on the port. The following compression algorithms are available: GZIP - The GZIP compression algorithm. ZLIB - The ZLIB compression algorithm.	GZIP
UseEncryption	A true/false flag that indicates whether the XML data sent to connected clients on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	False

Option	Description	Default
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to connected clients on the port. The following encryption algorithms are available: AES - The AES encryption algorithm. BLOWFISH - The Blowfish encryption algorithm. DES - The DES encryption algorithm. 3DES - The triple-DES encryption algorithm.	None
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the remote partner. If left blank, no initial handshaking is performed. The following encryption algorithms are available: RSA - The RSA encryption algorithm.	None
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the remote partner. If no encryption key is specified, the server will look for one in an encryption key file instead.	None
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	None
HandshakeEncryptionKeyHexEncoded	A true/false flag that indicates whether the handshaking encryption key is hex encoded.	False

Option	Description	Default
DefaultIdentityService	The name of a implemented identity management service to use as the default if the calling Secure Container does not specify one.	None

SERVER.IDENTITYSERVICE:XXX

The identity service section is used to implement SkyMobile identity management services. The ones configured here are the inbuilt Sky services. Customers can also implement their own service modules, in which case they implement their own custom sections.

Option	Description	Default
REGISTERSESSIONS	To enable advanced session security (only required when enabled in YVTI in the MEAP)	False
HOSTINTERFACE	The name of the host interface to use when communicating with SAP. (Only required if REGISTERSESSIONS is enabled)	None
DEFAULT:XXXX	A flag to allow clients to display the last used value for an identity parameter XXXX.	False
COPY:XXXX	To enable multiple targets for identity parameters, XXXX is the TARGET attribute, and the value of this configuration option is the SOURCE identity parameter, that is; ...COPY:USER = PARAMETER1, will enable the PARAMETER1 identity paremeter to target both PARAMETER1 and USER.	None

SERVER.IDENTITYSERVICE:LDAP

Option	Description	Default
SERVICETYPE	LDAP	

Option	Description	Default
USEDOMAIN	A flag to indicate whether to capture and/or use a domain as part of the login process.	False
DOMAIN	If USEDOMAIN is true, any value entered for this parameter will always be picked up as the user's domain. The user will not be prompted to enter their domain.	None
DEFAULTDOMAIN	If USEDOMAIN is true and DOMAIN is not specified, any value entered for this parameter will be the default value for the domain field when the user is prompted for the domain.	None
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.	0
SERVER	The name or address of the LDAP server.	None
PORT	The port on which to connect to the LDAP server.	389 if SSL is not enabled 443 if SSL is enabled
BINDCONTEXT	The initial bind context to use when attaching to the LDAP directory.	"/"

Option	Description	Default
TIMEOUT	The maximum number of seconds for which to wait for a response from the LDAP server before giving a timeout error.	10
USESSL	A flag to indicate whether or not to use SSL when communicating with the LDAP server.	False
SSLKEYSTORE	For use only when SSL is enabled, this is the pathname to the SSL key store file.	None
SSLKEYSTOREPASSWORD	For use only when SSL is enabled, this is the password to the SSL key store file.	None
SSLTRUSTSTORE	For use only when SSL is enabled, this is the pathname to the SSL trust store file.	None
SSLTRUSTSTOREPASSWORD	For use only when SSL is enabled, this is the password to the SSL trust store file.	None
SSLDEBUG	For use only when SSL is enabled, this is a flag to indicate whether SSL debugging should be enabled	false
LOOKUP:XXXX	Extract a value from LDAP and insert into identity parameter XXX, the value being the attribute within LDAP, that is; ...LOOKUP:PARAMETER1 = myLdapAttribute, will use the ldap attribute myLdapAttrubute as the value within the PARAMETER1 identity paremeter.	None

SERVER.IDENTITYSERVICE:SAP

Option	Description	Default	
SERVICETYPE	SAP		
HOSTINTERFACE	The name of the host interface to use when communicating with SAP.	None	
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.		0

SERVER.IDENTITYSERVICE:SKY

Option	Description	Default
SERVICETYPE	SkyMobile	
HOSTINTERFACE	The name of the host interface to use when communicating with SAP.	15
LOGFAILURES	Log unsuccessful identity check attempts	False
LOGSUCCESSES	Log successful identity checks	False
OFFLINELEASEHOURS	This is the number of hours for which securely stored login credentials remain valid for subsequent validation of offline logins. A negative or zero value means that the credentials remain valid indefinitely.	0
SENDPASSWORDTOHOST	Send the identity password to the host to be used by the server by user switching	False

SERVER.JAVA

This section controls invocation of the Java Virtual Machine (JVM).

Option	Description	Default
JAVACOMMAND	The path to the executable used to launch the Java Virtual Machine (for example, java.exe). Under Windows, this should be in short file name (MS-DOS 8.3) format, for maximum compatibility. If not specified, the value of the environment variable SKY_JAVA_CMD is used instead.	As noted.
JAVAARGUMENTS	Any additional arguments to be passed to the Java command line. Common examples are arguments to increase the maximum memory ceiling (-Xmx128m) or run with a reduced signal set (-Xrs).	- (no arguments)

SERVER.LOCALDATABASE

This section contains information pertaining to the locally cached database tables.

Option	Description	Default
MINIMUMNUMBERUPDATETHREADS	The minimum number of threads available for use in keeping the local database tables up to date.	1
MAXIMUMNUMBERUPDATETHREADS	The maximum number of threads available for use in keeping the local database tables up to date.	3
KEEPHOSTCONNECTIONS	A true/false flag that controls whether the local database processing threads attempt to maintain a continuous connection to the host.	True

Option	Description	Default
ERRORRETRYINTERVAL	If the server experiences an error when it is updating a table in the background, this is the number of seconds that it waits before it tries again.	3600 (seconds)
TRANSFERBUFFERSIZE	The maximum number of bytes that may be transferred to the Application Server from the host. This setting prevents large transfers from overwhelming devices with a low amount of memory or processing power. Instead, the transfer is broken into multiple 'chunks' which are transferred in multiple calls.	0 (no restriction)

SERVER.LOG

This section contains information about the logging and tracing capabilities.

Option	Description	Default
DIRECTORY	The directory into which the server will write its log files. If not set, this parameter defaults to "." (the current working directory).	As noted.

Option	Description	Default
STRATEGY	The logging strategy to be adopted. Must be one of the following: D - A new log file is to be created daily. W - A new log file is to be created weekly. P - A new log file is to be created for each process instance O - Overwrite the log file each time the server is restarted.	O
EXCLUDES	Indicates the types of log messages that should be excluded from the log file. There are 6 types of log messages - Information (I), Warning (W), Performance Track (P), Trace (T), Error (E) and Abort(A). The log message types to exclude should all be concatenated together. Thus, to exclude all Information and Trace messages from the log file, the appropriate setting would be "IT".	- (no types excluded)
MAXFILES	This setting restricts the maximum number of log files the Application Server keeps under D, W, and P strategies. If left blank or not set then no maximum is enforced. 0 indicates no maximum.	0 (no maximum)

Option	Description	Default
SHOWMILLISECONDS	A true/false flag that controls whether milliseconds are shown in log messages. This feature is useful for measuring sub-second timings and events.	False
PERFORMANCETRACKING	A true/false flag that controls whether or not performance tracking messages are written to the VTI log.	False

Option	Description	Default
MESSAGESEVERITYOVERRIDE:XXX	This configuration directive permits specific override of the behavior of individual log file messages. There can be multiple instances of this directive within the [Log] section. By default, XXX is assumed to be the name of a log message resource identifier (as listed in the vti.sample.properties file) – for example, VTI_MSGFMT_VTI_STARTED. However, if the value specified in place of the XXX is enclosed in double quotes, then it is instead assumed to be a text pattern to match against the log message text (for example, "Cannot read directory *". The value assigned to the directive should be one of the available types of log message (I, W, P, T, E, or A). This will cause all log messages that match either the specified resource identifier or the specified text pattern, to be treated as though they had the severity specified. Thus, individual error messages could be reduced in severity to warnings, for example, or just information messages. Additionally, a value of "X" can be specified that will cause matching messages to be completely ignored (excluded). ***Note:*** You must use this feature with caution as you are generally not recommended to ignore or downgrade error messages.	N/A

Option	Description	Default
SENDERRORLOGS	A true/false flag that controls whether the server sends error log messages up to SAP. This feature is useful for monitoring and support purposes.	False

SERVER.MONITOR

This section contains setup information for internal monitoring capability.

Option	Description	Default
USEMONITOR	A true/false flag that enables or disables the internal monitoring capability.	false
MONITORINTERVAL	The time interval (in seconds) between monitoring status checks.	60 (seconds)
GARBAGECOLLECT	A true/false flag that indicates whether Java garbage collection (which expends resources in an attempt to free up memory) should be performed as part of each monitoring status check. If set to true, garbage collection takes place each time a monitoring status check occurs. If set to false, garbage collection is managed by the Java Virtual Machine.	false
LOGMEMORYUSAGE	A true/false flag that indicates whether current memory usage information should be written to the log as part of each monitoring status check. If the both this option and the GarbageCollect option are set to true, memory usage information is logged both before and after the garbage collection takes place.	false

Option	Description	Default
LOGTHREADCOUNT	A true/false flag that indicates whether the number of active threads should be written to the log as part of each monitoring status check.	false
LOGTHREADNAMES	A true/false flag that indicates whether the names and states of active threads should be written to the log as part of each monitoring status check.	false
CLEARRFCTRACEFILES	A true/false flag that indicates whether to automatically delete logging/trace files created by the SAP RFC library.	false
RFCTRACEEXPIRYDAYS	The maximum number of days for which to retain SAP RFC trace files when automatic deletion is enabled.	7 (days)

SERVER.NETWORK

This section contains information on the servers interaction with the network.

Option	Description	Default
AcceptTimeout	The maximum period of time (in milliseconds) for which the thread associated with each port will block while waiting for new incoming connections. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)

Option	Description	Default
BindTimeout	The maximum period of time (in seconds) for which the server will wait while attempting to bind to its listening ports. If another process is listening on its designated ports, or if the bind process fails, then the server will not be able to establish the listening port. Once the timeout occurs, it will close down with an appropriate error message. If this parameter is set to zero or not specified, then the server will wait indefinitely to bind to its listening ports in the hope that they eventually become available.	0 (no timeout)
CheckForHostAddressChanges	A true/false flag that indicates whether the server should monitor the available network interfaces for changes to its IP address and/or host name. In situations where the network interface configuration is not subject to change, this can represent an unwanted processing overhead.	true
HostNameResolution	A true/false flag that indicates whether host name resolution is desired. Resolution of the host name can sometimes be a considerable overhead on performance. For this reason it is disabled by default. However, it should be enabled in situations where performance is not an issue, and display of the hostname rather than the IP address is felt to be of benefit.	false

Option	Description	Default
NetworkErrorSampleSize	The sample size for network error monitoring. The server maintains a moving window of the success or failure of the last n network calls, where n is the value of this parameter. If not set or left as zero, any network error is treated as fatal for the port or connection on which it occurs. For most well-tuned networks this should be the chosen setting.	0 (no error sampling)
NetworkErrorThreshold	The error threshold for network error monitoring. Must be less than or equal to the NetworkErrorSampleSize parameter. When this number of network errors occurs within the sample, the server will conclude that the connection or listening port has unrecoverable network problems and will close it down. It will then attempt to re-establish the connection or port in the hope of obtaining a more stable socket.	0 (no error sampling)
PreferredIpAddresses	A comma-delimited list of addresses and/or address ranges that the server will prefer to assume in a multihomed network scenario (in order of preference). These addresses must be in raw IP address form (xxx.xxx.xxx.xxx). An asterisk (*) can be used as a wildcard to substitute for any single value within the address. It is also possible to specify a valid range for a given octet by specifying the upper and lower boundaries, separated by a hyphen (for example, 192.168.2.100-200).	- (no preferred addresses)

Option	Description	Default
ReadTimeout	The maximum period of time (in milliseconds) for which the threads associated with connections on this port will block while waiting for input. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)
UseNativeSockets	A true/false flag that indicates whether native socket calls should be used instead of the capabilities provided by the Java virtual machine. This can sometimes provide additional information when errors occur, and will typically improve performance.	false
ValidIpAddresses	A comma-delimited list of addresses from which incoming connections will be accepted. These addresses can either be in humanreadable form, or in IP address form (xxx.xxx.xxx.xxx). An asterisk (*) can be used as a wildcard to substitute for any single value within the address. It is also possible to specify a valid range for a given octet by specifying the upper and lower boundaries, separated by a hyphen (for example, 192.168.2.100-200). If this parameter is specified, then any incoming connection from a host not listed will be rejected. If this parameter is left blank or not specified, then by default all incoming connections are considered valid regardless of the host from which they originate. Note that this parameter can be overridden for individual ports.	- (all addresses are valid)

Option	Description	Default
InvalidIpAddresses	A comma-delimited list of addresses from which incoming connections will not be accepted. This format of this list is identical to that described for the valid IP address list. Note that the invalid IP address list has a higher order of precedence than the valid IP address list, so a host which matches both lists is considered invalid. Note that this parameter can be overridden for individual ports.	- (no addresses are invalid)

SERVER.POLLDIRECTORY_XXX

These section(s) contain information pertaining to one or more poll directories that the Application Server should monitor for incoming files. When an incoming file is detected, a userwritten Java exit is automatically invoked to take any action that may be appropriate.

Option	Description	Default
IncomingDirectory	The pathname to a directory in which incoming files will be placed by an external process.	No default value.
ArchiveDirectory	The pathname to a directory in which files will be placed by the server once they have been successfully processed. If left blank or not present, then any successfully processed files will be deleted rather than being transferred to an archive directory.	As noted.

Option	Description	Default
ErrorDirectory	The pathname to a directory in which files will be placed by the server once they have been unsuccessfully processed (i.e. found to have errors). If left blank or not present, then any files which could not be processed will be deleted rather than being transferred to an error directory.	As noted.
IncomingFilePattern	A filename pattern that will be used to check whether files in the incoming directory are suitable for processing. Wildcard characters are permitted: an asterisk (*) matches any number of characters, and a question mark (?) matches any single character. If left blank or not present, then any file in the incoming directory is considered to be suitable for processing.	As noted.
PollFrequency	The time interval (in seconds) between polls of the incoming file directory.	No default value.
PollFrequencyMilliseconds	The time interval (in milliseconds) between polls of the incoming file directory. This interval is added to the number of seconds specified in the "PollFrequency" item.	0
LogIncomingFiles	A true/false flag that indicates whether log messages should be written for each incoming file that is detected.	false

Option	Description	Default
StopPollingOnError	A true/false flag that indicates the normal behavior for the poll directory when an error occurs. If set to true, all polling will stop when an error is encountered, and polling must be manually restarted from the web status page. If set to false, polling will continue to take place after an error is encountered.	false
ShellCommand	The command that should be executed to invoke a system shell. Normally the server is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh --s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by the server, try changing this parameter to "cmd.exe /c".	As noted.
SystemCommand	The command to be executed whenever an incoming file is detected. If not specified or left blank, it is assumed that no system command needs to be executed.	As noted.

Option	Description	Default
ProcessCommandViaInputStream	A true/false flag to control whether or not the system command should be read from an input stream. In most cases this flag should not need to be explicitly set as it defaults to behavior that is normally appropriate for the target OS.	As noted.
PollDirectoryExit	The fully qualified name of the user written Java exit to be invoked when an incoming file is detected. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.

SERVER.PRINTTEMPLATE_XXX

These section(s) contain information pertaining to one or more print templates that the Application Server will then make available to userwritten Java exits.

Option	Description	Default
TemplateFileName	The pathname of the text file containing the raw print template information.	No default value.
IgnoreUnknownVariables	A true/false flag that specifies whether or not any unknown substitution variables found in the template file should be ignored. Normally this would give rise to an error. However, if this option is enabled, processing will silently continue and the unknown substitution variable will be left unchanged. This flag can be useful in cases where the print template is a postscript file containing characters that resemble substitution variables.	false

Option	Description	Default
CreateWorkFiles	A true/false flag that indicates whether to create a work file containing the output from the print template when it is invoked.	true
WorkDirectory	The pathname to the directory into which work files should be written.	No default value.
WorkFileExtension	The file extension to use when creating work files.	.txt
DeleteWorkFiles	A true/false flag that indicates whether work files should be immediately deleted following processing.	true
WorkFileExpiryDays	The maximum number of days for which to retain work files (where they are not deleted immediately after being processed).	0
ExecutePrintCommand	A true/false flag that indicates whether a print command should be executed when the print template is invoked.	true

Option	Description	Default
ShellCommand	The command that should be executed to invoke a system shell. Normally the server is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh --s If you are running under Windows 2000, XP or 2003, and find that the print command you are executing works correctly from the command line, but does not work when invoked by the server, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
PrintCommand	The system command to be executed in order to print the contents of the work file. This system command may contain substitution variables that will be replaced with a corresponding runtime value. A list of the available substitution variables is as follows: &century& - The current four-digit century. &day& - The current two-digit day of the month. &filedir& - The pathname of the work file directory. &filename& - The name of the work file. &filepath& - The full pathname of the work file. &hour& - The current two-digit hour. &minute& - The current two-digit minute. &month& - The current two-digit month. &second& - The current two-digit second. &year& - The current two-digit year.	No default value.
ProcessCommandViaInputStream	A true/false flag to control whether or not the system command should be read from an input stream. In most cases this flag should not need to be explicitly set as it defaults to behavior that is normally appropriate for the target OS.	As noted.
PrintCommandTimeout	The maximum period of time (in seconds) for which to wait for a print command to finish executing. If the print command takes longer than this period of time to complete, it is terminated by the server. A timeout value of 0 means that no timeout is applicable.	10 (no timeout)

Option	Description	Default
LogPrintCommands	A true/false flag that indicates whether information messages should be written to the log each time a print command is executed.	True
SubstitutionVariable:XXX	Allows print template substitution variables to be specified in the configuration file. XXX should be the name of the substitution variable in question. For example, it would be possible to set a configuration option as follows: SubstitutionVariable:LARGEFONT = \027w This causes any instances of &LARGEFONT& in the print template to be replaced with the specified character sequence (an escape [ASCII 27] followed by a small 'w').	No default value.

SERVER.RELAYCONNECTIONS

This section controls whether (and how) the Application Server opens connections to a relay port running on a gateway. A relay connection allows remote manipulation of the server via an interface similar to that offered by the server's web status page.

Option	Description	Default
RELAYHOST	The name or address of the host on which the relay port is available (i.e. the gateway machine). If left blank or not specified, relay connections are not permitted.	No default value.

Option	Description	Default
RELAYPORT	The port number of the relay port. If left blank or not specified, relay connections are not permitted.	No default value.
ALLOWCONFIGCHANGESFROMWEBPAGE	A true/false flag that controls whether the configuration file can be modified via the relay web interface.	false
ALLOWFILEENQUIRYFROMWEBPAGE	A true/false flag that controls whether remote file management can be used to view files via the relay web interface.	false
ALLOWFILEUPDATESFROMWEBPAGE	A true/false flag that controls whether remote file management can be used to update files via the relay web interface.	false
ALLOWLDBENQUIRYFROMWEBPAGE	A true/false flag that controls whether local database tables can be viewed via the relay web interface.	true
ALLOWLDBUPDATESFROMWEBPAGE	A true/false flag that controls whether local database tables can be updated via the relay web interface.	

Option	Description	Default
AUTHORISATIONUSERNAME	The user ID to be used to secure the relay web interface. If not present or left blank, no logon security is enforced.	- (no logon security)
AUTHORISATIONPASSWORD	The password to be used to secure the relay web interface.	- (no password)
HTMLBUTTONLOCATION	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next/previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward etc.	Bottom
HTMLPAGESIZE	The maximum number of lines to display on the web pages.	15
SHOWLDBROWCOUNTS	A true/false flag that indicates whether, by default, row counts are listed against local database tables listed in the relay web interface.	false

Option	Description	Default
SUPPRESSLOGO	A true/false flag that allows suppression of the Sky Technologies logo on the relay web pages (to conserve bandwidth and improve performance).	false
USEHTMLCOMPRESSION	A true/false flag that indicates whether HTML compression should be used for web pages sent to browsers which support it.	true
RECONNECTSECONDS	The time interval (in seconds) between attempts to reconnect if the relay connection is severed.	30

SERVER.SERVEREXITS

This section contains information on user-written Java exits that should be invoked by the server. Server exits are not associated with any individual user session and effectively allow custom-written code to be executed in the background by the server.

Option	Description	Default
<SERVER EXIT CLASS NAME>	The fully qualified class name for each server exit class should be placed as a parameter in the "ServerExits" section to the left of the equals sign. The class in question must be available to the JVM in which the server is running (the class, or the jar/zip file containing the class, should be placed inside the "classes" directory). The value assigned to each server exit parameter is a true/false flag that tells the server whether or not the server exit is currently enabled.	No default value.

SERVER.STORAGE

This section contains information on where and how the Application Server should cache information that needs

to be persistently stored (including both the application cache and the local database).

Option	Description	Default
StorageType	The storage strategy to be adopted. Must be one of the following: File - Persistent data should be stored as files within a local file system (this is the default behavior). JDBC - Persistent data should be stored in a relational database. Communication with the database takes place via a JDBC driver. Database - Same as JDBC. SkyDB - Persistent data should be stored in the embedded SkyDB database. BlackBerry - Internal blackberry persistence storage.	File BlackBerry defaults to BlackBerry (internal persistance engine).
StorageRootDirectory	For use when the storage type is "File". This parameter tells the server which directory has been set aside for its use in storing persistent data. Defaults to "." (The current working directory for the application or web server into which the server has been loaded).	As noted.

Option	Description	Default
StorageDatabaseDriver	For use when the storage type is "JDBC". Should be set to the fully qualified name of the JDBC driver to be used in communicating with the storage database. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.
StorageDatabaseURL	For use when the storage type is "JDBC". The precise format of this URL will vary depending on the JDBC driver in use. Consult the documentation on your JDBC driver for further information.	No default value.
StorageDatabaseUser	For use when the storage type is "JDBC". The database user to use when establishing a connection to the storage database.	No default value.
StorageDatabasePassword	For use when the storage type is "JDBC". The password to use when establishing a connection to the storage database.	No default value.

Option	Description	Default
StorageNationalCharacter-SupportType	For use when the storage type is "JDBC". Describes the type of support for "National" characters (i.e. storage of nonEnglish or Unicode data). Three options are available: None - No such support is offered by the database. If running in Unicode mode against a database of this type, all data will be encoded/decoded on the way to/from the database. Standard - Indicates that data types such as NCHAR, NVARCHAR etc are fully supported by the database. Inbuilt - Indicates that Unicode data is handled transparently by the database with no special SQL syntax required.	Standard
StorageReservedWordProtection	For use when the storage type is "JDBC". This true/false flag helps circumvents problems with column names that may happen to be reserved words for the database currently in use. When enabled, all database column names are automatically prefixed with the specified prefix (see the next item).	true

Option	Description	Default
StorageReservedWordProtection-Prefix	For use when the storage type is "JDBC". This specifies the prefix to use in front of column names when reserved word protection is in use.	"_" (underscore)
StorageSqlLogging	For use when the storage type is "JDBC" or "SkyDB". This true/false flag indicates whether or not SQL sent to the storage database should be written to the log. This can be useful for debugging in some situations, but generates large log files and should generally be left off.	false
StorageCheckColumnSizes	For use when the storage type is "JDBC". This true/false flag enables or disables checking of column sizes at startup. For databases with dynamic column sizing (for example, H2), this check is not necessary and can in fact give unnecessary warnings.	true
StorageDatabaseInitialSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that will be executed when the server first starts (immediately after establishing its first connection to the database).	- (no custom SQL)

Option	Description	Default
StorageDatabaseFinalSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that will be executed when the server shuts down, just before it disconnects from the database for the last time.	- (no custom SQL)
StorageSessionInitialSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that will be executed each time a new connection is made to the database (before anything else occurs on the connection).	- (no custom SQL)
StorageSessionFinalSql	For use when the storage type is "JDBC". Allows entry of some custom SQL that will be executed just before a connection to the database is closed (after everything else has completed on the connection).	- (no custom SQL)

Option	Description	Default
StorageNullValueChecking	For use when the storage type is "JDBC". A true/false flag to indicate whether the database stores empty strings as nulls, hence necessitating null value checks whenever data is fetched from the database. Oracle is one example of a database that demonstrates this behavior. ***Note:*** This flag must be set to "true" if Oracle is in use.	false (no null value checking)

Option	Description	Default
StorageDatabaseFileName	For use when the storage type is "SkyDB". This is the name of the database file. ***Note:*** For blackberry devices, this configuration item does not specify the direct path to the database file. Instead it is expecting one of two values for this parameter. InternalFileSystem, or ExternalFileSystem. If InternalFileSystem is specified, the application will look for any root file system based on the blackberry internal 'store' memory, and secondly for any root file system based on the blackberry internal 'system' memory. If ExternalFileSystem is specified, the application will look for any root file system based on an 'sdcard', secondly it will look for any root file system based on a 'cfcard', and thirdly it will look for any root file system based on a 'memorystick'. Refer to BlackBerry Storage Options for more information.	No default value. BlackBerry defaults to InternalFileSystem.

Option	Description	Default
StoragePageCacheSize	For use when the storage type is "SkyDB". This is the maximum number of pages that will be cached in memory simultaneously. Each page is 4096 bytes in size, and also carries with it an in-memory representation of the items that it contains. Overall, a memory consumption of something less than 10 KB per cached page can typically be assumed. A zero or negative value turns page caching off, but this is not recommended for performance reasons.	1250
StorageReadLocks	For use when the storage type is "SkyDB". This true/false flag controls whether or not SkyDB takes out locks on a table during read operations (for example, selection of records).	false
StorageInternalLogging	For use when the storage type is "SkyDB". This true/false flag controls whether or not SkyDB writes information about its internal operations to the system log.	false

Option	Description	Default
StorageConsistencyCheck	For use when the storage type is "SkyDB". This true/false flag controls whether or not the database should be checked for consistency on startup. This slightly delays the startup process, but will report any problems encountered in the log.	false
StorageCompressOnShutdown	For use when the storage type is "SkyDB". This true/false flag indicates whether or not the database should be compressed during the shutdown process. This makes the database file smaller, but slightly delays the shutdown process.	true
StorageIndexDefragOnShutdown	For use when the storage type is "SkyDB". This true/false flag controls whether or not index defragmentation is performed during the shutdown process. This decreases the overall size of the database and also improves index performance, but slightly delays the shutdown process.	true

Option	Description	Default
StorageInternalizeStrings	For use when the storage type is "SkyDB". This true/false flag indicates whether or not string data should be "internalised". This consolidates shared strings and results in much lower memory footprint. It has a slight performance impact, which is typically offset by the much lower memory consumption. ***Note:*** Don't enable this feature when using the CrEme JVM, as too much string data results in internal overflows within the JVM that cause it to "crash" with the error "*** panic[21]: 16-bit string hash table overflow"	false
StorageIndexRebuild	A true/false flag that tells VTI whether to rebuild all indices (both primary and secondary) on all tables at startup. This flag is primarily intended for use during upgrades from a previous release. It should not be left enabled as a matter of course, as this has a performance impact.	false

Option	Description	Default
StorageFullEncryption	A true/false flag that controls whether the database contents get encrypted. If enabled the EncryptionAlgorithm, EncryptionKey/EncryptionKeyFile options become relevant. Currently only supported by SkyDB.	false
EncryptionAlgorithm	This option is currently only supported by SkyDB. It specifies the (symmetric) encryption algorithm to be used to encrypt secured data. The following values are permitted: AES - The AES (Rijndael) encryption scheme. BLOWFISH - The "blowfish" encryption scheme.	BLOWFISH
EncryptionKey	This option is currently only supported by SkyDB. It specifies the encryption key to be used to encrypt secured data. If no encryption key is specified, the server will look for one in an encryption key file instead.	As noted.
EncryptionKeyFile	This option is currently only supported by SkyDB. It specifies the name of the file containing the encryption key. If no encryption key is present here either, one will be internally generated by the server.	As noted.

Option	Description	Default
EncryptionKeyHexEncoded	This option is currently only supported by SkyDB. It is a true/false flag that specifies whether or not the encryption key is hexencoded.	false
StorageHandlerAllocationTimeout	The maximum period of time (in seconds) that the server will wait for access to its persistent storage cache during any given operation. If a storage handler cannot be allocated within this time period then the operation in question will fail.	10 (seconds)
StorageHandlerInactivityTimeout	The maximum period of time (in seconds) that the server keeps hold of an idle storage handler. If any given storage handler remains unused for this length of time, any associated database connection will be closed, and the storage handler will then be discarded.	60 (seconds)
MinimumNumberStorageHandlers	The minimum number of storage handlers that the server will keep on hand for use by processes needing to access the persistent storage cache. This parameter defaults to 0 for the "File" storage strategy, and 3 for the "Database" storage strategy.	As noted

Option	Description	Default
MaximumNumberStorageHandlers	The maximum number of storage handlers that the server will keep on hand for use by processes needing to access the persistent storage cache. If set to 0, no maximum is enforced. This parameter defaults to 0 for the "File" storage strategy, and 10 for the "Database" storage strategy.	As noted

SERVER.COMMANDPORT_XXX

These section(s) contain information on the command port(s), which are used to accept incoming commands such as stopping, restarting, reporting status, and so on. Normally only one command port is needed, but more than one can be set up if this is desired. The xxx should be substituted with a unique meaningful name for the command port.

Option	Description	Default
Port	The port number to listen on for internal commands.	No default value (normally set to 5070).

SERVER.HTTPCONTROLPORT_XXX

These section(s) contain parameters for the HTTP web status interface. Normally only one HTTP control port is needed, but more than one can be set up if this is desired. The xxx should be substituted with a unique meaningful name for the HTTP control port.

Option	Description	Default
Port	The port number to listen on for Web HTTP commands.	No default value (normally set to 5071).

Option	Description	Default
AllowConfigChangesFromWebPage	A true/false flag that controls whether the configuration file can be modified via the administration web interface.	false
AllowFileEnquiryFromWebPage	A true/false flag that controls whether remote file management can be used to view files via the administration web interface.	false
AllowFileUpdatesFromWebPage	A true/false flag that controls whether remote file management can be used to update files via the administration web interface.	false
AllowLdbEnquiryFromWebPage	A true/false flag that controls whether local database tables can be viewed via the administration web interface.	true
AllowLdbUpdatesFromWebPage	A true/false flag that controls whether local database tables can be updated via the administration web interface.	false
AuthorisationUsername	The user ID to be used to secure the web status page. If not present or left blank, no logon security is enforced when accessing the web status page.	- (no logon security)
AuthorisationPassword	The password to be used to secure the web status page.	- (no password)

Option	Description	Default
HtmlButtonLocation	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next/previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward etc.	Bottom
HtmlPageSize	The maximum number of lines to display on the web pages.	15
ShowLdbRowCounts	A true/false flag that indicates whether, by default, row counts are listed against local database tables listed in the administration web interface.	false
SuppressLogo	A true/false flag that allows suppression of the Sky Technologies logo on the web status pages (to conserve bandwidth and improve performance).	false
UseHtmlCompression	A true/false flag that indicates whether HTML compression should be sent to browsers which support it.	true

SERVER.GUIAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow GUI clients to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the GUI application port.

Option	Description	Default
PORT	The port number to listen on for incoming BPC/WPC application requests.	No default value (normally set to 5074).
APPLICATION	The identifying number of the application to be provided to BPC/WPC clients connecting on this port.	000
VERSION	The identifying number of the application version to be provided to BPC/WPC clients connecting on this port.	000
DEVICE	The identifying number to be assigned to BPC/WPC clients connecting on this port.	000
HOSTINTERFACE	The host interface definition to use.	No default value.
CLIENTCANREQUESTAPPLICATION	A true/false flag that indicates whether a BPC/WPC client is permitted to override the default application and/or version number with the application and/or version numbers specified in the WPC configuration file.	true

Option	Description	Default
SESSIONTIMEOUT	The maximum period of time (in minutes) for which the server will retain the context of an idle (inactive) session before flushing it from memory. Because BPC/WPC clients are not continuously connected, no notification is sent to the user when this occurs. If a user is inactive for greater than the specified timeout period, the next time they attempt to connect to the server via BPC/WPC, a new session will be established and they will be sent to the application logon screen. If this value is set to zero or a negative number, then no session timeout is enforced. This is not recommended since there is then no way for the server to remove idle sessions, resulting in increasing memory consumption over time.	60 (minutes)
USECOMPRESSION	A true/false flag that indicates whether the XML data sent to BPC/WPC sessions on the port should be compressed. This greatly reduces bandwidth consumption.	false
COMPRESSIONALGORITHM	The compression algorithm to use when compressing data sent to BPC/WPC sessions on the port. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.

Option	Description	Default
USEENCRYPTION	A true/false flag that indicates whether the XML data sent to BPC/WPC sessions on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	false
ENCRYPTIONALGORITHM	The encryption algorithm to use when encrypting data sent to BPC/WPC sessions on the port. The following encryption algorithms are available: BLOWFISH - The "blowfish" encryption scheme.	No default value.
ENCRYPTIONKEY	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
ENCRYPTIONKEYFILE	The file in which the encryption key is stored.	No default value.
ENCRYPTIONKEYHEXENCODED	A true/false flag that indicates whether the encryption key is hex-encoded.	false

SERVER.HTTPAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow browsers to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the HTTP application port.

Option	Description	Default
PORT	The port number to listen on for incoming HTTP application requests.	No default value (normally set to 5073).
APPLICATION	The identifying number of the application to be provided to browsers connecting on this port.	000
VERSION	The identifying number of the application version to be provided to browsers connecting on this port.	000
DEVICE	The identifying number to be assigned to browsers connecting on this port.	000
HOSTINTERFACE	The host interface definition to use.	No default value.
CLIENTCANREQUESTAPPLICATION	A true/false flag that indicates whether a browser is permitted to override the default application and/or version number using query string parameters (the relevant query string parameters are "appNum" and "verNum").	true

Option	Description	Default
SESSIONTIMEOUT	The maximum period of time (in minutes) for which the server will retain the context of an idle (inactive) session before flushing it from memory. Because a browser is not continuously connected, no notification is sent to the user when this occurs. If a user is inactive for greater than the specified timeout period, the next time they attempt to connect to the server via their browser, a new session will be established and they will be sent to the application logon screen. If this value is set to zero or a negative number, then no session timeout is enforced. This is not recommended since there is then no way for the server to remove idle sessions, resulting in increasing memory consumption over time.	60 (minutes)

SERVER.TERMINALAPPLICATIONPORT_XXX

These section(s) contain parameters for port(s) to allow terminalbased devices to access an application defined through the workbench in SAP. The xxx should be substituted with a unique meaningful name for the terminal application port.

Option	Description	Default
Port	The port number to listen on for incoming terminal connections.	No default value (normally set to 5072).

Option	Description	Default
Application	The identifying number of the application to be provided to devices connecting on this port. If terminal enquiry is in use then this is the default application number that will be used if the terminal enquiry process cannot determine the application number.	000
Version	The identifying number of the application version to be provided to devices connecting on this port. If terminal enquiry is in use then this is the default version number that will be used if the terminal enquiry process cannot determine the version number.	000
Device	The identifying number to be assigned to devices connecting on this port. If terminal enquiry is in use then this is the default device number that will be used if the terminal enquiry process cannot determine the device number.	000
ErrorBeeps	The number of "beeps" to be sent to a terminal device when an application error occurs. This allows errors to be made audibly distinguishable from beeps the device may issue for other reasons (for example, when scanning a bar code).	1
ExitDelay	The number of seconds for which the server should pause before closing down a connection. This gives the user a chance to read any error messages displayed in the event that the terminal device clears the screen (or scrolls off any information displayed) while resetting.	2

Option	Description	Default
HostInterface	The host interface definition to use.	No default value.
TerminalEnquiry	A true/false flag that indicates whether or not terminal enquiry is to be used for this port. Terminal enquiry allows the server to query any device connecting to it to establish the application number, version number, and/or device number to be used. The answerback string for each device making use of this feature should be set as follows: VTI[A=xxx,V=yyy,D=zzz] where xxx, yyy and zzz are (respectively) the application number, version number, and device number to be used. If the "ClientCanRequestApplication" parameter is set on, then any settings acquired through terminal enquiry override the default values set for the port. Where terminal enquiry is not used (i.e. this flag is set to "false") then the default values for the port are always used.	true
TerminalEnquiryTimeout	The maximum time period (in seconds) for which the server should wait for a response from a connecting terminal device during terminal enquiry. If the timeout expires, the server assumes that no further identifying information will be forthcoming from the device and then continues processing. This typically occurs when the format of the answerback string is incorrect. In the normal course of events the terminating right square bracket character "]" tells the server that the answerback string is complete and that it can proceed.	10 (seconds)

Option	Description	Default
TerminalType	The terminal type definition to use. This option enables custom and other terminal types to be implemented and supported. The server supports the following inbuilt terminal types: VT220 - VT220 emulation using 7-bit ASCII (same as VT220-7Bit) VT220-7Bit - VT220 emulation using 7-bit ASCII VT220-8Bit - VT220 emulation using 8-bit ASCII Additional variant terminal types can be set up in the configuration file through the TerminalType section (described hereafter).	VT220
ClientCanRequestApplication	A true/false flag that indicates whether a client connection can override the default application and/or version number using the terminal enquiry sequence.	true
ClearFieldValueOnReEntry	A true/false flag that tells the server whether or not to clear any existing value in a screen field when a new value is entered.	false
ColorEnabled	Whether the connected devices support VT220 colours.	true
SessionTimeout	The maximum period of time (in minutes) for which the server will permit an idle (inactive) connection to remain connected before disconnecting it. If the value is set to zero or a negative number, then no session timeout is enforced.	60 (minutes)

Option	Description	Default
TraceInput	A true/false flag that controls whether the server should write detailed information about incoming characters to the log. This is typically used for debugging device keystroke behavior (for example, function key mappings), and should generally be left disabled.	false

SERVER.XMLGATEWAYPORT

These section(s) contain parameters for port(s) on which the Application Server is to make an Access Gateway available. This allows SkyMobile to connect to a back-end host by using the Access Gateway port as an intermediary. For example, other instances of Application Server could connect to a back-end SAP instance via an XML port running on a central server.

Option	Description	Default
Port	The port number to listen on for incoming messages.	No default value (normally set to 5075).
HostInterface	The host interface definition into which the port will feed.	No default value.
DefaultIdentityService	The name of a implemented identity management service to use as the default if the calling Secure Container sends an identity management request and does not specify one.	None

Option	Description	Default
MaximumConnectedServers	The maximum number of separate servers permitted to connect through this gateway port. This is normally used to control load balancing. A value of 0 indicates that there is no maximum applicable.	0 (no limit)
MaximumMessageSize	The maximum permissible message size allowable on the port (in bytes). Messages greater than this size will cause the incoming connection that has issued them to be immediately closed. This parameter is primarily for use when the port is publicly exposed, and helps to protect the server from uninvited incoming traffic that isn't in the correct format for processing. A value of 0 indicates that there is no maximum message size.	0 (no limit)
UseCompression	A true/false flag that indicates whether XML data sent to connected clients should be compressed. This greatly reduces bandwidth consumption.	false
CompressionAlgorithm	The compression algorithm to use when compressing data sent to connected clients on the port. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value
Use Encryption	A true/false flag that indicates whether the XML data sent to connected clients on the port should be encrypted. Obviously, this is a more secure option, particularly over a public network.	False

Option	Description	Default
AllowUnencryptedProvisioning	A true/false flag that specifies whether an unencrypted provisioning request may still be accepted by this port even if encryption is enabled.	False
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to connected clients on the port. The following encryption algorithms are available: BLOWFISH - The "blowfish" encryption scheme (recommended). RSA - The "RSA" encryption algorithm.	No default value.
EncryptionKey	The encryption key to use when encrypting data. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
EncryptionKeyFile	The file in which the encryption key is stored.	No default value.
EncryptionKeyHexEncoded	A true/false flag that indicates whether the encryption key is hex-encoded.	False
HandshakeEncryptionAlgorithm	The encryption algorithm to use during the initial handshake with the remote partner. If left blank, no initial handshaking is performed. The following encryption algorithms are available: BLOWFISH - The "Blowfish" encryption algorithm. RSA - The "RSA" encryption algorithm (recommended).	No default value.

Option	Description	Default
HandshakeEncryptionKey	The encryption key to use during the initial handshake with the remote partner. If no encryption key is specified, the server will look for one in an encryption key file instead.	No default value.
HandshakeEncryptionKeyFile	The file in which the handshaking encryption key is stored.	No default value.
HandshakeEncryptionKeyHex-Encoded	A true/false flag that indicates whether the handshaking encryption key is hexencoded.	false
PingCheckIdleSeconds	The period of time (in seconds) for which a connected client can remain idle before a "ping check" is performed to establish whether the connection is still "live".	600 (10 minutes)
PingCheckTimeout	The period of time (in milliseconds) for which the server will wait for a response after performing a "ping check'. Once this time interval has passed, the connection is no longer considered "live" and is closed.	60000 (1 minute)
UseHttp	A true/false flag that indicates whether or not incoming communications will use the HTTP and/or HTTPS protocols. If enabled, the gateway port expects incoming SkyMobile communications to be bundled as HTTP/HTTPS traffic (potentially coming in via an HTTP/HTTPS proxy). Otherwise, normal SkyMobile communications are expected.	false

Option	Description	Default
XmlDebug	A true/false flag that controls whether the server writes information about incoming and outgoing XML received on the port, into its log. This option has a substantial performance impact and generates very large log files, so it should generally be left disabled.	false

SERVER.RELAYPORT_XXX

These section(s) contain parameters for port(s) on which the Application Server can accept incoming relay connections from other servers.

Option	Description	Default
PORT	The port number to listen on for incoming messages.	No default value (normally set to 5076).
IDLETIMEOUT	The maximum time interval (in seconds) that the server will allow a relay connection to remain unresponsive before concluding that is no longer working correctly and closes it.	3600 (5 minutes)
PINGTIMEOUT	The maximum time interval (in seconds) that the server allows a relay connection to remain idle before "pinging" it to make sure that it is still working correctly.	60 (1 minute)

SERVER.CUSTOMPORT_XXX

These section(s) contain parameters for port(s) with specially customized or unique behavior. Custom ports accept incoming messages through connections made on the port to which they are listening. Each time such a message arrives, the server will invoke a userdeveloped exit that is responsible for processing the message. This allows for extreme flexibility in the way in which the server can be made to interact with devices and applications conversing over a TCP/IP connection.

Option	Description	Default
Port	The port number to listen on for incoming messages.	No default value (normally set to 5077).
InputMessageType	The expected format for incoming messages. This tells the server when an incoming message is complete and thus ready for processing. This parameter must be one of the following: D - Delimited input messages. Incoming messages will be terminated with a specified delimiter. This is the default input mode. F - Fixed length messages. Incoming messages are all the same length. O - The message length is embedded in incoming messages at a specified offset position from the start of the message. In other words, the message contains a numeric field at a known position which contains the length of the message itself. U - Unformatted input. Any input received is processed immediately upon being received. This generally results in a message being sent for each incoming character, but is dependent on network behavior.	D (delimited input)
InputDelimiters	Only used for input message type "D". Specifies the character sequence(s) that are to act as delimiters for incoming messages. More than one value can be specified by separating the various delimiters with commas. Nonprintable characters can be specified by entering a backslash followed by the 3-digit ASCII value of the character. Thus, for example, a new line (ASCII value 10) would be entered as \010.	\010 (new line)

Option	Description	Default
InputIgnored	Only used for input message type "D". Specifies any character sequence(s) that are to be ignored and removed from incoming messages. More than one value can be specified by separating the strings to be ignored with commas. Non-printable characters can be specified by entering a backslash followed by the decimal ASCII value of the character.	\013 (carriage return)
InputStripDelim	Only used for input message type "D". Specifies whether or not delimiters should be stripped from the message prior to it being processed.	true
InputFixedLength	Only used for input message type "F". Specifies the length of incoming fixed length messages.	No default value.
InputOffsetStart	Only used for input message type "O". Specifies the offset from the start of the message at which to find the message length within the message. For example, if the message length were to be contained in characters 4-8 of the message, the offset would be 3, since character position 4 is 3 characters in from character position 1 (the start of the message).	No default value.
InputOffsetLength	Only used for input message type "O". Specifies the length of the message length field within the message. For example, if the message length were to be contained in characters 4-8 of the message, the length would be 5 (since it takes up character positions 4, 5, 6, 7, and 8, for a total of 5 characters).	No default value.

Option	Description	Default
CustomPortExit	The fully qualified class name for the Java custom port exit class to be invoked when a message is received. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.

SERVER.CUSTOMSERIALDEVICE_XXX

These section(s) contain parameters for serially connected devices that need to be integrated with the server. Custom serial devices accept incoming messages on a specified serial port. Each time such a message arrives, the server will invoke a user-developed exit that is responsible for processing the message.

Option	Description	Default
CommPort	The name of the communications port on which to listen for incoming messages (for example, COM1).	No default value.
Speed	The baud rate at which communication with the serial device should take place. If left blank or not specified, the current baud rate of the device is used.	As noted.
DataBits	The number of data bits used for communication with the serial device. Valid entries must be in the range 5-8. If left blank or not specified, the current number of data bits for the device is used.	As noted.
StopBits	The number of stop bits used for communication with the serial device. Valid entries are 1, 1.5, or 2. If left blank or not specified, the current number of stop bits for the device is used.	As noted.

Option	Description	Default
Parity	The parity settings to use during communication with the serial device. Valid entries are None, Odd, Even, Mark, or Space. If left blank or not specified, the current parity setting for the device is used.	As noted.
FlowControl	The flow control (handshaking) settings to use during communication with the serial device. Valid entries are as follows: None - No flow control. XonXoff - XON/XOFF handshaking is used. If left blank or not specified, the current flow control setting for the device is used.	As noted.
InitialConnect	A true/false flag that controls whether or not the server attempts to automatically establish and maintain a connection to the serial device. This may be undesirable in some situations, for example if communications with the device in question are initiated upon demand and controlled by a user exit.	true
DisconnectDelay	The time interval (in milliseconds) that the server will pause after disconnecting from the device. Some serially connected devices experience problems if sufficient time is not allowed to permit the device to completely shut down before it is used again. This pause is intended to ensure that such problems do not arise, but it can be reduced or removed in the interests of efficiency where it is not needed (or conversely, increased in cases where the device in question requires a long time to shut down properly).	500 (milliseconds)

Option	Description	Default
ReconnectInterval	The time interval (in seconds) to pause after an unsuccessful attempt to connect to the serial device, before trying again. This parameter is only used when the server is responsible for maintaining the connection to the device (i.e. InitialConnect is set to "true"). A negative or zero value indicates that no attempt to reconnect should take place.	0 (do not reconnect)
UseNativeComm	A true/false flag that controls whether the native serial communications library is used instead of the Java Comm Extension.	false
InputMessageType	The expected format for incoming messages. This tells the server when an incoming message is complete and thus ready for processing. This parameter must be one of the following: D - Delimited input messages. Incoming messages will be terminated with a specified delimiter. This is the default input mode. F - Fixed length messages. Incoming messages are all the same length. O - The message length is embedded in incoming messages at a specified offset position from the start of the message. In other words, the message contains a numeric field at a known position which contains the length of the message itself. U - Unformatted input. Any input received is processed immediately upon being received. This generally results in a message being sent for each incoming character, but is dependent on network behavior.	D (delimited input)

Option	Description	Default
InputDelimiters	Only used for input message type "D". Specifies the character sequence(s) that are to act as delimiters for incoming messages. More than one value can be specified by separating the various delimiters with commas. Nonprintable characters can be specified by entering a backslash followed by the 3-digit ASCII value of the character. Thus, for example, a new line (ASCII value 10) would be entered as \010.	\010 (new line)
InputIgnored	Only used for input message type "D". Specifies any character sequence(s) that are to be ignored and removed from incoming messages. More than one value can be specified by separating the strings to be ignored with commas. Non-printable characters can be specified by entering a backslash followed by the decimal ASCII value of the character.	\013 (carriage return)
InputStripDelim	Only used for input message type "D". Specifies whether or not delimiters should be stripped from the message prior to it being processed.	true
InputFixedLength	Only used for input message type "F". Specifies the length of incoming fixed length messages.	No default value.
InputOffsetStart	Only used for input message type "O". Specifies the offset from the start of the message at which to find the message length within the message. For example, if the message length were to be contained in characters 4-8 of the message, the offset would be 3, since character position 4 is 3 characters in from character position 1 (the start of the message).	No default value.

Option	Description	Default
InputOffsetLength	Only used for input message type "O". Specifies the length of the message length field within the message. For example, if the message length were to be contained in characters 4-8 of the message, the length would be 5 (since it takes up character positions 4, 5, 6, 7, and 8, for a total of 5 characters).	No default value.
CustomSerialDeviceExit	The fully qualified class name for the Java custom serial device exit class to be invoked when a message is received. The class in question must be available to the JVM in which the server is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the "classes" directory).	No default value.

SERVER.TERMINALTYPE_XXX

This Application Server configuration section contains definitions for additional terminal emulation modes (other than the inbuilt ones: VT220 and VT220-8bit) to be supported by SkyMobile. The xxx should be substituted with a unique meaningful name for the emulation mode. This name can then be assigned to the TerminalType parameter in a TerminalApplicationPort section.

Option	Description	Default
DisplaySequence:BEEP	The character sequence to display to cause the terminal to beep. As with all other sequences in this section, non-printable characters may be represented by a backslash ("\") followed by a three-digit decimal ASCII character value. For example, the terminal beep character is normally ASCII character 007 in decimal. This can be represented in the configuration file as \007.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:BOLD	The character sequence to display to cause the terminal device to turn bold mode on.	As per base terminal type if specified, otherwise no default.
DisplaySequence:BLINK	The character sequence to display to cause the terminal device to turn blinking mode on.	As per base terminal type if specified, otherwise no default.
DisplaySequence:CLEAR	The character sequence to display to cause the terminal device to clear the screen.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:SETCURSOR	The character sequence to display to cause the terminal device to set the cursor position to specified X and Y co-ordinates. This setting interacts with the SetCursorXOffset, SetCursorYOffset, SetCursorXToken, and SetCursorYToken parameters (see below).	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOBOLD	The character sequence to display to cause the terminal device to turn bold mode off.	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOBLINK	The character sequence to display to cause the terminal device to turn blinking mode off.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:NOREVERSE	The character sequence to display to cause the terminal device to turn reverse (inverse video) mode off.	As per base terminal type if specified, otherwise no default.
DisplaySequence:NOUNDERLINE	The character sequence to display to cause the terminal device to turn underline mode off.	As per base terminal type if specified, otherwise no default.
DisplaySequence:REVERSE	The character sequence to display to cause the terminal device to turn reverse (inverse video) mode on.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
DisplaySequence:TERMENQ	The character sequence to display to initiate terminal enquiry (i.e. to trigger the terminal device to response with its answerback string).	As per base terminal type if specified, otherwise no default.
DisplaySequence:UNDERLINE	The character sequence to display to cause the terminal device to turn underline mode on.	As per base terminal type if specified, otherwise no default.
FunctionKeySequence:xxx	The character sequence that will be generated by the terminal device to signify that function key xxx has been activated.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:BACKSPACE	The character sequence that will be generated by the terminal device to signify that a destructive backspace has been requested.	As per base terminal type if specified, otherwise no default.
KeySequence:BACKTAB	The character sequence that will be generated by the terminal device to signify that a backtab (tab backwards) has been requested.	As per base terminal type if specified, otherwise no default.
KeySequence:INSERT	The character sequence that will be generated by the terminal device to signify a request to toggle between insert and replace modes.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:DELETE	The character sequence that will be generated by the terminal device to signify a request to delete the character over which the cursor is currently positioned.	As per base terminal type if specified, otherwise no default.
KeySequence:UP	The character sequence that will be generated by the terminal device to signify that the up arrow key has been pressed.	As per base terminal type if specified, otherwise no default.
KeySequence:DOWN	The character sequence that will be generated by the terminal device to signify that the down arrow key has been pressed.	As per base terminal type if specified, otherwise no default.

Option	Description	Default
KeySequence:RIGHT	The character sequence that will be generated by the terminal device to signify that the right arrow key has been pressed.	As per base terminal type if specified, otherwise no default.
KeySequence:LEFT	The character sequence that will be generated by the terminal device to signify that the left arrow key has been pressed.	As per base terminal type if specified, otherwise no default.
KeySequence:TAB	The character sequence that will be generated by the terminal device to signify that a tab (forwards) has been requested.	As per base terminal type if specified, otherwise no default.
SetCursorXOffset	The X offset to use when directing the cursor to a particular position on the screen. By default the first column on the screen is considered to be column 0. If the cursor addressing system for the emulation mode treats the first column to be column 1, a value of 1 will be required for the X offset.	As per base terminal type if specified, otherwise 0.

Option	Description	Default
SetCursorYOffset	The Y offset to use when directing the cursor to a particular position on the screen. By default the first line on the screen is considered to be line 0. If the cursor addressing system for the emulation mode treats the first line to be line 1, a value of 1 will be required for the Y offset.	As per base terminal type if specified, otherwise 0.
SetCursorXToken	The token used to represent the position of the X co-ordinate value inside the DisplaySequence:SETCURSOR string.	<X>
SetCursorYToken	The token used to represent the position of the Y co-ordinate value inside the DisplaySequence:SETCURSOR string.	<Y>

Go back to the top

3.9.12.10 Example Configurations

A number of different example configurations are provided here to show typical implementations. Select from the list on the right.

Example Configurations

- Gateway SAP RFC Connection
- Container Connection to Gateway
- Gateway Port Configuration
- Windows Mobile Secure Container Configuration

Gateway SAP RFC Connection

The following examples show the configuration to connect to a SAP host through a direct RFC connection and also through a SAP Logon Load Balancing. Refer to config for more information.

Specific Host RFC Connection

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAPRFCCLIENT.JNILOGGING = false
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MINIMUMNUMBERHANDLERS = 5
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPHOST = nnn.nnnn.nnn.nnn (or name)
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBGROUP =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBHOST =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPPASSWORD = password
```

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPSYSTEM = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSELB = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSER = CPIC_USER
```

SAP Logon Load Balancing RFC Connection

```
SERVER.HOSTINTERFACE:SAPRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAPRFCCLIENT.JNILOGGING = false
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAPRFCCLIENT.MINIMUMNUMBERHANDLERS = 5
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPHOST =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBGROUP = SAP Logon Group Name
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBHOST = nnn.nnnn.nnn.nnn (or
name)
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPLBSYSTEM = SAP SID
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPPASSWORD = password
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPSYSTEM =
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSELB = 1
SERVER.HOSTINTERFACE:SAPRFCCLIENT.SAPUSER = CPIC_USER
```

Go back to the top

Container Connection to Gateway

This example configuration is for a Secure Container connecting to a Access Gateway port without using encryption. Refer to config for more information on the options. If multiple Access Gateways are being used in a fail over configuration, then all the gateway TCP/IP addresses need to be listed against the XMLGATEWAYHOSTS option. See here for more details.

Example Gateway Host Interface Configuration

```
SERVER.HOSTINTERFACE:XMLCLIENT.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONALGORITHM =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEY =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYFILE =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYHEXENCODED =
SERVER.HOSTINTERFACE:XMLCLIENT.ENCRYPTIONKEYSTRENGTH =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONALGORITHM =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEY =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEYFILE =
SERVER.HOSTINTERFACE:XMLCLIENT.HANDSHAKEENCRYPTIONKEYHEXENCODED =
SERVER.HOSTINTERFACE:XMLCLIENT.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:XMLCLIENT.MAXIMUMNUMBERHANDLERS = 20
SERVER.HOSTINTERFACE:XMLCLIENT.MAXIMUMOPTIMIZATIONLEVEL =
SERVER.HOSTINTERFACE:XMLCLIENT.MINIMUMNUMBERHANDLERS = 0
SERVER.HOSTINTERFACE:XMLCLIENT.USECOMPRESSION = true
SERVER.HOSTINTERFACE:XMLCLIENT.USECONDENSEDXML =
SERVER.HOSTINTERFACE:XMLCLIENT.USEENCRYPTION = false
SERVER.HOSTINTERFACE:XMLCLIENT.USETRANSFERFIELDS =
SERVER.HOSTINTERFACE:XMLCLIENT.XMLDEBUG = false
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYHOSTS = nnn.nnn.nnn.nnn (or
name)
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYHOSTSELECTION = Sequential
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYPORT = nnnnnn
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYRESPONSETIMEOUT = 30
SERVER.HOSTINTERFACE:XMLCLIENT.XMLGATEWAYSTALLTIMEOUT = 30
```

Go back to the top

Gateway Port Configuration

This example configuration is for an Access Gateway port. Refer to the config section for more information.

Example Access Gateway Port Configuration

```
SERVER.XMLGATEWAYPORT:PORTnnnnnn.COMPRESSIONALGORITHM = GZIP
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEYFILE =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.ENCRYPTIONKEYHEXENCODED =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.HOSTINTERFACE = xxxxxxxxxx
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PINGCHECKIDLESECONDS =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PINGCHECKTIMEOUT =
SERVER.XMLGATEWAYPORT:PORTnnnnnn.PORT = nnnnnn
SERVER.XMLGATEWAYPORT:PORTnnnnnn.USECOMPRESSION = true
SERVER.XMLGATEWAYPORT:PORTnnnnnn.USEENCRYPTION = false
SERVER.XMLGATEWAYPORT:PORTnnnnnn.XMLDEBUG = false
```

Go back to the top

Windows Mobile Secure Container Configuration

This example configuration is for a secure container running on Windows Mobile. Refer to the config section for more information.

Server configuration

```
CLIENT.GENERAL.ALLOWMULTIPLEINSTANCES = false
CLIENT.GENERAL.DISABLELOGOFFMESSAGE = false
CLIENT.GENERAL.DISABLESOUND = false
CLIENT.GENERAL.HOSTSTATUSMESSAGES = All
CLIENT.GENERAL.KEYBOARDONLY = false
CLIENT.GENERAL.UNICODEENABLED = true
CLIENT.LOG.LOGACTIVE = true
CLIENT.LOG.LOGFILEDIRECTORY = ./log
CLIENT.LOG.MAXFILES = 7
CLIENT.LOG.PERFORMANCETRACKING = false
CLIENT.LOG.SHOWMILLISECONDS = false
CLIENT.LOG.STRATEGY = Daily
CLIENT.PRESENTATION.AUTOSUGGEST = false
CLIENT.PRESENTATION.CHARPADDINGHEIGHT = 2
CLIENT.PRESENTATION.CHARPADDINGWIDTH = 1
```

```
CLIENT.PRESENTATION.DISABLEABOUTMENU = false
CLIENT.PRESENTATION.DISABLEMINIMIZEMENU = true
CLIENT.PRESENTATION.DISABLESELFTESTMENU = false
CLIENT.PRESENTATION.DISABLESHUTDOWNMENU = false
CLIENT.PRESENTATION.DISPLAYIMAGESIZES = false
CLIENT.PRESENTATION.DISPLAYLAYOUTGRID = false
CLIENT.PRESENTATION.FIXEDSIZELINEHEIGHT = 25
CLIENT.PRESENTATION.FONTNAME = Arial
CLIENT.PRESENTATION.FONTSIZE = 9
CLIENT.PRESENTATION.HIGHRESOLUTIONGRAPHICS = true
CLIENT.PRESENTATION.LAYOUTHORIZONTAL =
CLIENT.PRESENTATION.LAYOUTVERTICAL =
CLIENT.PRESENTATION.LEFTMARGIN = 10
CLIENT.PRESENTATION.LOGIMAGESIZES = false
CLIENT.PRESENTATION.MAXIMISEONSTARTUP = false
CLIENT.PRESENTATION.MESSAGEBOXNOFOCUS = false
CLIENT.PRESENTATION.SELECTTEXTONFOCUSCHANGE = true
CLIENT.PRESENTATION.SHOWFUNCTIONKEYSINMENU = false
CLIENT.PRESENTATION.SHOWHORIZONTALSCROLLBAR = true
CLIENT.PRESENTATION.SHOWMENU = true
CLIENT.PRESENTATION.SHOWVERTICALSCROLLBAR = false
CLIENT.PRESENTATION.SHOWWINDOWCONTROLS = true
CLIENT.PRESENTATION.STATUSBARERRORCOLOR = red
CLIENT.PRESENTATION.STATUSBARICONSET = Basic
CLIENT.PRESENTATION.STATUSBARINFOCOLOR = green
CLIENT.PRESENTATION.TOPMARGIN = 10
CLIENT.PRESENTATION.WINDOWHEIGHT = 500
CLIENT.PRESENTATION.WINDOWHORIZONTALALIGNMENT = Centre
CLIENT.PRESENTATION.WINDOWVERTICALALIGNMENT = Centre
CLIENT.PRESENTATION.WINDOWWIDTH = 500
CLIENT.SERVER.COMPRESSIONALGORITHM = GZIP
CLIENT.SERVER.LAUNCHSERVER = true
CLIENT.SERVER.SERVERCOMMANDPORT = 5070
CLIENT.SERVER.SERVERHOSTNAME = localhost
CLIENT.SERVER.SERVERPORT = 5074
```

```
CLIENT.SERVER.SERVERRESPONSETIMEOUT = 30
CLIENT.SERVER.USEENCRYPTION = false
CLIENT.SERVER.XMLDEBUGTOSCREEN = false
GLOBAL.CONFIGURATION.EDIT = TRUE
GLOBAL.CONFIGURATION.ENCRYPT = FALSE
GLOBAL.STARTUP.PROVISIONCHECK = FALSE
SERVER.APPLICATIONCACHE.APPLICATIONREFRESHINTERVAL = 3600
SERVER.APPLICATIONCACHE.FUNCTIONCACHESIZE = 10
SERVER.APPLICATIONCACHE.REFRESHBYFUNCTION = true
SERVER.BACKUPRESTORE.COMPRESSIONALGORITHM = GZIP
SERVER.BACKUPRESTORE.USECOMPRESSION = true
SERVER.BINARYFILES.APPLICATIONBINARYREFRESHINTERVAL =
SERVER.BINARYFILES.DOWNLOADDIRECTORY = /Program Files/Sky
Technologies/VTI/downloads
SERVER.BINARYFILES.DYNAMICBINARYREFRESHINTERVAL = 180
SERVER.BINARYFILES.FILELOCATION:GRAPHICS = /Program Files/Sky
Technologies/VTI/graphics
SERVER.BINARYFILES.FILELOCATION:SOUNDS = /Program Files/Sky
Technologies/VTI/sounds
SERVER.BINARYFILES.PACKAGEREFRESHINTERVAL = 0
SERVER.BINARYFILES.UPLOADDIRECTORY = /Program Files/Sky
Technologies/VTI/uploads
SERVER.COMMANDPORT:PORT5070.PORT = 5070
SERVER.GENERAL.AUTORESTART = false
SERVER.GENERAL.DATEFORMAT = DD/MM/YYYY
SERVER.GENERAL.SERVERGROUP = DEMO
SERVER.GENERAL.SERVERID = WPC0005327
SERVER.GENERAL.SHUTDOWNTIMEOUT = 30
SERVER.GENERAL.TENANT = 00000
SERVER.GENERAL.TIMEFORMAT = HH:MM:SS
SERVER.GENERAL.TIMEZONE =
SERVER.GENERAL.UNICODEENABLED = true
SERVER.GUIAPPLICATIONPORT:PORT5074.APPLICATION = 900
SERVER.GUIAPPLICATIONPORT:PORT5074.CLIENTCANREQUESTAPPLICATION = true
SERVER.GUIAPPLICATIONPORT:PORT5074.COMPRESSIONALGORITHM = GZIP
```

```
SERVER.GUIAPPLICATIONPORT:PORT5074.DEVICE = 000
SERVER.GUIAPPLICATIONPORT:PORT5074.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.GUIAPPLICATIONPORT:PORT5074.ENCRYPTIONKEYHEXENCODED = false
SERVER.GUIAPPLICATIONPORT:PORT5074.HOSTINTERFACE = SKYCLOUD
SERVER.GUIAPPLICATIONPORT:PORT5074.PORT = 5074
SERVER.GUIAPPLICATIONPORT:PORT5074.SESSIONTIMEOUT = 60
SERVER.GUIAPPLICATIONPORT:PORT5074.USECOMPRESSION = true
SERVER.GUIAPPLICATIONPORT:PORT5074.USEENCRYPTION = false
SERVER.GUIAPPLICATIONPORT:PORT5074.VERSION = 001
SERVER.GUIAPPLICATIONPORT:PORT5074.XMLDEBUG = false
SERVER.HEARTBEAT.HEARTBEATCONNECTEDINTERVAL = 15
SERVER.HEARTBEAT.HEARTBEATINTERVAL = 15
SERVER.HEARTBEAT.HEARTBEATLOSSSEVERITY = I
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:APPLICATION =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:BINARYFILES =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:HEARTBEAT = true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:LOCALDATABASE =
true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:PACKAGES = true
SERVER.HOSTFUNCTIONALITYSUPPORT.FUNCTIONALITYSUPPORT:SESSIONMANAGER =
true
SERVER.HOSTINTERFACE:SKYCLOUD.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:SKYCLOUD.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:SKYCLOUD.MAXIMUMNUMBERHANDLERS = 50
SERVER.HOSTINTERFACE:SKYCLOUD.MINIMUMNUMBERHANDLERS = 0
SERVER.HOSTINTERFACE:SKYCLOUD.USECOMPRESSION = true
SERVER.HOSTINTERFACE:SKYCLOUD.USEENCRYPTION = false
SERVER.HOSTINTERFACE:SKYCLOUD.XMLDEBUG = false
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYHOSTS =
skycloud.skytechnologies.com
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYHOSTSELECTION = Sequential
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYPORT = 30000
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYRESPONSETIMEOUT = 120
```

```
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYSTALLTIMEOUT = 120
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWCONFIGCHANGESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWFILEENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWFILEUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWLDBUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.HTMLBUTTONLOCATION = both
SERVER.HTTPCONTROLPORT:PORT5071.PORT = 5071
SERVER.HTTPCONTROLPORT:PORT5071.SECURE = false
SERVER.HTTPCONTROLPORT:PORT5071.SHOWLDBROWCOUNTS = true
SERVER.LOCALDATABASE.ERRORRETRYINTERVAL = 300
SERVER.LOCALDATABASE.MAINTAINHOSTCONNECTIVITY = true
SERVER.LOCALDATABASE.MAXIMUMNUMBERUPDATETHREADS = 3
SERVER.LOCALDATABASE.MINIMUMNUMBERUPDATETHREADS = 1
SERVER.LOCALDATABASE.TRANSFERBUFFERSIZE = 250000
SERVER.LOG.LOGFILEDIRECTORY = /Program Files/Sky Technologies/VTI/log
SERVER.LOG.LOGSTRATEGY = D
SERVER.LOG.MAXIMUMNUMBERLOGFILES = 14
SERVER.LOG.PERFORMANCETRACKING = false
SERVER.LOG.SHOWMILLISECONDS = false
SERVER.LOG.STOREDPROCDUMP = false
SERVER.LOG.STOREDPROCLOGGING = false
SERVER.MONITOR.CLEARRFCTRACEFILES = true
SERVER.MONITOR.GARBAGECOLLECT = true
SERVER.MONITOR.LOGMEMORYUSAGE = true
SERVER.MONITOR.LOGTHREADCOUNT = true
SERVER.MONITOR.LOGTHREADNAMES = false
SERVER.MONITOR.MONITORINTERVAL = 60
SERVER.MONITOR.RFCTRACEEXPIRYDAYS = 7
SERVER.MONITOR.USEMONITOR = true
SERVER.RELAYCONNECTIONS.ALLOWCONFIGCHANGESFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWFILEENQUIRYFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWFILEUPDATESFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWLDBUPDATESFROMWEBPAGE = false
```

```
SERVER.RELAYCONNECTIONS.HTMLBUTTONLOCATION = Both
SERVER.RELAYCONNECTIONS.HTMLPAGESIZE = 20
SERVER.RELAYCONNECTIONS.RELAYHOST = skycloud.skytechnologies.com
SERVER.RELAYCONNECTIONS.RELAYPORT = 30001
SERVER.RELAYCONNECTIONS.SHOWLDBROWCOUNTS = true
SERVER.SSL.SSLDEBUG = false
SERVER.STORAGE.STORAGECOMPRESSONSHUTDOWN = true
SERVER.STORAGE.STORAGECONSISTENCYCHECK = false
SERVER.STORAGE.STORAGEDATABASEFILENAME = /Program Files/Sky
Technologies/VTI/db/vti.db
SERVER.STORAGE.STORAGEINDEXDEFRAGONSHUTDOWN = true
SERVER.STORAGE.STORAGEINDEXREBUILD = false
SERVER.STORAGE.STORAGEINTERNALIZESTRINGS = true
SERVER.STORAGE.STORAGEINTERNALLOGGING = false
SERVER.STORAGE.STORAGEREADLOCKS = false
SERVER.STORAGE.STORAGESQLLOGGING = false
SERVER.STORAGE.STORAGETYPE = SkyDB
```

Go back to the top

3.9.12.11 Connecting to SAP

The Windows Secure Container may connect to one or more 'back end' systems (for example, SAP RFC), or connect to one or more gateway ports. In either case, one or more HostInterface sections need to be configured in the configuration. Refer to Configuration options for a detailed description of these sections and their options. Once the host connection sections have been declared, the application port configurations refer to the SAP connection to use.

By default, SkyMobile connects to SAP using a special CPIC (Interface) user specified in its configuration file "skymobile.cfg". This user cannot log onto SAP directly and is only used to process business transactions. If you wish mobile users to log on directly to SAP using their allocated SAP user ID and password, then you need to implement a logon screen which captures the user and password. The logon screen then executes a ABAP verify screen exit, which performs a special SAP function call and if all is ok, performs a logon context switch; instructs SkyMobile to re-logon to SAP using the supplied user and password. A sample ABAP exit (/SKY/YVTIVLOG) is provided, which contains all the code necessary. This option is only valid for real-time (online/synchronous) connections with SAP. Offline connections (asynchronous) always use the default SAP user, as specified in the Application Server configuration file.

> ***Important:*** It is not advisable to use direct user connections with SAP as this can consume system resources and impact performance. It is far more efficient for all container connections to utilize a shared RFC connection pool and pass through the user credentials as part of the SkyMobile transmission protocol. You can use this user ID as meta-data in SAP transaction processing.

3.9.12.12 Tracking Application Servers Users and Devices

With mobile applications, it is important to be able to identify which devices are connected and their current status. SkyMobile provides facilities to identify specific Application Servers and client connections.

Server ID and Group

A unique Application Server ID should be specified in the skymobile.cfg configuration file. This may be up to 15 characters. If no server ID is specified, it defaults to the current computer host name or is automatically provisioned. The Server ID is especially useful for managing selective downloads and refreshes of data and application definitions; if the Server ID is xyz, then download a specific customer, product profile. It is also necessary to restrict application and local database definitions.

In addition an Application Server group can be specified in the configuration file. This also may be up to 15 characters. The server group is useful to identify servers for a specific application, and geographic group. It may be used in the workbench to define configuration at a group level and effectively filter selection lists.

skymobile.cfg

```
SERVER.GENERAL.SERVERGROUP = Mobile_Sales
SERVER.GENERAL.SERVERID = PRODUCTION1
```

The Device number

A three digit device ID may be allocated to each client to help identify specific connections. The device ID may be configured for Windows presentation (WPC) and VT220 text terminal modes. For WPC, this is specified in the "skymobile.cfg" configuration file. For VT220 text terminals the device id may be passed via an answer-back string defined in the telnet client, for example, VTI[d=nnn) where the 'd' indicates device and 'nnn' the unique three digit number. The device number may also be specified through a screen procedure or programmatically through the ABAP or Java APIs. Once specified, the device number is linked to all requests.

Sessions

Every connection to SkyMobile is allocated a unique session ID. This is used internally to identify transactions and link them to specific users and devices. With a SAP back end there are two session IDs, a local session id and a SAP session ID. The SAP session manager displays all connected sessions and their attributes. When an application is logged off, the associated session is de-activated.

User Identification

User identification is achieved through populating the user ID variable, which is stored and passed with all requests. Along with the user ID, a password, and work area/group may also be populated and used for security checks, transaction identification, data selection etc. These variables may be populated from procedures and/or via the ABAP/Java programming APIs.

3.9.13 General Troubleshooting

This section lists some of the more common problems with installations and configuration. Contact Kony support (support@kony.com) if you cannot resolve the issue.

3.9.13.1 Windows Service Does not Always Start At Reboot

This situation is observed in cases where a database such as Microsoft SQL Server is in use. If the operating system attempts to launch the service before the database is ready, the Application Server is unable to connect to the database, and this causes the service to shut down. The solution to this problem is to create a dependency between the SkyMobile service and the database service, so that the SkyMobile service always launches after the database service is started. This requires use of a registry editor (such as regedt32). You must add the following value to the registry:

Registry Entry

```
HKLM\SYSTEM\CurrentControlSet\Services\<service name>\DependOnService
Where <service name> is the name of the SkyMobile service (generally
SKY_VTI).
```

The value to be created must be a REG_MULTI_SZ (multi-string) value, and you should set to the name of the database. In the case of Microsoft SQL Server, the database service is named "MSSQLSERVER".

3.9.13.2 Server/PC/PDA Hangs Whilst Downloading/Refreshing Data

The Application Server uses memory to uncompress and prepare downloaded transfer data from SAP. It uses as much memory as is available for this process. However for large sets of data, there may not be enough memory available to complete the operation and the machine and/or the Application Server may hang. To overcome this, a maximum transfer limit may be imposed that causes SAP to buffer up the data; send it down in chunks. Set the 'TransferBufferSize' configuration option in bytes to prevent this from occurring. Refer to the Application Server configuration section for more details.

3.9.13.3 Download/Refresh Option Fails in SAP

If a data extract exceeds the ABAP heap size, the RFC call is terminated and is backed out. The Application Server or Access Gateway writes the RFC failure details to their system logs. The SAP system log also has various SQL and backout messages. The only definitive message is written to the work process log for example, ABAP heap size exceeded. The only way around this issue is to re-design the download/refresh processing to optimize the amount of data extracted and/or sent.

3.9.13.4 User Interface Cannot Connect to a Application

This problem is mainly due to an incorrect IP/Port connection, or the specified application version not being available to the Application Server. Perform the following checks:

- Check that the Application Server runs by displaying its Web status page
- Check that the port, its type and designated application version is available in the 'ports' section
- Check the log to see if the application is loaded
- Check that you are not overriding the application or version using an answer back string.

If the port, port type and application version are correct and are available to the Application Server, check that the application version has a 'initial function' configured.

3.9.13.5 How Do I Find Out The IP Address of a Application Server?

Every time a Application Server starts, it issues a heartbeat request to the back end host system, that is SAP. Part of this heartbeat, is the IP address and command port details. You may view these connection details online in SAP using the YECJ transaction. Once you have the IP address, you may then use the Web status page to connect to the server: http:/{IP address}:{HTTP control port}. This web status page displays all the attributes, ports, configuration and log entries.

3.9.13.6 Certain Function Keys Don't Work in VT220 Emulation Mode

Check the telnet client configuration and install VT220 emulation and function key support. If unsure, contact the vendor for more information. As a last resort and to support specialized terminals, you may configure your own text terminal type, specifying the function key escape sequences.

3.9.13.7 Scalability Issues Under Windows

If you are running more than a handful of users against a central server or gateway, the central machine should be running a "server" operating system. Normal Windows 2000, XP, 2003 installations are not designed to scale under load in the same way as their equivalent server versions, for example, 2003 server.

3.9.13.8 Connectivity Problems Over a GPRS Network

Instances are observed over some GPRS networks where the default MTU (Maximum Transmission Unit) setting on the machine acting as gateway can be a problem. This is a setting in the TCP/IP stack that controls the maximum allowable size of a TCP/IP packet. It applies equally to both Windows and UNIX machines, although the method of setting it varies depending on the platform in use. If you are experiencing connectivity problems over your GPRS network, it may be worth reconfiguring the machine(s) on which your gateways run as follows:

Config

```
MTU Setting: 576
Path MTU Discovery: Disabled
```

There are a number of freeware and shareware utilities available that can configure these parameters in Windows.

In UNIX, the MTU settings are often kernel parameters. Changing these parameters require the intervention of UNIX systems administration staff. The exact method of configuring the parameters depends on which flavor of UNIX is in use.

3.9.13.9 Local Database Cannot be Used By The Application

This usually occurs if the LDB definition is maintained and fields are removed or changed. The Application Server attempts to automatically adjust the table, but some databases may not have a high enough level of JDBC compliance or do not support the SQL functionality to do this. If this is the case, the server disables the LDB table from use and you must resolve the issue manually; perform the changes yourself, or drop the table and the server automatically recreates it.

3.9.13.10 Application Or Local Database Is Not Available

Check that specific cross reference configuration is not configured in the SAP workbench to restrict which applications go to which Application Servers. This is linked to the server-id. Similarly, you must configure local database definitions that are not directly linked to a application for all servers, or configure to maintain for a specific Application Server through its server-id.

3.9.13.11 A Application Server Won't Start after Changing SAP Connection Definition Name

An Application Server maintains a unique local database for every SAP destination; tables may not be shared. If you change the SAP destination, the Application Server detects that it has an 'orphan' definition and stops. You must drop the unused table definitions, or use the same SAP connection definition name to continue.

3.9.13.12 Local Database Table Data Offsets are Wrong (SAP 4.x systems)

If the underlying SAP DD structure changes, you must resave all dependent LDB field definition. This is because the server stores the offset in order to perform LDB processing using dynamic SQL in SAP. Many ABAP LDB programming options also use offset to retrieve and select data values. This only applies to previous non-ECC releases of SAP.

3.9.13.13 SAP Download/Refresh Tasks Locking Out on NRIV Table

If database download/refresh tasks hang in SAP (use transaction SM50 to view) and if they wait on locks (enqueue) on table NRIV, this is because the YVTI_TFRID number range is not buffered. Use transaction SNRO to update the definition and specify memory buffering. You should do this as a local change, since transporting number ranges can cause inconsistencies across environments

3.9.13.14 Difficulties Communicating with a Serial Device

Communication with serial devices is a thorny area, because there are many variables at work, any one of which can result in communications being scrambled or lost. If you have difficulties, the following suggestions may help:

- It is essential to have on hand the documentation for the serial device that you attempt to integrate with.

- Ensure that any dip switch settings on the device match the communications parameters you have entered in the relevant section of the server configuration file.
- Sometimes the Application Server does not appear to receive information because the incoming messages are not in the anticipated format, so it keeps accumulating data without ever invoking the associated exit. To check whether this may be the case, try setting the InputMessageType parameter to "U" (unformatted), then write a custom serial device exit that simply writes everything coming in to the log. Check to see what (if anything) is actually received.
- On a Windows platform, you may need to use the MODE command from a DOS command window to correctly set the communications settings for the device. Check the help on the MODE command for further information.
- If all else fails, there are tools for debugging serial device communications available. One such tool that is freeware is called Portmon and you can download from http://www.sysinternals.com.

3.9.14 Automatic Provisioning Process

Kony for SAP fully supports the automatic provisioning of the Secure Container from a central MEAP server that you just download and start and you are prompted to:

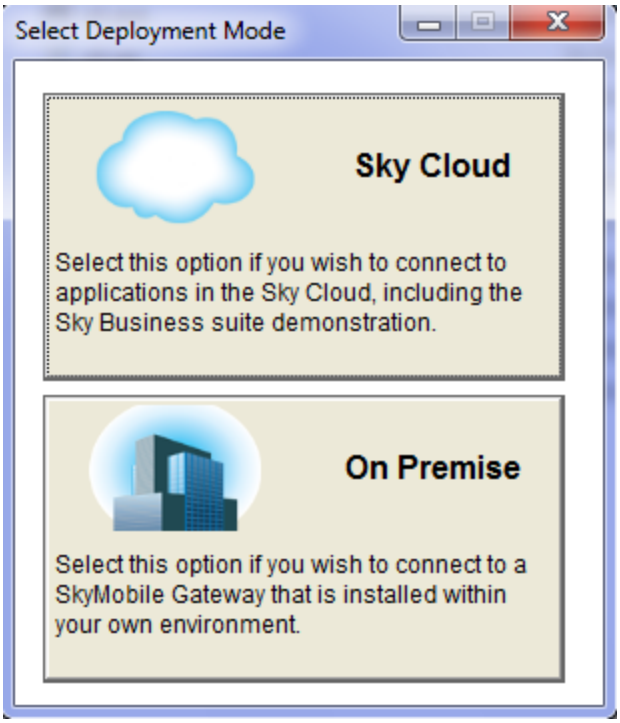
- Connect to either the Sky Cloud or On premise MEAP server
- Type in a profile and optional pass code (PIN) to use.

In this way, there is no need for any manual intervention and the complete deployment of the Container software, applications and data is automatic and transparent to the user.

> ***Note:*** You can totally automate the above by using a server profile "white list" entry in the Sky Cloud for the unique device type and ID, that is, the user is not prompted at all and is started directly into the application. For more information on automatic provisioning of the Secure Container, consult the Secure Container provisioning section in the Management guide.

Once the above attributes are verified, all the configuration, applications and data profiling requirements are automatically configured and downloaded to the Application server, that is, the user does not have to configure anything; everything is centrally managed and is automatic.

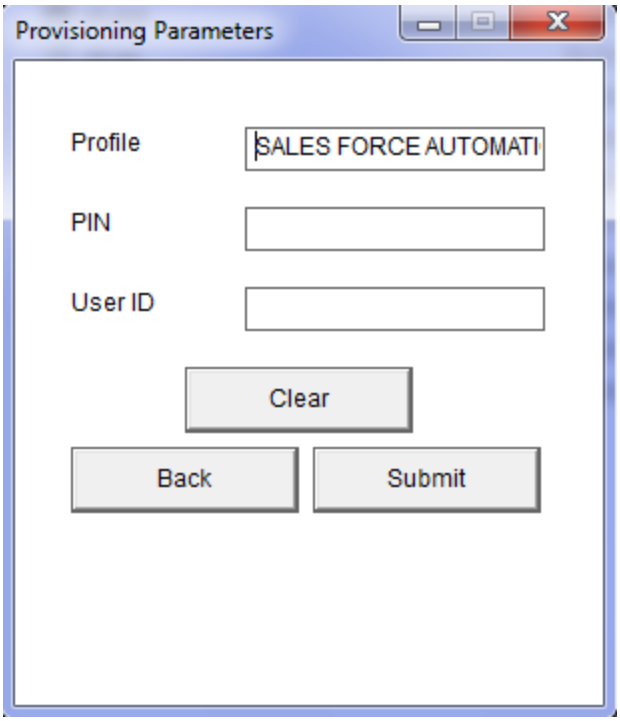
The basic screen flows are as follows:



Click either the "Sky Cloud" or "On Premise" options.

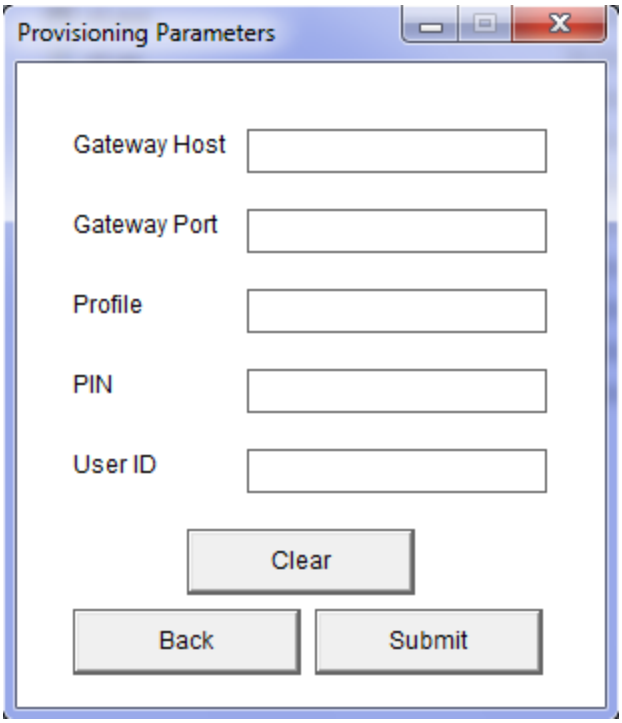
> ***Note:*** If the Sky Cloud is not available, the "On Premise" prompts appear straight. Select Sky Cloud for all demonstration and SaaS (Software as a Service) options.

3.9.14.1 Sky Cloud



You need to specify a valid Profile and PIN (optional) to use. Your administrator or Sky contact should give these details.

3.9.14.2 On Premise



On premise requires you to display a valid Profile name and PIN (optional) and the on premise gateway address and port to connect to. Your administrator or Sky contact should give these details.

3.9.14.3 Initialization

Once entered, the associated configuration, applications, and data profiling is all automatically generated and downloaded. The Application Server then initializes, downloads all definitions and data and displays the first screen.

3.10 Database Guide

This database guide gives you an overview of the steps involved in setting up a relational database for use with a SkyMobile Secure Container, including the in-built SkyDB database system. It focuses on how to set up some common databases, such as HSQLDB, SQL Server, Oracle, and SkyDB. It also covers troubleshooting problems and errors that may possibly occur during this process.

> ***Note:*** In the majority of cases, the in-built SkyDB database suffices and Kony recommends that you only use a alternate database engine if necessary.

3.10.1 JDBC Requirements

If you are plan to use SkyDB to store persistent data, there are no special requirements that you need in place. SkyDB is embedded into SkyMobile.

If you are plan to make use of an external database, ensure that the following prerequisites are addressed:

- Database is available
- JDBC driver suitable for the database is used
- A Database user configuration with appropriate permissions
- Database connection parameters

3.10.1.1 Database

You need to install the relational database, configure and it should be operational before proceeding any further. Refer to your database provider documentation for details on how to install and configure the database.

Back to the #top

3.10.1.2 JDBC Driver

A JDBC version 4 compliant driver for the relational database is required to allow the Application Server to communicate with and manage the database. Often, the database vendor provides a JDBC driver with the database software itself.

The JDBC drivers typically come in one of two formats:

- A zip file (for example, Oracle's 10G JDBC driver)
- A JAR file (for example, the UNA2000 JDBC driver for SQL Server)

Once sourced, you need to place a copy of the JDBC driver file in the `<<SKY HOME>>/vti/classes` subdirectory.

The table below details the recommended JDBC drivers for some common database platforms.

Database	Recommended JDBC Driver	Website
HSQLDB	hsqldb.jar	http://www.hsqldb.org
Microsoft SQL Server Or SQL Server Express	SQL Server JDBC driver v1.2+ or UNA2000	http://msdn2.microsoft.com/en-gb/data/aa937724.aspx
Oracle	Oracle 10G JDBC Driver	http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

For details on how to install the chosen JDBC driver, refer to the appropriate vendor documentation.

Back to the #top

3.10.1.3 Database User

The Sky Application Servers require a database user with appropriate permissions to be able to:

- connect to the database
- create any required tables and indexes
- insert/update/delete/select table records

The procedure to follow varies depending on the database that is used. You should follow the instructions that the vendor of the database you have selected provided.

Back to the #top

3.10.1.4 JDBC Connection URL

To connect to a database using a JDBC driver, you require two key parameters:

- JDBC driver class name

The table below provides the parameters for some of the common JDBC drivers.

JDBC Driver	Driver Class Name	JDBC URL
HSQL DB	org.hsqldb.jdbcDriver	jdbc:hsqldb:file:./db/vti;hsqldb.default_table_type=cached
SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<<host>>:<<port>>
UNA2000	com.inet.tds.TdsDriver	jdbc:inetdae7:<<host>>
Oracle 10G	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<<host>>:<<port>>:<<SID>>

Back to the #top

3.10.2 Database Configurations

The following section provides sample configurations for a range of databases and also technical hints when using those databases with a Sky Application Server.

- #SkyDB
- #HSQLDB
- #SQL Server
- #Oracle

3.10.2.1 SkyDB

SkyDB is the Sky Technologies owned embedded database system that comes as default with SkyMobile. It is a high performance database system that is available across all SkyMobile supported platforms. For performance, data integrity and portability reasons, Sky highly recommends that you use SkyDB by default.

Sample Configuration

Example SkyDB Storage Configuration

```
SERVER.STORAGE.STORAGECOMPRESSONSHUTDOWN = true
SERVER.STORAGE.STORAGECONSISTENCYCHECK = true
SERVER.STORAGE.STORAGEDATABASEFILENAME = ./db/vti.db
SERVER.STORAGE.STORAGEINDEXDEFRAGONSHUTDOWN = true
SERVER.STORAGE.STORAGEINDEXREBUILD = false
SERVER.STORAGE.STORAGEINTERNALIZESTRINGS = true
SERVER.STORAGE.STORAGEINTERNALLOGGING = false
SERVER.STORAGE.STORAGELOCKTIMEOUT = 10000
SERVER.STORAGE.STORAGEPAGECACHESIZE =
SERVER.STORAGE.STORAGETYPE = SkyDB
```

Hints and Troubleshooting

- SkyDB achieves high performance through a combination of different strategies. Some of the factors contributing to its speed are as follows:
    - Because SkyDB operates exclusively as an embedded database, many of the overheads associated with inter-process communication are eliminated.
    - SkyDB uses a delayed write mechanism where the task of flushing data to disk is delegated to a dedicated background thread.
    - Experience on various hardware configurations show that availability of free RAM is generally not a major consideration, whereas there are normally significant overheads associated with reading data from disk. SkyDB therefore maintains a size able page cache.
    - The memory footprint of the database is limited wherever possible by obtaining what are known as "canonical representations" of shared strings. This means that rather than holding duplicate copies of many identical strings, the database consolidates them so that they all point to the same memory address. The garbage collection mechanism can then free the unused memory that the unused duplicates consume.
- SkyDB uses variable length records within a fixed page size of 4096 bytes. It is currently unable to store data where a single row consumes more than 4096 bytes. If you have a need for a very "wide" table row, you may need to consider an alternative storage mechanism.
- The SkyDB page cache configuration is measured in number of pages. It defaults to 1250 pages. As a general rule of thumb, a cached page can be considered to consume approximately 10 KB. This is somewhat variable depending on the actual contents of the page. You require 4096 bytes for the "raw" page contents, and an in-memory representation of the page contents is also stored with the page. Because Java stores data in Unicode (where each character consumes 2 bytes), this in-memory representation of the page contents can in theory consume up to 4096 x 2 = 8192 additional bytes. However, in practice not all pages are completely full, and there is also a certain amount of storage overhead within the page itself that is not transferred to the in-memory representation of the data. Hence, 10 KB is a useful rule of thumb when performing page cache sizing calculations.

- The `temp` file that is sometimes present in the same directory as the database file contains a cross-reference of the database. When the database is properly shut down, this file is merged with the "main" database file and then deleted. If the database is not shut down cleanly, the `temp` file is used to ensure that the database is brought back to a consistent state the next time it is opened.
- You can enable or disable the logging of SkyDB internal activity through the "StorageInternalLogging" parameter. This information is at a very low level, but may occasionally be helpful in diagnosing problems.
- You can enable or disable the shared string memory management by the "StorageInternalizeStrings" parameter.

> *Important:* You must not enable the **StorageInternalizeStrings** option when using the CrEme JVM on Windows Mobile or Windows CE, as too many internalized strings cause a "panic" condition that completely shuts down the JVM.

Back to #top

3.10.2.2 HSQLDB

HSQLDB is a freeware database system that supports JDBC.

> *Important:* Field experience shows that the HSQLDB database is subject to serious intermittent corruption. When this problem occurs, the database becomes unusable and you must rebuilt from scratch. Sky therefore strongly recommends that you should use HSQLDB only for development and testing purposes, as it is not robust enough for use in a production setting.

Sample Configuration

Example HSQLDB Storage Configuration

```
SERVER.STORAGE.STORAGETYPE = Database
SERVER.STORAGE.STORAGEDATABASEDRIVER = org.hsqldb.jdbcDriver
SERVER.STORAGE.STORAGEDATABASEURL =
```

```
jdbc:hsqldb:file:./db/vti;hsqldb.default_table_
type=cached;hsqldb.cache_scale=10
SERVER.STORAGE.STORAGEDATABASEUSER = sa
SERVER.STORAGE.STORAGEDATABASEPASSWORD =
SERVER.STORAGE.STORAGENATIONALCHARACTERSUPPORTTYPE = Inbuilt
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTION = true
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTIONPREFIX =
SERVER.STORAGE.STORAGESQLLOGGING = false
SERVER.STORAGE.STORAGEUSETABLENAMEINDROPINDEX = false
SERVER.STORAGE.STORAGEDATABASEINITIALSQL =
SERVER.STORAGE.STORAGEDATABASEFINALSQL = shutdown compact
```

Hints and Troubleshooting

- HSQLDB writes all SQL updates to a `.log` file that is only applied to the database when the database is closed. This process can sometimes take up to several minutes to complete. If the database is not closed down properly (for example, in the event of a process "crashing"), the database is instead updated when a connection is next made to it.
- HSQLDB has a variety of performance-related configuration parameters. These are of particular importance when operating on a resource constrained device (for example, under Windows Mobile). Consult the HSQLDB documentation for further information on the best use of these options. Some of the most important ones are as follows:
    - hsqldb.default_table_type
    - hsqldb.cache_scale
    - hsqldb.log_size
- HSQLDB is thread-safe, but the kernel of the database engine is single-threaded. This means that it is not a good choice for use in a multi-user environment.

Back to #top

3.10.2.3 SQL Server

Microsoft SQL Server (both the full blown version and Express version) are fully supported and good for high volume, complex database scenarios. There are a number of good JDBC drivers available from both Microsoft and third party vendors.

Sample Configuration

Example UNA2000 JDBC Driver Configuration

```
SERVER.STORAGE.STORAGETYPE = Database
SERVER.STORAGE.STORAGEDATABASEDRIVER = com.inet.tds.TdsDriver
SERVER.STORAGE.STORAGEDATABASEURL = jdbc:inetdae7:localhost
SERVER.STORAGE.STORAGEDATABASEUSER = vti
SERVER.STORAGE.STORAGEDATABASEPASSWORD = vti
SERVER.STORAGE.STORAGENATIONALCHARACTERSUPPORTTYPE = Standard
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTION = true
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTIONPREFIX =
SERVER.STORAGE.STORAGESQLLOGGING = false
```

Example Microsoft SQL Server JDBC Driver Configuration

```
SERVER.STORAGE.STORAGETYPE = Database
SERVER.STORAGE.STORAGEDATABASEDRIVER =
com.microsoft.sqlserver.jdbc.SQLServerDriver
SERVER.STORAGE.STORAGEDATABASEURL = jdbc:sqlserver://localhost:1433
SERVER.STORAGE.STORAGEDATABASEUSER = vti
SERVER.STORAGE.STORAGEDATABASEPASSWORD = vti
SERVER.STORAGE.STORAGENATIONALCHARACTERSUPPORTTYPE = Standard
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTION = true
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTIONPREFIX =
SERVER.STORAGE.STORAGESQLLOGGING = false
```

Hints and Troubleshooting

- By default, Microsoft SQL Express is installed without TCPIP access enabled. To access the database through JDBC, you need to use the SQL Server Configuration Manager tool to enable TCPIP access through port 1433.
- Once created, you need to grant the database user access to the appropriate database.
- Protocol not installed. This situation only arises with Microsoft SQL Server, for which you must install network protocols in order to use. Typically you at least need the TCP/IP protocol. Check your Microsoft SQL Server documentation for further information on how to install a network protocol for use by the database.
- Mixed mode authentication not enabled. You can configure the database server to use either Windows authentication, or mixed mode authentication. If you are using Windows authentication, your username and password are effectively ignored as your Windows username determines your level of access to the database.

Back to #top

3.10.2.4 Oracle

The Oracle database is fully supported and is good for high volume, complex database scenarios. There are a number of good JDBC drivers available from both Oracle and third party vendors.

Sample Configuration

Example Oracle Storage Configuration

```
SERVER.STORAGE.STORAGETYPE = Database
SERVER.STORAGE.STORAGEDATABASEDRIVER =
oracle.jdbc.driver.OracleDriver
SERVER.STORAGE.STORAGEDATABASEURL =
jdbc:oracle:thin:@localhost:1521:XE
SERVER.STORAGE.STORAGEDATABASEUSER = vti
SERVER.STORAGE.STORAGEDATABASEPASSWORD = vti
SERVER.STORAGE.STORAGENATIONALCHARACTERSUPPORTTYPE = Standard
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTION = true
```

```
SERVER.STORAGE.STORAGERESERVEDWORDPROTECTIONPREFIX = vti
SERVER.STORAGE.STORAGESQLLOGGING = false
SERVER.STORAGE.STORAGENULLVALUECHECKING = true
```

Hints and Troubleshooting

- Oracle treats empty strings differently from NULL values. Do not forget to enable the **StorageNullValueChecking** option, otherwise SkyMobile may not ignore empty timestamp fields and not upload data.
- Oracle has a limitation of 30 characters for table names, column names and index names. If you exceed this, it generates an error and the Application Server aborts. Also note that SkyMobile adds a suffix "_pkey" to LDB names for the primary index so your LDB names cannot exceed 26 characters!
- You cannot use a range of characters (for example, % _) as the first character in table names or column names. To get around this, use a couple of characters (for example, vti) in the **StorageReservedWordProtectionPrefix** option.

Back to #top

3.10.3 Database Troubleshooting

This section lists some of the more common database-related problems. Also refer to the relevant database 'hints and troubleshooting' section for more information. If you cannot resolve the issue yourself, contact Kony support.

3.10.3.1 Database Connectivity

Database connectivity issues typically manifest with the Sky product aborting shortly after startup with the following message written to the log: "Error initializing storage". There can be many possible reasons why you cannot establish a connection to the database. Some of the more common ones are given below:

Potential Problem	Solution
Database server is not running	Start the database server.
Malformed database connection URL	Check your JDBC driver documentation to ensure that the connection URL is correctly formatted.
Incorrect port number	Check to ensure that you are attempting to connect to the right port. In a pinch, you can even use a telnet client to determine whether the database server is listening on the port in question (if you don't get a connection, then it isn't running or is using a different port number).
Username/password not valid	Check to ensure that your username and password are valid and have the permission necessary to connect to the database.

3.10.3.2 StorageException Problems

If the product is writing StorageException stack traces to the application logs, this means that the JDBC driver is not called successfully. There can be a number of reasons for this. The most common are listed below:

Potential Problem	Solution
Underlying database problems	Check the database logs for any errors.
Insufficient database privileges	Ensure that the user you are connecting to the database wish has sufficient privileges.
SQL syntax/compatibility errors	Enable SQL Logging (through the StorageSqlLogging configuration file parameter) to capture the violating statement. If you cannot resolve the problem yourself, forward a copy of the log to Kony.

3.11 Recommended Architecture and Configuration

Kony for SAP is flexible and you may implement in many different ways. You may use the following high level recommendations as a guide to the most common and flexible architecture and configuration.

- Install the SkyMobile SAP MEAP into all SAP environments where you wish to execute SkyMobile applications and Sky ECS interface management. If you have distributed SAP systems for example, ERP and CRM, then you may also consider implementing SkyMobile SAP XAI cross system support.
- Implement a primary and secondary SkyMobile Access Gateway behind the corporate firewall (or in the DMZ) to facilitate switch over and disaster recovery
    - Use the HTTPS networking option if you are implementing the gateways behind a proxy server
    - Implement identity management to authorize the device user using your preferred service.
- Application design:
    - Develop your SkyMobile applications in a modular fashion aligned along business process that is, use multiple SkyMobile application versions
    - Use data objects with generic data profiles wherever possible
    - Use ECS interface management with SAP user switching and background server parallel processing wherever possible
    - If intending on using multiple languages, review the multi-language support first for guidelines.

- Configure automatic provisioning:
    - Configure a server profile to define the Secure Container run time environment:
        - Use a "master" configuration with specific overrides (if any) for specific logical platforms / device types
        - Use HTTPS (URL) to connect to a access gateway
        - Configure the gateway identity management requirements
        - Implement the SkyMobile application console (910/001) to launch the Kony for SAP applications and browser applets (if any)
        - Consider implementing a redirection profile in the Sky Cloud and white list device entries to help automate the provisioning process.
    - Configure user profiling for role based application provisioning:
        - Load work groups and user definitions
        - Define roles with application references and data profile generation rules
        - Assign work groups to roles as required
- Install the Kony for SAP secure container onto the required devices from either the public application stores, private enterprise stores or manually through a vendor tool.
    - When the provisioning process prompts, select the appropriate profile
    - When identity management prompts, log on as a valid user
    - If everything is configured correctly, the associated applications and data are automatically synchronized with the device.

4. Development and Customization

This section describes how to design, develop and maintain Kony for SAP application definitions using the MEAP IDE (Integrated Development Environment).

4.1 Key Topics

4.2 Terminology

Term	Description
Group	For easier management and listing purposes, you may allocate application versions, database table definitions and binary objects to a group. You may then selectively list definitions by specifying the group in selection screens.
Application	A high level application area, usually associated with a distinct area of the business, for example, CRM, Field Service, Warehouse Management. Each application area is given a unique three digit number for easy reference and management.
Version	A unique applet (logical process) within a application area, for example, Delivery Picking within Warehouse Management. Each version is given a unique three digit number for easy reference and management.
Function	An application version consists of screen functions and the application flow logic to manage screen processing. Function definitions contain the layout (fields, images, tables, function keys), database integration (select, modify, delete), procedures and programming exit controls to perform more complex processing. These features and how to implement are described in detail in the function builder section.
Action	A screen function may have one or more actions (events) that the user input triggers, for example, pressing of a function key or button, scanning a value into a field or clicking a hot link. These actions then subsequently perform processing such as database operations, execute procedures, execute custom ABAP or Java programs, calling other functions.
Procedure	Procedures enable you to specify a sequence of one or more built-in commands to control the behaviour of the screen. You can associate a procedure with any action (event) and can call procedures from within procedures. The idea is that you can use procedures to avoid having to write custom exit programs, thus making the applications totally device independent.

Term	Description
LDB (local Database)	Kony for SAP fully supports the definition of its own SQL database tables, in which to store application data. You may imbed these definitions into functions in-order to perform automatic selection and maintenance functions. This built-in capability eliminates having to develop complex programming code and helps to considerably reduce the complexity of applications. You may use Local database definitions (LDB) to define data download, refresh and upload rules and schedules. The Secure Container also automatically creates and maintains Local database definitions that support its own built-in database (SkyDB) or a JDBC compliant SQL database, for example, SQL Server, Oracle, and DB2.
DOB (Data Object)	You can also manage local database tables as SkyMobile data objects (DOBs) that allow you to configure inter-table dependencies and perform transaction level data synchronization. Thus you can process multiple tables within a single synchronization operation. Data objects are not referenced directly from application screens, but are rather configured as complimentary objects (see below).
Data Profiles	Data profiles are used to effectively automate the distribution of data between remote containers on the device and the back end host using configurable rules, what data goes to which container. The data profile workbench is used to configure all the definitions. You may configure profiles directly against data objects to automatically invoke in data download, refresh and upload operations.
Server Profiles	These are used to fully automate the provisioning of Secure Containers on mobile devices. They allow you to centrally manage the configuration, applications and data profiles for a device and support the use of white/black lists, encryption objects.
User Profiles	These are used to implement the "role based provisioning" of applications and data; users only get what applications and data they are authorized to.
Identity Management	This is the process by which a user is authenticated against a specific device and is performed before the Secure Container is started.

Term	Description
Application Console	A standard SkyMobile application that is used to launch and manage multiple applications within a secure container. The console may be use as is, or copied and customized. For more information on the application console, click here.
BOB (Binary Objects)	The SkyMobile framework supports the concept of a 'binary repository' where various types of file may be imported and exported, for example, graphics, word documents, software, signatures (drawing objects), and data. A workbench and SDK is provided to support the management of binary objects (BOBs) and standard integration with SkyMobile and various SAP sub-systems, such as GOS (Generic Object Services) is supported. Binary objects may also be associated directly with data object processing; synchronize files associated with data base data.
Complimentary Objects	SkyMobile supports rich programming APIs (application programming interfaces) for both the Java and ABAP programming languages. ABAP is used for SAP programming and Java for Application Server programming. The software development kits (SDK) are designed to considerably reduce the programming effort and skill level required, by providing built-in functions to interact with SkyMobile. Programming exits are optional and are only required where more complex logic, interfaces to other external systems are required. As a further aid, the Workbench can generate code templates based on a screen function. The programming exit concept is very powerful and enables complex applications to be constructed with relative ease. In our experience a relatively low skill level is required, thus customers can develop and maintain their mobile applications without requiring specialist help.

Term	Description
Exits (custom Programs)	Whilst SkyMobile supports "code free" applications using procedures, you may imbed custom Java and ABAP code. SkyMobile supports rich programming APIs (application programming interfaces) for both the Java and ABAP programming languages. ABAP is used for SAP programming and Java for Application Server programming. The software development kits (SDK) are designed to considerably reduce the programming effort and skill level required, by providing built-in functions to interact with SkyMobile. Programming exits are optional and are only required where more complex logic, interfaces to other external systems are required. As a further aid, the Workbench can generate code templates based on a screen function. The programming exit concept is very powerful and enables complex applications to be constructed with relative ease. In our experience, a relatively low skill level is required, thus customers can develop and maintain their mobile applications without requiring specialist help. ***Important:*** Custom exits are not supported for the iOS platform at this stage. Kony recommends that exits should only be used in exceptional cases where the functionality is not possible with procedures.
SOF Standard Object Framework	Standard objects are common programs (exits) that may be shared across applications for example, launch a URL, phone a number, load an attachment. The SOF workbench is used to create your own custom objects as well as view the standard Sky supplied (internal) objects. This is a effective mechanism to abstract platform dependent common routines away from applications and the core Sky software. All SOF object program source is supplied as part of the Sky installation.

4.3 Common Transaction Codes and Icons

Code	Description
YVTA	Complimentary colours
YVTB	Binary Object Workbench
YVTC	Application Server / local database cross reference
YVTD	Data Object (DOB) Workbench
YVTE	Start the SAPGUI emulator
YVTF	Style manager
YVTG	Generic fields
YVTH	Colour manager
YVTI	System Management Console
YVTL	Local Database (LDB) Workbench
YVTM	Manual application synchronisation
YVTN	Number range maintenance
YVTO	Complimentary binary objects cross reference
YVTP	Binary package management
YVTQ	Binary group cross reference
YVTS	SAP Session manager

Code	Description
YVTT	Export (transport) definitions
YVTU	Complimentary styles
YVTV	Value list maintenance
YVTW	SkyMobile Workbench
YVTX	Application server / application version cross reference
YVT4	Data profiler
YVT5	Server profiler
YVT6	User profiler

4.4 Standard Toolbar Icons

	SAPGUI emulation set up		Local Database(LDB) Workbench
	SAPGUI emulator		Generic field manager
	Switchboard manager / Clipboard manager		Transport definitions
	Session manager		Binary object workbench
	Intelligent store and forward		Style manager
	Workbench		Documentation text
	Function list		Logs
	Choose a level or definition		Colour manager
	Display more information. Hints and tips.		Test utility
	Value list manager		Parameter value manager
	Data Object (DOB) Workbench		Complimentary Object(s)
	Trace facility		Where used cross reference
	Maintain version (free format) build information		Number range manager

	Data Profiler		Server Profiler
	Language editor		

The following SAP standard maintenance icons are used throughout.

4.5 How are Applications and Data Managed?

Before you start developing or deploying applications, it is important to understand how SkyMobile manages and executes mobile applications. The integrated development environment (IDE) is used from within the MEAP Server to design, develop and deploy mobile applications from a central location. The application "meta-data" is stored in the IDE database and is downloaded to the Secure Containers on the device to execute; is interpreted at run-time. Applications can operate in a mixture of modes and configurations that best suits the requirement. In any case, the Secure Containers connect to a central MEAP Server through the Access Gateway and automatically 'pull' their application definitions from the IDE. This means that changes are made centrally and are then deployed out to the remote containers. Role based application provisioning is performed through the user profiler and is effectively managed on the device using the Application Console.

4.5.1 Data Objects

As well as applications, you may also automatically synchronize database data and binary objects (files) using data objects. The synchronization of applications and data is managed through the use of timestamps and you may configure to happen automatically, triggered on a event or manually controlled. There are three synchronization events that are used to transfer data objects between the Secure Container and the back end host system:

Event	Description
DOWNLOAD	This event is triggered when the Secure Container is first initialized, or may be triggered by a application or system event. Its function is to initially populate the container database from the back end host, for example, master data.
REFRESH	The refresh function applies delta data object changes based on timestamp differences. You may configure this to periodically check the back end host on a interval (polling) for and updates, or you may push out the changes to the Container on the device. The Secure Container maintains an internal timestamp of when the last refresh occurred (specified by the MEAP Server) and you may use this to check for any updates since.You may configure a logical deletion indicator technique for example, ignore data that is flagged as deleted.

Event	Description
UPLOAD	The upload event transfers data from the Secure Container to the back end host. You may configure this periodically to check the containers database to select data to be uploaded to the back end host. Typically, this is done using a timestamp field on the database table; the Container uploads all data where the timestamp is not populated and then updates the timestamp locally once the back end hosts all the data. The upload may also receive data back from the back end host, for example, deletions to apply, and refreshed data.

4.5.2 Profiling

Profiling is used to provision the Secure Container environment and enable the "role based provisioning" of applications and data; the user only gets what they are authorized to. There are three different profiler functions in the IDE:

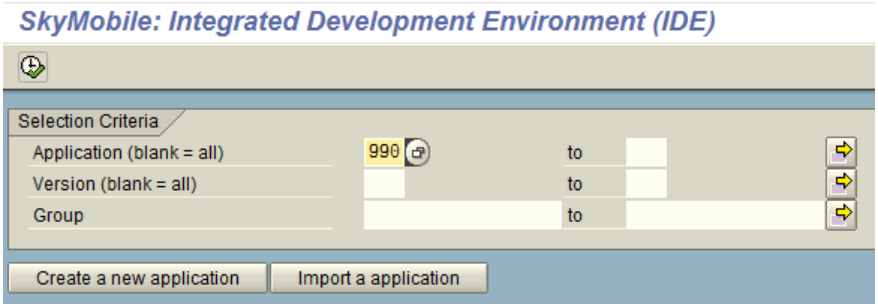
Data	Determines the synchronization rules of data objects; who gets what.
Server	Used to provision the Secure Container environment; its run-time configuration, network security, generic applications and data profiles.
User	Associates users with roles identifying on demand applications and data profiles. This is used in conjunction with the Application Console.

4.5.3 Identity Management

Users are authenticated and associated with secure containers on the device through a process called Identity management. This occurs before the container starts so that any subsequent synchronizations are performed against a "known user".

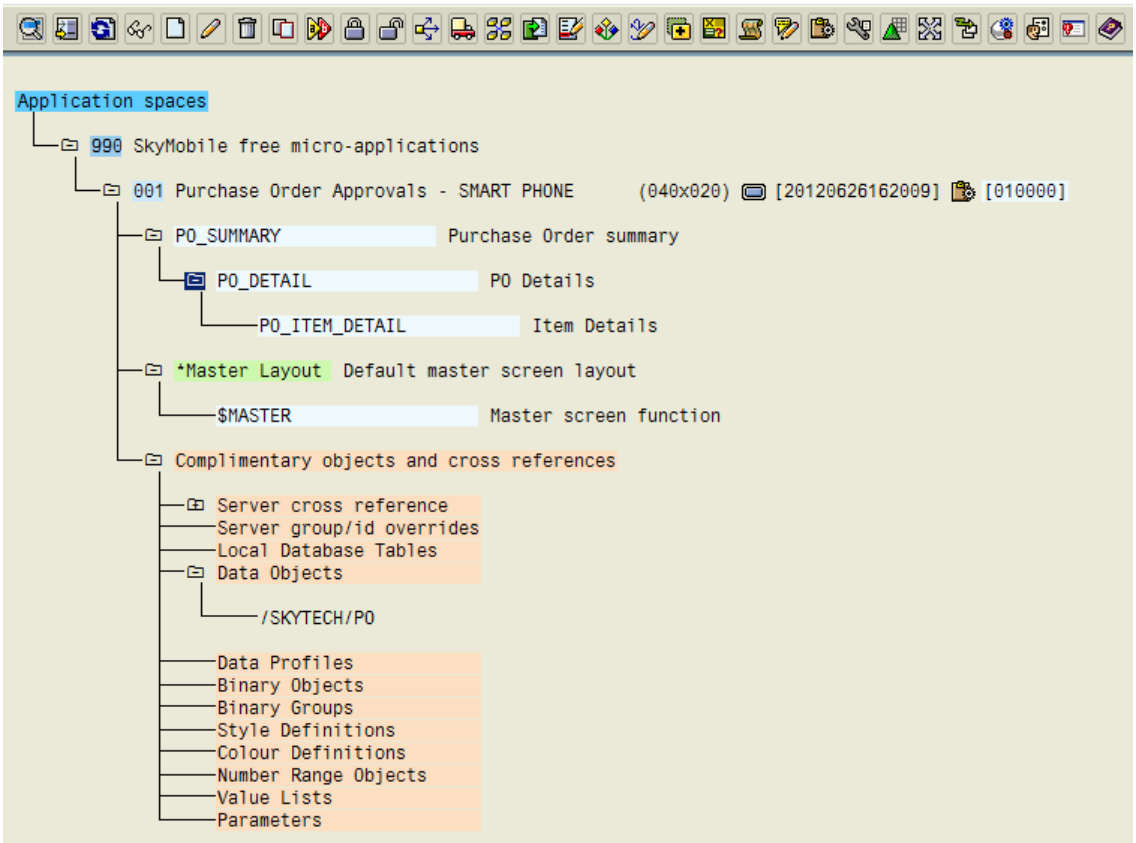
4.6 The IDE Workbench

The SkyMobile integrated development environment is used to develop applications and their supporting objects. It enables you to develop screens, procedural logic, data definitions, and integration. The SAP IDE workbench is started in SAP using the YVTW transaction. A selection screen first appears to enable you to restrict the view.



Here you may select specific, or a range of, applications and versions. You may also select all applications that are configured in a group. It may also be convenient to create a new application or import an application from an exported definition. When you execute the selection screen, the main hierarchy display appears.

SkyMobile Integrated Development Environment (IDE)



The hierarchy displays all the selected application areas, versions, functions and complimentary objects in a expandable/collapsible format. The display also attempts to show the call levels of screen functions, for example, logon screen, main menu, and options by analyzing the internal events; what screen functions call what? To prevent recursive loops and massive displays, hierarchy levels that are already displayed are not expanded and have the message "[Already expanded]" against them. Nodes are expanded using the +/- node connectors or by positioning the cursor on a node and clicking on expand and collapse icons on the application toolbar. Either way, the workbench attempts to remember the expansion and position so that it is re-displayed "as is" when you return to it. To navigate, you may: double click any entry on the hierarchy, position your cursor on the relevant line or level and click the relevant application tool bar icon or use drop down menu options to branch to relevant areas of functionality.

> ***Note:*** In more complex applications, where you can repeatedly call the same screen functions from many places, the hierarchy can be become difficult to navigate and screen functions hard to find. In these cases, the function list (see the following description) may be useful.

From the hierarchy, you may perform the following:

- Maintain and manage application components:
  - Application area
  - Version (applet)
  - Create function and function lists
  - Transport applications to another MEAP Server or export them to an external file
  - Test applications using the SAPGUI emulator
- Invoke other workbenches to maintain and manage:
  - Local databases
  - Data objects
  - Binary objects
  - Data profiles
  - Server profiles
  - User profiles
- Maintain complimentary objects and cross references:
  - Server cross reference
  - Server group/ID overrides
  - Local Database Tables
  - Data Objects

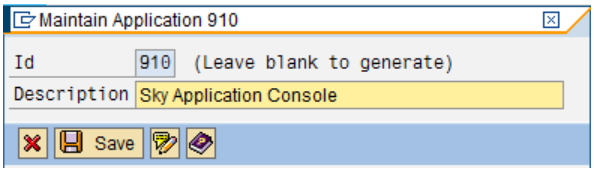
- Data Profiles
- Binary Objects
- Binary Groups
- Styles
- Colours
- Number Range Objects
- Value Lists
- Parameters

4.6.1 Application and Version Numbers

Kony for SAP executes mobile applications at the application version level, identified by the three digit identifiers, for example, 996/003. You may specify these unique numbers manually, or automatically generated when you create. You may re-assign the application and/or version numbers using the rename option on the main workbench screen. This application version numbering system is what is configured to connect to applications. This configuration differs depending on the emulation method used, and in some cases you may specify dynamically. Refer the appropriate installation and configuration guide for more details. As a general rule, Kony reserves application numbers from 900 to 999 for its internal, sample and template applications and you should avoid these. When you create an application and version, you can allow the system to automatically generate the next available free number for you.

4.6.2 Application

An application is the grouping mechanism available to "organize" common applications. The combination of application and version forms the unique identifier for an individual application. To create an application, go to the workbench, and place your cursor on the "SkyMobile applications" heading. Click the **Create** button on the application toolbar and the following pop-up appears:

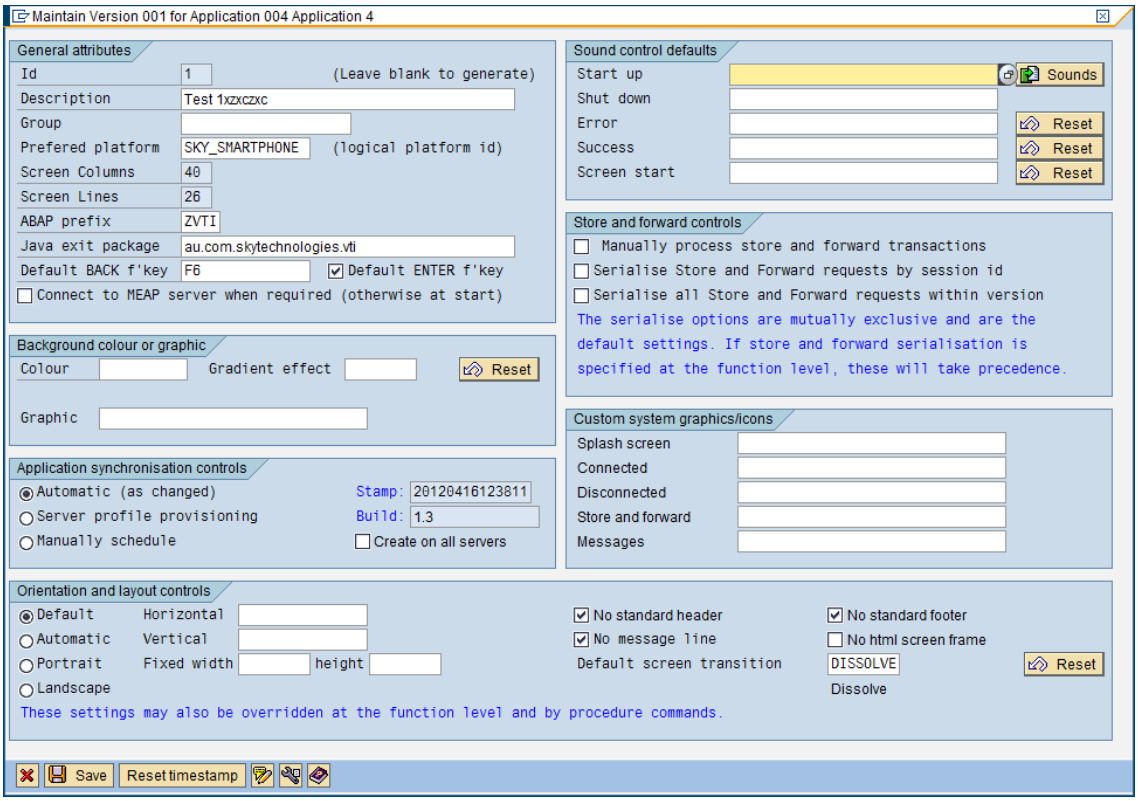


Enter an application description, and (optionally) an application ID that must be numeric. Click **Save** to create the application definition. To modify an existing application, simply "double click" it, or alternatively place your cursor on the application and click the **Change** button on the application toolbar. You may use the text icon to further document the application level as to its usage.

> ***Note:*** Kony for SAP generates an application ID if you leave the field blank.

4.6.3 Version

A version is created within a unique application area to identify a unique application or applet. Typically, a version is used to cater for different screen sizes, different variations of functionality and/or specific emulation modes. To create a version, place your cursor on the relevant application, and click the **Create** button on the application toolbar. The following pop-up appears:



Field	Description
Id	A unique three digit version number. When creating a new version, you may leave this blank to generate the next available unique version number in the application.
Description	A unique description of the version.
Group	An optional group to assign the application/version to.

Field	Description
Preferred platform	Specify the preferred logical platform that the application version is designed for. If specified, the screen columns and lines are defaulted from this and the total column and line entry fields are made display only (greyed out).
Total screen columns/lines	The screen dimensions, for example, 26x10 for RF, 40x16 for PDA, 80x24 for PC. SkyMobile can handle screens as small as 5x5 to as large as 999x999. If a logical platform is specified, these fields are greyed out.
ABAP prefix	A standard four character prefix used to generate default ABAP exit names. If you are planning to use the generated names for SAP ABAP exits, the prefix should start with a 'Z'.
Java exit package	SkyMobile gives you the ability to write Java exits, for processing that is not reliant on SAP. As such you need to define a package for the exits. A common definition is "au.com.xxx" where "xxx" is a project or company name.
Default BACK and ENTER function key	When a new screen function is created, these function keys are automatically generated.
Connect to MEAP server when required	If this option is set, SkyMobile only initiates a SAP connection when one is required. This is only really applicable for offline (local) applications. The default behaviour is to attempt to connect to the MEAP host when the application first starts.
Default background colour or graphic	The default background colour or graphic to use for all screens.

Field	Description
Sound control defaults	These are the default start up, shut down, error, success and screen start sounds. ***Note:*** The error, success and start sounds are just defaults and are not automatically applicable to existing screen functions unless you explicitly "reset" the corresponding function and event values. Use the **Sounds** button to maintain your library of sounds in the SkyMobile binary repository.
Orientation and layout controls	These are the defaults to use when screen functions are created; what layout management, transition, orientation is to be used. If nothing is specified here or at the function level, then the defaults are taken from the secure container configuration or run time OS defaults. By default, each screen reserves the 1st line for a header (title), the last line for messages and the second last line as a footer to display event actions (for example, VT220). The no header/message line/footer/screen frame options are used to suppress the use of the standard components, thus enabling more screen space or a different screen design. The screen frame only applies to Web browser applications; draw a frame around the HTML screen definition.
Synchronization controls	By default, SkyMobile automatically synchronises changes to an application version with any connected Application Servers. If you wish not to do this and control synchronisation manually, specify the 'Manually schedule...' option. The current timestamp and free format 'build' information is also displayed. If all synchronisation is managed via Server Profiles i.e. provisioning, then select the "Server profile provisioning" option. ***Important:*** If you specify "Create on all servers", be aware that the application version is downloaded to any secure container that connects to this SAP MEAP.
Custom system graphics and icons	You can customize various graphics and icons that appear at runtime; Splash screen, connected/disconnected icon, store and forward icon, and messages icon.

Field	Description
Store and forward controls	Control the general sequencing of store and forward requests that this application generates. You can further refine these, using the ISF classification options in each screen function definition.
	Maintain a text description of the version, for example, high level application description.
	Maintain free format build information. This is viewable on the client and is useful to see what version of the application is currently loaded.
Reset Timestamp	This updates the version timestamp to the current system date/time.

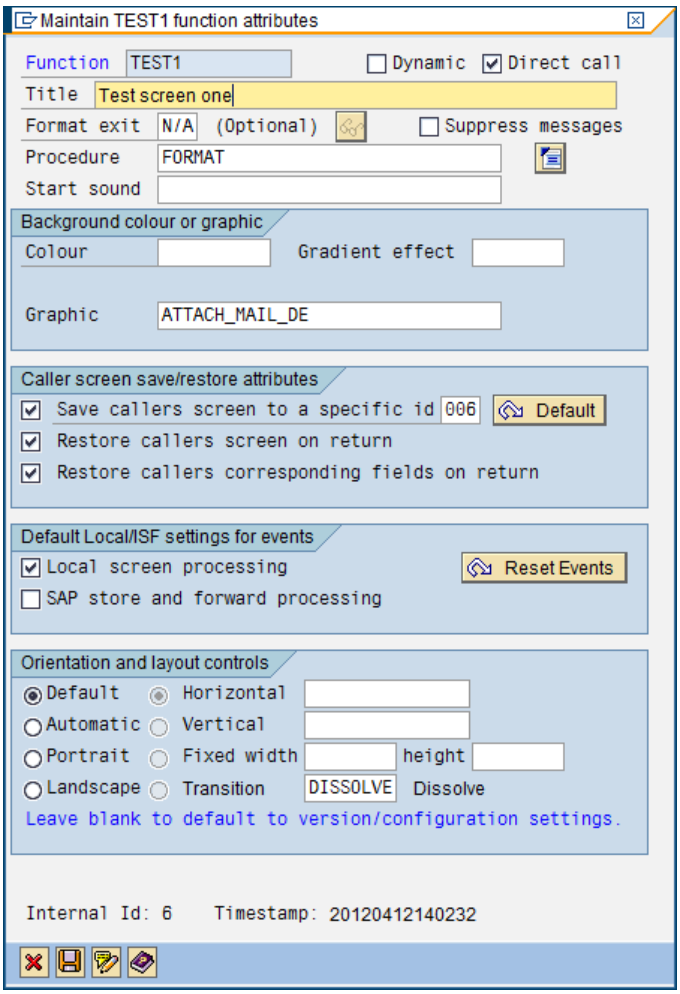
4.6.4 Create Function

> ***Note:*** To maintain an existing function, simply double-click the entry in the hierarchy or on the function list.

Function objects contain both the layout and processing logic for a screen display. They may be either created from the main workbench by positioning the cursor on a relevant version and clicking the **Create** icon on the application toolbar, or by configuring a call to another screen function from within a screen function. In either case, a pop-up is invoked, that prompts for the basic screen function attributes and then the user is transferred to the function builder that is used to define the layout and processing controls of the screen function.

> ***Note:*** Because of its breadth of functionality, the function builder is documented as its own separate help node and not under the IDE that covers the overall framework. Developers spend most of their time using the function builder to design and maintain application screens and their logic.

The function attributes are configured either when you first create a function definition, or by clicking the **attributes** icon (hat) on the application toolbar from within the screen painter. You may configure a "master" screen layout that is automatically copied whenever you create a new function definition. In this way, you may implement a common look and feel. See the description in this section on the 'Master Layout' node. If a new function is created from the hierarchy, the following pop-up appears:

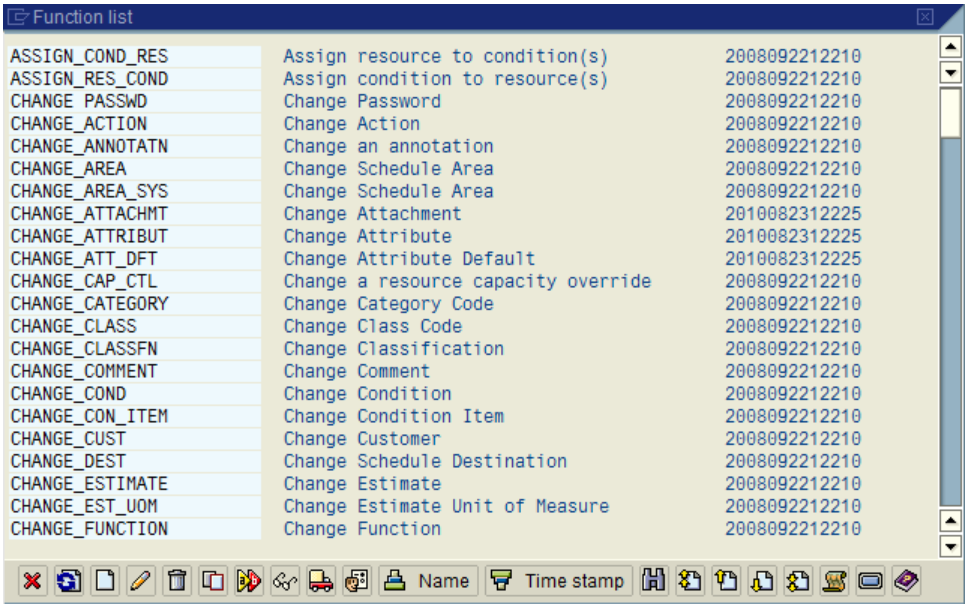


Field	Description
Function	A unique function name within the application version
Dynamically called	Documents that the function is dynamically called; is not directly linked to another screen function, but is invoked using a procedure or programming exit.
Direct call	Indicates that you may call the screen function directly using XAI (external application integration).
Title	A description. This is used as the standard screen header (if configured). If using a standard header, you may imbed substitution variables, identified by a "&'.

Field	Description
Format exit	See the section on programming exits for more details. The format exit allows you to call either a custom ABAP or Java exit program when the screen layout is generated for the first time. They are useful to initially populate or alter the screen layout.
Procedure	Refer the section on procedures for more information. This procedure is executed when the screen layout is generated for the first time. They are useful to initially populate or alter the screen layout.
Start sound	This is the sound file to play when the screen is formatted for the first time.
Background colour or graphic	This is the background colour or graphic to apply to the screen. The default depends on the presentation client used. If this is not specified, the default background colour or graphic on the application version definition is used.
Caller screen save/restore attributes	These options are only applicable for sub-screen type processing, where you may invoke the screen function from many points from within the application version. They instruct SkyMobile to save an image of the callers screen and restore it upon return. You may also alter the values of corresponding (same) fields upon return. These options are also useful if you want to keep the values on the callers screen. By default, SkyMobile uses an internal save id, but you may require a unique save id if a chain of screens are processed and you require to keep their context.
Default local/ISF settings for events	The local and store and forward settings only really apply to each event performed in screen processing, for example, press a button. These settings are the defaults and you may use the **Reset** button to apply the defaults to all existing events.
Default orientation and layout controls	You may fix various display options that are to be applied for the current screen function. If nothing is specified here, then the defaults from the application version and base configuration are taken. You can dynamically override most of these options at run time using procedure commands.
	Maintain a full text description of the screen function.

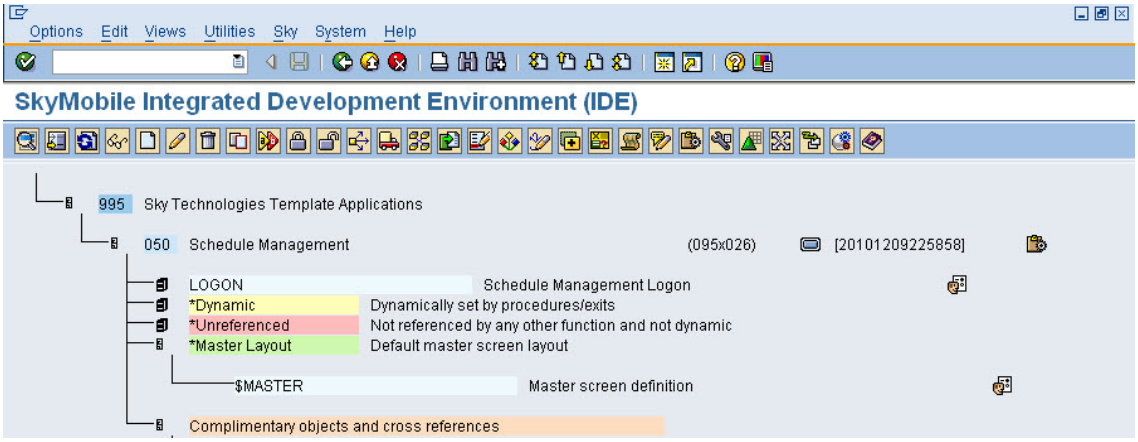
4.6.5 Function List

The icon invokes the function list from either positioning your cursor on a version and clicking on the application toolbar, or clicking the icon at the version level. This is effectively a list of screen functions. You may sort the list into either alphabetic (default) or most recently used (time stamp) order. You may create and maintain the screen functions in exactly the same way as the hierarchy. You position your cursor on the function and select the con from the application toolbar, or double-click the name to launch the screen painter. The find facility is very useful to search and locate functions.



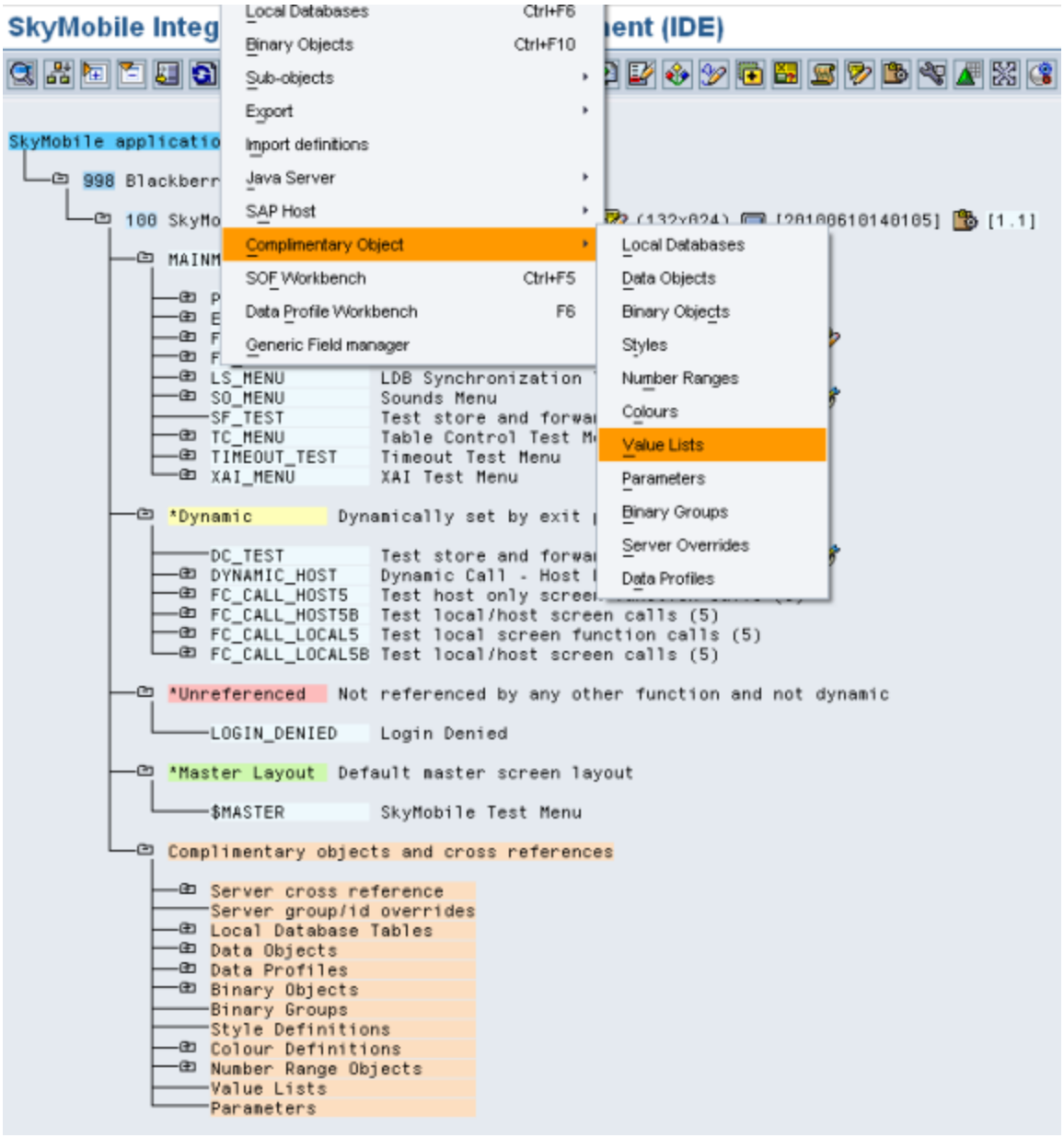
4.6.6 Master Layout

You may use the master layout to define a standard screen function that is used as the basis whenever you create a new screen function definition. In this way, standard headings,and buttons are automatically copied. The master layout is essentially a dummy screen function with a special internal name of $MASTER. You create and maintain master layouts in exactly the same way as normal screen functions.



4.6.7 Complimentary Objects and Cross References

There are certain screen objects that are not statically referenced in screen function definitions, but are rather specified at run time and/or manipulated by screen processing exits. In order for the definitions to be associated with the application so that they may be downloaded, transported, you must declare them as complimentary objects to the application version. You maintain the cross reference definitions from either the **Utilities** > **Complimentary Objects** menu options or by double-clicking the complimentary object nodes in the hierarchy. The various cross references also appear, for example, what Secure Container groups and ids are configured to download the application.



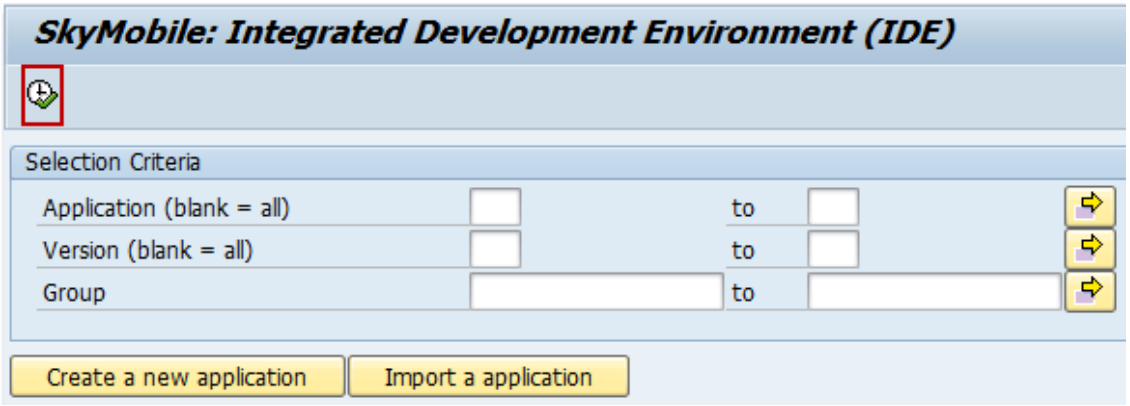
Refer to the section on complimentary objects in this guide for more details on how to maintain the various object types associated with an application.

4.6.8 Working with a Text Description for a SkyMobile Application

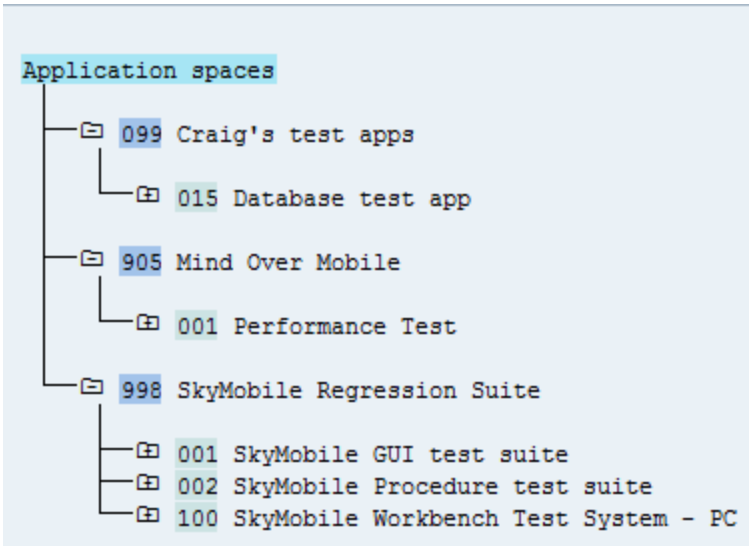
You may add extended text descriptions for a SkyMobile application.

To work with text description for a SkyMobile application, follow these steps:

1. Execute the transaction, **/SKY/YVTW (YVTW)** in SAP system.
   The **SkyMobile: Integrated Development Environment (IDE)** window appears.

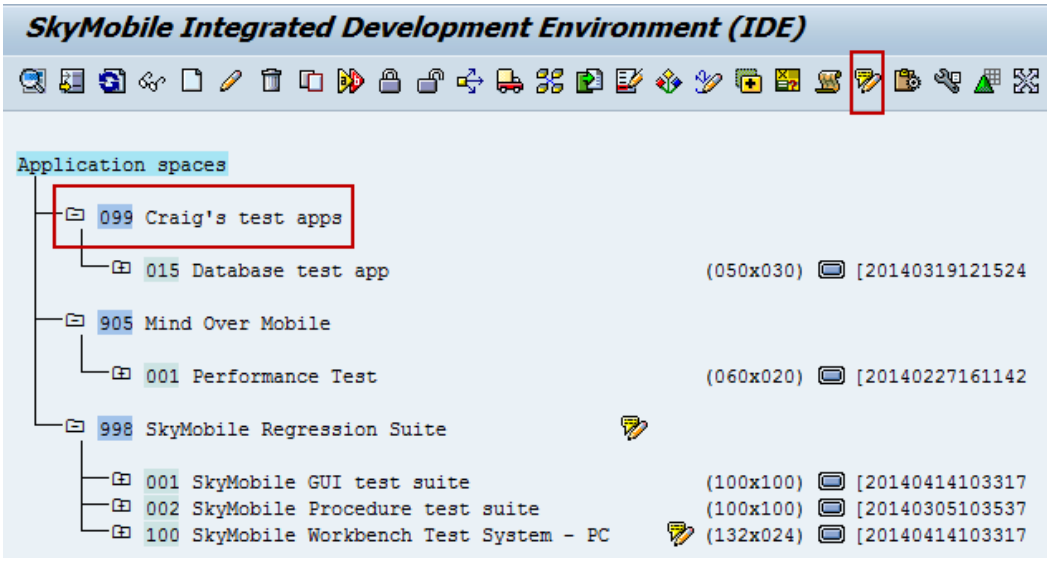


2. Click **Execute**.
   The **Application spaces** dialog appears with the list of SkyMobile application spaces.

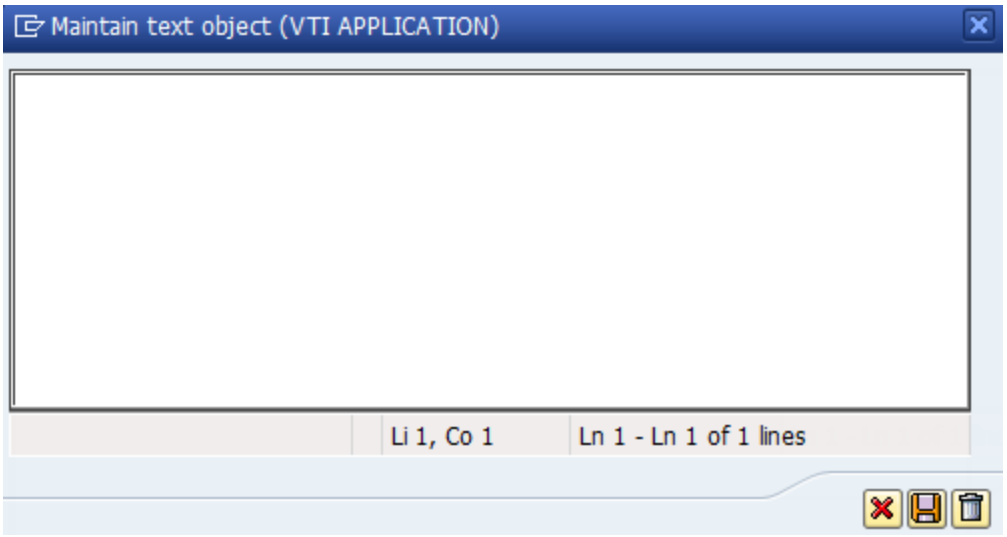


3. To create a text description for a SkyMobile application, follow these steps:

   a. Select a SkyMobile application from the list and click **Maintain text (Ctrl + Shift + F5)**.

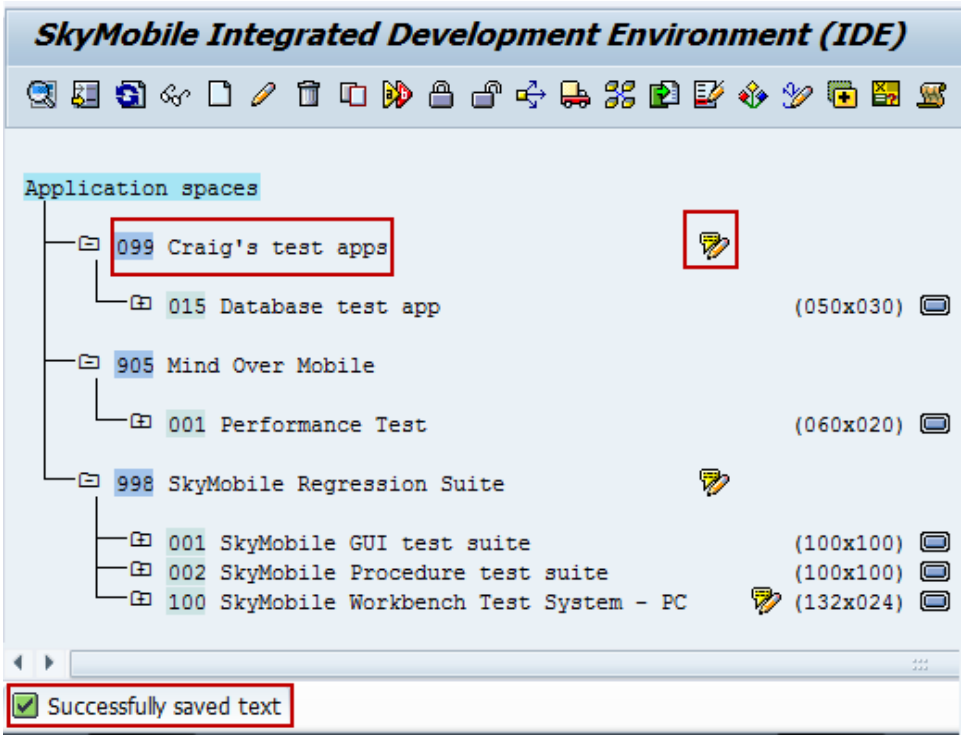


   The **Maintain text object(VTI APPLICATION)** text editor appears.



   b. Enter the text description you want to add to the SkyMobile application and click **Save**. The **Maintain text** icon appears next to the selected SkyMobile application name. The message, "Successfully saved text" appears at the bottom of **SkyMobile Integrated**

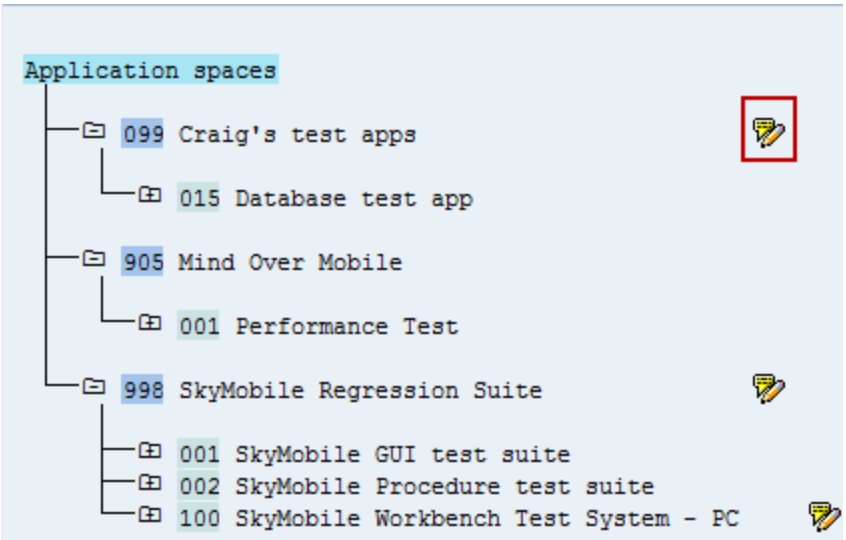
Development Environment (IDE).



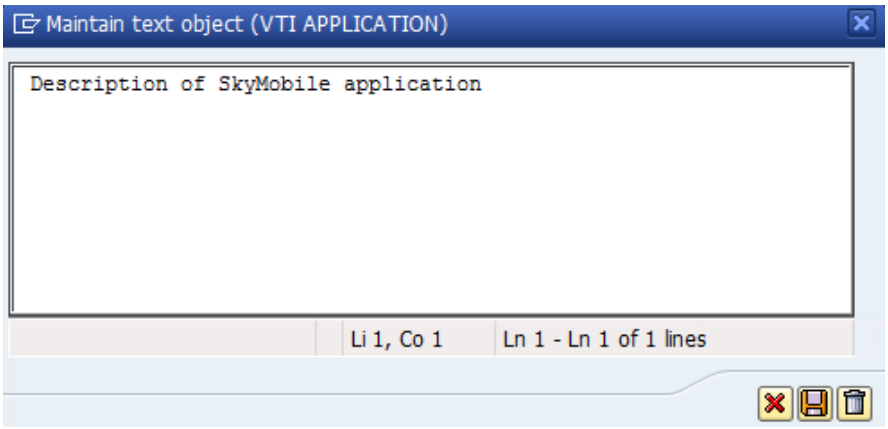
4. To change the description text for a SkyMobile application, follow these steps:

    a. Select a SkyMobile application name from the list that has the **Maintain text** icon next to the SkyMobile application.

b. Click the **Maintain text** icon.

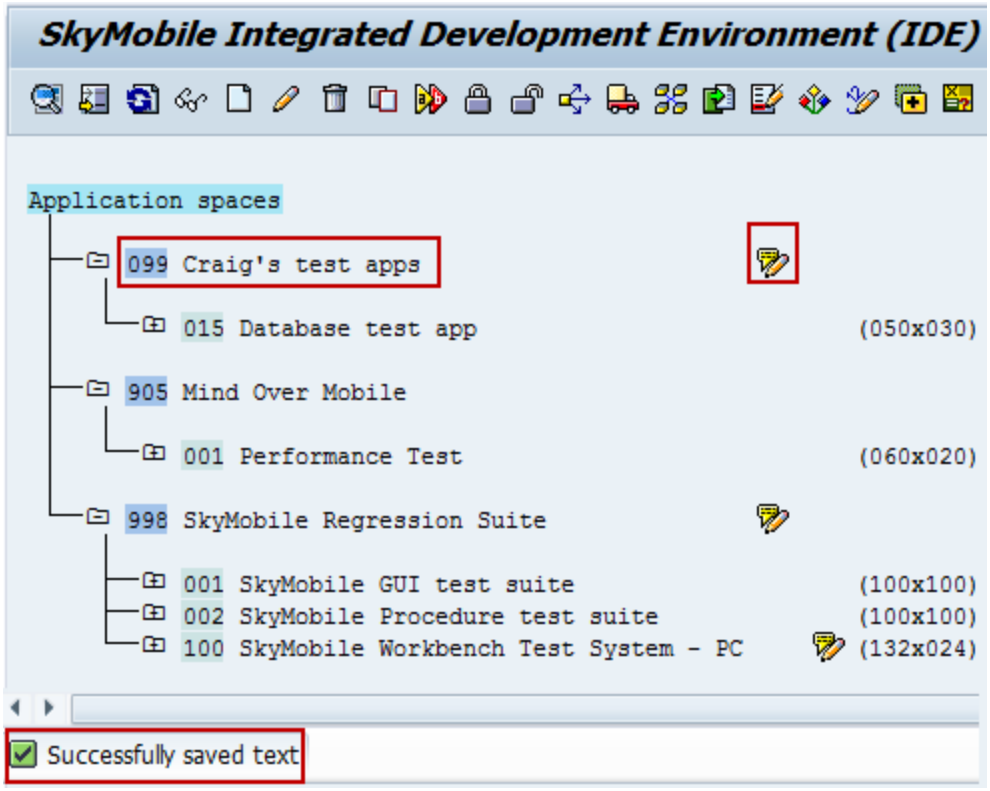


The **Maintain text object (VTI APPLICATION)** appears with the existing description text.



c. You may change the description.

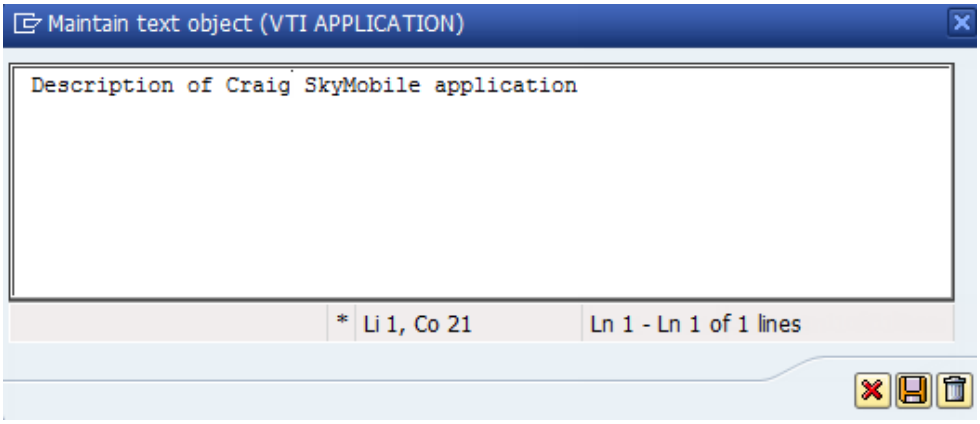
d. Click **Save**.
The system saves the changes. The message, "Successfully saved text", appears at the bottom of **SkyMobile Integrated Development Environment (IDE)**.



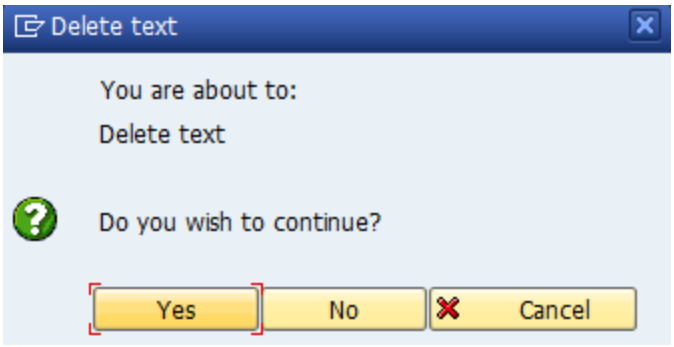
5. To delete the description text of a SkyMobile application, follow these steps:

    a. Select a SkyMobile application name from the list that has the **Maintain text** icon next to the SkyMobile application.

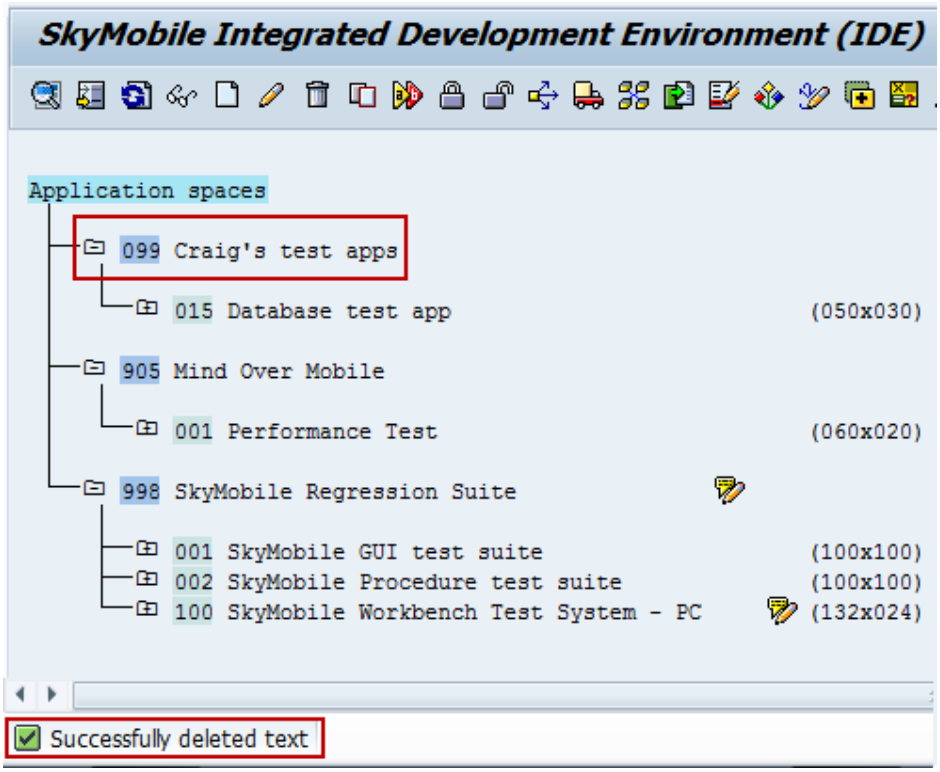
    b. Click the **Maintain text** icon.
    The **Maintain text object (VTI Application)** dialog appears with the existing description.



c. Click **Delete (Shift+ F1)** on the **Maintain text object** dialog to delete the text.
The **Delete text** dialog appears.



d. Click **Yes** to continue.
The **Maintain text** icon no longer appears next to the SkyMobile application name. The message, "Successfully deleted text" appears at the bottom of **SkyMobile Integrated Development Environment (IDE)**.



4.6.9 Maintaining Text Objects

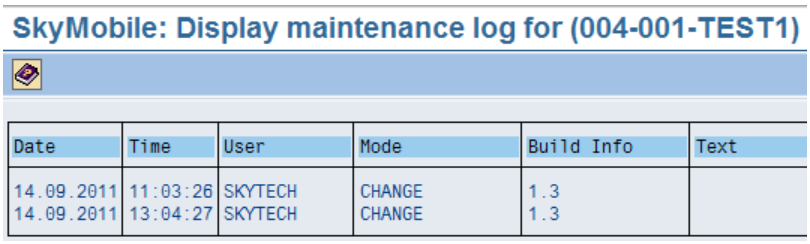
You use this option to maintain extended texts for the following entities:

- Local Database Tables (LDB) (Transaction: /n/SKY/YVTL)
- Data Objects (DOB) (Transaction: /SKY/YVTD)
- SkyMobile Applications (Transaction: /SKY/YVTW).

Use the  icon available on the application toolbar for maintenance of the transactions of the aforementioned objects.

4.6.10 Function Maintenance Log

This utility displays a change log of all maintenance against application version functions. This is usually invoked from the main IDE hierarchy screen by positioning your cursor on a function and clicking on the  icon or the option in the edit menu. You may also invoke it from within the function builder itself. A list of all changes appears.

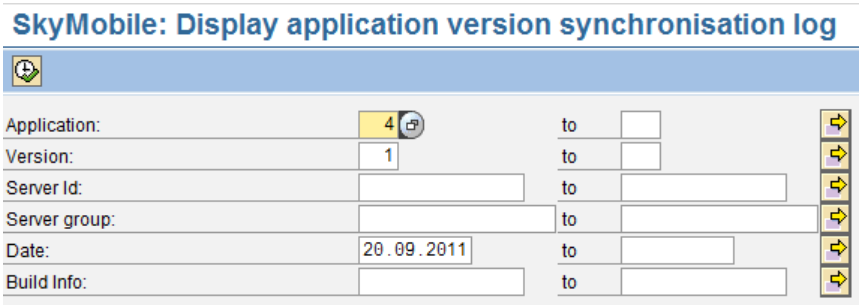

SkyMobile: Display maintenance log for (004-001-TEST1)

Date	Time	User	Mode	Build Info	Text
14.09.2011	11:03:26	SKYTECH	CHANGE	1.3	
14.09.2011	13:04:27	SKYTECH	CHANGE	1.3	

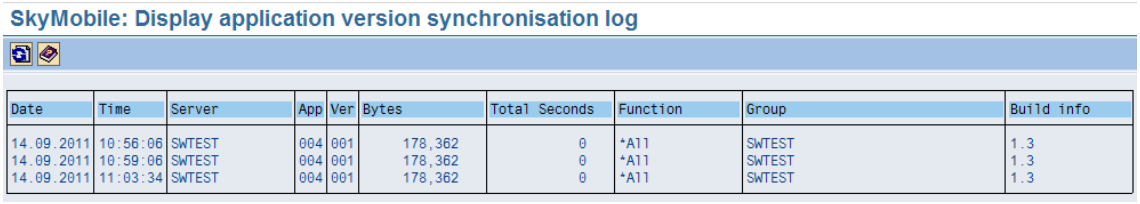
This report is useful to see all the changes that are made to a function, including transports.

4.6.11 Application Version Synchronization Log

This utility displays a list of all application version synchronization requests. You usually invoke this from the main IDE hierarchy screen by positioning your cursor on a application version and clicking the  icon. A selection screen appears to help filter the list entries.



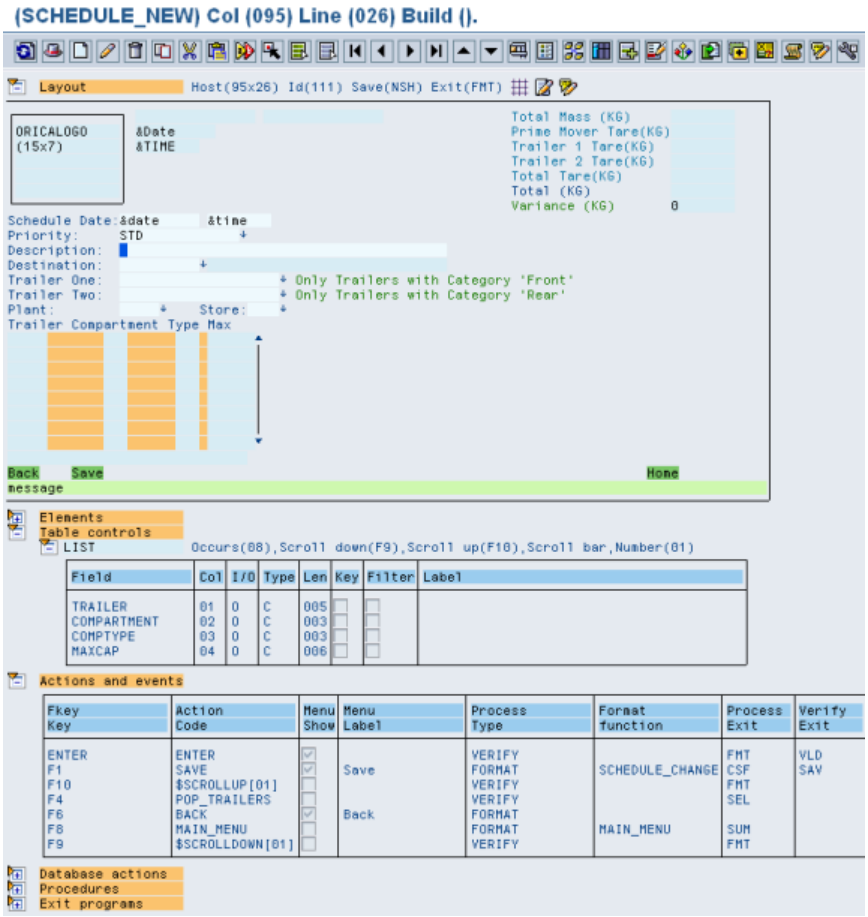
This report is useful to see all the synchronization requests processed for a Secure Container and when they took place.



Date	Time	Server	App	Ver	Bytes	Total Seconds	Function	Group	Build info
14.09.2011	10:56:06	SWTEST	004	001	178,362	0	*All	SWTEST	1.3
14.09.2011	10:59:06	SWTEST	004	001	178,362	0	*All	SWTEST	1.3
14.09.2011	11:03:34	SWTEST	004	001	178,362	0	*All	SWTEST	1.3

4.7 The Function Builder (Screens)

The screen function builder is used to configure the layout and logic of a screen (function) definition. This includes screen fields, database processing, procedures, programming exits, screen action/events. You invoke the screen painter from the application workbench when you create a new function or change an existing function, double-clicking a function name or positioning your cursor on the function and clicking the **Change** icon. The screen painter has distinct sections that you may expand and collapse as required. You maintain definitions by either double-clicking or positioning your cursor and selecting an application toolbar icon. Each screen section is described in detail, later on.



> ***Note:*** Since the screen painter is limited by the capability of the SAPGUI, you cannot view certain elements such as graphics, and drawing objects. These may appear only using the GUI presentation layer, for example, WPC. It is common practice for developers to have this running and switch back and forth between the SAP screen painter and GUI representation.

4.7.1 How Screen Processing Works

Screen functions are dynamic, that is, the Secure Container interprets the screen configuration and renders the user interface according to the layout management configuration and device capability. The events that trigger determines the application screen flow. You may invoke the event by either directly pressing on the screen, a function key or by association with another screen control such as a button, and scroll bar. You define the screen object using the function builder.

4.7.1.1 The Concept of the Initial Screen

You need to configure the "initial screen", the first screen to invoke for an application. This is effectively the anchor point from which the application starts, for example, logon screen or main menu. The initial screen is not configured automatically, and you must manually nominate. To define an initial function, select the screen function that you want to set as the initial function by highlighting with the cursor. Then use the menu path Edit -> Make Initial Function. Save your setting and exit. The initial function now appears directly beneath the application version definition, and is marked as the initial function.

4.7.1.2 How Values are Copied between Screens

When you invoke another screen function from an existing screen function, SkyMobile automatically copies the value of all screen elements that have the same name. You may save the context of previous screens by instructing SkyMobile to save an image (see screen function save/restore attributes), restore it and optionally restore corresponding field values on return. If you have to save and restore later, then you should specify a unique screen id in the function header attributes. This unique id is used internally to identify the saved definition, so if it is duplicated, you may get unpredictable results.

> ***Note:*** Only field definitions of the same name are copied. If you wish to copy fields of different names, and tables use the export/import procedure commands.

4.7.1.3 Generic Fields

A generic field enables you to create a generic "reference" field definition that you can use when defining screen elements. They help to provide consistency across screen definitions and cross referencing to see where fields are used in screens. The generic field manager is used to maintain generic field definitions. It is described in detail later in the section. The "like" field option is used to relate a screen element to a generic field.

4.7.1.4 Local Versus Host Applications

SkyMobile supports both online (real-time) and offline (local) screen functions, or a mixture of both. This is a design option that depends whether the application is still available if there is no connection with the back end host system. You may configure each screen definition to be local or not. If it is not local, the Secure Container passes the request directly to the MEAP Server to process. If the MEAP Server is not available, the screen function is not available. If the screen function is local, it is always available. You may also configure local screen functions that perform real-time transactions with the MEAP Server. All options are configured using the workbench that is described in detail in the workbench section.

4.7.2 The Order of Screen Processing

When you process each screen, the following primary sequence of events take place:

1. Initialization (format):
    a. Save the calling screen
    b. Format the screen layout
    c. Copy any same name fields from the calling screen
    d. Perform any local database operations configured for the format action
    e. Perform any procedure configured for the format action
    f. Perform any programming exits configured for the format action
    g. Import any drawing objects.
2. Event (action) processing:
    a. Perform input screen formatting and value checking
    b. Perform any local database operations configured for the action
    c. Perform any procedure configured for the action
    d. Perform any programming exits configured for the action
    e. Export any drawing objects
    f. Perform output screen formatting.
3. Return to caller:
    a. Process the current screen first (if configured)
    b. Restore the calling screen (as configured)

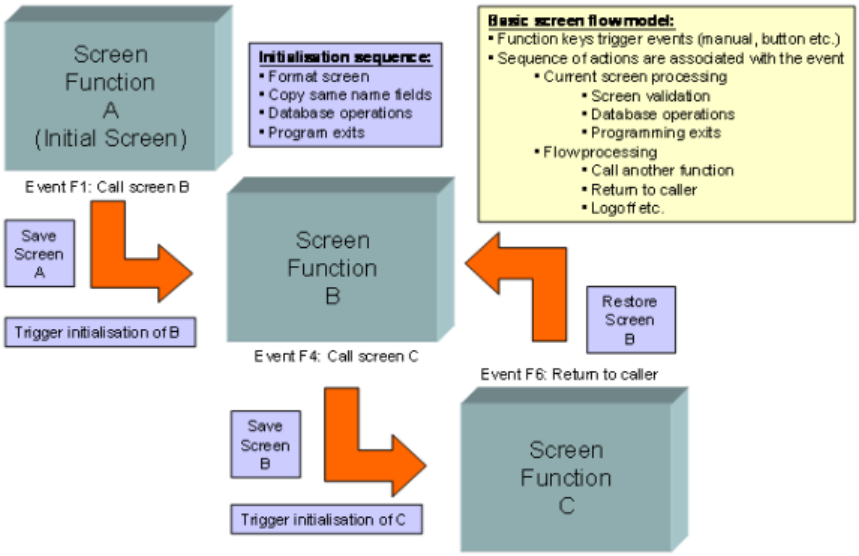
c. Copy any same name fields back to the restored screen (if configured)

d. Re-invoke the restored screen initialization (format)

4. Switching functions:

a. The current screen processing is completed as per event (action) processing

b. The new screen is processed as per initialization (format)



The above diagram represents a very basic screen flow. There are many configuration options that dictate how Kony for SAP is to handle the flow, saving and restoring of screen definitions; these are described in more detail in the Screen Painter section.

4.7.3 Local and Host Processing Modes

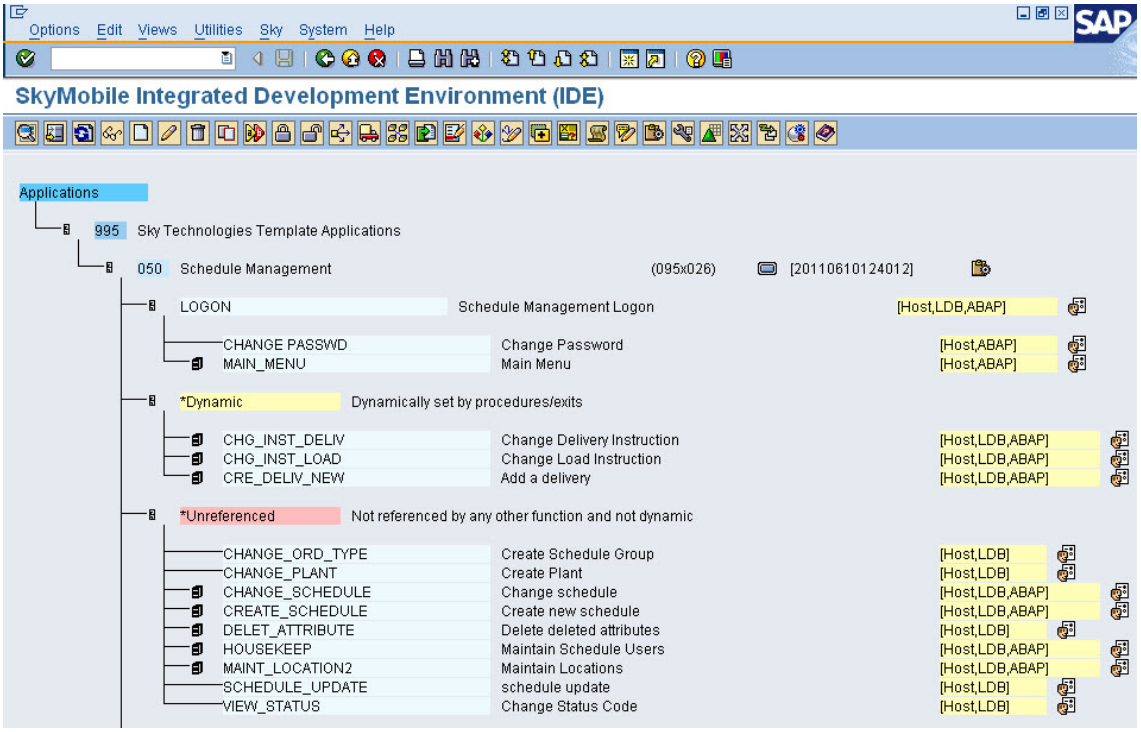
Local, means that the Secure Container attempts to process as much of the screen function locally, without going to the host MEAP Server. A screen function can be totally local (offline), totally host dependent (online) or a mix of the two (local with host dependent actions). The 'local / host' flag configuration determines the operating mode of the overall screen and its actions on the function itself, the events (actions), the type of exit (ABAP/Java) and/or the database operation. You are recommended that you should define all functions as local with host dependent actions as required. When you save the screen function, SkyMobile analyzes all the settings and sets the appropriate internal 'post local' flags on the action events, for example, if you have a local screen that invokes a ABAP, verify exit when a button is pressed, the associated action/event is marked as 'post local'; all local processing is performed first and then the host MEAP Server is called.

You may also configure actions on local screens to be 'store and forward' enabled, if the host is not available, store the entire screen operation in a queue and then process it asynchronously against the host when it becomes available. This technique is useful when real time execution of MEAP Server host transactions is preferred, but you can allow data capture to continue if the MEAP is not available; stored and processed later. Read the "SAP host management > Store and forward manager" section in the SkyMobile systems management guide for more information on this mechanism.

> *Important:* You should take care when mixing local and host functions that use 'return to caller' functionality, because SkyMobile automatically saves the caller screen definition and uses this to restore the position/context when a return to caller event is triggered. Special unique screen ids are used for this purpose. So if a local function triggers a host only function and if a host only function triggers a local function, the SAP and Java session managers may not have the original saved session and a internal 'saved area not found' message appears. The workbench attempts to detect these conditions and issues warnings to that effect.

If the host system is not available, the screen still appears, but the message 'Host not available' appears whenever a host dependent action processes. If the screen is not local, the message 'Host not available' appears at the point where the screen is invoked. Local 'offline' screen processing needs careful design consideration; all the LDB data required and any programming exits must also be fully local for it to be effective.

> ***Note:*** The local and SAP dependencies may be listed against the hierarchy on the main workbench screen. Select local attributes option from the Views menu. The local, LDB and exit attributes appear on the right hand side of the application hierarchy. This feature is very useful to see the overall local/SAP dependencies for all functions in an application version.



When the host is not available, the user is notified that the application version is going into offline mode. Similarly, when the host is available again, the user is notified that the application version is going into online mode. If you wish to only connect to the host when required, select the 'only connect to SAP when required' option on the application Version. The initial function of an application version is always local.

4.7.3.1 Configuring Processing Modes

You determine whether the overall screen function is local in Function attributes of the screen (hat icon on the application toolbar). However, this does not mean that all screen actions are local. When you maintain each action/event, you may override the host/local screen setting.

If the local function has host dependencies; ABAP exit and/or SAP only LDB definition, the post local processing indicator is automatically set on the event that indicates that a call to the host system is required after all the local processing successfully completes. If you expand the "Actions and events section" all exit, local and post local settings appear in the list.

By default, the callers action/event inherits the local/host configuration of the function that is called.

4.7.3.2 Configuring Processing Modes

You determine whether the overall screen function is local in Function attributes of the screen (hat icon on the application toolbar). However, this does not mean that all screen actions are local. When you maintain each action/event, you may override the host/local screen setting.

If the local function has host dependencies, ABAP exit and/or SAP only LDB definition, the post local processing indicator is automatically set on the event which indicates that a call to the host system is required after all the local processing has been successfully completed. If you expand the "Actions and events section" all exit, local and post local settings are displayed in the list.

By default, the callers action/event inherits the local/host configuration of the function that is called.

4.7.3.3 Store and Forward Processing

The concept of store and forward is to temporarily store a SAP transaction in a queue, when the SAP system is not available to process it. Transactions continue to be stored in this manner until the Application Server re-connects with the SAP system. When this happens, the Application Server drains the store and forward queue by submitting the stored transactions for processing in SAP. The application remains in store and forward mode until the store and forward queue has no entries, upon which the mode automatically changes back to online mode for SAP transactions. The message 'Transaction stored' appears to the user. The SkyMobile store and forward mechanism is known as the ISF (Intelligent Store and Forward) manager. The SAP transactions also continue to be automatically stored and forwarded in SAP, until the outstanding SAP dependencies are processed. You must carefully think through the store and forward processing and its implications as to its impact on the application and the inter-dependencies/sequencing of the transactions. With ISF processing, you are effectively making an assumption that the transaction may be successfully processed asynchronously with no dependency on the operator. You may view the active ISF processing queue using the Utilities->SAP Host->Store and forward or YVTI (administration) options. See systems management for details.

Specification

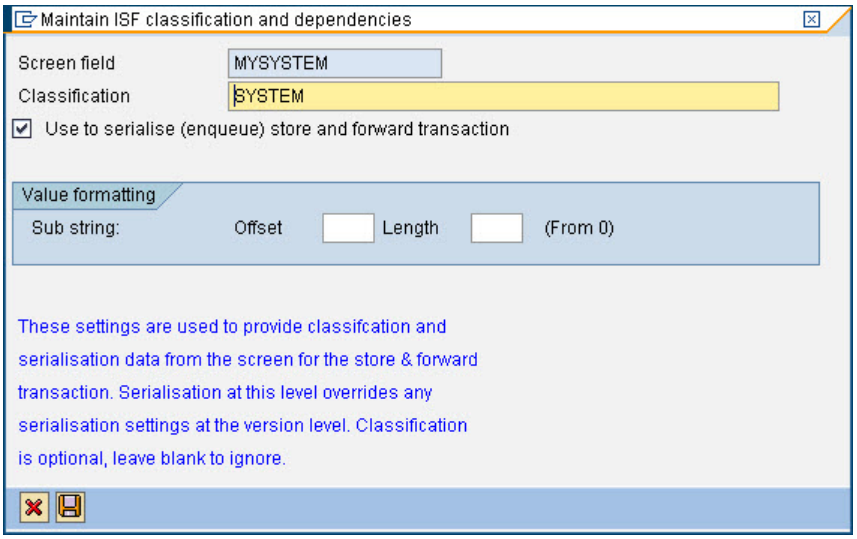
A default value for store and forward is configured in the screen function attributes (hat icon on the application toolbar). This may be overridden at each action definition level. The screen function and the function key must be 'local' to specify store and forward. You may view the current store and forward settings on the function key list of a screen on the right hand side. The store and forward settings for the entire application version may appear using the local attribute view on the main workbench screen.

Serialization and Classification

There is an extra configuration to control the order in which store and forward transactions in the back end SAP system. You may configure configuration to serialize ISF transactions within each SkyMobile session or all ISF transactions (FIFO) on the application version. This determines the level that store and forward processing in SAP is halted if an error occurs. In addition to this, you may classify store and forward transactions for a screen according to screen field values. You may use these fields to

further determine the order of processing and/or the level of enqueue (logical locking) that is to occur.

When store and forward is activated for any screen action, a new control appears to configure specific screen field serialization and enqueue controls:



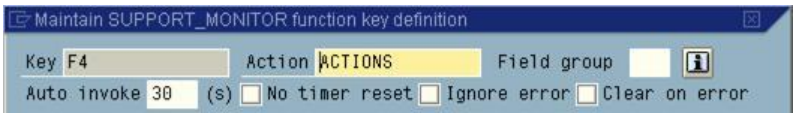
4.7.4 Partial Screen Updates - Auto Refresh

It is possible to instruct SkyMobile to trigger an action automatically after a defined period of time, for example, refresh the screen data every minute. You may apply this to the whole screen or just a group of fields on the screen. You may also define multiple "timer events" to process different parts of the screen. Local database operations and/or custom program exists are associated with each action as per normal.

> ***Note:*** You may reset the timer when there is any screen activity or configure to happen regardless of screen processing.

An example application is to automatically re-display the recorded weights from a number of connected weighbridges on a regular basis, regularly check and display the status of orders, automatically refresh a support issues display, and update a management dashboard. Also refer to actions and events section for more information.

You define the timing and conditions against the screen action/event definition that you want to automatically trigger. For example,



In the above example, the screen action "ACTIONS" automatically triggers every 30 seconds.

4.7.4.1 Auto Invoke

This option automatically triggers the action after 001-999 seconds. The default behaviour is to automatically interrupt the timer on every screen I/O; only executes when the screen is idle. Use the "Don't reset timer" option to ignore this and process the action irrespective of any other screen I/O. You may specify multiple actions with auto-invocation controls on a screen, in this way, you can refresh dynamic values such as times, counters, weights, and database values on a regular basis.

4.7.4.2 NoTimer Reset

By default, the auto execute timer is reset automatically whenever any screen activity (typing, and event) takes place. By checking this option, the auto execution happens regardless of anything else happening on the screen. If the scope of this includes possible input fields, you may loose recently typed in data.

4.7.4.3 Ignore Error

Irrespective of any error that the event processing raises, a ok return code and no message is returned to the caller.

4.7.4.4 Clear on Error (Clear Screen Fields or Field Group)

If the event returns an error, automatically clear the screen fields (or field group). This is designed for circumstances where an auto refresh may fail and no returned value is indicated by nothing).

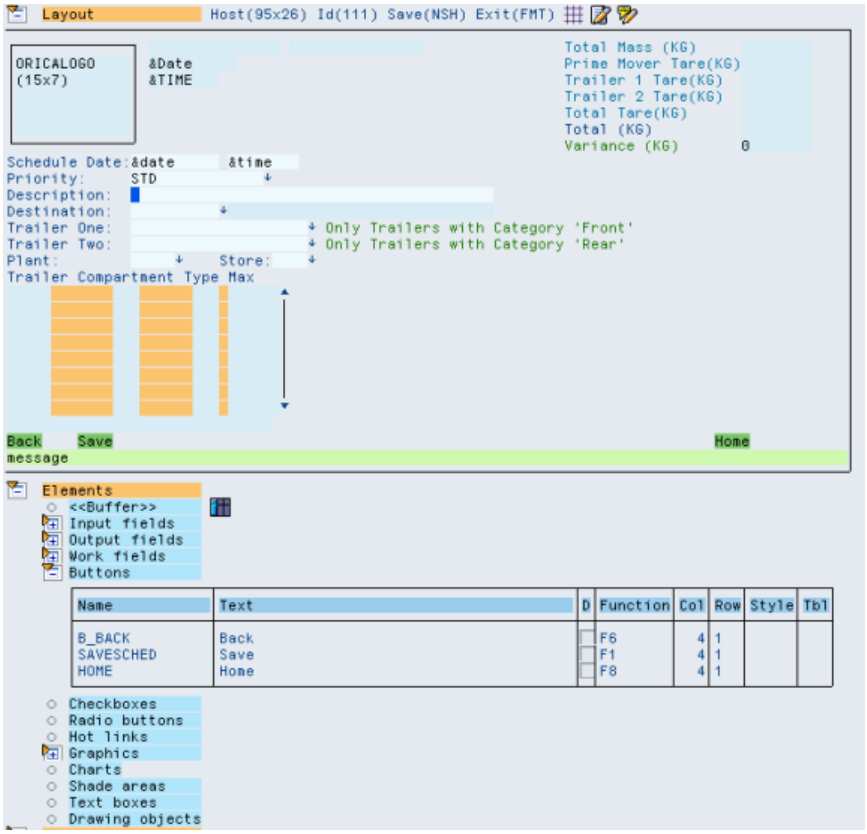
4.7.4.5 Field Group

Restrict the fields to refresh to just those in a specific field group. If left blank, all fields on the screen may be updated. This option is useful to leave portions of the screen alone, whilst updating others, for example, update the total number of orders, whilst retaining the current user input.

4.7.5 Layout Section

The screen layout section is used to define the screen display or map, input. SkyMobile supports an extensive range of screen elements that you may define to provide a rich display. Whilst you may use the same application in many different emulation modes, it is important to understand the limitations of each mobile platform in your overall design, for example, graphics, buttons, hot links, buttons, and hotlinks. The screen painter provides a semi-graphic view of what the screen looks like, depending on the column/line coordinates configured in the version definition. The layout section provides a work area (pallet) with which you may define the screen elements. If configured, the standard header and footer sections also appear. A basic grid and field summary may appear by clicking the grid icon on the screen layout section. You may only define fields in the designated area. A basic colour scheme is used to help differentiate the different screen elements in the layout, for example, graphics, drawing areas have boxes drawn around them and are white if input and pale blue if output; same goes for general fields.

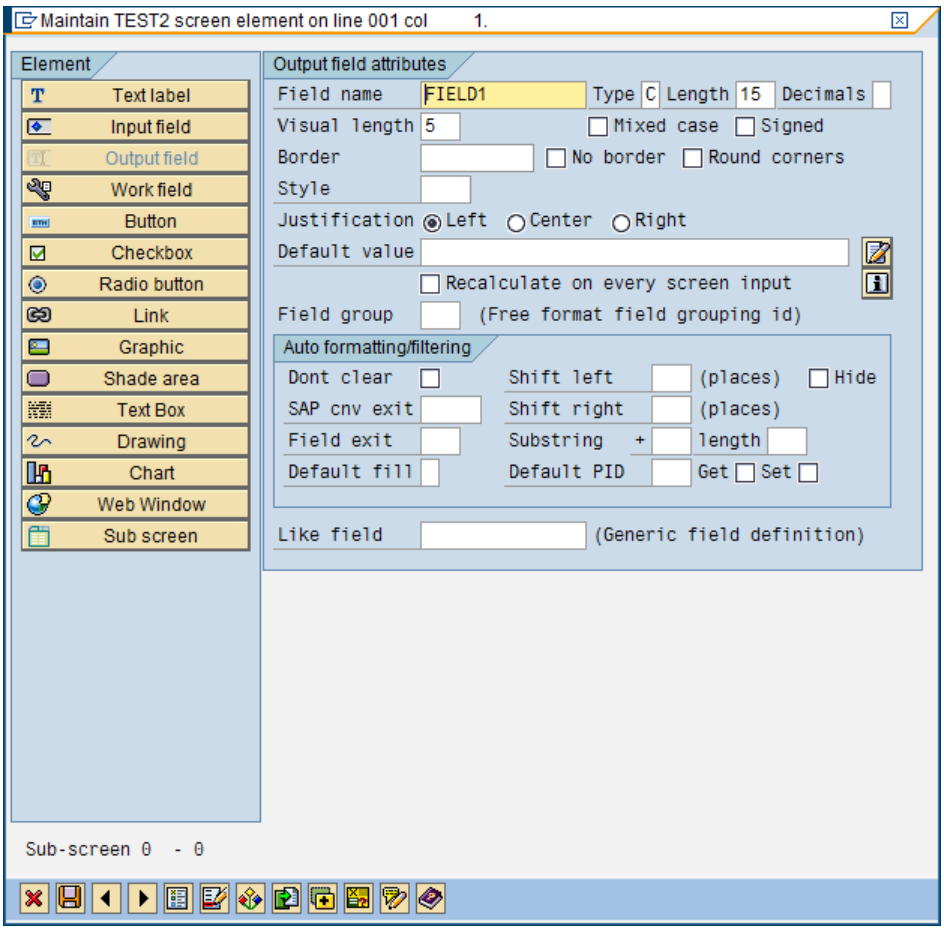
> *Important:* The layout editor provides a near enough representation of the screen. This may vary subtly depending on the platform and the dynamic layout management option used. Sky recommends that you develop in conjunction with actual devices or simulators for best results.



To create a screen element, go into the function (screen) in change mode. "Click" the screen where you want to position the field. Select the create button. This invokes a pop-up to create all of the various screen element types. To maintain a screen field, simply double-click it or position your cursor on it and select the change option.

4.7.5.1 Screen Element Pop-up

When you create or change a screen element (field), a pop-up appears from which you choose the type of screen element and configure its attributes. Simply select the type, using the type buttons on the left hand side and specify the attributes. You may maintain generic graphic, field and style definitions using the associated icons on the application toolbar. You may assign one or more elements to a "field group" that is useful to process a collection of fields in screen event processing, for example, auto refresh operations.

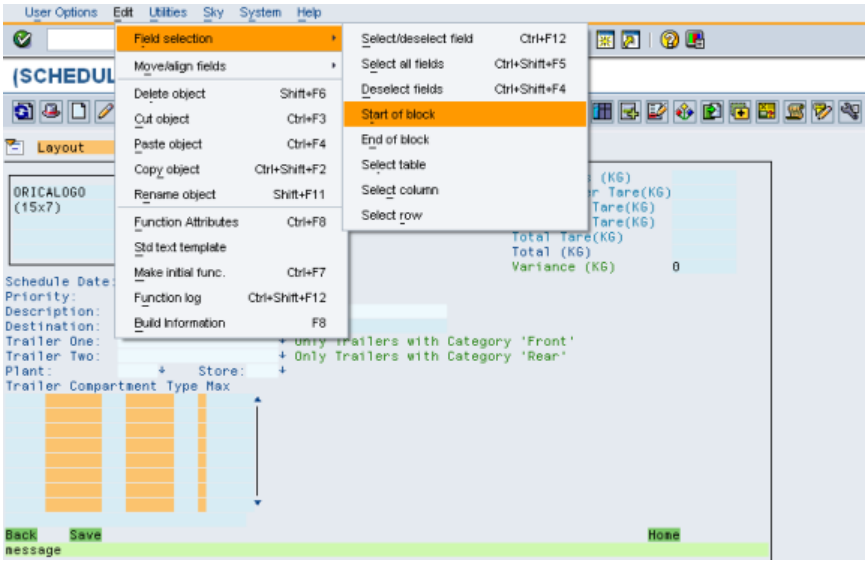


Element Type	Description
Text label	A static text description.
Input field	A field that can accept input characters and/or numeric.
Output field	A field that is display only.
Work field	A non-display field that you may use to store values for processing purposes.
Button	A push button that can perform a function.

Element Type	Description
Checkbox	A tick box providing a simple on/off capability to select an option.
Radio button	A single or defined group of radio buttons provide on/off switches to select options.
Link	Similar to the Web, text that invokes an action when clicked upon.
Graphic	An imbedded binary file image, picture (jpg, gif, and tif). A graphic may also be active, invoke an action when clicked upon.
Shade Area	Colour a specified area of the screen. Mainly used to highlight a group of fields.
Text box	A text box definition.
Drawing	A drawing object is a free format area of the screen that is used to capture a binary image, signature. Controls on how to manage it are also included.
Chart	Imbedded pie, bar, line, charts.
Web Window	An area within the screen within which to execute Html, and Java script. This is only currently supported on the IOS and Android platforms. It allows you to mix both SkyMobile and browser functionality.
Sub-screen	This is a designated area of the main screen within which you can switch between different displays (for example, like a tab control, but far more flexible).

4.7.5.2 Manipulating Screen Elements

Once defined, you may select and move around the screen elements using a variety of ways. The most common is to position your cursor on the element and then click the toolbar icon, for example, move one position left. The edit menu also contains a range of options under field selection and move/align fields.



The following toolbar options are commonly used:

	Copy		Cut
	Paste		Select single item
	Select all		De-select all
	Left align		Move left 1 column
	Right align		Move right 1 column
	Move up 1 row		Move down 1 row

	Switch to previous sub-screen sequence		Switch to next sub-screen sequence

Moving Elements around the Screen

Once defined, you can easily move around screen elements using the arrows on the application toolbar. The cut and paste icons are also useful, just position the cursor on the field, click the cut icon, position the cursor where you want it and click the paste icon.

> ***Note:*** If you move a sub-screen definition, all its related elements are also moved. In addition, you cannot move a sub-screen element outside of the sub-screen boundary.

Selecting Multiple Elements at a Time

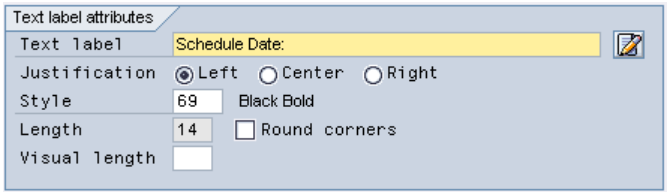
You may select more than one field for processing by using the field selection options on the edit menu. You may select or deselect fields individually using the select/deselect, select all, deselect all icons on the application toolbar. In this way, you may move, delete, cut, copy and paste the multiple fields at a time.

Copying Field Definitions

You may also copy the field definitions by positioning the cursor on the field, clicking the copy icon on the application toolbar, positioning your cursor where you want to copy the field to and clicking the paste icon on the application toolbar. You may also copy field definitions from various other sources, for example, another screen, local database definition, SAP data dictionary, and generic field definition using the field clipboard facility. See the documentation for using the clipboard in the field definitions or clipboard manager sections.

4.7.5.3 Text labels

A text label enables you to add a description (label) to the screen. A text label has no field name and thus cannot be referenced or manipulated by local database operations or program exits. Text labels also cannot be imbedded in table controls. If this is required, use output fields with default values instead.

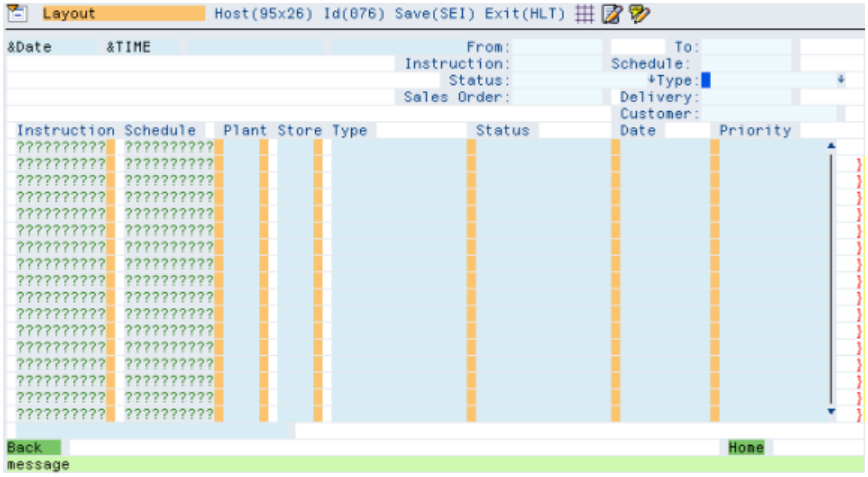


Field	Description
Text label	Descriptive text. Use the extended value icon to enter a long text description.
Left/Right Justification	When the end client will be a GUI interface (for example, WPC, HTML or SAP) you may choose to left or right justify the text.
Style	A style definition to apply to the text, for example, text color.
Length	Indicates the internal maximum length of the text.

Note: You may define dynamic text that is populated at run time, by using an output field with a default value instead, with the 'no border' option specified An output field may also be required to support double byte language options, for example, Chinese.

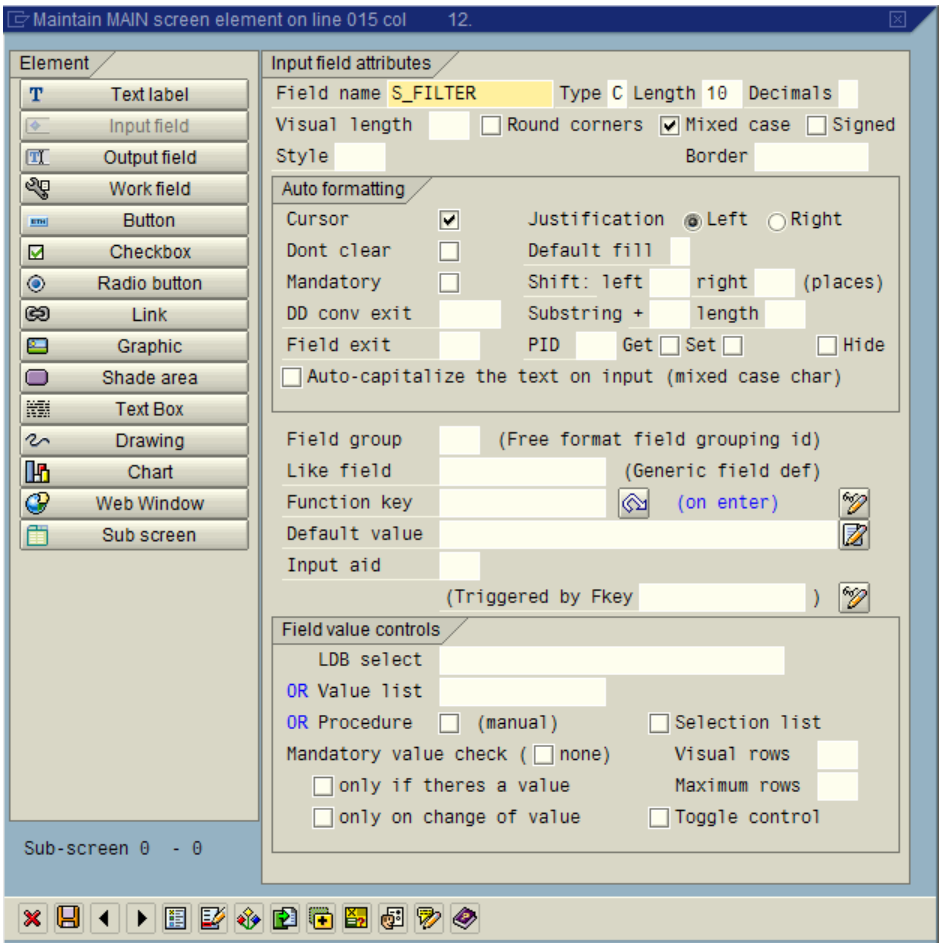
Full Screen Text Input

Creating individual text elements can be a tedious task. By clicking the icon on the layout line, all available free screen space is made available for free format text input. This is a simple quick way to define all the text in a full screen mode. To turn off the option, simply click the icon again.



4.7.5.4 Input fields

An input field allows user input to be captured within a screen. To create an input field, select the **Input** tab/button. The following attributes appear:



Field	Description
Field Name	A unique field name within the screen. ***Note:*** If you plan to perform LDB processing, it is advantageous to give your screen names the same name as the corresponding fields in the LDB. If you do this, SkyMobile attempts to "align" the screen fields to the LDB fields during the definition creation, hence saving time mapping the field definitions.
Type	Field Type. Available field types are: • C - Character field that converts data to upper case. • I - Invisible character field that allows upper/lower case but replaces input with an *. • N - Numeric field. Allowing decimal place and sign. • D - Date field (format as per SAP or operating system default) • T - Time field (format hh:mm:ss) • Special client processing: ◦ '#' - Phone number ◦ '@' - Email address ***Note:*** A (alphabetic) is still supported but is gradually phased out.
Length	The length of the field. There are some restrictions on the length allowed for certain field types: C - 999, I - 999, N - 18, D - 10, T - 8.
Visible Length	The display length of the field on the screen.

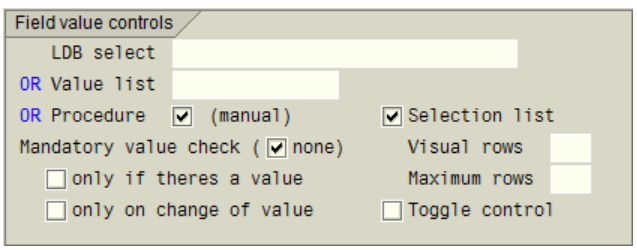
Field	Description
Decimals	The number of decimal places reserved for a numeric field. ***Note:*** The length is inclusive of the decimal places and so the number of decimals must be less that the total length of the field.
Style	A style definition to apply to the content, for example, bold, underline, red.
Mixed Case	Character fields type only. Indicates that the field can contain mixed case, upper and lower case characters. ***Note:*** Select the Input field attribute, **Mixed Case** check box, when you use input field as work-field to launch an URL.
Signed	Numeric field types only. Indicates that the numeric field can contain a signed value.

Field	Description
Auto Formatting:	There are a number of "auto format" options that are automatically performed upon input/output screen processing: • Cursor: Position cursor on this field when the screen is first drawn. • Left/Right Justify • Do not Clear: Ensures that when a "clear screen" is issued, the field is ignored. • Default Fill: enable "padding" of a field with a particular character. • Mandatory: Field must be maintained. • Shift Left/Right: After user input, shift the details x places. • SAP Conv Exit: SAP field conversion exit (SAP host calls only) • Field exit: Call a user exit to format/validate the field. • Substring + length: After input, the input sub-string is edited • PID: Parameter ID processing enables you to save and restore the field value. Once defined you can "set" the value or "get" the value. This allows a field to be "defaulted" to the last value for a particular user.
Field Group:	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout e.g. auto refresh.
Like Field:	Reference your field definition to a global definition. (Generic fields are detailed below).
Simulate Fkey:	When the cursor is positioned within this field, and "ENTER" is pressed, a function key can be processed. This is particularly useful in VT220 processing with RF scanning. For example the RF scanner can be configured to send an ENTER at the end of the scanned input. Therefore, if an F1 was entered in this configuration option, F1 would be triggered by the RF scan and hence any processing associated with F1.

Field	Description
Default Value:	A default value can be assigned to the field. There are a number of internal variables that may be automatically substituted at run time. For example, &TIME, &DATE. Use the extended value icon to enter a long default value. Hints and tips are available for default values. Refer to the Default Values section for more information.
Input Aid:	A number of standard input aids are available to assist data capture. For example, a numeric keypad to assist with the input of numeric values, calculator, and digital (capture signature). ***Note:*** Instead of an icon appearing beside the input field to trigger the input aid, you can implement your own graphic that triggers a function key and specify the function key in the "triggered by function key" field. This is useful for applications where larger buttons and graphics are required. Refer Input Aid for more information.
Field Value Controls	This set of configuration enables SkyMobile to automatically restrict/check or suggest a range of values, and/or make them available as a toggle or drop down box. Click here for more details.

Value Checking Selection Lists and Toggle Controls

When configuring input fields, you can automatically check, and/or suggest, a value from a list of data. The list may be associated with either a local database, or a static value list. All these mechanisms are designed to assist the user capture and validate data. The following configuration on the input field attributes controls the value checking and selection mechanisms.



Field	Description
LDB select, value list or procedure	You may nominate either an LDB (local database), a static value list, or through a procedure to populate the list of available values. When populating through a procedure, you need to add the values to the fields selection list using the "append value list" command. By default, SkyMobile checks that the value entered into the field matches a value on the list.
Mandatory value check	These options determine whether the data value check if only performed if there is a value, don't allow no-value and whether the value check is only performed if the value changes. You may deactivate the mandatory value check using the 'No mandatory value check' option.
No mandatory value check	Specify this option if you just want to provide a drop-down selection list, but allow the user to enter new values that don't occur on the list.
Selection list	Instructs SkyMobile to enable a 'drop down' list capability for the values, so the user can select one. You may also configure the maximum number of values to show at a time in the list (scrollable) and the maximum number of rows to be selected for the selection list.
Toggle control	As apposed to a 'drop down' style of list, a toggle control automatically populates the field with the next available value. When the end of the list is reached, the first value is returned.

Including other screen fields to qualify the selection list

The LDB configuration automatically generates the LDB function required based on a one to one screen field / LDB field relationship. You may then update this to include other fields as required and even return additional data vales. Once generated, the LDB function is not overwritten, thus you may customize it. See the section on local database screen functions for more details.

> ***Important:*** You cannot mix any other LDB operations with the generated LDB operation on the same field event. You may however configure other LDB operations on a triggered screen event.

Prompt for screen field / LDB field map

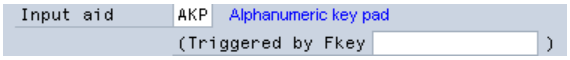
The workbench automatically maps the input screen field to an existing LDB field of the same name. If none exists, a pop-up appears to enter one.

Input Aids

There are several built in input aids that can assist users in data capture operations. Some of these are automatically implied, such as date and time selection pop-ups, and others you must explicitly configure, such as a calculator. Sky is adding new input aids all the time; use the drop down to see what is available. The following are some samples.

> ***Important:*** Some input aids, for example, date/time are associated automatically. If you do not want a default input aid to be associated with a field, explicitly specify a "NON" (no input aid) value.

> ***Note:*** You must check to see if the presentation client you are using supports the input aid that you intend to use. Refer to the client capability matrix for more details.

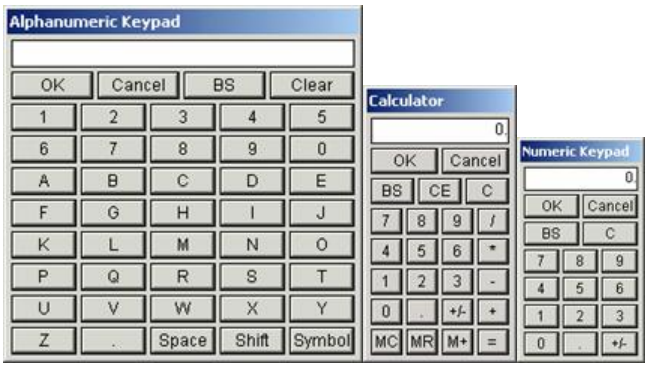


Date/Time fields:

If a screen input field is configured as type D (date), SkyMobile automatically associates a popup option beside the field. If this is pressed, a selection pop-up is invoked. For type T (Time) fields, tick the imbedded change value box and select the hour, minute or second value. You may now use the up down arrows to change.

Optional data capture aids:

There are built-in input aids to assist data capture. A specific input aid is activated through the 'input aid' selection on the screen input field attributes. For example, you may associate a calculator with a numeric quantity field. When an input aid is used, an icon appears next to the relevant field that displays a popup input aid window when pressed. For example:



The following optional input aids are available:

Input Field	Description
AKP	Alphanumeric key pad
CAL	Calculator
DAT	Date calendar invocation (date fields only)
FIL	File name selector (explorer). Navigate and list the local file system and select a file name to return to the field. The full path name is returned.
INC	Numeric value incrementor/decrementor
NKP	Numeric key pad
NON	No input pad

Input Field	Description
QTY	Qwerty style key pad
RTE	Rapid Text Entry. This facility is designed for the rapid scanning and capture of information into a list that is then returned to the application for processing, for example, high speed scanning of barcodes in a warehouse. You may configure a default delimiter in the associated configuration section that is defined in the main presentation client configuration.
TIM	Time HH:MM:SS entry
TMD	Time duration capture

Launching/Triggering input aids

You may also launch input aids from other screen events, for example, graphic. This may be useful in situations where the input aid icon is not big enough for easy selection, for example, industrial touch screen. In this case, you may use the "triggered by fkey" option to indicate that the input aid for the field is actually triggered by another screen object, for example, button, graphic, and link. This gives you flexibility to define the type and position of the field to use as the "trigger".

Controlling the display and layout of input aids

Each input aid has a corresponding configuration section in the CLIENT section of the configuration file. This contains all the options available to both control the display size and behavior of the aid. For example:

Rapid Text Entry Client Configuration

```
CLIENT.INPUTAID-RAPIDTEXT.FONTNAME =
CLIENT.INPUTAID-RAPIDTEXT.FONTSIZE =
CLIENT.INPUTAID-RAPIDTEXT.FONTWEIGHT =
CLIENT.INPUTAID-RAPIDTEXT.BUTTONPADDINGHEIGHT =
CLIENT.INPUTAID-RAPIDTEXT.BUTTONPADDINGWIDTH =
CLIENT.INPUTAID-RAPIDTEXT.BUTTONSPACING =
```

```
CLIENT.INPUTAID-RAPIDTEXT.WINDOWHEIGHT = 200
CLIENT.INPUTAID-RAPIDTEXT.WINDOWWIDTH = 150
CLIENT.INPUTAID-RAPIDTEXT.ALLOWDUPLICATES = false
CLIENT.INPUTAID-RAPIDTEXT.INPUTDELIMITER = @
CLIENT.INPUTAID-RAPIDTEXT.OUTPUTDELIMITER = ,
CLIENT.INPUTAID-RAPIDTEXT.SOUNDFAILURE =
CLIENT.INPUTAID-RAPIDTEXT.SOUNDSUCCESS =
```

Important: See the input aid section in the client configuration section for more information.

Default Values

You may specify combinations of static and dynamically substituted values. If you require a longer default value than the screen allows, click the text button at the right hand side of the default value to display a full text entry window:

You may specify the following substitution variables:

Variable	Description
&amp/&ampersand	The value '&'
&appid	The current SkyMobile three digit application Id
&appstamp	The application timestamp of the application currently running
&build	The application build info
&clienthost	The Application Server host name
&clientip	The Application Server TCP/IP address
&clienttype	The type of client (APC, BPC, IPC, WPC)

Variable	Description
&cldate	The current client date
&cltime	The current client time
&clutc	The client's UTC offset
&date	Current date in the default format
&deviceid	The unique 14 digit device identifier
&devicemake	The device hardware manufacturer
&devicemodel	The device model
&devicenumber	The three digit device number
&devid	The three digit device number
&function	The current SkyMobile function name
&hostutc	The SAP system UTC offset
&language	The current language configured in the header attribute
&os	The client operating system
&osversion	The client operating system version
&p1 - &p8	The p1 to p8 parameters from the user session
&platform	The logical platform the client has been allocated to
&plant	The header plant value

Variable	Description
&serverid	The Application Server ID
&servergroup	The Application Server group
&serverutc	The application server UTC offset
&sessionid	The current client session ID
&tenant	The tenant of the current user
&time	Current time in hh:mm:ss format
&timestamp	Current timestamp in yyyymmddhhmmss format, 14 character
&title	The current screen functions title
&userid	The header user ID
&verid	The current SkyMobile three digit version ID
&workarea	The header work area value
&workgroup	The workgroup of the current user
&(field name)	Screen field value, for example, &(order_number).
=(section.key)	Parameter value, for example, =(salescfg.salesorg), =(&(region).salesorg).

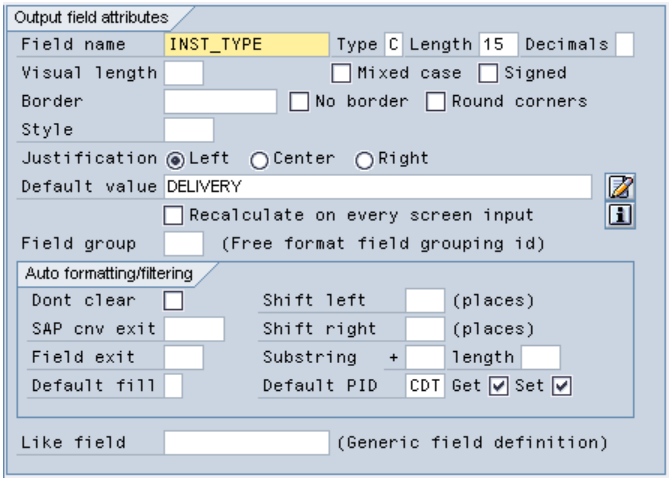
If you want to re-assess the default value each time the screen refreshes, then specify the recalculate default value option.

4.7.5.5 Output Fields

If you want the default value to be re-assessed each time the screen is refreshed, then specify the recalculate default value option.

Output fields enable you to display data that cannot be modified. Select the **output** button.

> ***Note:*** The attributes and auto formatting/filtering are the same as input fields, only the differences are listed here.

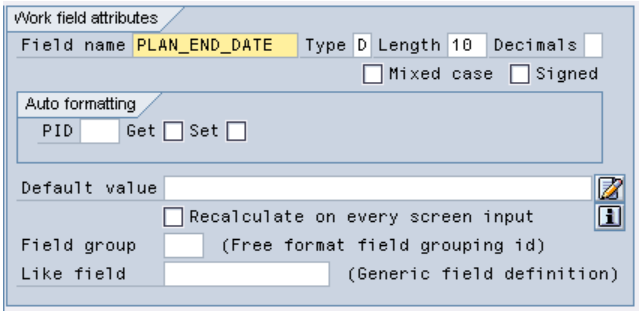


Field	Description
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
No border	Suppresses the displaying of a field border around the field, similar to a text label
Default value	You can assign a default value to the field. There are a number of internal variables that you may automatically substitute at run time. Example, &TIME, &DATE. Use the extended value icon to enter a long default value. Hints and tips are available for default values. Refer to the Default values section for more information.

Field	Description
Recalculate default value	As well as specifying a static default field value, or internal substitution variable, you may instruct SkyMobile to default the field on every screen input, for example, &TIME then automatically updates the time on every screen output operation.

4.7.5.6 Workfields

Work fields enable you to create non-display screen fields that are used to store "working data" that you do not want to display on the screen. Examples of this are: user IDs, plant, date, time, server ID. You may require these types of values for LDB and/or programming exit processing. The listed attributes are the same as input/output fields. There is a limit of 999 work field definitions. If more are required or you require storing structures or table definitions, use the export/import data program APIs.



Note: To maintain a "work" field, expand the list of fields available beneath the screen to show the list of work fields and double-click the field you wish to maintain. You can maintain all field types in this fashion.

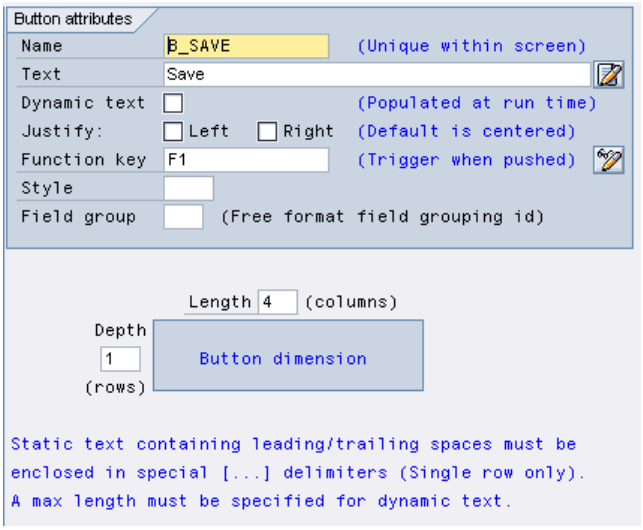
Field	Description
Type	Field Type. Available field types are: • C - Character field that converts data to upper case. • N - Numeric field. Allowing decimal place and sign. • D - Date field (format as per SAP or operating system default) • T - Time field (format hh:mm:ss). ***Note:*** The same length, decimals, mixed case and signed attributes apply as input and output fields. Because work fields are non-display, type I (invisible) is not applicable.
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Default value	You can assign a default value to the field. There are a number of internal variables that you may automatically substitute at run time. For example, &TIME, &DATE. Use the extended value icon to enter a long default value. Hints and tips are available for default values. Refer to the Default values section for more information.
Recalculate default value	As well as specifying a static default field value, or internal substitution variable, you may instruct SkyMobile to default the field on every screen input, for example &TIME then automatically updates the time on every screen output operation.

How Work Fields are Handled

Work fields of the same name are automatically copied between screen functions, in the same manner as input and output fields. You may also imbed work fields into table controls and thus store on each row of the table. In fact, an entire table may be made up of only work fields and thus never display on the screen. Work fields are made available to custom programs in exactly the same manner as input fields.

4.7.5.7 Buttons

You can configure push buttons onto your screen. Note that while they display in VT220 mode, they are aimed at a GUI interface, such as HTML, WPC, Blackberry or SAP. To create a button, select the "button" tab or double-click an existing button definition.

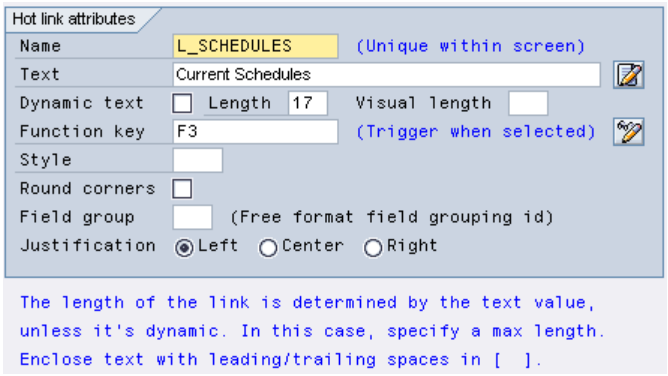


Field	Description
Name	The name of the button. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "b_".
Text	Any text to display on the button. Yo can padd with spaces by defining the text within square brackets, for example, [Push Button] pads the button (hence making it larger) with spaces. Use the extended value icon to enter a long text value
Dynamic Text	The text on the button can be dynamic; populated as part of the screen processing. To enable this process, select this box and configure the max length of the button. ***Note:*** If you configure the button for dynamic text, but do not populate this text (usually through screen exit or LDB function) the button does not appear (as it as a text length of zero).

Field	Description
Function Key	When you press the button, a function key triggers. All required processing then associates with the function key event.
Length	Button length. You must maintain for dynamic text. SkyMobile calculates it if the button is not dynamic text.
Depth	The depth in rows of the button. This value defaults to 1 for normal buttons but if you require larger buttons, specify a larger depth.
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Style	A style definition to apply to the text, for example, background, foreground text colours.

4.7.5.8 Links

Links enable a string of text to be active, and behave like a hyperlink. To create a "link", select the link button. Links are very useful when processing in a handheld device, where navigation is through a stylus.



Field	Description
Name	The name of the link. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "l_".
Text	Any text to display on the link. You can pad this with spaces by defining the text within square brackets, for example, [Link] pads the link text (hence making it larger) with spaces. Use the extended value icon to enter a long text value.
Dynamic Text	The text on the link can be dynamic, populated as part of the screen processing. To enable this process, select this box and configure the max length of the link. ***Note:*** If you configure the link for dynamic text, but do not populate this text (usually through screen exit or LDB function) the link does not display (as it as a text length of zero).
Length	Link length. You must maintain for dynamic text. Workbench calculates it, if the link is not dynamic text.

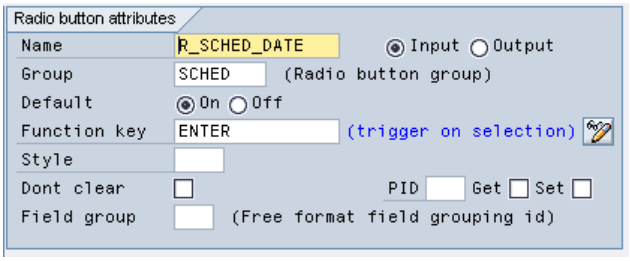
Field	Description
Function Key	When you press the link, a function key is triggers. All required processing is then associated with the function key event.
Field Group:	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Style	A style definition to apply to the text, for example, text colour.

Note: You may present a list of links easily by placing a link field in a table control and specifying the dynamic text option. Values may be populated directly into the table from a local database or programming exit. The same can be performed using buttons.

4.7.5.9 Radio buttons

You can add radio buttons by selecting the "Radio" button, like push buttons they are aimed at a GUI interface, and not text terminals. You may place radio buttons anywhere on the screen, need not be contiguous.

When you place inside tables, radio buttons can behave in two different ways. If there is one radio button in a radio button group, then the radio buttons act as 'row selectors', when the radio button in one row is selected, the radio buttons in the other rows are cleared. However, if multiple radio buttons are added to a radio button group, then the radio buttons toggle independent of any other row.

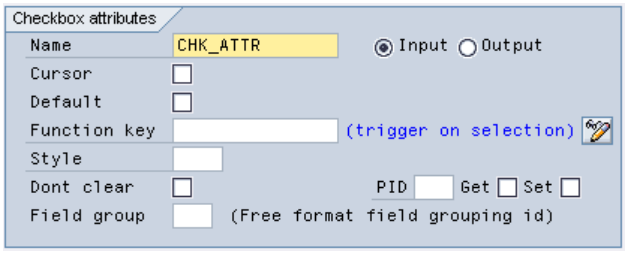


Field	Description
Name	The name of the radio button. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "r_".
Input/Output	Configure whether its output only or allows input.
Group	Often you configure a number of radio buttons (each with a different field name) in a "group" where you can select only one. Within screen processing, you have only a single radio button active at any time within a group.
Default On/Off	Configures the default state for the Radio button. You may default only one radio button for the group.
Function Key	When you press the radio button, a function key is triggers. All required processing is then assigned to the function key.

Field	Description
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Style	A style definition to apply, for example, colour.
Do not clear	Explicitly bypass the field when clearing the screen
PID	Get/configure the default value using a declared field parameter ID.

4.7.5.10 Checkboxes

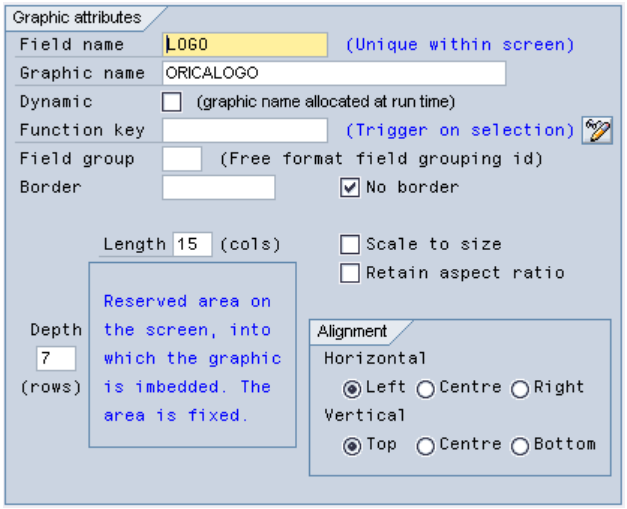
A checkbox (tick box) is an on/off option. It is a single byte field that is blank for off and 'X' (true) for on.



Field	Description
Name	The name of the radio button. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "x_".
Input/Output	Configure whether its output only or allows input.
Cursor	Position the cursor on this field, by default.
Default On/Off	Configures the default state for the check box.
Function Key	When you press the checkbox, a function key is triggers. All required processing is then assigned to the function key.
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Style	A style definition to apply, for example, colour.
Do not clear	Explicitly bypass the field when clearing the screen.
PID	Get/Set the default value using a declared field parameter ID.

4.7.5.11 Graphics

An imbedded graphic is effectively a binary picture in jpg, gif, tif, png, bmp format. You use the binary object manager to import graphic files into SAP, and these are distributed to the Application Servers. A graphic may be fixed or dynamic, meaning, you may populate it at run time from a LDB or screen exit. You can associate the graphic with a function key, trigger an action when you click it upon. The following attributes are available for graphics. You may produce graphics easily using most drawing tools, for example, MS-PowerPoint; however for best results, you should use a professional tool, such as Paint Shop Pro, or that a graphic artist provides.

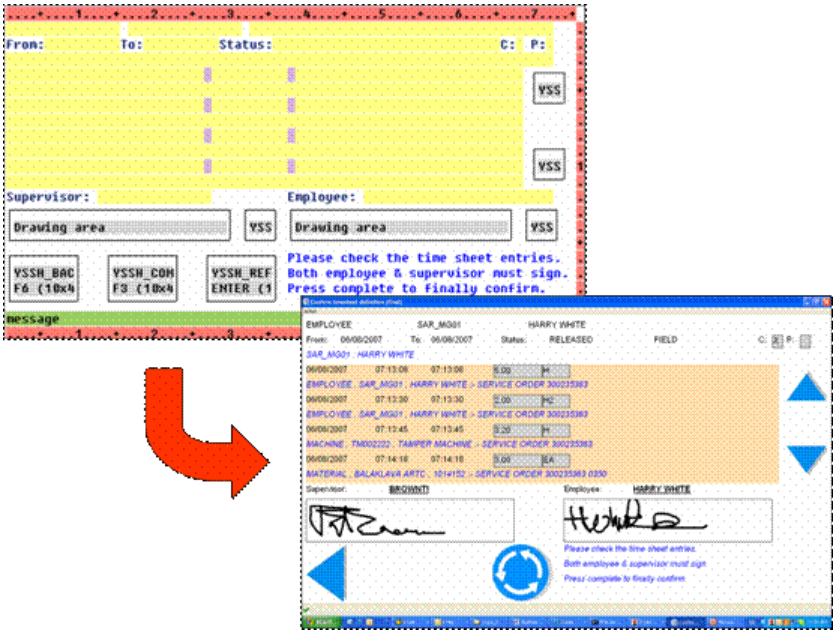


Field	Description
Field name	The name of the graphic in the screen definition. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "g_".
Graphic name	The name of the binary object to display. This is a static reference that is used as the default, if the name is dynamic. Graphics are loaded into SAP through the binary object manager. The logical name they are stored as is the name used here. ***Note:*** You can use only graphic binaries stored as version 0 in screens.

Field	Description
Dynamic	The binary graphic file name to use can be dynamic; populated as part of the screen processing at run time.
Function Key	Flags the graphic as active; when you press the graphic, a function key triggers. All required processing is then assigned to the function key.
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Length	The horizontal length (columns) of the graphic on the screen.
Depth	The vertical length (rows) of the graphic on the screen.
Formatting and alignment	These attributes govern how SkyMobile is to render the image on the screen: • Scale to size -perform a best fit, depending on the length and depth of the graphic area. The image may be shrunk or stretched as a result. • Retain aspect ratio - the aspect ratio is the relationship between the width of the picture and the height of the picture. By retaining the aspect ratio, the system attempts to retain the original height/width ratio as best it can. • No border - do not display a border around the graphic • Horizontal / Vertical - SkyMobile displays the graphic as is in the space provided. The top, center and bottom configurations determine how the image is aligned if it is smaller than the designated space.

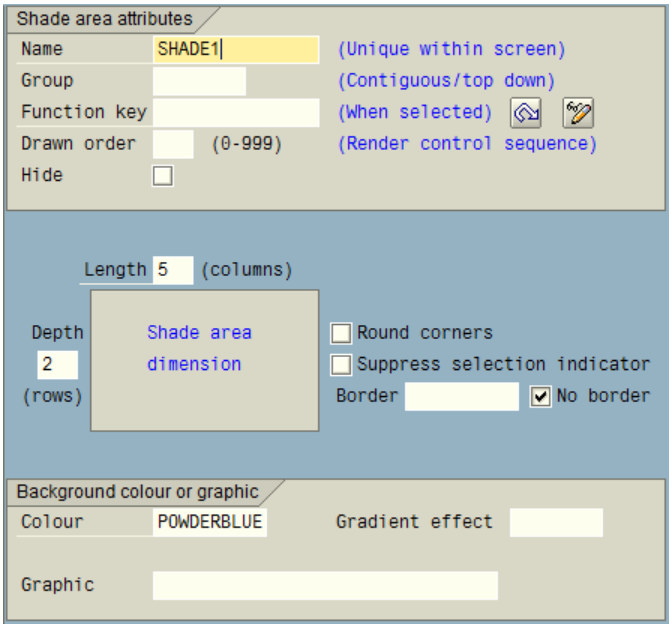
Note: SkyMobile reserves an area on the screen based on the configured length and depth. These settings are fixed, you cannot dynamically determine at run time. No other field may share this space.

You should test the design and display of graphic images should by running the application on a Application Server with a graphic presentation option, for example, WPC as you work. The actual graphic representation may subtlety vary from the more basic workbench display.



4.7.5.12 Shade Areas

A shade area highlights the specified area of the screen with a certain colour. These are useful to highlight a group of screen elements. You may use shade areas effectively to present and select complex data. The following attributes are available:



Field	Description
Field name	The name of the shade area in the screen definition. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "s_".
Group	If a shade area group is specified, then the list of associated shade areas is treated as a contiguous top down list (similar to a table). This is useful to present more complex layouts where the entire group of attributes may be selected.
Drawn order	This is the order in which shade areas will be drawn and overlayed on the screen.

Field	Description
Function key	You may assign a function key to be activated when anything within the shade area is pressed. If an element within the shade area has its own function key, then it takes preference.
Length / Depth	The horizontal length (columns) on the screen. The vertical length (rows) on the screen.
Round Corners	By default, shade areas have square corners. This option draws rounded corners and is useful for 'softer' displays.
Suppress selection indicator	On many platforms, a symbol, for example, '>' is used to denote further navigation. Check this option to ignore any default selection indicator displays.
Border	The no border option suppresses a 'box' border from being written around the shade area. Otherwise, a box is drawn and you may specify a colour.
Colour	The background colour of the shade area.
Gradient effect	The direction of a gradient if desired.
Graphic	The name of a binary object to be used as a background image.

Because a shade area is essentially 'background' on a screen, some special processing rules apply:

- The first row/column position is reserved; you can position nothing here. A small 'box' symbol is written here by the screen painter.
- If you double-click the body of the shade area, the screen painter assumes that you wish to maintain a screen element within the shade area. To position on the shade area definition itself, double-click the 'box' symbol in the top left hand corner of the shade definition.

- When shade areas are overlayed, they are drawn in accordance with the "Drawn order" field. If the drawn order is not specified, the assumed sequence of overlay is the same as defined on the screen painter.

4.7.5.13 Text Boxes

A text box enables you to display and maintain a large description on the screen with automatic wrap around and scrolling. The screen control allows you to configure the dimensions of the text box.

? Unknown Attachment

Field	Description
Field name	The name of the text box in the screen definition. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "t_".
Style	A style definition to apply, for example, background/foreground colour.
Input/Output	Whether text input is allowed or its display only.
Max bytes	The maximum number of characters that can you can input into the text box. Anything over this is truncated. ***Note:*** A text box definition may be larger than the displayable area on the screen. If you specify "0", the size is unlimited.
Vertical scroll bar	If the actual text is more than the displayable size, an optional scroll bar may appear to assist scrolling.
No border	Suppresses the drawing of a border (box) around the text area.
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example,auto refresh.

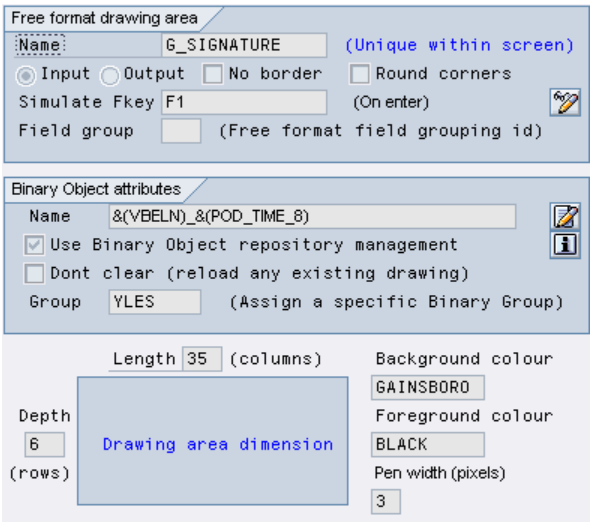
Field	Description
Input controls	These attributes configure the input behaviour of the text box and are identical to those of a normal input field, for example, you can set the default cursor position, make it mandatory, utilize an input aid. Configure a default value. Refer to input fields for more details on these options.
Length	The horizontal length (columns) on the screen.
Depth	The vertical length (rows) on the screen.

The text field is treated in exactly the same way as a character field; you may use it in database operations and screen processing exits.

SkyMobile reserves an area on the screen based on the configured length and depth. These settings are fixed; you cannot dynamically determine at run time. No other field may share this space.

4.7.5.14 Drawing Objects

This is a free format area of the screen used to capture a binary image, for example, a signature. The drawing object also has controls on how to manage the captured image; create a file and/or synchronise it with the binary manager. See the section for more information on how to manage the uploading and downloading of drawing objects.



Field	Description
Field name	The name of the drawing object in the screen definition. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "d_".
Input/Output	Whether the drawing object can be input or is display only.
No border	Suppress drawing a border around the drawing area.
Fkey	If specified, the drawing is not processed until you initiate this function key. If you do not specify this, the drawing object is processed on every screen I/O. Drawing object processing is performed after LDB and screen exit processing.

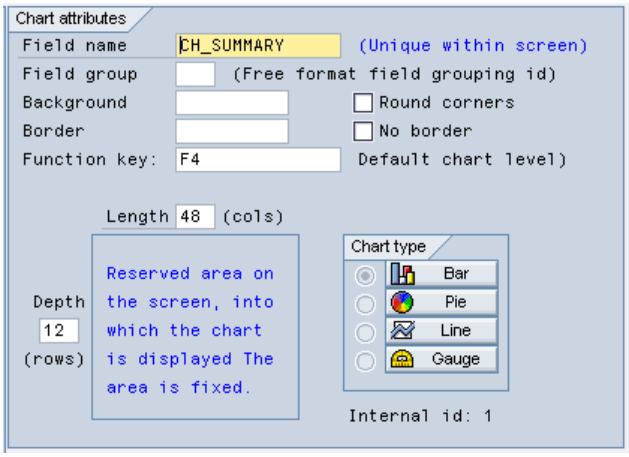
Field	Description
Field Group:	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Obj name	This is used as the file name and binary object name (see below). If in local mode, the DOCUMENTS directory is used. If it is a host function, the default working directory (as per the system defaults) is used. You may specify multiple dynamic substitution variables in the name in the same way as other default values, for example, &DATE, &TIME, &SERVERID. You may also reference screen field values by using the &(field name) notation. If using the binary obj repository, the name is limited to 32. Use the extended value icon to enter a long value. Hints and tips are available for using drawing objects.
Use the binary object repository	This option will automatically invoke the binary file manager to store the binary drawing object data. When processing locally, the data is written to the file name specified and this is automatically synchronised (uploaded) to the host, where it is stored centrally in the binary object database using the unique file name. If the screen function is host only, the drawing object is loaded directly into the central binary object database. The object name is used both as the file and the binary object name and is limited to 32 characters.
Group	By default, the binary group specified in skymobile.cfg or on the version is used. A specific alternate group may be configured here. The group specified here may only be 8 characters long. This is essentially the group that captured binary images will be assigned to in the binary object manager. Groups are used to better organise and segregate drawings. The binary object group cross reference mechanism is used to determine how drawing objects are synchronised with servers.
Don't clear	If specified, the drawing area will not be initialised and any existing image will be re-loaded and/or re-displayed.
Length	The horizontal length (columns) on the screen.
Depth	The vertical length (rows) on the screen.

Field	Description
Background/ Foreground colour	Enable you to control the background colour of the drawing area and the colour of the "ink" or pen.
Pen width	Controls the thickness of the drawing pen.

The drawing object is internally compressed and stored as a bit map (bmp). You may control the processing of drawing objects programmatically within screen processing exits. You can test a special indicator "drawn in" as to whether a drawing is actually made. See the ABAP and Java programming guides and the binary object section in this guide for more details.

4.7.5.15 Charts

This control allows you to embed charts directly into screens, for example, Pie, Bar, and Line graphs; gauges. These allow you to implement basic analytics. You may specify multiple charts on a single screen. When a chart screen element is maintained, an equivalent table control and work attributes are also created. These are used to store the chart data and the behaviour of the chart definition. See the full description of the internal tables / attributes below.

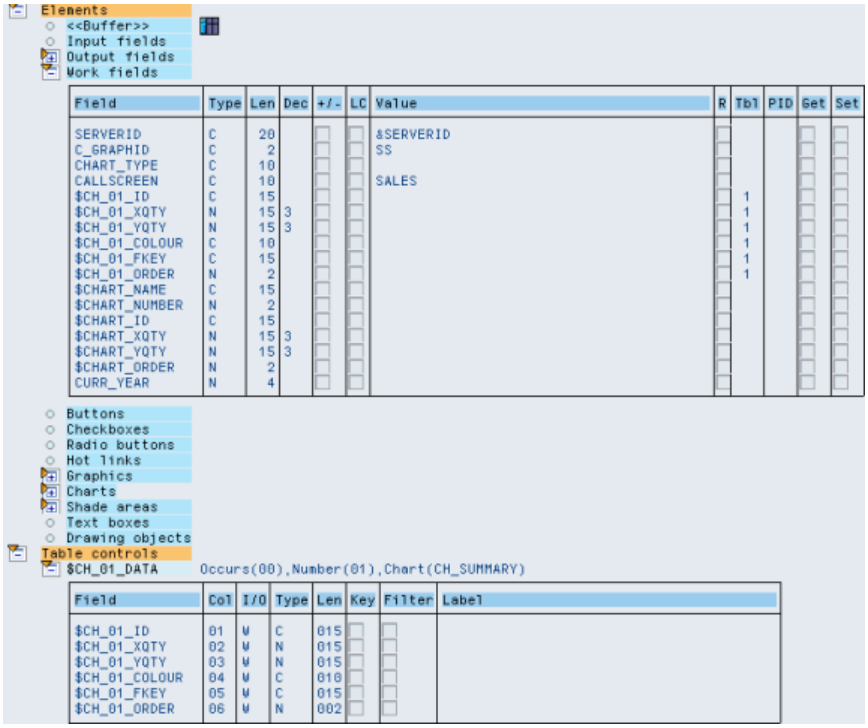


Attribute	Description
Field name	The name of the chart object in the screen definition. Must be unique within the screen. ***Note:*** Sky recommends a name prefixed with "c_".
Field Group	Associates the field with a collection of other screen elements to help restrict operations and events to only certain definitions on a screen layout, for example, auto refresh.
Background	The background colour of the chart area.
Round corners	Use rounded corners for the chart background area.

Attribute	Description
Border	The 'no border' attribute suppresses drawing a border around the chart area. Otherwise a border is drawn and you may configure the colour.
Function key	Enables a function key trigger to define against the entire chart area; you may click/touch anywhere on the chart. You may also configure function keys to trigger against the chart elements/points by specifying these in the internal $CH_nn_DATA table.
Length / Depth	The horizontal length (columns) on the screen.
Depth	The vertical length (rows) on the screen.
Chart type	This is the type of chart that is dynamically drawn in the area provided.

Internal Work Areas

Whenever you maintain a chart control, internal work fields and a table control are also automatically maintained for you. You use these to populate the data and control the behaviour of the chart definition. Each chart is allocated a unique internal id, for example, 01. This is used to allocate a unique table control (data table) and work fields starting with '$CH_' and '$CHART'. In the following example, you can clearly see the internal work fields starting with '$CH' and the internal table control that are generated for chart 01.



$CH_nn_ ID	The unique ID of the chart segment/bar/line node. This is also used to identify the legend.
$CH_nn_ XQTY $CH_nn_ YQTY	The X/Y quantity used to define and draw the chart area. The Y quantity is populated in all cases. The X quantity is populated for line charts.
$CH_nn_ COLOUR	The colour to be used when drawing the chart segment / line. If left blank, a unique colour is automatically assigned.
$CH_nn_ FKEY	The function key to trigger if you click/touch this part of the chart.
$CH_nn_ ORDER	The order in which to draw the chart areas. If left blank, the default is used to use the order of the table data.

The $CHART.... Internal work fields are automatically populated when a chart is selected; you can analyze these in a procedure / custom exit program.

Populating Data

To specify the chart data, you simply add rows to the $CH_nn_DATA internal table, specifying the unique ID and other attributes as required. You may do this directly from a LDB action, procedure, or custom exit in the same way as any other screen table.

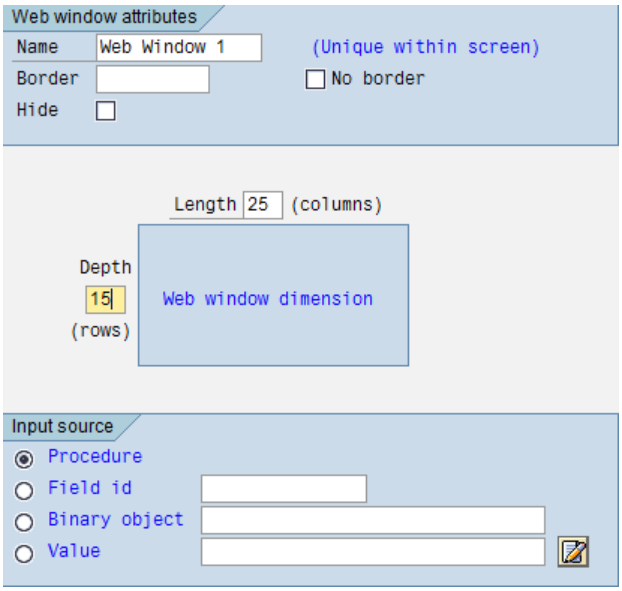
You can configure the value for the "needle" in a gauge chart by assigning a value to the $CHART_nn_ZQTY field in the $CH_nn_DATA internal table.

4.7.5.16 Web Windows

These are fixed areas of the screen function that are designated to display browser input; Html, Java script. You typically use them to execute embedded web applications or more fancy text labels using Html. The idea is that you define one or more web windows on the screen and then feed in html, Java script into it through a procedure, field value, binary object or static value (default). This enables you to effectively mix SkyMobile and browser functionality within SkyMobile applications all running within a SkyMobile secure container.

> ***Note:*** Web Window functionality is only currently supported on the IOS (IPC) and Android (APC) platforms.

To create a Web Window definition, select the Web Window tab/button. The following attributes appear:



Option	Description
Name	A unique name for the web window within the screen function

Option	Description
Border	The no border option suppresses a 'box' border from being written around the shade area. Otherwise, a box is drawn and you may specify a colour.
Hide	This defaults the entire web window to hide when the screen is initialized.
Length/depth	The horizontal length (columns) on the screen and vertical length (rows) on the screen of the reserved sub-screen area.
Input source	This is where the web window gets its input; Html, and Java script. • Procedure: a special type of procedure that allows you to edit a list. It is more suitable for more complex scripts. • Field id: input is from a SkyMobile field. Useful for URLs that you may pass in from a table, dynamically created. • Binary object: input from a binary object in the SkyMobile binary repository • A fixed default value, for example, a URL

Substituting Values

In all cases of input, standard SkyMobile variable substitution is performed on all &, &(...) and = (xxx.xxx)" type values before control is passed to the browser.

Interfacing between SkyMobile and Web Windows

Internal java scripts are provided that enable you to get/set data values and trigger events (fkey's):

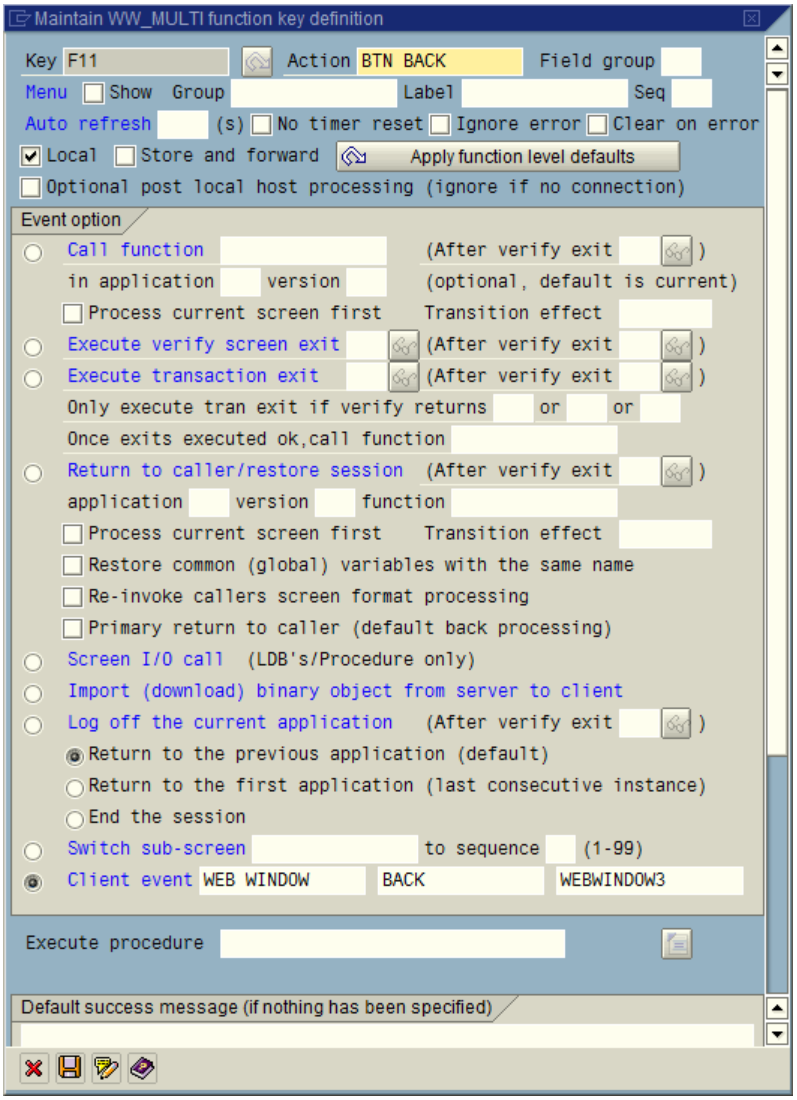
- Sky.getFieldValue("xxxxx"); where xxxxx is the SkyMobile screen field to retrieve the value from for example, var input = Sky.getFieldValue("INPUT");
- Sky.setFieldValue("xxxxx", yyyyy + "!"); where xxxxx is the SkyMobile screen field to insert the value into and yyyyy is the html variable to take the value from, for example, Sky.setFieldValue

("INPUT", work + "!");

- Sky.trigger("xx"); where xx is the SkyMobile screen trigger/fkey, for example, Sky.trigger("F1");

Browser Controls and Limitations:

The whole concept of the secure container web window is to limit the capability of the browser so that users cannot take control of the application. By default, the SkyMobile web windows are totally restricted and you have options to implement your own back, forward, and refresh controls as "client events" associated with graphics/buttons.



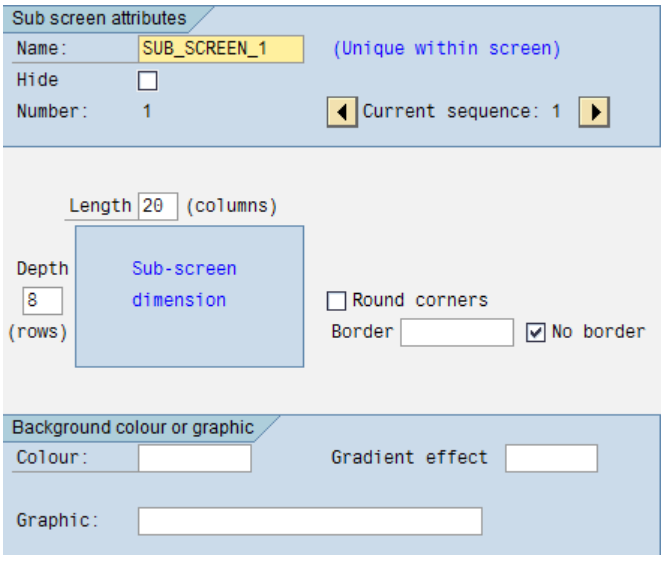
The client event type "WEB WINDOW" supports the BACK, FORWARD, RELOAD and STOP actions to apply to a "target" web window.

4.7.5.17 Sub-Screens

A sub-screen enables you to switch between many different displays within a single area of a screen (a type of tab control). This can be a flexible and powerful way to utilize the same screen real estate to display many different combinations without having to design separate screen functions. The concept is, that you allocate a dedicated area of the screen to a sub-screen definition, and then maintain different display sequences within it. Each sub-screen sequence is a distinct list of screen elements that only appear when that sequence (01-99) is activated. For example, you may have a sub-screen to display order information: sequence 01 displays the order header details, sequence 02 displays the order items, sequence 03 displays the order delivery schedule. You flip between the different sequences using screen events, procedure commands or programming exits. There is no limit as to how many different sub-screen definitions you may have on a screen and each sub-screen may have 01-99 unique instances called sequences.

> ***Important:*** You cannot embed a sub-screen within another sub-screen.

To create a sub-screen, select the "Sub screen" tab/button. The following attributes appear:



Option	Description
Name	A unique name for the sub-screen within the function.

Option	Description
Hide	Default the entire sub-screen to hide when the screen is first formatted.
Number	This is the unique internal number allocated to the overall sub-screen definition. You refer to this number when using events and procedure commands. This defaults to 1, when you create the sub-screen for the first time.
Current sequence	The currently active sequence (or instance) of the sub-screen that appears in the layout manager. You may use the arrows to scroll to different sequences.
Length / Depth	The horizontal length (columns) on the screen and vertical length (rows) on the screen of the reserved sub-screen area.
Round corners	By default, shade areas have square corners. This option draws rounded corners and is useful for 'softer' displays.
Border	The no border option suppresses a 'box' border from being written around the shade area. Otherwise, a box is drawn and you may specify a colour.
Colour	The background colour of the shade area.
Gradient effect	The direction of a gradient if desired, for example, left to right. Refer to the section Colours and gradients for more details.
Graphic	The name of a binary object to use as a background image.

Navigating between Different Sequences in the Workbench

Once you have created the sub-screen definition, a section of the screen is highlighted within which you can then start creating unique screen elements for the current active sequence (instance). Like a shade area, a special symbol appears in the top left corner that you can double-click to get to the sub-screen definition. To move to the next sequence, for example, from 01 to 02, you can either position your cursor anywhere in the sub-screen area and use the blue left/right arrows in the tool bar or use the left/right icons in the sub-screen definition itself.

Switching to Different Sequences at Run Time

You may use events, procedure commands and/or programming exits to dynamically switch to different sequences of a sub screen at run time. In this way, you can attach different displays to buttons, graphics, procedure logic. Because each sub-screen screen element is unique within the screen (just as per normal), you can pre-populate them in advance or when the sequence is switched.

4.7.5.18 Function Attributes

You can configure the screen function attributes by clicking the "attributes" icon (hat) on the application toolbar from within the screen painter. The attributes control the default behavior of the screen definition.

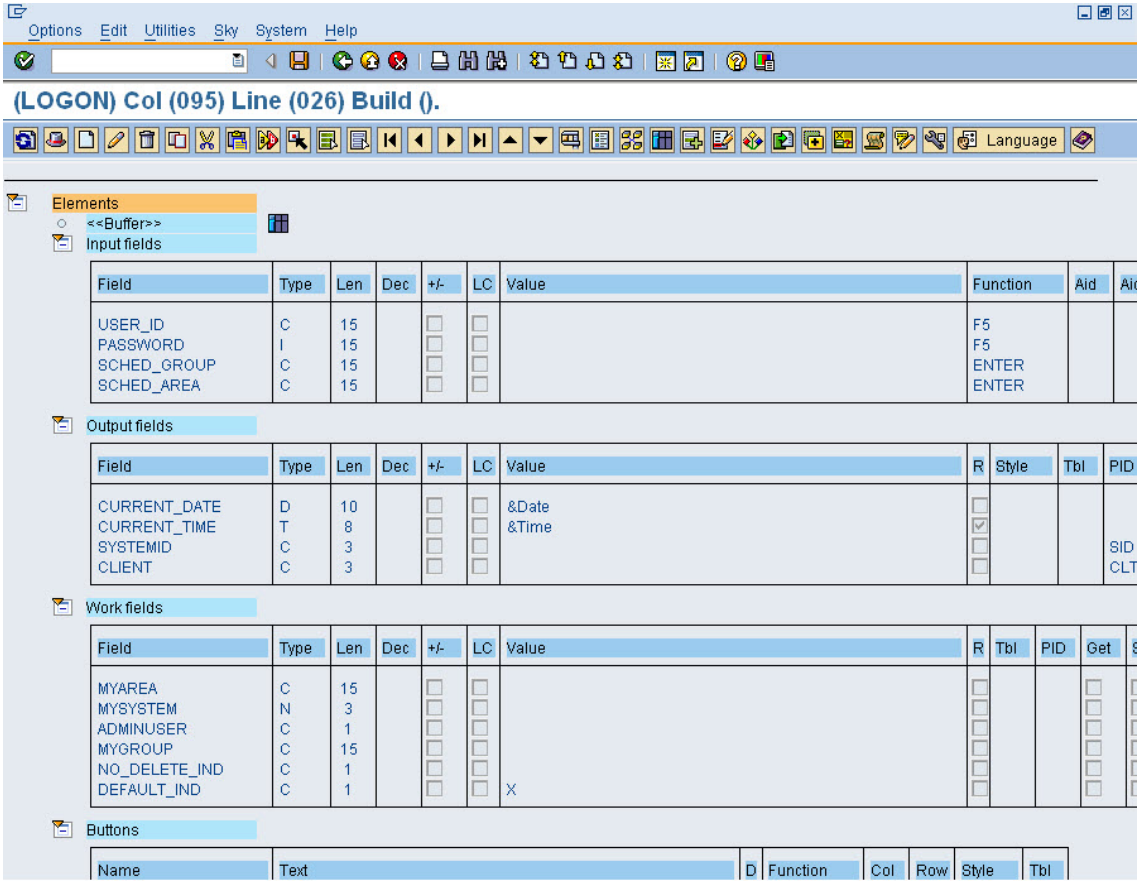
Field	Description
Function	A unique function name within the application version. This is specified when the function is created. As well as a name, each screen function is allocated a unique internal number within the application version that is known as a "funid". This appears at the bottom of the pop-up as "internal id".
Dynamically called	This configuration indicates that the function is designed to be invoked through a procedure or custom user exit using the "next function" technique; is not called directly by another screen function. It is "information only" to indicate in the hierarchy display that the function is referenced.
Direct call	Indicates that you can invoke the function directly by a external application through the XAI interface. In this case, the normal format exit and local database (LDB) processing is not performed and the internally generated '$GOTO' event is triggered. Default field values are still populated in the same manner from PIDs and the current screen (if any). The XAI interfaces enable SkyMobile functions to be directly invoked from external systems, for example, programs, URLs, browser, and email.
Title	A description. This is used as the standard screen header (if configured). If using a standard header, you may imbed substitution variables, identified by a "&'.
Format exit	Refer to the programming exits for more details.The format exit allows you to call either a custom ABAP or Java exit program when the screen is generated for the first time. They are useful to initially populate or alter the screen layout. Sky recommends that you use procedures (see below) wherever possible and only use programming exits in exceptional circumstances.

Field	Description
Procedure	Refer procedures for more information. This procedure is executed when the screen layout is generated for the first time. They are useful to initially populate or alter the screen layout. A procedure consists of built-in commands and logic operations that allow you to control how screens are processed and displayed. If the procedure does not exist, you are prompted to create it. You may also edit existing procedures using the maintenance icon on the right hand side.
Start sound	This sound is played whenever you first format the screen. You must first load the sound file into the SkyMobile binary object repository. Refer to the binary objects for more information on how to do this.
Background	This is the background colour or graphic to apply. The default depends on the presentation client used. If you do not specify, the default background colour on the version is used. You may also specify a gradient effect to apply for a colour, for example, fade from one colour to another from the left to the right. In order to do this, you must set up the colour definition with a gradient colour. Refer colours and gradients for more details.
Caller screen save/restore attributes	These options are only applicable for inter-screen processing, where you may invoke the screen function from many different points within application versions. They instruct SkyMobile to save an image of the callers screen and restore it upon return. You may also alter the values of corresponding (same) fields upon return. These options are also useful if you want to keep the values on the callers screen. By default, SkyMobile uses an internal save ID, but you may require a unique save ID if a chain of screens are processed and you require to keep their context.

Field	Description
Default settings for events	Local screen processing indicates that the screen function (by default) is to be processed locally by the Application Server. This configuration is used as the default for all action/events as well. Any calling screen function also takes this setting into account; whether to use the local Application Server or the host to initially format the screen. Refer to the section on local/host processing modes for more information. Store and forward indicates that the entire screen and action to be performed is stored locally in a queue if the host connection is not available; and then is executed asynchronously in the background when the connection is re-established. This setting is only valid for actions that are performed locally but have post local host dependencies, for example, call an ABAP exit. Refer to the section on Store and forward processing for more information on the store and forward manager.
Orientation and layout controls	You may use these settings to override the default settings in either the configuration file or application version attributes, for example, set the orientation of the screen, the SkyMobile layout manager to use and the transition method from the calling screen. For information on these options, refer to the general section on Orientation and layout controls and how they are defaulted and processed.

4.7.6 Element Section

This section displays a summary of all the screen elements. Note that only those types that have definitions have an expand/collapse icon next to them. You may also create and maintain screen elements and associated objects from here by double-clicking on them or positioning your cursor and clicking on the toolbar option.



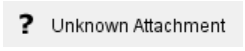
Important: You may not create certain screen elements that require a screen position from this section, but you must create directly from the screen layout section; double-click where you want it. In most cases, SkyMobile tries to find the first available space on the screen to accommodate the definition.

You may convert work fields to input/output fields on the screen by simply clicking on one and then selecting the input or output tab. The screen painter then automatically finds the next available space. You can then cut and paste it from there.

4.7.6.1 Using the Clipboard Buffer

A temporary buffer of commonly used fields may be used to copy and paste definitions to the screen. You can add existing screen fields to the <<BUFFER>>, or use the clipboard manager to copy fields into the <<BUFFER>> from a variety of sources. By using the <<BUFFER>> or personal clipboard, you can save considerable time when defining screen fields. Refer to the clipboard manager section for more information on this feature.

Fields are copied from the <<BUFFER>> by positioning your cursor on the buffer field and clicking on the copy icon on the application toolbar. The field may be then pasted onto the screen layout by positioning your cursor where you want it and clicking on the paste icon on the application toolbar. Screen fields may be added to the <<BUFFER>> using the "add to buffer" icon and removed from the <<BUFFER>> using the delete icon.

Click the 

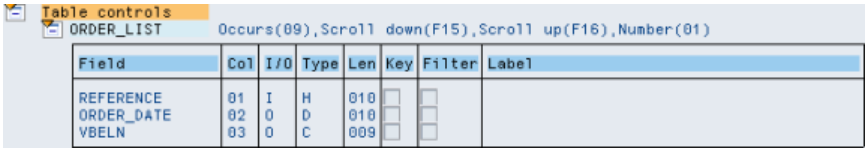
icon to be transferred to the clipboard manager where you can maintain the list of field definitions from other object types, for example, other screens, database definitions, SAP dictionary objects.

4.7.7 Table Control Section

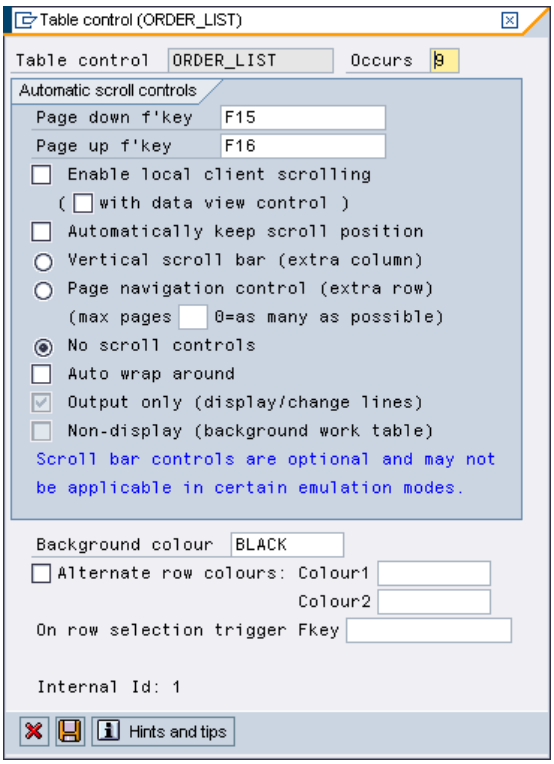
This section lists all the table controls (lists) that are defined for the screen. A table control is basically a repeating list of values, for example, order lines. A table control may span many lines and you may imbed graphics.

> ***Note:*** Before you create the actual table control, you should first create and align all the fields that are included in the table, design the first row.

Table controls are useful to manage lists of data on the screen. You may define multiple table controls on a screen (up to 99) and each is given a unique number. First create all the displayable and work field definitions on the screen that you want to include in the table control; then align the first row as you would like it to display (not applicable for work/display only tables). Table controls support multi-line rows. Once all the fields are defined, create the table control, specifying how many rows to display (occurs) and add fields to it. You may have single row table controls as well. You cannot use text labels in table controls, you should use output fields with default values instead. Internally, SkyMobile manages a table data list that is populated either from local database operations (LDB's) or screen exit programs. This data is then mapped to the screen depending on the "top row" value; starting row for the page. Special $TBL fields are automatically generated by the session manager and you may control these fields on an individual basis through the SkyMobile ABAP and Java APIs.



To create a table control, expand the table control section. Place your cursor on the text beneath the title of "Table Controls". Select the Create button. Similarly, to change a table control, position your cursor on the name and click the change icon on the application toolbar (or double-click it).



Field	Description
Occurs	The number of displayable rows on the screen. ***Note:*** This does not limit the number of rows actually stored in the table.
Page down/up keys	The function keys allocated to automatically scroll through the table entries. The workbench automatically generates internal $SCROLLUP[nn] and $SCROLLDOWN[nn] action/event definitions for scrolling. You may assign custom LDB operations and program exits to process on these events, giving the extra flexibility to manipulate the screen, for example, change colours, and assign dynamic graphics. If you are not using a scroll bar, you may trigger these events manually or through buttons, links, and graphics. You may even assign the PGUP and PGDN keyboard keys to scroll a single table on a screen.

Field	Description
Enable local client scrolling / data view control	This option causes all table information to transmit to the presentation client and all subsequent scrolling is performed locally without having to call the server. You should take care to limit the size of the table since this may affect performance. In addition, any special server side scroll processing (for example, exits) are not performed. ***Note:*** Check the capability of the presentation client that you use to see if it supports this option. The data view control option is only available with local client scrolling and provides built in functionality to dynamically filter the table entries that appear. The data view control requires that there is a free line area directly above the table reserved for it.
Automatically keep scroll position	The current 'top row' position is saved and restored when another screen function is called. ***Note:*** It is the top row number and not the row contents that are saved; if the data is changed, then the table display may be different on return. If there is no data to display from the saved point, the table is reset to row 1.
Vertical scroll bar	GUI emulators only. Provides an up/down GUI scroll bar immediately to the right of the table. The workbench automatically creates the function key definitions for this. Presentation clients provide both page and line at a time scrolling functionality; if you click the up/down arrow, this scrolls a line, if you click in the body of the bar this scrolls a page.
Page navigation control	Instead of a vertical scroll bar, this generates a horizontal list of page numbers at the bottom of the table; you navigate to an exact page by clicking the number. The max pages setting controls how many page numbers appear at the bottom of the screen.

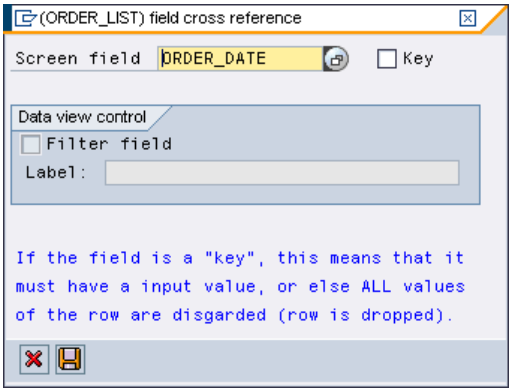
Field	Description
Auto wrap around	Directs SkyMobile to automatically start scrolling again from the start of the table, if the bottom is reached or the bottom if the top is reached.
Output only	If the table has display/change only rows. This option suppresses input rows at the end of the table and only displays the data available.
Non-display (work table)	The workbench maintains this setting automatically if it detects that the table only contains 'work' type fields; no displayable screen elements. Work tables are useful to store data during screen processing.
Background colour	A colour to assign as the table background. The default is grey.
Alternate row colours	If this is specified, each row of the table is assigned a different colour; colour 1 / colour 2. This is useful to highlight rows.
Trigger fkey	This designates a function key that is activated if the table row is selected. If an element within the row also has a fkey, this takes precedence.
Internal number	Each table definition is assigned a unique number by the system from 01 to 99. This number is cross referenced with fields.

Note: A table control definition has no relevance to the physical layout of the table on the screen. This is determined by the position of the screen fields assigned to it.

4.7.7.1 Allocating Elements

The screen elements must already exist in order to add to the table control. The screen elements to include in the table should be lined up according to the first row in the table. Position your cursor on the table control name, and click the "create" icon. A pop-up appears, into which you specify a screen element name. A single table row may span multiple screen lines; multi-line tables are fully supported.

> ***Note:*** You should take care of the expected dimension of the table when including multi-line screen elements, for example, graphics, as these can cause significant blow outs in table size and controls.



Field	Description
Screen field	The screen field to add to the table control.
Key	This indicates that the field provides a unique key to the table i.e. unique value. The implication of this is that empty or duplicate rows are dropped automatically by the server when table rows are processed.
Filter field / label	If the table has been configured with a data view control, then you may nominate which fields may be used as filters by checking this box and providing an alternate more descriptive label.

You must first remove the work fields from a table before they are converted to input/output fields. This is because work fields have no screen position as such (row/column coordinates) and thus you may position first and then add back to the screen table definition.

4.7.7.2 Table Control Design Considerations

The following are the table control design considerations:

Populating Table Data

As well as user input, you may populate tables directly by local database functions, procedures and/or programming exits. See the section on Local database, Procedures and the ABAP and Java programming guides for more details on the techniques and APIs available. You may populate tables using multiple passes; first load the order details and then loop through the table and populate the product details. This is achieved by defining multiple local database (LDB) operations with reference to the table and input table fields. The LDB built-in aggregate functions are extremely useful to summarize database data into tables.

How Table Data is Managed

When you define a table control, you specify only the number of displayable rows that the user may view on the screen at any one time. There is no imposed limit as to how much data may be loaded into the table. For instance, 30 orders may be loaded from a local database, but only five orders may be viewed on the screen. SkyMobile stores and controls the entire table, so if the user scrolls, they effectively scroll to the next viewable part of the table. The local and host session managers automatically compress table data to remove empty rows and columns. If the table is flagged as "output only" empty rows are suppressed. One or more table fields may be designated as keys meaning that if they have no value or are not unique (duplicate) the row is removed from the list.

> ***Important:*** Because the entire table is associated with the screen, you should take care when in host processing mode on how much data is returned to the client to display, how many table rows are selected and returned to the Application Server to scroll through locally. Performance problems can occur if the total size of the table control is excessive.

Table Control Work Areas and the Active Row

Every table control has a work area (header) that contains the working field definitions. These are populated with the values from a specific row if it is activated on the screen. For example, if a table row contains a link and the user clicks it, the row is activated and its contents are automatically copied to the table works area. This makes the data available to transfer to another screen, programming exits, and local database functions.

Configuring the Entire Screen as a Single Table Row

You may configure the entire screen as a single table row. If there is only one displayable row, SkyMobile automatically activates it; it is always the active row. In this way, you may load and scroll through data and easily initialize other screen functions passing the current data displayed, for example, show the attributes of one order header on the screen and if the user presses F1 or a button, call another screen to display the items, and passing the order number.

Using Work Tables

As well as configuring tables for display purposes, you can configure work tables that are used solely to store, process and aggregate data. You create the table in the same manner as usual, except that all the fields are work fields. When you save the table definition, the workbench flags the table as non-display, a background work table.

4.7.8 Actions and Events Section

Any sort of action invoked from a screen is related to a function key. Even the pressing of a button triggers a function key that in turn triggers an action (or event) to occur. A screen action is a unique code, upon which you can configure various combinations of processing such as: call another screen function, logoff, execute a programming exit, invoke a database operation, execute a procedure. SkyMobile supports a range of pre-configured function keys (including enter) per screen. Function keys are created and maintained using either: association with a screen element, for example, button, the standard screen footer (if available), or through the 'Action' section on the screen painter.

> ***Note:*** Function keys are logical triggers that may or may not be associated with function keys on the keyboard for example, F1.

The processing sequence when an action is triggered is:

1. Local database (LDB) operations
2. Procedure
3. Custom exit program
4. Built-in option (for example, logoff)

4.7.8.1 Action Section

Expand the action section and position your cursor anywhere on the heading to create a new action, or on an existing line to change or delete. You may also double-click on a line to maintain the definition.

Actions and events

Fkey Key	Action Code	Menu Show	Menu Label	Process Type	Format function	Procedure Definition
F1	NEXT	☐		FORMAT	TEST3	
F10	HIDE_F8	☐		SCREENIO		HIDE_F8
F11	TEST6	☐		SCREENIO		TEST6
F12	TEST1	☐		SUBSCREEN	(SS1-01)	TEST
F13	TEST1B	☐		FORMAT	TEST1	
F2	LIST_CHART_DETL	☐		SCREENIO		LIST_CHART_DETAIL
F3	NEXT2	☐		FORMAT	TEST5	
F4	NEXT3	☐		SCREENIO		NEXT_FUNCTION
F5	TEST	☐		SCREENIO		TEST
F6	BACK	☐		FORMAT		
F7	REMOTE_CALL	☐		SCREENIO		TEST_REMOTE_CALL
F8	CHECK_IF	☑	check if	SCREENIO		CHECK_IF
F9	SM_PROCEDURE	☐		TRANSACTION		

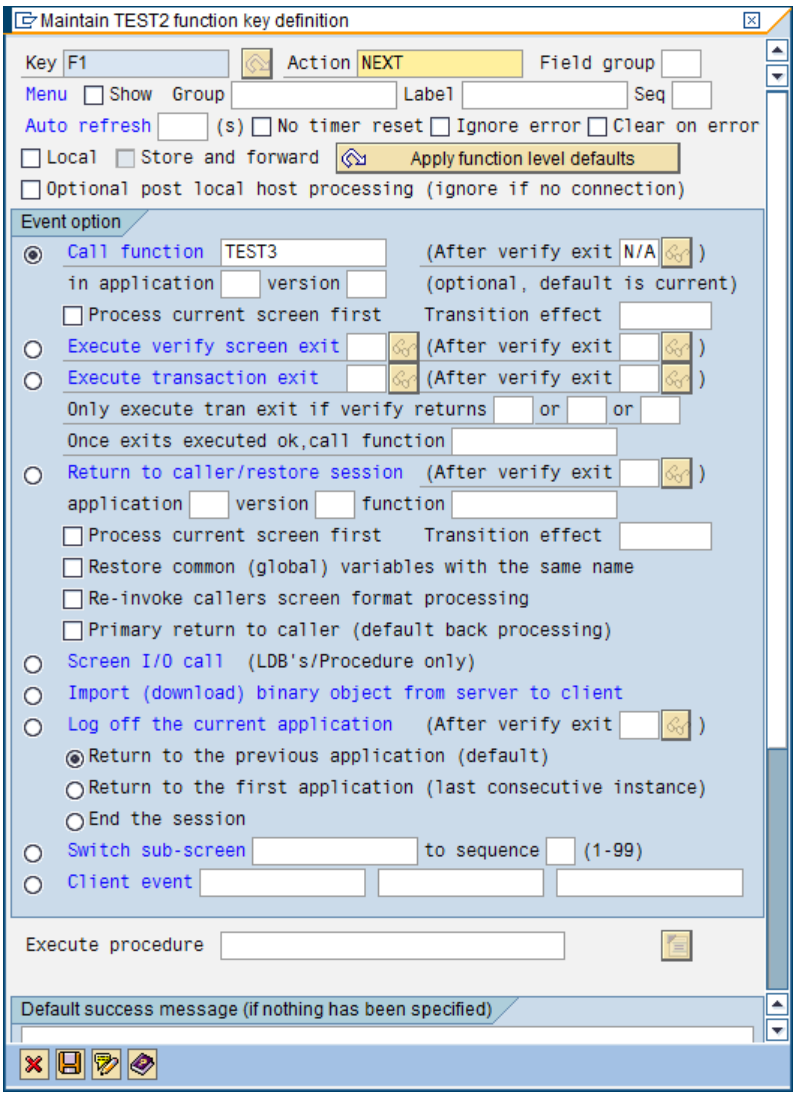
> ***Note:*** The action section has some 'hot' fields that you may use to branch to other definitions, for example, another screen function, and procedure definition.

4.7.8.2 Screen Actions

An action code is the link between a function key and an internal event. For example, if the user presses F1, this is associated with a user defined action code and an internal event is triggered to process all associated database and program exits associated with the action.

4.7.8.3 Action Definition Screen

This pop-up screen defines what the action is to do, when you press.



Field	Description
Function key	A unique function key id (F1-F30/ENTER/ESC/PGDN/PGUP).

Field	Description
Action code	A unique name to identify the 'event' that is triggered. Other functions such local database requests may be associated with this. The action name cannot be the same as any field name or the name of the function. This is because these names are used for special actions, for example, a field drop-down request and screen initialization. ***Note:*** The screen painter automatically generates some internal actions such as $SCROLLUP and $SCROLLDOWN. You may still customize these to call program exits giving you the flexibility to further control the screen formatting, for example, change colours, and dynamic graphics.
Field group	A field group representing one or more fields that is to be associated with the action. This means that only those fields that belong to the group are considered for event/screen processing. You must therefore assign all required fields to the group.

Field	Description
Auto invocation controls (auto refresh)	This option automatically triggers the action after 001-999 seconds. The default behavior is to automatically interrupt the timer on every screen I/O; only executes when the screen is idle. Use the "Don't reset timer" option to ignore this and process the action irrespective of any other screen I/O. You may specify multiple actions with auto-invocation controls on a screen in this way, and can refresh dynamic values such as times, counters, weights, and database values on a regular basis. Use these options to control partial screen updates. No timer reset: By default, the auto execute timer is reset automatically whenever any screen activity (typing, event) takes place. By checking this option, the auto execution happens regardless of anything else happening on the screen. If the scope of this includes possible input fields, you may loose recently typed in data. Ignore error: Irrespective of any error that the event processing raises, a ok return code and no message is returned to the caller. Clear on error (clear screen fields or field group): If the event returns an error, automatically clear the screen fields (or field group). This is designed for circumstances where an auto refresh may fail and no returned value is indicated by nothing).

Field	Description
Menu controls	The show, menu group, label and sequence options enable you to define and control application menus. These menus have different behavior depending on the presentation client used. Read the following notes on each emulation method. If show is specified and no menu group and/or label text is specified, SkyMobile generates appropriate default text, for example, the action code for GUI emulators and the fkey concatenated with the action code for VT220. Show indicator: This must be set to true, for a menu entry to be generated. If turned off the menu group, label and sequence are initialized by the workbench. Menu group The menu group is free format text and many be used to specify different menu displays. If left blank, any menu labels are displayed under the default 'action' or system menu. Label: The label text is free format text and is used to associate a action with a menu option. If left blank the label text defaults to the action code associated with the function key or event. Sequence: The sequence number may be used to order the menu option within a menu group or the default system menu. If not specified, the order will be as processed i.e. no specific order. Behaviour: GUI: Menu groups are displayed as distinct separate menus on the menu bar/options display. This may appear either on the top or bottom of the screen. VT220 (text terminal): A function key bar may be configured on the SkyMobile version. If so, a list of labels is displayed. If there is no label text, the fkey concatenated with the action code is used. Menu group is ignored.

Field	Description
Local	Whether the action is to be processed locally by the Application Server. ***Note:*** This may be different from the local/host mode of the overall function. ***Note:*** If the action is local but references host dependencies, for example, LDB and/or ABAP programs, the event is flagged as "post local" meaning that all local processing is performed first and then a call is made to the host to perform the "host specific" processing. Refer to the local/host processing modes section on for more information.
Store and forward	Whether the action is to be stored locally and then automatically forwarded to the host (in the background) if a host connection is not currently available. Refer to the Store and forward processing section for more information.
Apply defaults	Apply the defaults for local and store and forward from the screen function attributes.
Optional post local host processing	If the function key event is configured as local and has some host dependencies, ignore the post local host processing if no connection to the host is available, carry on as normal.
Call function	Call another screen function. After verify exit allows you to check the contents of the current screen, before starting the next function. Process current screen prior to next, validates the current screen first. Refer to the switching applications and functions for more details.
Execute verify screen exit	Execute a verify exit definition. After verify exit, allows you to invoke two separate verify exits within the one action.

Field	Description
Execute transaction exit	Executes a transaction exit definition. After verify exit, allows for a verify exit to check the screen input prior to processing the transaction. Execution of the transaction exit is ignored if the verify exit issues a return code. You may also configure the reverse, only execute the transaction exit, if the verify exit issues a specific return code. You may also invoke another screen function once the transaction exit successfully executed.

Field	Description
Return to caller	Returns control back to the screen that called it. This is only necessary for sub-functions that may invoke from multiple functions. If the function is only called from one place and there is no data that you require on the callers screen, use the call function option to return back. Return to caller has the effect of storing the callers screen in memory and this option instructs SkyMobile to restore the screen. You should specify a unique save id in the function attributes. Refer to the section on 'Maintaining function attributes' for more details. • Restore common global variables copy matching screen fields from the current screen back to the restored screen. • Process current screen prior to next, validates the current screen logic first and then restores the caller screen. • Reinvoke callers screen format processing instructs SkyMobile to reformat the callers screen. Refer to the switching applications and functions for more details. There are considerations when using this option. The following is a list of the most common problems encountered: Saved screen is overwritten Return to caller uses the save/restore configuration on the function header to automatically save and restore VTI sessions. If you do not specify a unique save id, a default name of spaces is used. The danger of this is that another save using the default overwrites any existing saved definition and thus the restore point is lost. To avoid this, use a unique save id for each screen. Import from memory failed (MEAP) Another design consideration with return to caller, is when using it in conjunction with $LOGON processing in SAP. The $LOGON processing instructs the VTI Application Server to close the current RFC connection with SAP, and open another using the supplied userid and password. Since the SAP session manager stores its saved screen definitions in ABAP memory, any saved screen definitions saved prior to the $LOGON are lost. If the application then attempts to restore any screen prior to the $LOGON, it receives an internal error "import from memory failed". In these cases, you should use direct calls to the function instead. Mixed local and host save/restore processing (saved session not found)

Field	Description
Screen I/O call only	A screen I/O call is a basic input/output call to the SkyMobile session manager. It gives the opportunity to perform scrolling, built-in field checking, and local database processing that are associated with the action.
End session (logoff)	This ends or logoffs the SkyMobile session. You may configure a verify exit definition to process prior to the logoff event. This may perform special logoff processing or prevent the logoff from occurring by setting a return code.
Switch sub-screen	Initiates a switch to another sub-screen sequence on the screen.
Client event	These are special presentation client side events to perform various tasks such as integrating with the devices camera, controlling a embedded browser window (web window). These tend to be APIs that are OS/Device specific and you cannot generically implement at the server level.
Procedure	The name of a procedure to execute when the event action is triggers. If the procedure does not exist, you are prompted to create one.
Default success message	Use this message if the return code is ok, and no other information message is specified. You may also configure the severity of the popup, I/W/E.
Confirmation prior to execution	This instructs SkyMobile to pop-up a confirmation box prior to executing the function key action. This gives the user the opportunity to cancel the request, for example, "Are you sure you wish to cancel the order?"
Default sound controls	You may configure start, success and error sounds to play when the event triggers and when control is returns. These are termed default, because the procedure or exit processing can potentially override the sounds.

Important: If you configure the action to call another function and the nominated function does not exist, the current screen function is saved and the user is prompted to create the new function as per the workbench. In the same way, users are prompted to create custom program exit definitions if they don't exist.

4.7.8.4 Client Camera Event

The SkyMobile client CAMERA client event is used to capture either a photograph or a video and automatically add it to the SkyMobile binary repository. You control the attributes through special "$PHOTO-" and "$VIDEO-" work fields, some of which get automatically generated for you. You have a choice of either a PHOTO or VIDEO action. The client event is defined through a function event/action that is triggered from the screen. You should use the classification fields, clsfield and clsdata, to associate the captured data with a data object, for example, ORDER 1234567890. You should also be careful of the size of captured video and use the transfer buffering and asynchronous options with data objects.

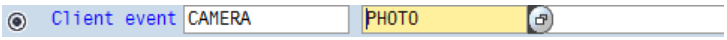


Photo Work Fields and Operation

Work field	Required	Description
$PHOTO_ FILENAME	YES	The binary filename that you should use within the binary repository. On returning, the binary filename is stored within this field if successful, else it is blank upon cancellation or failure. If this field is left blank, a binary filename of SERVID_ YYYYMMDDHHMMSS is generated. ***Note:*** You must mark this screen field as MIXED_CASE in the workbench.
$PHOTO_ LOCATION	NO	The binary location, either one of the defaults such as DOCUMENTS, PHOTOS, VIDEOS.. or a configuration defined binary location that can point to other folder locations. If this field is not present, left blank, or does not match any available binary location, the location defaults to PHOTOS.

Work field	Required	Description
$PHOTO_QUALITY	NO	The quality of the image to be stored. Valid options are QVGA (320x240), VGA(640x480), 1M(1280x960), 2M(1600x1200), 3M (2048x1536), blank, or a user defined resolution. If this field contains a value of the format WWWWxHHHH, then it uses the WWWW as the width, and the HHHH as the height when saving the captured image. If this field is not present, left blank, or does not match any of the other options, the image resolution defaults to the maximum the camera can take. ***Note:*** You should specify all dimensions in landscape, width larger than height, and depending on the orientation of the camera when the image is captured, the resulting saved image is rotated accordingly. Windows Mobile finds the closest resolution equal to or above the resolution requested and the image is not scaled to the exact resolution if you do not find an exact match.
$PHOTO_FORMAT	NO	The format to save the captured image as. Valid options are BMP, PNG and JPG. If this field is not present, left blank, or does not match any of the other options, the encoding defaults to JPG. ***Note:*** BlackBerry OS prior to 4.6.0 only supports saving photos in BMP format, 4.6.0 and onwards supports saving in BMP, PNG and JPG. Windows Mobile only supports saving images in JPG.
$PHOTO_RESULT	NO	If this field is present, upon returning it contains OKAY, CANCEL or FAIL, to indicate the choice the user made whilst capturing the image.
$PHOTO_VERSION	NO	The version of the binary filename to store within the binary repository. If this field is not present, left blank, or does not contain a valid integer, it defaults to 0.

Work field	Required	Description
$PHOTO_ KEEPDAYS	NO	The number of days to keep the captured image within the central binary repository in SAP. If this field is not present, left blank, or does not contain a valid integer, it defaults to 0 which means unlimited.
$PHOTO_ CATEGORY	NO	This is the binary object repository category for the image. If left blank, it is defaulted based on the extension.
$PHOTO_ GROUP	NO	The binary group under which to store the image within the binary repository. If this field is not present or left blank,it defaults to the activated fields parent group. If this still has no group it defaults to the application versions binary group. If this still has no group it tries to use the configuration files SERVER.BINARYFILES.DEFAULTBINARYGROUP value. If this still has no group, then no group is associated with the image.
$PHOTO_ CLSFIELD	NO	The classification field under which to associate with the image within the binary repository. If this field is not present,it defaults to blank.
$PHOTO_ CLSDATA	NO	The classification data under which to associate with the image within the binary repository. If this field is not present,it defaults to blank.
$PHOTO_ UPLOAD	NO	A flag indicating whether to upload the captured image immediately or not. If this field is not present or does not contain the value 'X', the resulting image is not uploaded immediately.

Video Work Fields and Operation:

Work field	Required	Description
$VIDEO_ FILENAME	YES	The binary filename that to use within the binary repository. On returning, the binary filename is stored within this field if successful, else it is blank upon cancellation or failure. If this field is left blank, a binary filename of SERVID_YYYYMMDDHHMMSS is generated. ***Note:*** You must mark this screen field as MIXED_CASE in the workbench.
$VIDEO_ LOCATION	NO	The binary location, either one of the defaults such as DOCUMENTS, PHOTOS, VIDEOS.. or a configuration defined binary location that can point to other folder locations. If this field is not present, left blank, or does not match any available binary location, the location defaults to VIDEOS.

<table>
<tr><th>Work field</th><th>Required</th><th>Description</th></tr>
<tr><td>$VIDEO_
QUALITY</td><td>NO</td><td>The quality of the video to be stored. Valid options are different for each platform. For Android devices there are two options for the $VIDEO_QUALITY
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>LOW</td><td>Lower resolution if the hardware supports encoding at multiple resolutions, and smaller file size or shortened recording length.</td></tr>
<tr><td>HIGH</td><td>Higher resolution if the hardware supports encoding at multiple resolutions, and larger file size or extended recording length.</td></tr>
</table>
For IOS devices there are four options available for the $VIDEO_QUALITY
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>LOW or
480x360</td><td>Video captured at 480 by 360 of recording length permitted by the file systems free space.</td></tr>
<tr><td>HIGH
1920x1080</td><td>Video captured at 1920 by 1080 of recording length permitted by the file systems free space.</td></tr>
<tr><td>192x144</td><td>Video captured at 192 by 144 of recording length permitted by the file systems free space.</td></tr>
<tr><td>640x480</td><td>Video captured at 640 by 480 of recording length permitted by the file systems free space.</td></tr>
</table>
For BlackBerry and Windows Mobile devices the options are specific to the devices capabilities for $VIDEO_QUALITY
<table>
<tr><th>Value</th><th>Description</th></tr>
</table></td></tr>
</table>

Work field	Required	Description
$VIDEO_ RESULT	NO	If this field is present, upon returning it contains OKAY, CANCEL or FAIL, to indicate the choice the user made whilst capturing the video.
$VIDEO_ VERSION	NO	The version of the binary filename to store within the binary repository. If this field is not present, left blank, or does not contain a valid integer, it defaults to 0.
$VIDEO_ KEEPDAYS	NO	The number of days to keep the captured image within the binary repository. If this field is not present, left blank, or does not contain a valid integer, it defaults to 0 which means unlimited.
$VIDEO_ CATEGORY	NO	This is the binary object repository category for the image. If left blank, it is defaulted based on the extension.
$VIDEO_ GROUP	NO	The binary group under which to store the video within the binary repository. If this field is not present or left blank,it defaults to the activated fields parent group. If this still has no group, it defaults to the application versions binary group. If this still has no group it tries to use the configuration files SERVER.BINARYFILES.DEFAULTBINARYGROUP value. If this still has no group, then no group is associated with the video.
$VIDEO_ CLSFIELD	NO	The classification field under which to associate with the video within the binary repository. If this field is not present,it will default to blank.
$VIDEO_ CLSDATA	NO	The classification data under which to associate with the video within the binary repository. If this field is not present, it defaults to blank.
$VIDEO_ UPLOAD	NO	A flag indicating whether to upload the captured image immediately or not. If this field is not present or does not contain the value 'X', the resulting image is not uploaded immediately.

Note: The format of the video saved under both IOS and Android is in .mp4 format.
The format of the video saved under BlackBerry is in .3gp format.
The format of the video saved under Windows Mobile is dependent on the encoders that the OS provides, and can be either .wmv, .avi or .mp4 format.

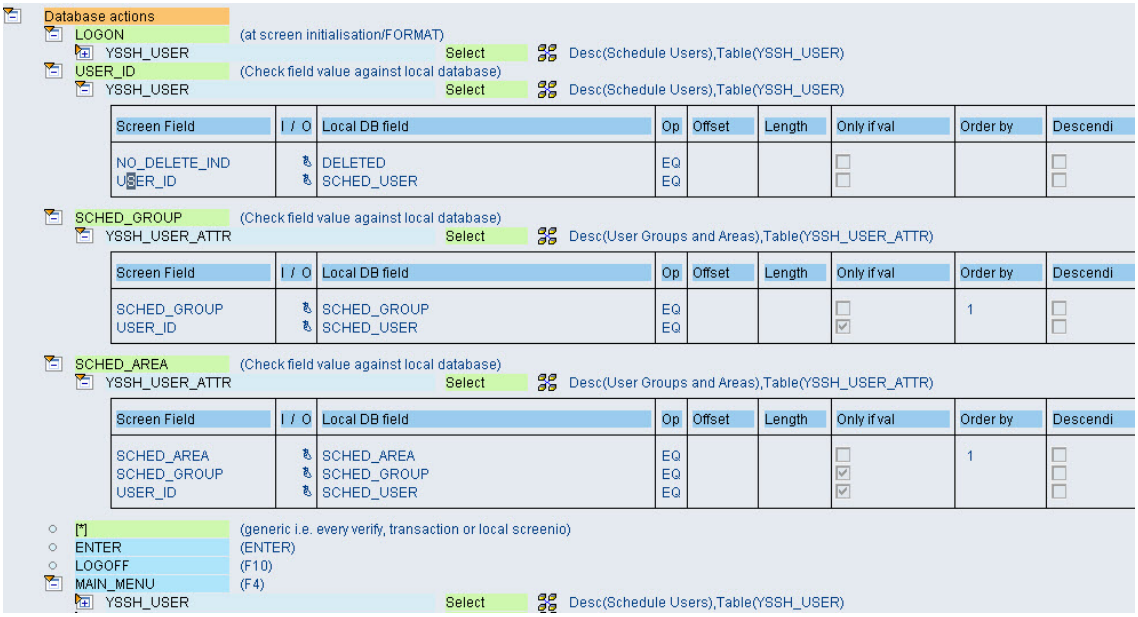
4.7.9 Database Actions Section

These functions allow you to imbed database operations directly into screen processing, to select and maintain database data. This is useful to display information, verify input data, update database information; and transfer data to the host. Before you may use LDBs in a screen, they must be defined using the LDB workbench. Expand the local database section to view all existing actions and show all associated LDB functions.

As well as a user defined event action code, there are four other types of internal action codes that may be used to trigger LDB functions:

1. Function initialisation; when the screen function is first formatted
2. For fields that have associated LDB value or value list checks (generated internally by field definition)
3. Stand alone, triggered by a procedure 'Call LDB action ...' command
4. Generic action '[*]' that is performed on every action irrespective.

Local database screen functions are either associated directly with event actions, or are stand alone. If you wish to create a LDB operation associated with an event action, you position your cursor on the "action" node and click create. If you wish to create a "stand alone" LDB action, you position your cursor on the top level "Database actions" node and click create.

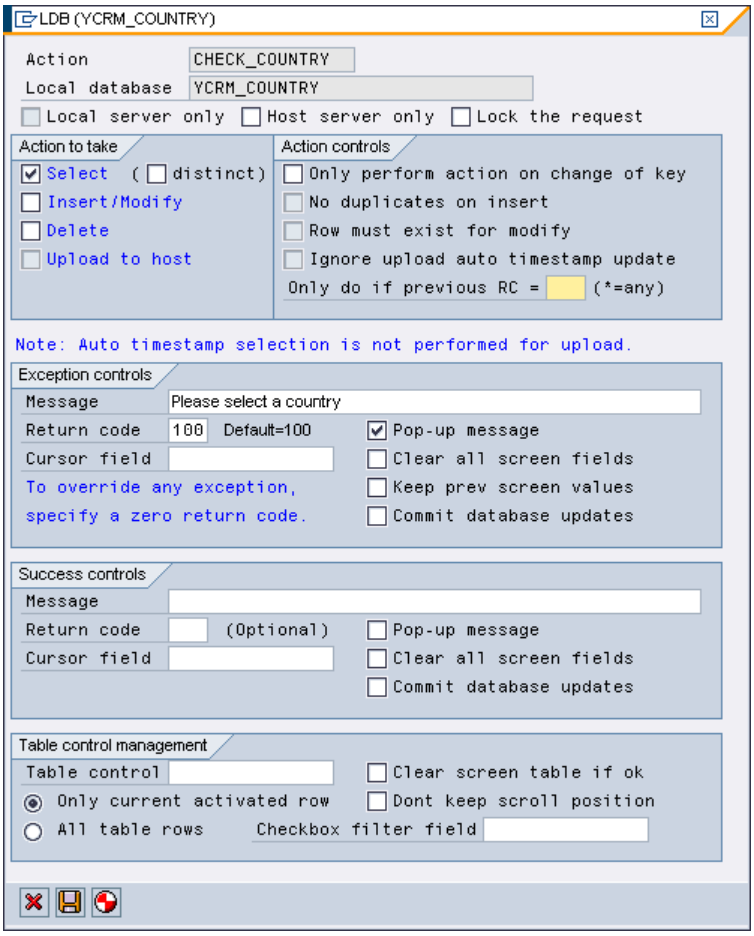


There are two maintenance screens involved, the first is for the LDB definition and the second is to map screen fields to LDB fields. To create a new LDB definition, position your cursor on the action to associate it with and click the create icon on the application toolbar. To change an existing definition, position your cursor on the line and click the change icon, or alternatively double-click the line.

> ***Note:*** You may define one or multiple LDB functions within one action code. Use the up/down arrows on the application toolbar to alter the sequence in which they are executed. You may configure condition code dependencies between them to selectively process or bypass processing depending on the result of the previous operation.

4.7.9.1 Database Function

This is effectively the header that defines the type of operation (select, and delete) and the behaviour, for example, load the data into a table control.



Attribute	Description
Local database	The local database table to use. This must already have been created using the LDB workbench.

Attribute	Description
Host/local processing only	Instruct SkyMobile that the database function is to be performed only on the host, or locally. The 'local only' option is only available for local screen definitions. The SAP only option is useful to force a Application Server to process the LDB request against a SAP table. Click here for more information.
Lock the request	Effectively enqueues the LDB request; make sure only one can happen at a time.
Action to take	This effectively equates to the SQL operation to perform against the LDB; select data, insert/modify data, delete data or invoke an upload operation to SAP. Against the select, there is a distinct option that applies to all fields selected.
Action controls	Only do on change of key Only perform the LDB function if the values of the designated screen input fields changed. No duplicates (insert/modify only) / Row must exist Indicates that no duplicate row must exist for a insert operation; effectively nullifies modify. The converse of this is 'row must exist' that effectively nullifies insert. Only do if previous RC = nnn Only perform the LDB function if the previous LDB function has issued a specific return code. This enables basic conditional return code processing to be implemented. A '*' indicates any non-zero return code.
Exception controls	A return code and/or message to set if the LDB function failed. You can also default the cursor to a specific screen field. 'Clear screen fields' automatically clears all the screen field values that are not explicitly marked to be kept (no-clear). Keep previous values Restores the screen to that of prior to the LDB request.

Attribute	Description
Success controls	A return code and/or message to set if the LDB function is successful. You can also default the cursor to a specific screen field. 'Clear screen fields' automatically clears all the screen field values that are not explicitly marked to keep (no-clear).
Table control management	This indicates that the LDB function is to work with a screen table control for either input or output values; populate a table with values or use table control data to update a LDB. Only active row or all table rows Indicates that the LDB function is to receive its data only from the current active (selected) row, or, the request applies to all rows in the table. Checkbox filter You can use a checkbox in the table to filter out specific table rows for processing; only use those rows where the checkbox is true ('X') for LDB processing. This feature is extremely useful for selective processing, for example, only delete those rows selected. Clear rows if ok If the LDB operation is successful, automatically clear all the table rows.

Database Logical Unit of Work (Commit)

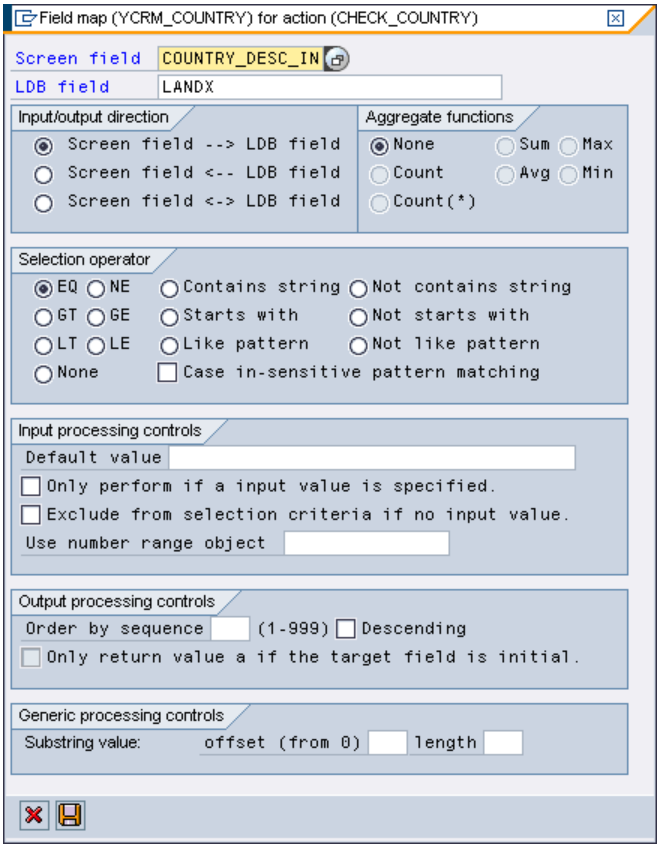
By default, SkyMobile does not issue a commit work at the end of each LDB operation, but rather when all the LDB functions for the action are successfully processed. A commit work commits all updates to the database, you cannot back out them past this point. If you require that this is required and the logical unit of work is not extended into the next LDB function, then you must specify the 'Commit database updates' on the exception or success controls, or both.

Automatically Generating Screen Field / LDB Field Maps

You may use the generate icon on the pop-up toolbar to automatically generate mapping from the screen to the local database by matching name. Screen input fields are generated as input key field maps and output fields as result maps. This function is only useful if the same field names are used, otherwise you should perform the mappings manually.

4.7.9.2 Field Maps

Once the LDB function is created, you must create the field maps between the screen and the LDB. These determine what data is to be input to the LDB operation and what is output back to the screen. They also determine if the result is to be populated into a table control, or input is from a table control. Position the cursor on a LDB definition and click the **Create** icon on the application toolbar.



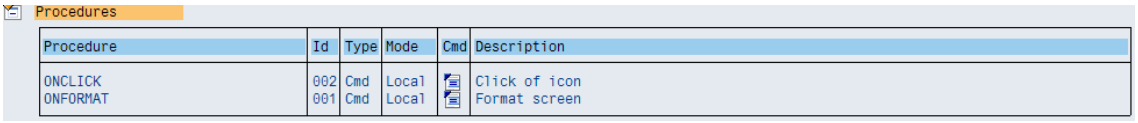
Field	Description
Screen field	The screen field to use for input or output. This field may be in a table control.
LDB field	The corresponding input/output field on the LDB

Field	Description
Input/output direction	The direction of the data flow; from the screen to the LDB (->), from the LDB to the screen (<-) or bi-directional (<->).
Operator	The SQL operator to apply. This is not applicable for insert/modify/delete.
Aggregate functions	You can apply sum, max, min, avg, count SQL aggregate functions to the output field value. These are useful for summary tables, charts. If aggregates are configured, any non-aggregate output fields are automatically used in the group by.
Default	If the input or output field has no value, use this default value.
Only perform if a input value is specified	The entire LDB operation is performed if the nominated screen field has a value.
Exclude from selection criteria if initial	This option allows for dynamic selection criteria, only include this criteria in the SQL where clause if the input screen field contains a value.
Number range object	Use the nominated number range object to populate the LDB field. This feature is useful to automatically populate the field with a next number value, for example, next order number. The value is returned to the nominated screen field. ***Note:*** The number is only automatically incremented if a database row does not already exist that is a new number is only returned if a record is inserted.
Order by	A sequence number representing a sorted order. This only applies to LDB select operations and controls the order that data is returned in. The default is ascending, but descending may be applied.

Field	Description
Only set if output field is initial	Only populates the screen field if there is currently no value; empty.
Substring source value	You may apply a portion of the field data, for example, first five bytes.

4.7.10 Procedures Section

This section lists all the procedures that are defined for the screen. Each procedure has a unique name up to 32 characters that cannot contain spaces. Procedures are usually defined on the function attribute (on format) pop-up and/or on the event action pop-up. A procedure is basically a list of built-in commands that you can use to control the behavior of screen processing. The aim is to be able to implement more complex logic and dynamic controls without having to write custom programs exits. A complete range of commands is available to control just about every facet of screen processing. As mentioned, procedures are usually defined using the function attribute and event action pop-up. You may also maintain them from this list by double-clicking the entry or positioning your cursor and using create, change and delete icons on the application toolbar. Each procedure is made up of a header definition and a sequence of commands maintained by a full screen Procedure command editor. You may easily navigate to a procedure by clicking the "hotlink" names in the layout or event action sections, double-clicking the line in the procedures section, or double-clicking the icon to go straight to the command editor.


Procedures

Procedure	Id	Type	Mode	Cmd	Description
ONCLICK	002	Cmd	Local		Click of icon
ONFORMAT	001	Cmd	Local		Format screen

4.7.10.1 When Procedures are Called

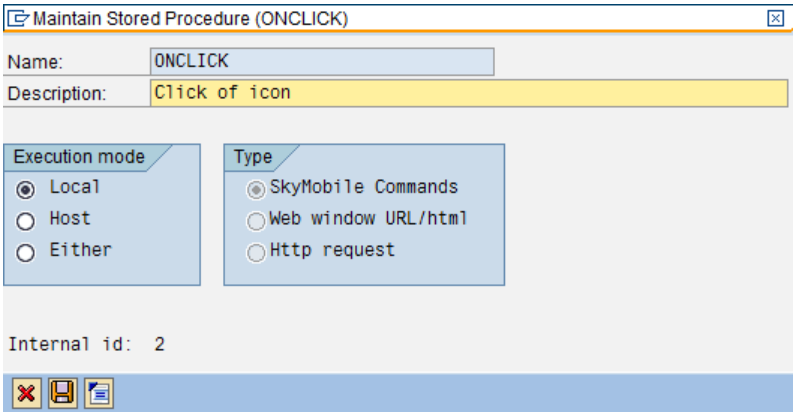
Procedures are invoked after LDB screen map processing and before custom exit processing. Secure Container or the MEAP Server invokes the procedures either by depending on the local/host configuration. A format procedure is called when the screen layout is generated and any procedure associated with a event action are invoked when action is triggered. You may configure a procedure as local, host or either, can perform directly by the container or by the MEAP. If an action or screen function is configured as local and a host procedure is configured, SkyMobile detects that there are 'post local' processing dependencies and calls the procedure on the host, once all the local screen processing is performed.

You may call procedures from within procedures to make processing more modular.

Important: You cannot mix local and host commands in the same procedure. You must call a separate procedure, for example, a local procedure then calls a host procedure and then continues processing when control is returned.

4.7.10.2 Procedures Definition

This pop-up is used to create and maintain the procedure header definition.



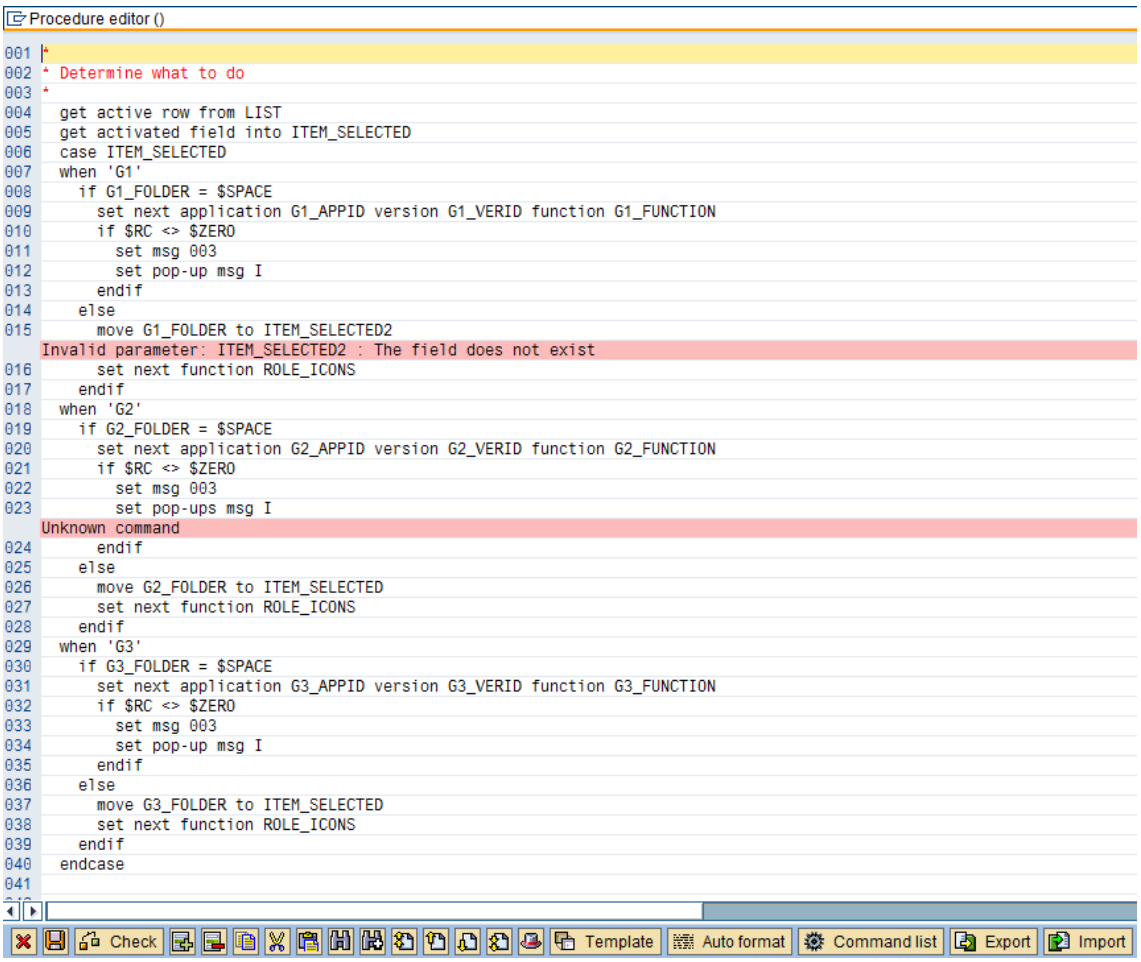
Attribute	Description
Name	A unique 32 character name (within the screen function) that cannot contain any spaces.
Description	A free format description of the procedures function
Execution mode	Whether you must execute the procedure locally, by the host, or by either
Type	This is reserved for future use
	Invoke the command editor. This is done automatically when you create a new procedure.

Once you created the procedure, you are automatically taken to the procedure command editor.

4.7.10.3 Procedure Command Editor

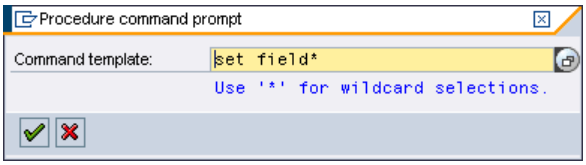
Once you create your procedure, you may specify a list of built in commands that you have to execute. You can do this through a full screen command editor that has various formatting and template generation features to make things easier. Each line is a unique command; a command cannot span lines. You insert, delete, cut, copy, paste by positioning your cursor and using the relevant icon on the toolbar. Pressing enter can also insert a new line prior to or after the current cursor position. If you do not know the syntax, you simply double-click a blank line or on part of a command and the command template generator, or a parameter selection pop-up invokes to assist you. The command editor has a check facility that makes sure that the syntax and parameter references are correct. You should successfully check the complete command list before you can save it. Any errors clearly appear just below the command, highlighted in red. If you wish to see a list of all available command definitions, click "Command list" in the toolbar.

> ***Note:*** For more information on the available procedure commands and examples of usage, refer to the procedure command reference section.

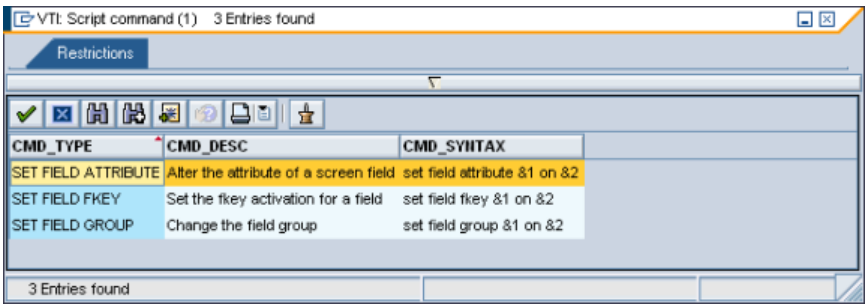


Template Generator

This feature is invoked by clicking a blank line or an incomplete command. It allows you to search for and select a command, and then generates the syntax for you.



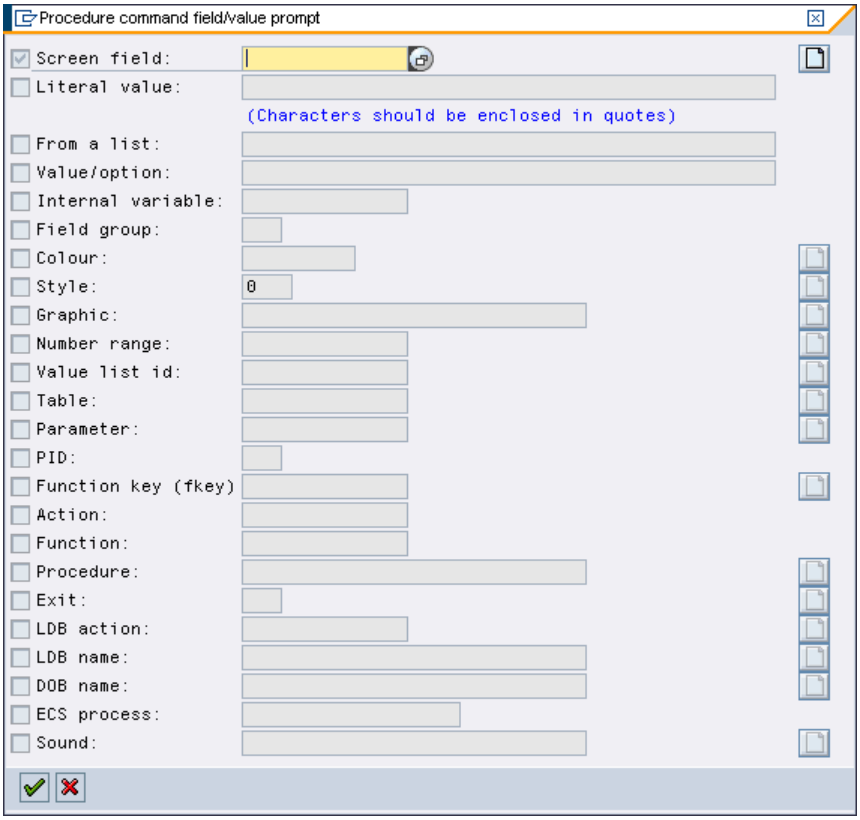
You may use wildcard selections using '*' and then click the drop-down or click F4 to display the search result list.



Double-click the required entry to copy it back into the command template selection.

Parameter Selection

Required parameter place markers in a command are identified by '&', for example, set field attribute &1 on &2. If you then double-click the '&' value, the parameter selection aid is invoked. This aid intuitively knows what type of parameter is required through the command type definition. You may even create new definitions from here by clicking the create icons on the right hand side, for example, create a new screen work field without having to exit procedure maintenance.

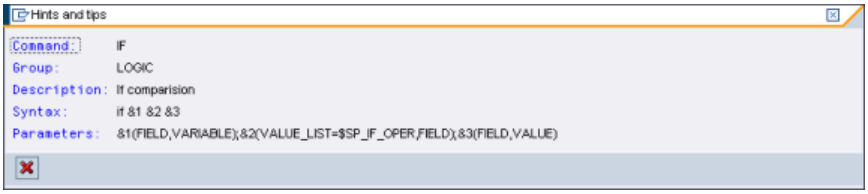


Comments

To make the command list more readable and to document processing, you may insert comments by placing a '*' in the first column.

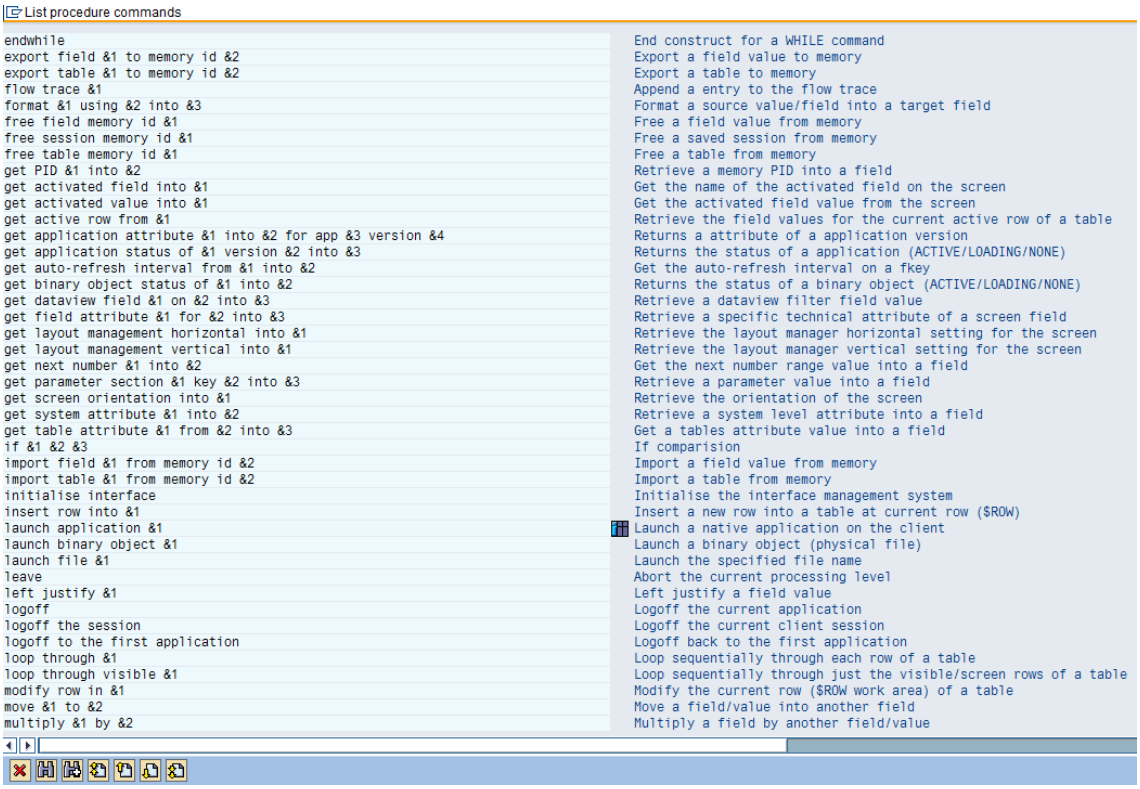
Command Definition

If you double-click any part of a complete command (not a parameter), a pop-up of the command definition appears.



Command List

You may list all available procedure commands from the editor by clicking the Command list icon on the toolbar. The following pop-up appears:



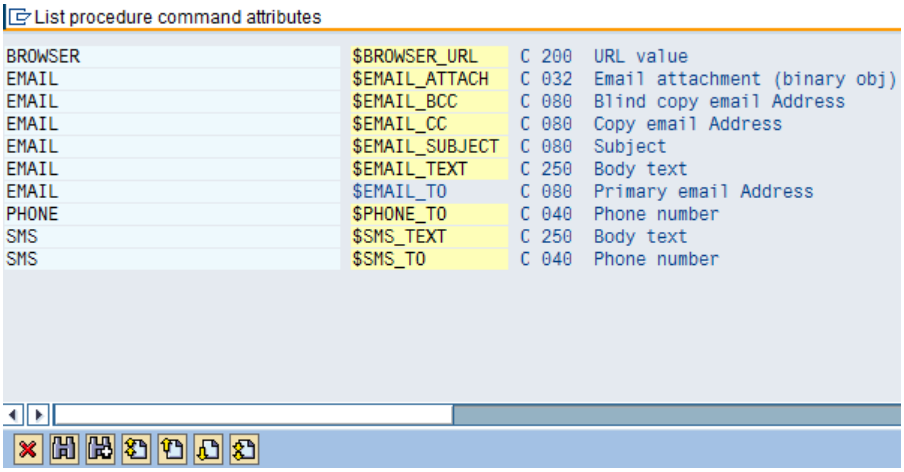
> ***Note:*** The find function in the application tool bar is useful to search for specific values.

Copying a Command Back into the Editor

If you wish to copy a command into the editor, double-click the command and it is copied into a temporary buffer. You then return to the editor and use the paste icon in the toolbar to copy in the command where you want it.

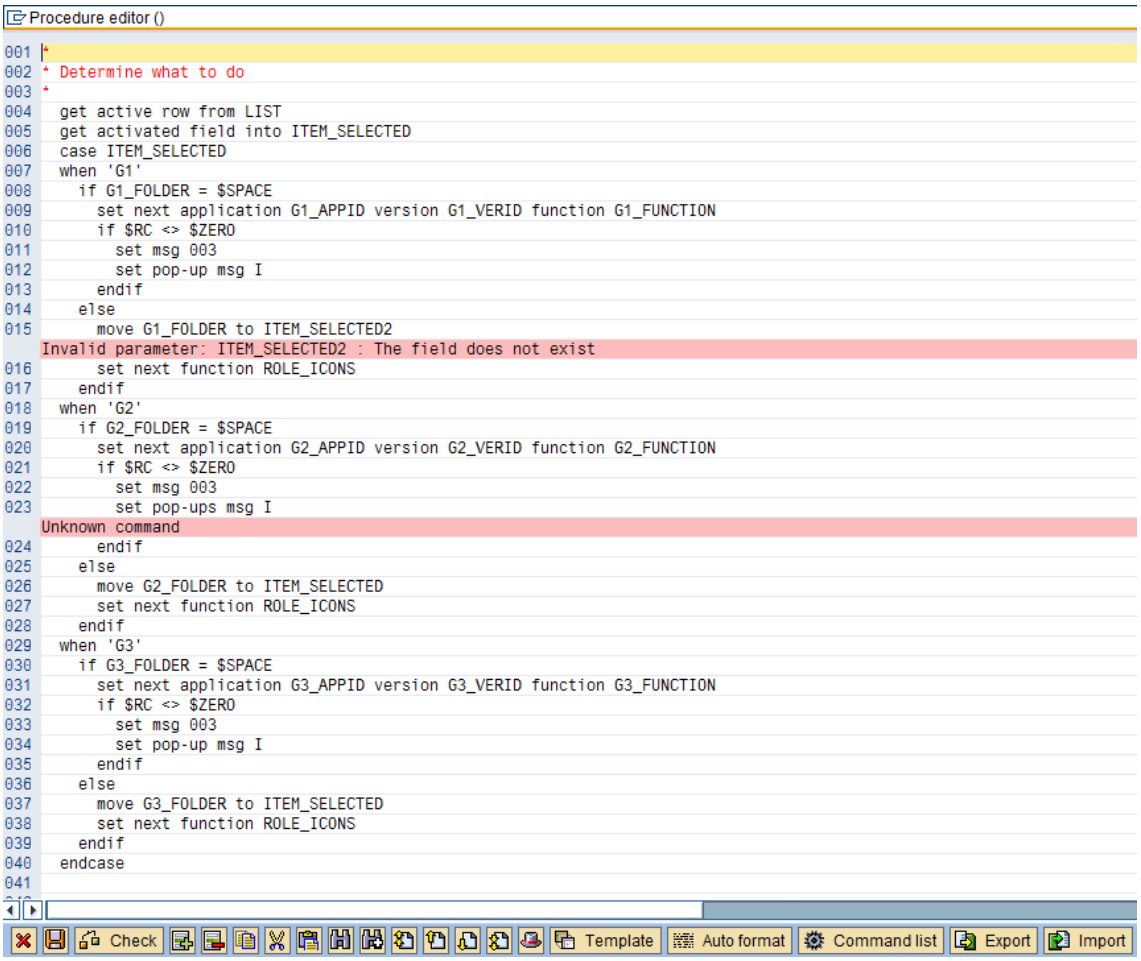
Special Control Work Fields (Command Attributes)

The icon to the left of the command indicates that the command has special $ control work fields associated with it. Click this icon to display a list and their attributes. If the work field does not exist in the function, it is highlighted in yellow. You may automatically generate the work field definition by double-clicking it.



Command Check Facility

You you can save any procedure command list, you must successfully check. You can do either manually through the Check icon on the toolbar, or automatically when you click the **Save** icon.



Any error messages are highlighted in red and appear immediately under the command line.

4.7.10.4 Procedure Command Reference

The structure of a procedure command is made up of English like text interspersed with substitutional parameters identified by a "&", for example, move &1 to &2. Each substitution parameter has internal meta data that controls what type of value is acceptable, for example, a field name, and static value. If you double-click the parameter, a pop-up of acceptable types of values appear. If you double-click the 'english' section of the command syntax, then a pop-up of the command meta-data appears. All procedure commands are allocated into broad categories for easier reference. Examples and extended descriptions are available to help describe usage and options of the more complex command constructs.

Command Categories

- Field
- Table
- Screen
- System
- Logic

Levels and Constructs

Some commands are constructs, they have a beginning and an end, for example,

```
IF FIELD1 = 'X'
OR FIELD1 = 'Y'
....
ELSE IF FIELD1 = 'Z'
....
ELSE
....
ENDIF
```

If the condition is true, then all commands between the IF and ENDIF are executed. The same is true for CASE/ENDCASE, LOOP/ENDLOOP, DO/ENDDO, and WHILE/ENDWHILE. Constructs dictate the level of the dependent commands, the IF is at level 1 and the dependent commands are level 2. If you then perform a LEAVE command, you exit the current level and resume processing at the next command on the previous/upper level. The CONTINUE, CHECK and LEAVE logic commands are all useful to control processing within constructs. You may nest constructs within each other.

Table Screen Field References

You have to use a (*) notation following a table field (cell) to refer to the internal field definition ($tb...) and not the table work field, for example, configure colour red on priority(*). If you use an assignment, for example, Move against a (*) field this updates both the visible list field and the table data, thus all assignment operations must take this into account.

Field Commands

Field commands are used to directly manipulate and format function elements and their data such as fields, graphics.

Refer to the example sections at the end of this page for more information on the more complex constructs.

Syntax	Description
add &1 to &2	Add a value/field to another field
append value list &1 from &2	Append a value to a fields value list
calculate &1 &2 : &3 &4	Perform a special calculation for a specific field type, for example, subtract values from date and time fields. For example, calculate 'date' v_date : subtract v_days
calculate UTC &1 &2 &3 &4 giving &5 &6	Calculate the UTC date time using a specific date, time and offset. See the example section for more information on how to use this command.
center &1	Center a field value
check value list &1 using &2	Check that a fields value exists in a value list
clear field &1	Clear a fields value
clear field group &1	Clear the values of a entire field group
clear screen	Clear the values of the entire screen (including tables)
clear value list &1	Clear a value list for a field

Syntax	Description
compare &1 &2 with &3 giving &4	Compare two values together, for example, dates/times
compare field group &1	Compares field values with a saved field group
concatenate &1 and &2 into &3 separated by &4	Concatenate two values into a target field ($NULL = contiguous)
condense &1	Remove consecutive blanks from within a value
divide &1 by &2	Divide a field by another field/value
format &1 using &2 into &3	Format a source value/field into a target field
get field attribute &1 for &2 into &3	Retrieve a specific technical attribute of a screen field
get next number &1 into &2	Get the next number range value into a field
get parameter section &1 key &2 into &3	Retrieve a parameter value into a field
get PID &1 into &2	Retrieve a memory PID into a field
left justify &1	Left justify a field value
move &1 to &2	Move a field/value into another field
multiply &1 by &2	Multiply a field by another field/value
no-gaps &1	Remove all blanks from within a value
refresh field &1	Re-instate the field into the screen definition

Syntax	Description
refresh field group &1	Re-instate the entire field group into the screen definition
refresh screen	Re-instate the entire screen definition
remove field &1	Remove/delete the field from the screen definition
remove field group &1	Remove/delete the entire field group from the screen definition
remove leading zeros from &1	Remove leading zeros from a field value
replace &1 with &2 into &3	Replace a value in a field with another value
restore field group &1	Restores field values from a saved field group
right justify &1	Right justify a field value
save field group &1	Saves the values of a field group to memory
search &1 for &2	Search a field for a value. Returns $POSITION.
set background colour &1 on &2	Change the background colour of a field
set border colour &1 on &2	Change the border colour of a field
set chart type &1 on &2	Dynamically sets the chart type
set checkbox &1 to &2	Configure a check box field to true/false
set colour &1 on &2	Change the foreground colour of a field
set cursor on &1	Positions the cursor on a specific field

Syntax	Description
set field attribute &1 on &2	Alter the attribute of a screen field
set field fkey &1 on &2	Set the fkey activation for a field
set field group &1 on &2	Change the field group
set graphic &1 on &2	Change the graphic name dynamically
set parameter section &1 key &2 from &3	Set a parameter value from a field or value
set PID &1 from &2	Set a memory PID value from a field/value
set radiobutton &1 on	Set a radio button field to true
set style &1 on &2	Change the style of a field
set text &1 on &2	Change the text of a screen element dynamically
split &1 at &2 into &3	Separate a delimited list of values into a table
substring &1 offset &2 length &3 into &4	Sub-string a value from a source field into a target field
subtract &1 from &2	Subtract a value/field from another field
upper-case &1	Convert a character field to upper case values
zero pad &1	Right justify and zero pad a field value

Examples:

- UTC time zone calculations
- Formatting values

- Date and time calculations
- Saving and comparing fields

Date and Time Calculations

You may use the calculate procedure command to add and subtract days and seconds from date and time fields.

```
calculate 'date' v_date : 'subtract' v_days
calculate 'time' v_time : 'add' v_seconds
```

In the above calculate examples:

- the first parameter indicates the type of operation, for example, date or time
- the second parameter is the target field
- The third parameter is the operation, for example, add or subtract
- The fourth parameter is the numeric value to apply.

Formatting Values

The procedure format command is useful to perform a number of format functions against a value, such as date/time manipulation and masking.

4.7.10.5 Example Format Commands

```
Format v_date using 'WEEK-NUMBER' into v_week
Format v_cost using 'NMASK($_,__,_._#) into v_cost_display
```

A range of formatting options are available. Click one of the links below:

- Date formats
- Time formats
- Other format options
- Character mask (CMASK)
- Numeric mask (NMASK)

4.7.10.6 Date Formats

Format	Description
YYYY/MM/DD	date display format
DD/MM/YYYY	date display format
DD/MM/YY	date display format
MM/DD/YYYY	date display format
MM/DD/YY	date display format
MM/YYYY	date display format

Format	Description
DD/MM	date display format
DDMMYY	date display format
MMDDYY	date display format
YYMMDD	date display format
YYYY	date display format
YY	date display format
MM	date display format
DD	date display format
YYYYMM	date display format
YYMM	date display format
MMDD	date display format
MONTH-NAME-SHORT	e.g. JAN, FEB etc.
MONTH-NAME-LONG	e.g. January, February etc.
DAY-NAME-SHORT	e.g. MON, TUE etc.
DAY-NAME-LONG	e.g. Monday, Tuesday etc.
DAY-NUMBER	Monday is taken as 1
WEEK-NUMBER	The week number of the input date

Format	Description
DAY-NUMBER-FROM-NAME	e.g. MON is 1, TUE is 2 etc.
MONTH-NUMBER-FROM-NAME	e.g. JAN is 1, FEB is 2 etc.

Click here to go back to the top.

Time formats

Format	Description
HH:MM:SS	display time format
HH:MM	display time format
MM:SS	display time format
HH	display time format
MIN	display time format
SS	display time format
AM/PM	whether the time is AM or PM. 120000/000000 is AM

Click here to go back to the top.

Other format options

Format	Description
MD5HASH	Calculate the MD5 hash of a value
CHARTOHEX	Convert a character string to character hex representation
HEXTOCHAR	Convert a character hex representation to a character string

Click here to go back to the top.

Character mask (CMASK)

FORMAT FIELD1 WITH CMASK(...) INTO FIELD2

Format a character string into the designated pattern from left to right. The input field (FIELD1) can only contain any alphanumeric value.
The character mask string can contain the following:

- "_" is a fixed place marker, is always filled from FIELD1.
- "?" is an optional value mask. Each variable mask occurrence (contiguous group of ?) is trimmed of spaces from front and back.
- "!" indicates that the masked character is unwanted and you should drop from the string.
- The mask can contain additional text containing none of the above.
- \ is used as the escape character.

4.7.10.7 Examples

Input	Mask	Result
102306	_::_	10:23:06
120	The altitude is ?????? feet	The altitude is 120 feet
Abcde	<???>	<Abc-de->

Click here to go back to the top.

4.7.10.8 Numeric Mask (NMASK)

FORMAT FIELD1 WITH NMASK(...) INTO FIELD2

Format a numeric value into the designated pattern. The input field (FIELD1) should only contain values (0-9,+,-,',comma,dot), support signed numeric values with decimal places.

> ***Note:*** The input numeric string is treated exactly as is with no special treatment of decimals. It is the responsibility of the developer to pre-format the input as to the sign, and decimal places.
> The numeric mask string can contain the following:

- "_" is a fixed place marker i.e. is always filled from FIELD1.
- "?" is an optional value, preceding and trailing zeros are compressed out as long as there is no numeric value (1-9).

  > ***Note:*** Single "," and * * "." grouping and decimal values are also compressed out.

- "#" is the sign. If negative, its value is "-". If positive, it is ignored (compressed out). If a negative value is input and there is no sign placeholder the sign is ignored.
- The mask can contain additional text containing none of the above.
- \ is used as the escape character.

Examples:

Input	Mask	Result
123456789	__,_,_._	1,234,567.89
012345	__,_,_._	0,123.45
-123456	$?,?,???.__#	$123.45-
120345600	?,?,???.__#	1,203,456.00
012034500-	#?,?,?.	-120,345
-123.4500	?,?,?.#	123.45-
-123.4500	$#?,?,?.	$-123.45

Input	Mask	Result
123.4500	$#?,?,?.	$123.45

Click here to go back to the top.

Saving and Comparing Fields

It is often useful to save one or more field values on a screen and then compare them to see if anything changed. Two procedure commands are available to do this:

- save field group &1
- compare field group &1
- restore field group &1

Both commands require a field group reference as input, save all screen elements associated with the field group that is set up in the function workbench. When doing a save, all the values are stored in memory and then you may compare with the current values in the field group. You may only save, compare and restore a field group within a function. The restore command resets the current values of the screen elements from the last saved definition.

UTC Time Zone Calculations

You may use the calculate UTC procedure command to provide the ability to calculate client & host UTC date and times to fully support time zone processing. There are internal ($) system and (&) substitution variables available to provide the current UTC offset of the MEAP Server and Secure Container and you may use these to calculate the correct host and client times respectively.

The procedure command “calculate UTC &1 &2 &3 &4 giving &5 &6” is added to calculate the actual UTC date and time from a given date, time and offset.

calculate UTC v_date v_time v_utc_offset false giving v_utc_date v_utc_time

In the above example:

- The input v_date and v_time parameters can be directly from database data, static values, field references with default value substitution variables (&date, &time, &cldate, &cltime) or direct reference to internal system variables ($date, $time, $cldate, $cltime).
- The v_utc_offset parameter is in the format “+-hh:mm”, offset from UTC 0 (GMT).

  > ***Note:*** The “:mm” component is optional and the “hh” component may be a single digit, for example, “+8”, “+04”, “-1:30”.

- The true/false indicator (&4) dictates whether you should subtract the UTC offset from the current date and time, or incremented (default).
- The target v_utc_date and v_utc_time fields contain the new calculated values.

Table Commands

You use table commands to manage and manipulate function table controls, maintain values, loop through the table data, and alter attributes. The important things to realize about table processing is that when a row is referenced, the contents are automatically copied to and from the table work fields and you should always use these in procedure processing; the visible portion of a non-work table is controlled by the top row attribute and the occurs, number of visible rows. For example, when you read a table row, its data elements are copied into the work fields and the $ROW internal variable is maintained. When you update a row, the data is copied from the table work fields into the internal table data elements. If you wish to refer to the technical attributes of a table field (colour, style, hide) then you must use the special "fieldname(*)" notation, differentiate between the data processing and technical attribute processing by suffixing the field name with "(*)".

Syntax	Description
append row to &1	Append a row (work area) to the end of a table
clear dataview &1	Clear the dataview control associated with a table
clear table &1	Clear all the table data rows
clear table work fields &1	Clear the field work area (header) of a table
delete row from &1	Delete the current row ($ROW) from table &1
get active row from &1	Retrieve the field values for the current active row of a table
get dataview field &1 on &2 into &3	Retrieve a dataview filter field value
get table attribute &1 from &2 into &3	Get a tables attribute value into a field
insert row into &1	Insert a new row into a table at current row ($ROW)

Syntax	Description
loop through &1	Loop sequentially through each row of a table
loop through visible &1	Loop sequentially through just the visible/screen rows of a table
endloop	End construct for a loop through
modify row in &1	Modify the current row ($ROW work area) of a table
read row &1 from &2	Reads a specific table row into the work area
refresh table work fields for &1	Refreshes the table work area field default values
set active row for &1	Configures the current row of a table to active
set dataview field &1 on &2 to &3	Configure a dataview filter field to a value
set table attribute &1 on &2 to &3	Configure a tables attribute value
set table colour on &1 to &2	Configure a tables background colour
sort table &1 by &2 order &3	Sorts a table control

Examples:

- Looping through tables

Looping through Tables

The procedure commands 'LOOP THROUGH' and 'LOOP THROUGH VISIBLE' are used to process all the rows of a table, or just the visible portion of a table. When READ or LOOP explicitly references any table row, the $ROW internal variable is set to the current row number and the table work fields are automatically populated with the row data. If you have nested table loops, then you must save the $ROW value on each nested level if you require to use it in any MODIFY ROW/DELETE ROW operation; use SET TABLE ATTRIBUTE ACTIVEROW row_save.

If you are altering the technical attributes of a table field, for example, hide the field, then you must suffix the field name with "(*\)". This differentiates between the table work field that contains just the data value and the actual table element on the screen or scrollable data values.

For example:

The following procedure sets the colour on a field of the visible (screen) portion of the table.

```
001  loop through visible ORDERS
002       case order_status
003            when 'DELIVERED'
004                 set colour 'green' on order_number(*)
005            when 'OUTSTANDING'
006                 set colour 'red' on order_number(*)
007            when 'INPROGRESS'
008                 set colour 'yellow' on order_number(*)
009       endcase
010  endloop
```

4.7.10.9 Screen Commands

Syntax	Description
call exit &1	Call a custom exit program
call LDB action &1	Invoke a screen LDB action
call procedure &1	Invoke another stored procedure
call remote procedure &1 in function &2	Invoke another stored procedure from another function
get activated field into &1	Get the name of the activated field on the screen
get activated value into &1	Get the activated field value from the screen
get auto-refresh interval from &1 into &2	Get the auto-refresh interval on a fkey
get layout management horizontal into &1	Retrieve the layout manager horizontal setting for the screen
get layout management vertical into &1	Retrieve the layout manager vertical setting for the screen
get screen orientation into &1	Retrieve the orientation of the screen
refresh fkey &1	Re-instate a function key definition into the current screen
refresh fkey action &1	Refresh a action/fkey into the current screen
refresh menu group &1	Re-instate a menu group definition into the current screen

Syntax	Description
remove fkey &1	Remove a function key definition from the current screen
remove fkey action &1	Remove a action/fkey from the current screen
remove menu group &1	Remove a menu group definition from the current screen
set auto-refresh interval on &1 to &2	Configure the auto-refresh interval on a fkey
set fkey attribute &1 on &2	Configure a fkey attribute
set fkey menu group &1 on &2	Configure a fkey menu group
set fkey text &1 on &2	Configure a fkey text description
set layout management horizontal to &1	Configure the layout manager horizontal option for the screen
set layout management vertical to &1	Configure the layout manager vertical option for the screen
set screen colour &1	Configure the current screen background colour
set screen graphic &1	Configure the current screen background graphic
set screen orientation to &1	Configure the orientation of the screen
set screen title &1	Configure the current screen title
substitute title value &1 with &2	Substitute a replacement value into the header title

4.7.10.10 System Commands

Syntax	Description
check application &1	Check for a native application on the client
check DOB start on &1 for &2	Check if a triggered DOB synchronization request is finished
check host connected	Check to see that the back end host is connected
check LDB start on &1 for &2	Check if a triggered LDB synchronization request is finished
classify binary object &1 with field &2 data &3	Apply classification to an existing binary object
commit work	Issue a database commit work
create binary object &1 from file &2	Create a new binary object from a input file
delete binary object &1	Logically delete a binary object
delete binary object by classification field &1 data &2	Delete a binary object using its classification
export field &1 to memory id &2	Export a field value to memory
export table &1 to memory id &2	Export a table to memory
flow trace &1	Append a entry to the flow trace
free field memory id &1	Free a field value from memory
free session memory id &1	Free a saved session from memory

Syntax	Description
free table memory id &1	Free a table from memory
get application attribute &1 into &2 for app &3 version &4	Returns a attribute of a application version
get application status of &1 version &2 into &3	Returns the status of a application (ACTIVE/LOADING/NONE)
get binary object status of &1 into &2	Returns the status of a binary object (ACTIVE/LOADING/NONE)
get system attribute &1 into &2	Retrieve a system level attribute into a field
import field &1 from memory id &2	Import a field value from memory
import table &1 from memory id &2	Import a table from memory
initialise interface	Initialise the interface management system
launch application &1	Launch a native application on the client
launch binary object &1	Launch a binary object (physical file)
launch file &1	Launch the specified file name
logoff	Logoff the current application
logoff the session	Logoff the current client session
logoff to the first application	Logoff back to the first application
restore session from memory id &1	Restore a saved session from memory

Syntax	Description
return to caller : restore variables &1 : re-format screen &2	Restore the screen that called this function. See switching applications and functions for more details.
return to function &1 app &2 ver &3 : variables &4 : format &5	Restore back to a specific application version function. See switching applications and functions for more details.
rollback work	Issue a database rollback / backout
save session to memory id &1	Save the current session to memory
send progress message &1	Send an interactive progress message to the client status bar
set flow trace &1	Toggle the host flow trace setting
set interface attribute &1 to &2	Configure a interface attribute
set language &1	Configure the header language
set message &1	Configure the header message text
set msg &1	Configure the header message text
set next application &1 version &2 function &3	At end, switch context to another application (not immediate). Refer to switching applications and functions for more details.

Syntax	Description
set next function &1	At end, switch context to another screen function (not immediate). Refer to switching applications and functions for more details.
set password &1	Configure the header password
set pop-up message &1	Indicate that its a pop-up message with a blank/I/W/E severity
set pop-up msg &1	Indicate that its a pop-up message with a blank/I/W/E severity
set rc &1	Configure the header return code
set return code &1	Configure the header return code
set sound &1	Configure the header sound
set system attribute &1 from &2	Configure a system level attribute
set tenant &1	Configure the header tenant id
set userid &1	Configure the header user id
Configure workgroup &1	Configure the header work group
shutdown	Initiate a application server shutdown (local only)
start binary object &1 for &2	Trigger a Binary Object synchronisation request
start DOB &1 for &2	Trigger a DOB synchronisation request
start heartbeat	Initiate a heartbeat call (local only)

Syntax	Description
start interface &1	Start a interface run
start LDB &1 for &2	Trigger a LDB synchronisation request
substitute message value &1 with &2	Substitute a replacement value into the header message
wait &1 seconds	Wait a specified number of seconds

4.7.10.11 Logic Commands

Syntax	Description
if &1 &2 &3	If comparision
and &1 &2 &3	and condition
or &1 &2 &3	or condition
elseif &1 &2 &3	Else if comparision for a IF command
else	Else condition for a IF command
endif	End construct for a IF command
case &1	Case construct with when conditions
when &1	When condition for a case construct
when others	Otherwise condition for a CASE command
endcase	End construct for a CASE command
do	Perform/loop until a leave command is detected
do times &1	Perform/loop the specified number of times
enddo	End construct for a DO command
while &1 &2 &3	Do while the condition is true
endwhile	End construct for a WHILE command
leave	Abort the current processing level

Syntax	Description
stop	Immediate stop of all further procedure processing
check &1 &2 &3	Only proceed if the condition is true, otherwise ignore (continue)
continue	Ignore further processing and continue on with the next iteration

4.7.10.12 Profile Commands

Syntax	Description
call data profile &1 with command &2	Invoke a data profile
clear data profile attributes	Clear the data profile attributes table
set data profile attribute &1 from &2	Configure a data profile attribute value

4.7.10.13 Internal Variables

You may reference these internal variables in any command as a standard internal field, starts with '$'.

Variable	Description
$APPID	Current application ID
$VERID	Current application version ID
$FUNCTION	Current function name
$BUILD	The work bench build information of the current function
$ACTION	Current event action
$PROCEDURE	The current stored procedure name
$LINE	The current stored procedure line being executed
$RC	The internal return code of the last command executed
$MSG	The internal message associated with the last command executed
$ROW	The current table row
$DATE	Current system date
$TIME	Current system time
$TIMESTAMP	Current date + time timestamp value
$SERVERID	Current server ID (header)
$SERVERGROUP	Current server group (header)

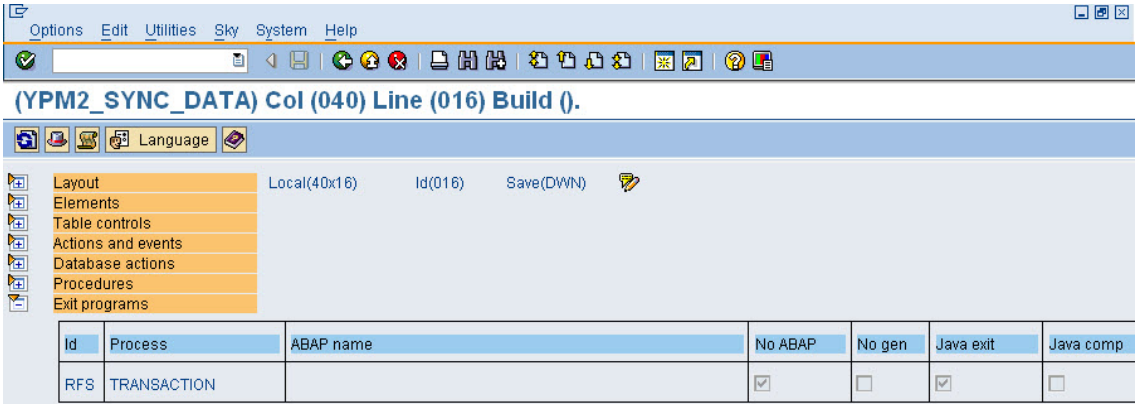
Variable	Description
$USERID	Current user ID (header)
$WORKGROUP	Current work group (header)
$WORKAREA	Current work area (header)
$SPACE	Single space
$ZERO	0
$COMMA	,
$ASTERISK	*
$QUOTE	'
$TRUE	X
$FALSE	Space
$POSITION	Offset from 0 of search found position
$SYS-RC	Header RC value
$SYS-MSG	Header MSG value
$INDEX	Current iteration of a do/while loop

4.7.11 Exit Programs Section

This section lists all the program exit definitions that are defined for the screen. Each exit is uniquely identified by a three character id. Exits are usually defined using the function attribute, element attribute and action pop-up screens; format, verify and transaction exits. You may also maintain them from this list by double-clicking the entry or positioning your cursor and using the application tool bar, in which case the exit definitions pop-up appears.

Usually exit programs are associated only with a single screen action, but it is possible to share them across multiple screens and actions. It is also possible to refer to standard object (SOF) definitions that are common sky supplied or custom utility programs.

> ***Important:*** In either case with shared programs, you must take utmost care of the screen field parameters requirements; if dependent screen field definitions change.



4.7.11.1 Overview

SkyMobile screen functions have many features to automatically process screens, based on configuration, for example, field value checking, scrolling tables, and local database controls. However, if more complex processing is required, such as to perform difficult calculations or execute an interface, functionality is provided to invoke custom programs at specific points in the screen processing cycle. You may write these custom programs in either ABAP or Java and are known as

exits. The SkyMobile ABAP and Java programming guides describe the SkyMobile SDK and available APIs. They also contain example code to demonstrate how the exit programs interact with SkyMobile. Before you may configure exit programs, you should configure a SAP exit prefix (for example, ZVTI) and/or a Java exit package name against the application version definition.

> ***Note:*** The SkyMobile SAP housekeeping routine automatically deletes the un-referenced screen exit definitions.

4.7.11.2 Types of Exit

Even though SkyMobile has no direct control over what is actually performed in the exit, it provides a concept of an exit type to help structure processing and identify the function of the exit. When you create an exit definition, you need to specify the type of exit as follows:

Exit Type	Description
Format	The exit is invoked in screen pre-processing to pre-format the screen layout and/or data, prior to display for the first time. You specify a format exit in the main function attributes (hat icon). See the Function attributes section for more information.
Verify	Validates the screen input and displays any error messages. Verify exits are specified on action definitions.
Transaction	Posts any business transactions. Transaction exits are specified on action definitions.
Field	Common routine to check/manipulate the contents of a single field. Field exits are specified on input screen field definitions.

An exit definition may also base on a standard object framework (SOF) definition. These are screen exits that are common and you may share across multiple screen functions. Refer the main documentation on the Standard Object Framework for more details.

4.7.11.3 When Screen Exits are Called

Screen programming exits are either specified on the function attributes (screen format exit), a screen field (field exit) or on a function key definition. Either the Application Server or the SAP session manager invokes all exits depending on the exit configuration. A format exit is called when the screen layout is generated, field exits are called when the screen is input and you can perform any exit associated with a function key are invoked when the function key is pressed. You may configure an exit as ABAP, Java or both; can be performed directly by the Application Server or in SAP. If you configure an action or screen function as local and an configure ABAP exit, SkyMobile detects that there are 'post local processing SAP dependencies' and calls the ABAP exit in SAP, once all the local screen processing is performed. You may configure both a Java and an ABAP exit program on the same exit definition.

In all cases, exits are called after any database (LDB) processing. Refer to The order of screen processing for more information.

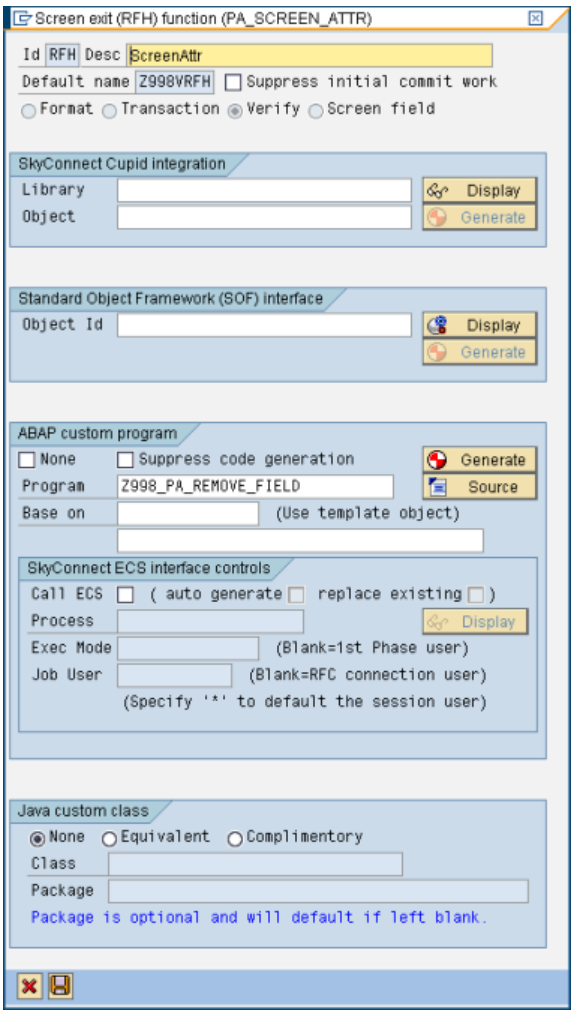
4.7.11.4 Exit Names

Each exit definition is identified by a three character abbreviation, for example, GRP (goods receipt purchase order). This three character id is specified, wherever an exit is invoked. You may also share a single exit definition across many screen functions. A default program name is generated in the form '{program exit prefix}{exit type}{exit suffix}. You may override this default name by either a Java class name or a free format ABAP program name.

4.7.11.5 Exit Definition Screen

You use this screen to create and maintain screen exit definitions for ABAP programs and Java classes. You may launch the pop-up from various areas of the screen painter, for example, function attributes (format exit), field definitions (field exit), and actions (verify/transaction). Each exit definition is unique within the screen function and you must allocate a process type; format, verify, transaction and screen field. This process defines the purpose of the exit and when it is invoked.

> ***Note:*** You may share the same ABAP programs and Java classes across multiple functions. In this case, you must take that the exit is compatible as per screen field requirements and that the 'original' (owner) screen function is the only one that can maintain it, for example, regenerate ABAP code.



Field	Description
Id	A unique three character id used to identify the exit definition.
Default name	A default program name is generated based on the program prefix (configured for the application version), the exit type (one character identifier) and the exit suffix. This name is used, if no specific ABAP or Java name is used.
Suppress commit	Suppress issuing a database commit work just prior to executing the exit. You may wish to do this to enable rollback to a previous point.

Field	Description
Exit type	The type of exit. This is used to categorize the purpose of the exit, format, verify, transaction and screen field.
SkyConnect Cupid Integration	Here, you can specify the name of a SkyConnect Cupid object to automatically trigger when the exit definition invokes. Cupid is essentially a data translator with many options to integrate with standard SAP business functions. See the documentation on SkyConnect Cupid for more information. You must configure the Cupid object with a VTI screen input data type.
Standard object (SOF)	Select a standard object to invoke as the exit. The drop-down list displays all the standard (SOF) objects available. You may define your own SOF objects; see main section on the "Standard Object Framework" for details. Once you have based your exit on a SOF object, the ABAP and Java attributes are greyed out. You may also be prompted for the input/output fields that the object; parameter mapping requires.

Field	Description
ABAP custom program	These attributes are specific to ABAP exit definition. 'None' explicitly indicates that there is no ABAP exit; otherwise a skeleton ABAP exit is automatically generated when the exit definition is generated. The automatic generation of ABAP code is suppressed using the 'Suppress code generation' option. You may specify a custom ABAP program name , this must start with 'Z'. You may use the 'Display' button to invoke the ABAP editor to view and maintain the ABAP program source. The base on attributes generates additional code to call SAP BAPIs, and function modules. SkyMobile automatically attempts to match required parameter values with the names of screen fields. Table parameters are also fully supported. You may configure the SkyConnect ECS options to invoke ECS interface processing instead of a direct synchronous call. The ECS option is only valid for transaction type screen exits. You can override the execution mode of the process, job/server/synchronous and also specify the SAP user id to use ('*' derives the user name from the SkyMobile logged on user when using user level client connections). The default is to use the current SAP RFC user, for example, CPIC_VTI. ***Note:*** When using 'server' execution mode, the SAP processing user is that of the SkyConnect background server. Refer SkyMobile ABAP programming guide for more details on SDK and how to program screen exits.

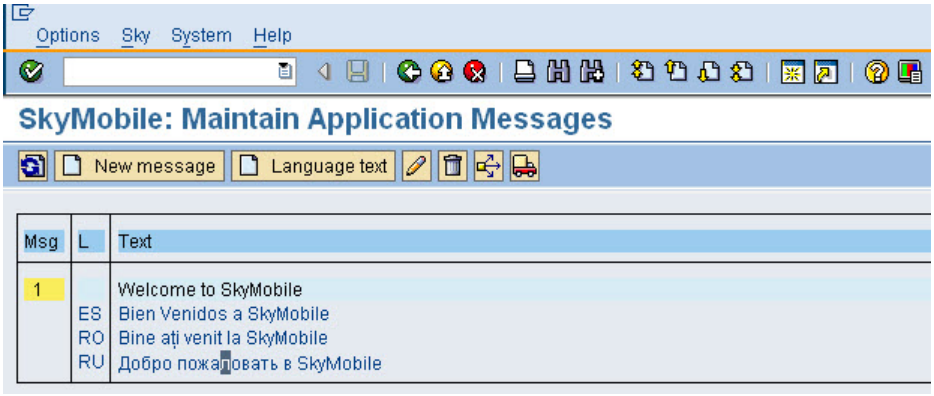
Field	Description
Java custom class	These attributes are specific to implementing a Java exit. 'No Java exit' explicitly indicates that there is no Java exit associated with the exit definition. 'Equivalent Java exit' indicates that you should use this exit in place of any ABAP exit when executing in local mode. The 'Complimentary Java exit' indicates that you should as well call the the exit as the ABAP exit (if any) when executing in local mode. The class name of the exit, otherwise the default name is used. You may also enter a specific package name for the class. If you leave blank, the package name defined at the application version level is used. This enables you to effectively load classes from multiple different Java packages. See the SkyMobile Java programming guide for more details on SDK and how to program screen exits.

Application Server 'service' Exits

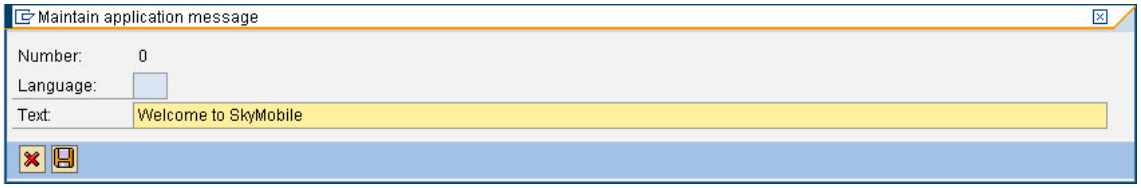
The Application Server supports the concept of 'service exits'. These special programs start when the Application Server starts and stops when it shuts down. These exits are used mainly for system tasks, for example, monitoring I/O for a specific device, and polling a directory for files. See the Java programming guide for more details.

4.7.12 Application Messages

Application messages are unique within a specific application version and are referenced from procedures and ABAP/Java custom programs through a unique number, for example, 001, and 002. You may have a maximum of 999 messages per version. You may launch the Application message editor through the procedure editor (double-clicking a message number) or through the utilities menu on the main screen function builder screen. All the messages associated with the application version appear. From here you may maintain the message definition and any associated language text. You can share the same message across screens, and programs.



To maintain the messages and language texts, simply position the cursor and select create, change or delete from the toolbar. There are two separate create buttons, one to create new messages and another to create language texts for an existing message. You may simply change existing values by double-clicking them. You use the following pop-up to maintain message text:



> ***Note:*** You can dynamically substitute values into messages by inserting placeholders into the text and using the 'substitute message value &1 with &2' procedure command, for example, substitute message value '#1' with order_number.

4.7.13 Switching between Applications and Functions

There are multiple ways to invoke the next function and return back to the caller. You have options to start the next function within the current application/version, or branch to a function in a different application/version. You can achieve this by using the:

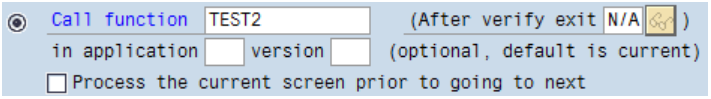
- Built-in options on the event definition
- Set next application/function, return to caller/function procedure commands
- Save/restore session procedure commands
- APIs in the ABAP and Java SDKs.

For switching to work effectively, you must understand how function data is saved and restored. Each function has a unique "save id" that is usually the unique internal three digit function ID, you specify this in the function header attributes and is what is referred to in the configuration, procedure, API. Specifying a save ID is optional, but highly recommended. When you call another function, the current functions state is saved in a memory stack and is restored when a 'return to caller' operation is invoked. There are extra options to control whether field values get copied, and whether the format processing is invoked again. Because of possible saved session dependencies, you must be aware of the flow when mixing local and host functions, a session that is saved locally is not available for a host restore and vice versa. You may also perform your own save/restore session handling using procedure commands.

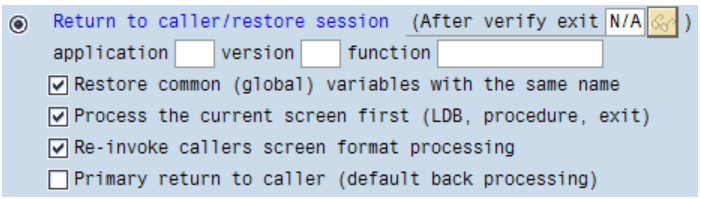
4.7.13.1 Event/Actions:

One of the most simple mechanisms to switch functions is to use the built-in event options to "call function" and "return to caller". These are static and are defined on the event/action attributes.

You use the following section to call another function either in the same application version, or another.



You use the following section is to return to a previously saved definition. By default, this is the last caller, but you can direct to another function in the same or another application/version.



4.7.13.2 Procedure Commands

You may use the following procedure commands to dynamically set the next function or return to caller:

- Configure next function (name)
- Configure next application (appid) version (verid) function (function)
- Return to caller: restore variables (true/false): re-format screen (true/false)
- Return to function (function] app (appid] ver (verid): variables (true/false): format (true/false)

> ***Important:***
> You may make the application, version and function totally dynamic using work fields. When you return to caller, all saved session beyond that point are freed from the saved stack to conserve memory.

4.8 Data Storage and Processing

4.8.1 About

This section highlights the SkyMobile functionality used to store database data and binary objects both centrally on the MEAP server and locally on the device. It discusses how to use the various workbenches to define the structure of data objects, synchronization rules, how the binary repository works and how definitions are managed.

> ***Note:*** Kony recommends that you use data objects associated with data profiles for all database data and binary object synchronization. While you perform LDB and binary object synchronization independently, data objects provide the simplest and most efficient mechanism.

Key Topics

Local Database Tables
Data Objects
Binary Objects

4.8.2 Local Databases (LDB)

Kony for SAP fully supports the definition of its own SQL database tables, in which to store application data either on the MEAP Server or locally on the Secure Container. You can also use Local database definitions (LDB) to define data synchronization (download, refresh and upload) rules and schedules. The Secure Container automatically create and maintains them on the device that supports its own inbuilt database (SkyDB) and JDBC compliant SQL database, for example, SQL Server, and Oracle. You may develop custom download, refresh and upload ABAP function modules using the SkyMobile ABAP SDK to perform specialized extractions and updating SAP data. You may also imbed these definitions into screens in-order to perform automatic selection and maintenance functions. This built-in capability eliminates having to develop program code and helps to considerably reduce the complexity of applications.

The local database workbench enables developers to define database table definitions and rules and logic for transferring data between the Secure Container on the device, the central MEAP Server and back end host systems. In the case of a SAP MEAP Server, the local database manager is fully integrated with the SAP data-dictionary, allowing tables, views and structures to reference. Secure Containers use these local database (LDB) definitions to dynamically create local tables and synchronize data. In addition, you may imbed LDBs directly into screen functions to perform the automatic selection and modification of data. This further eliminates the need for program code and procedural logic.

4.8.2.1 SAP Dictionary Definitions

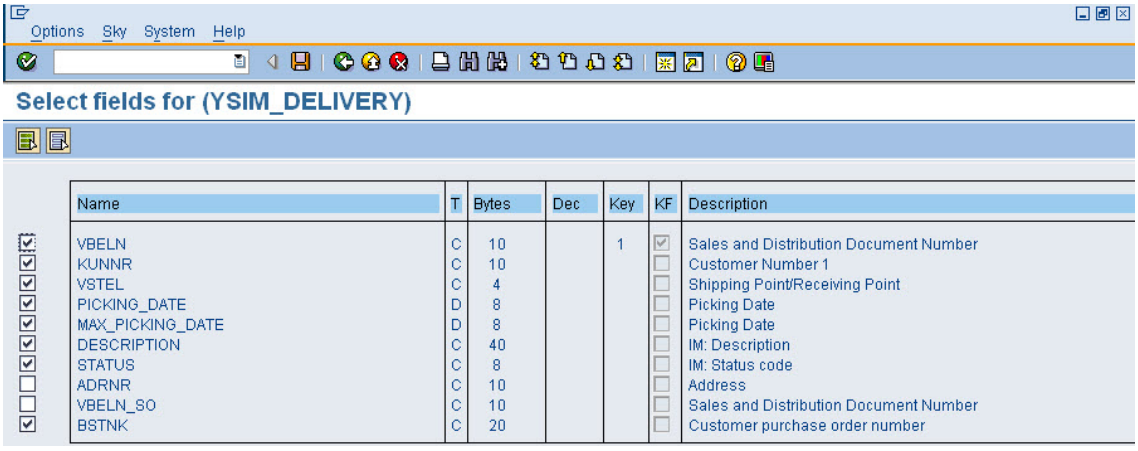
All local database definitions are based on a single SAP data dictionary table, view or structure. Therefore, the SAP data dictionary is used to define all base field definitions. Even if the LDB is not used in SAP, you must define in the SAP DD as a custom table or structure. You must not use SAP DD definition all fields in the LDB, thus helping to restrict the volume of data stored and transferred to a local system. You can also use LDBs directly in SAP; through applications that are executed using the SAPGUI emulator. This feature is useful for the development and testing of mobile applications and for production applications to run through the SAPGUI.

4.8.2.2 Data Objects (DOBs)

One of the biggest problems with database table level synchronization is the lack of an overall "transaction", for example, if a order consists of a header, item and schedule tables, how do you confirm that all the table data is synchronized. SkyMobile addresses this issue and simplifies the overall synchronization process through the use of data objects (DOBs). Multiple LDB definitions may be incorporated into one or more data objects (DOBs). Data objects enable inter-table relationships to be defined and provide the ability to synchronize data as an object; a complete (whole) transaction. Be aware that if a LDB is incorporated in to a DOB, the data object's scheduling takes precedence. Note that the LDB synchronization modes and data processing mechanisms are used as the underlying database methods for the data object. Think of a data object as a 'wrapper' around multiple LDB database tables. Refer the section on data objects for more information. All data profiling, determining what data is synchronized with which remote Application Servers, is configured at the data object level using the Data Profile manager.

4.8.2.3 Including Fields

When you create a LDB, you are automatically prompted to create the fields. Otherwise, the LDB fields are maintained by positioning the cursor on an existing LDB definition and selecting the 'fields' icon on the application toolbar. The following field selection screen appears:



You may select or de-select the fields by selecting the associated check box.

> ***Note:*** The field lengths appear in terms of bytes and not number of characters / digits. Application toolbar functions are also provided to select/deselect all fields.

4.8.2.4 LDB Hints and Tips

Local Database (LDB) definitions are created and maintained using the LDB Workbench in the MEAP Server. It is used to define fields, keys and data transfer rules. The Secure Container automatically creates and maintains these definitions as required; if LDB definitions are associated with an application or configured to maintain on all or a specific container. You may use the LDB definitions in back end host applications. After creating LDB tables, the Secure Container then invokes the download, refresh and upload rules for automatically synchronizing data with the back end host in a 'pull' operation, or by 'pushing' data out to the device from the host using the heartbeat command queue.

> ***Note:*** If the LDB is part of a data object definition (DOB), then the download, refresh and upload options of the DOB override those of the child LDB.

As well as supporting Sky's own in-built database (SkyDB), Secure Containers on Windows and UNIX platforms conform to the Java JDBC specification. Thus, you may use any JDBC compliant driver and/or SQL database. The options for this are located in the \Storage\ section in the Container configuration file. Sky recommends using its own in-built SkyDB. Refer to the database section in the installation guide for more details.

The Container partitions LDB tables for every host connection definition; you may share tables between different host environments. You should not share the same database between Containers. In the same way, that applications are automatically synchronised, LDB definitions are time stamped and changed definitions are automatically downloaded and implemented. How much the Secure Container can do however, depends on the database and its level of JDBC and SQL compliance, for example, some databases do not let you alter field lengths using SQL commands or drop existing fields that are in the primary key. The aim of the Container is to fully automate this maintenance, but there are always areas where you may require manual intervention.

Data Classification

You may nominate a LDB field for classification; provide the ability to track transfer operations based on its values. For example; If you wish to track which Secure Container processed certain orders, you can classify the LDB order number field and then use the classification list screen to view when the order is downloaded, refreshed, and uploaded. Refer to changing field attributes for more details on how to apply classification to a field.

Automatic Number Range Tracking

You may associate a field directly with a number range definition so that its latest maximum value is tracked whenever a upload of data takes place from the secure contained to the MEAP. This is useful to ensure that the last known maximum value is used if the containers environment is rebuilt at any stage. Refer to changing field attributes for more details on how to apply number range tracking and renumbering.

Primary Key Buffering

By default, the MEAP Server LDB processor saves the entire data set and then returns a buffer at a time, or everything if no buffer is specified; all the data is selected at once. This is efficient when processing small to medium volumes of data, but can be inefficient when processing large volumes. In these cases, you may use this option to select a buffer of data, save the primary key position and then restart greater than this position at the next call. This means that the MEAP Server never processes or stores more than one buffer full of data at a time, thus avoiding having to store the entire data set. So, when processing large volumes of data on download/refresh operations (1K+ rows), Kony highly recommends using the "primary key buffer selection" option in conjunction with the "transfer buffer" configuration on the Container. Whilst still requiring the same number of calls to the host, this avoids having to select and store all the data on the first call. You can crudely calculate the amount of memory using the following formula: (("fields" * 57) + "object data") * "rows", where object data is an allowance for long fields that exceed 40 bytes. Efficient indexes are vital for performance and use the LDB trace statistics for fine tuning.

Lock/Enqueue the Request

This option effectively single threads the LDB operation; only one can run at a time. You can use in conjunction with screen LDBs and custom ABAP to ensure that updates and retrievals are properly synchronized across all concurrent host processing. Sky recommends you only specify locking to avoid definite issues as "single threading" can impact overall performance. If you cannot obtain a lock, an error is raised and further processing is ignored.

Clear all Existing Data

Instructs SkyMobile to delete all data in the receiving system prior to performing the data transfer request. Use with caution.

Perform at Every Start Up

The request is processed every time the Secure Container starts.

Automatic Data Retention

The retention option allows you to specify a retention period in days for a specific LDB (table) definition. This instructs the MEAP Server housekeeping routine to physically delete records that are older that this value from the current date , for example, if you specify a retention of 30 days, the housekeeping routine looks for records older than (current date - 30). You must have configured a "timestamp" type field for the LDB. You can do this from the workbench by double-clicking the field and ticking the timestamp checkbox. You may only have one timestamp type field for this option. If you have also configured a "logical delete indicator" type field, the housekeeping routine only selects records where the logical deletion indicator is set; is true (X).

Force Host Data Conversion

By default, the LDB sub-system detects if LDB data elements are already in the correct format. It does this by checking the first occurrence and if it is already in the correct internal format, the remaining occurrences are ignored. If you wish to force every data element to be individually checked, then specify this option. This incurs a performance overhead, particularly with large sets of data.

Dont use SAP in-Memory Processing (one off large volumes, use with caution)

By default, all data elements are stored in MEAP Server memory using internal tables. This is efficient until large volumes of data are processed that can cause the processes/threads to run out of memory or be paged out/in. This option causes internal tables to bypased and a database table used instead to store the data elements to transfer. This is slow, but is suitable when processing large data sets to help minimise internal SAP memory usage and/or WP failures. You should rather consider a primary key traversal in this case, unless it is a requirement that the entire data set is selected at once.

Logical Deletion Indicator Processing

Download/Refresh all

Only rows where the specified logical deletion indicator field is false (space) is selected. In the case of a refresh operation, any deleted rows that have occurred since the last refresh are flagged for deletion on the Java Server.

> ***Note:*** You should only specify a deletion indicator for a refresh if a timestamp is configured as well. This functionality only applies to in-built dynamic SQL. Custom functions need to implement their own processing logic.

Upload all

If a logical deletion indicator field is configured for a upload all, the MEAP Server LDB processor scans the uploaded data rows for deleted entries and then generates delete requests (D) as reply data to instruct the Secure Container to physically delete the rows once they are successfully maintained in the MEAP Server central database.

Schedule Overrides by Server

You may specify different interval and specific times for different servers by schedule overrides. Click on the **by server** button to maintain the list.

SQL+ Conditions

This facility allows you to configure additional SQL conditions for the operation. The the LDB session manager automatically generates these when a download or refresh is processed. In the case of a custom function module, the conditions are generated and passed through in the IT WHERE table. It is up to the custom code as to how they are applied. The SQL conditions support sub-queries; values derived from other LDB selections and dynamic variable substitution. See the hints and tips in the SQL conditions for more details.

Ignore Upload if no Data

Instructs the Secure Container to ignore calling the SAP LDB upload routine if there is no data to upload, thus saving unnecessary calls to SAP. You should not use this option if independent reply data is generated by the upload routine for example, reply back with deleted (D) entries.

Trigger Interface after Upload

This option automatically triggers the specified SkyMobile interface (ECS) process after the host upload processing is completed successfully. The LDB upload header and the primary key data values are exported to ECS memory and thus a ABAP program may use to perform additional processing. Refer to the sample ABAP program /sky/yvti8022 for details of how to retrieve the data.

> ***Note:*** The primary key values are as the SAP LDB processor receives and so if a custom function further filters these or manipulates, you may need to perform your own ECS start processing. Refer to the "VTI SAMPLE CONTACTS" LDB example.

4.8.2.5 How are LDBs Managed

The LDB Workbench creates and maintains the Local Database (LDB) definitions in the MEAP Server. It is used to define fields, keys and data transfer rules. The Secure Container automatically creates and maintains these definitions as required; if LDB definitions are associated with an application or configured to maintain on all or a specific container. You may use the LDB definitions in back end host applications. After creating LDB tables, the Secure Container then invokes the download, refresh and upload rules for automatically synchronising data with the back end host in a 'pull' operation, or by 'pushing' data out to the device from the host using the heartbeat command queue.

What Databases and Storage Mechanisms are Supported?

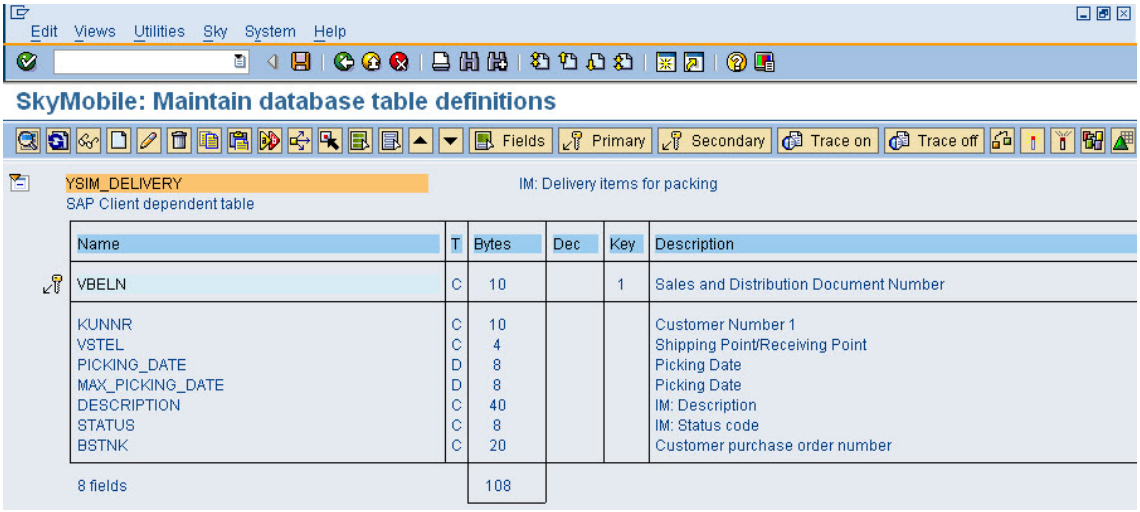
As well as supporting their own in-built database (SkyDB), Secure Containers conform to the Java JDBC specification. Thus, you may use any JDBC compliant driver and/or SQL database. The options for this are located in the [Storage] section in the Container configuration file. Sky recommends using its own in-built SkyDB. See the external database section in the installation guide for more details.

SAP Connections and Synchronization

The Container partitions LDB tables for every host connection definition; you may not share tables between different host environments. You should not also share the same database between Containers. In the same way that applications are automatically synchronised, LDB definitions are time stamped and changed definitions are automatically downloaded and implemented. How much the Secure Container can do however, depends on the database and its level of JDBC and SQL compliance, for example, some databases do not let you alter field lengths using SQL commands or drop existing fields that are in the primary key. The aim of the Container is to fully automate this maintenance, but there are always areas where you may require manual intervention.

4.8.2.6 The Local Database Workbench

You may use a standard icon: on the application toolbar to invoke the local database workbench. You may use a selection screen to restrict the LDBs that appear. To select a specific LDB to process, you position your cursor on the name and click the relevant icon on the application toolbar. You may also start directly the LDB workbench using the YVTL transaction, or through the utility and application toolbars on the main workbench and screen painter screens.



Menu Button	Functionality Description
	Maintain local database definitions, field definitions and attributes
	Define primary keys
	Define secondary keys
	Activate the trace facility

Menu Button	Functionality Description
	Deactivate the trace facility
	Test data transfer operations
	See where a LDB is used (where used)
	Generate Gateway Service

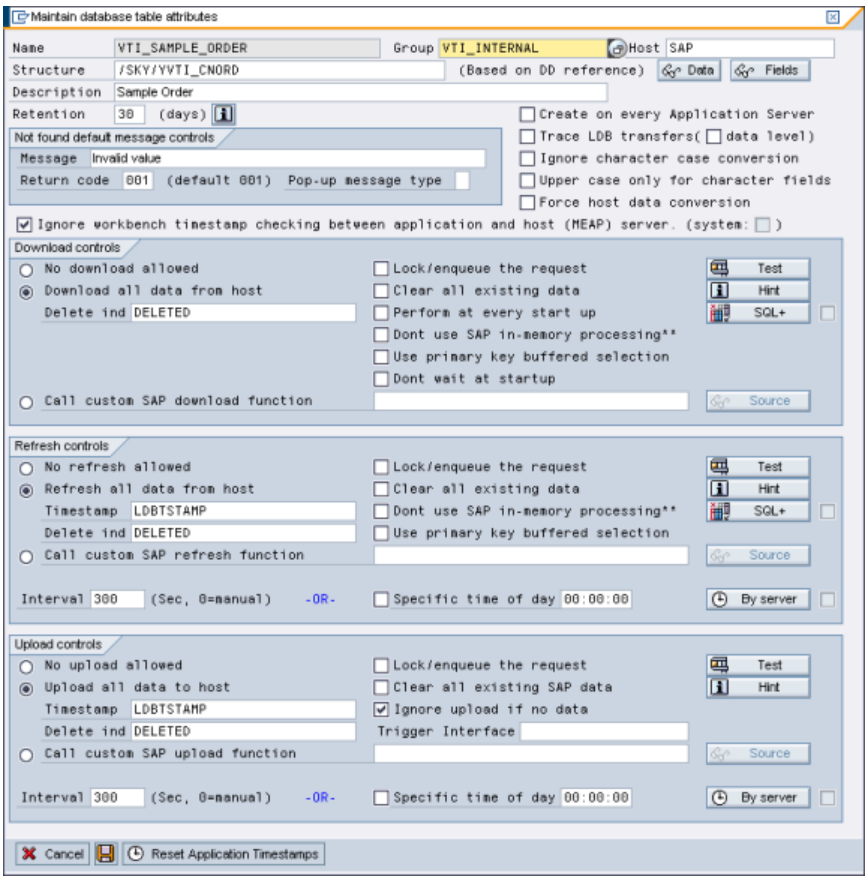
Additional functionality - Maintain replication rules

Saving Changes and Time Stamping

When you create or change a definition, it is not updated immediately to database, but only when a final save is issued. Changed definitions are flagged with a '*' that appears to the left hand side of the name. When a local database definition is finally saved, its time stamp is updated. The workbench then prompts to change the timestamps of any dependencies, for example, applications and data objects. In the case of an application, this causes the new definition to automatically download unless the application version is configured to synchronise manually.

Creating or Maintaining the LDB Definition

Click the create icon on the application toolbar, or position the cursor on an existing name and click the change icon. You may also copy or rename the existing definitions. The following pop-up appears:



This definition screen contains all the controls for the LDB; its name, description, what SAP DD table/view/structure it is based on and the data synchronisation controls.

> ***Important:*** It is important to note that if the LDB is part of a data object, the data objects download, refresh and upload scheduling attributes supersedes that of the LDB when the definition is implemented on the Application Server.

Field	Description
Structure	The name of a table, view or structure upon which the LDB definition is based. If specified, it must exist in the SAP data dictionary.
Host system	The type of host system for the LDB definition. Specify SAP if SAP is the host; central store.

Field	Description
Not found default controls	This message and return code is defaulted, whenever a selection operation fails in LDB screen processing.
Retention	Automatically delete LDB data entries whose timestamp expired past the configured number of days. If a logical deletion indicator is specified, then only records where this is set is deleted. You must configure a timestamp field and optionally a delete indicator field to use this option.
Create on every Application server	The LDB definition has to automatically download to all Application Servers. ***Note:*** Specify this only if the definition is truly generic.
Trace transfer requests (Data level)	Activate or deactivate the SAP LDB trace facility. The data trace option creates a trace file for every transfer request containing the internal header, field and data definitions.
Ignore character case conversions	By default, SkyMobile honours case conversions with LDB fields. By setting this flag, SkyMobile ignores any case conversions.
Upper case only for character fields	By setting this flag, SkyMobile treats all character fields as upper case only, irrespective of their mixed case settings.

Field	Description
Force host data conversion	By default, the SAP LDB sub-system performs a basic sample scan of the data to ensure that it is in the correct internal format. If ok, then the rest of the data set is ignored, otherwise it performs a conversion check on all the data elements. By setting this flag, you force SkyMobile to always perform a data conversion check. ***Note:*** This can impact performance when dealing with large volumes of data.
Ignore workbench timestamp checking	By default, the LDB definitions must match between the SAP host and the Application Server. If they are found inconsistent, then an error is raised and the transfer is ignored. This option suppresses the timestamp check; the application server converts the data as required. If the LDB is used at the data object level, rather use the equivalent DOB (data object) setting or configure it system wide using the SkyMobile System Manager.
Download controls	Downloading of data is designed to initially populate new table definitions with data from SAP. You may either download all data automatically from the related SAP table/view or call a custom function to retrieve the data (see ABAP programming guide). You may configure the download to occur every time the Application Server starts by setting the "Perform at every startup" indicator. By default, the start of the Application Server is delayed until all required data is downloaded. If you set "don't wait at startup", then processing continues and the download continues in the background. Delete Indicator If a LDB field is nominated as a logical delete indicator, only rows where the delete indicator is not set (not X) are selected. ***Note:*** Use the hints button on the right hand side to view help information about all of the processing options available. Also see the common descriptions below. It is recommended that you use a primary key traversal wherever possible.

Field	Description
Refresh controls	In contrast to the one off download concept, refresh operations are intended for incremental update; periodically check the SAP system for changes. You may either refresh all data automatically from the related SAP table/view or call a custom function to retrieve the data (see ABAP programming guide). You may schedule the refresh to occur every xxx seconds or at a specific time of day. If no time or interval is specified (0), the you may invoke the refresh manually from the Application Server Web status page. Timestamp If a LDB field is nominated as a timestamp control field, only rows whose timestamp is greater than or equal to the last refresh timestamp is selected for processing. Delete Indicator If a LDB field is nominated as a logical delete indicator, only rows where the delete indicator is not set (not X) are selected. ***Note:*** Use the hints button on the right hand side to view help information about all of the processing options available. Also see the common descriptions below. It is recommended that you use a primary key traversal wherever possible.

Field	Description
Upload controls	Upload operations are intended for incremental update; periodically send local LDB changes to SAP (push). You may either upload all data automatically into the related SAP table/view or call a custom function to update the data (see ABAP programming guide). You may schedule the upload operation to occur every xxx seconds or at a specific time of day. If no time or interval is specified (0), you may invoke the upload manually from the Application Server Web status page. An upload can also return data in the same way as a refresh operation to create update or delete data. If ignore upload if no data is set, the Application Server ignores calling SAP if there is no data selected. Timestamp If a LDB field is nominated as a timestamp control field, the rows are automatically updated with the upload timestamp. The timestamp field is also used to help "buffer" uploads of large data sets. ***Note:*** It is therefore very important to define an upload timestamp field when the number of records to upload are potentially large. Delete Indicator If a LDB field is nominated as a logical delete indicator, any rows that are flagged as deleted are returned to the calling Application Server so it can delete them locally. If used in conjunction with a timestamp field, only rows deleted since the last upload timestamp are selected. Trigger ECS Use this option to automatically start a SkyConnect ECS interface after a successful upload. The ECS program passes all the primary key values of the data set that is maintained. See the ABAP programming guide for more details. ***Note:*** Use the hints button on the right hand side to view help information about all of the processing options available. Also see the common descriptions below.

Field	Description
	The test facility is useful to simulate a download, refresh and/or upload operation and review the resulting data. This is particularly useful if you implemented a custom ABAP function module and you wish to debug it online. Refer to Testing LDB requests for more information.
	This provides a quick online help reference of the options available and their behaviour.
	This option provides the facility to configure SQL conditions against LDB definitions. You can use these then in the data selection process. ***Note:*** The nested calls to other LDB definitions are also supported; feed the result set of another LDB into the 'where IN' condition of the current LDB. Refer to Configurable SQL conditions for more information.
	In some cases you may wish to override the scheduling interval or time depending on the Application Server. This enables you to configure schedule overrides that take precedence over the default.
Clear all existing data	Instructs SkyMobile to delete all data in the receiving system prior to performing the data transfer request. ***Note:*** Use with caution.

Field	Description
Use primary keyed buffered selection	By default, the SAP LDB processor saves the entire data set and then returns a buffer at a time, or everything if no buffer is specified; all the data is selected at once. This is efficient when processing small to medium volumes of data, but can be inefficient when processing large volumes. In these cases, you may use this option to select a buffer of data, save the primary key position and then restart greater than this position at the next call. This means that SAP never processes or stores more than one buffer full of data at a time, thus avoiding having to store the entire data set. When processing large volumes of data on download/refresh operations (1K+ rows), Sky recommends using the "primary key buffered selection" option in conjunction with TransferBuffer configuration on the Application Server. Whilst still requiring the same number of calls to SAP, this avoids having to select and store all the data on the first call. You can crudely calculate the amount of memory you can use using the following formula: (({fields} * 57) + object data) * {rows}, where object data is an allowance for long fields that exceed 40 bytes. Efficient indexes are vital for performance and use the LDB trace statistics for fine tunning.
Donot use SAP in-memory processing	***Note:*** This option is designed for special cases with one off large volumes, use with caution and only if necessary. By default, all data elements are stored in SAP memory using internal tables. This is efficient until large volumes of data are processed that can cause the work process to run out of memory or page out/in. This option causes internal tables to bypass and a database table used instead to store the data elements to transfer. This is slow, but is suitable when processing large data sets to help minimize internal SAP memory usage and/or WP failures. You should rather consider a primary key traversal in this case, unless it is a requirement that the entire data set is selected at once.

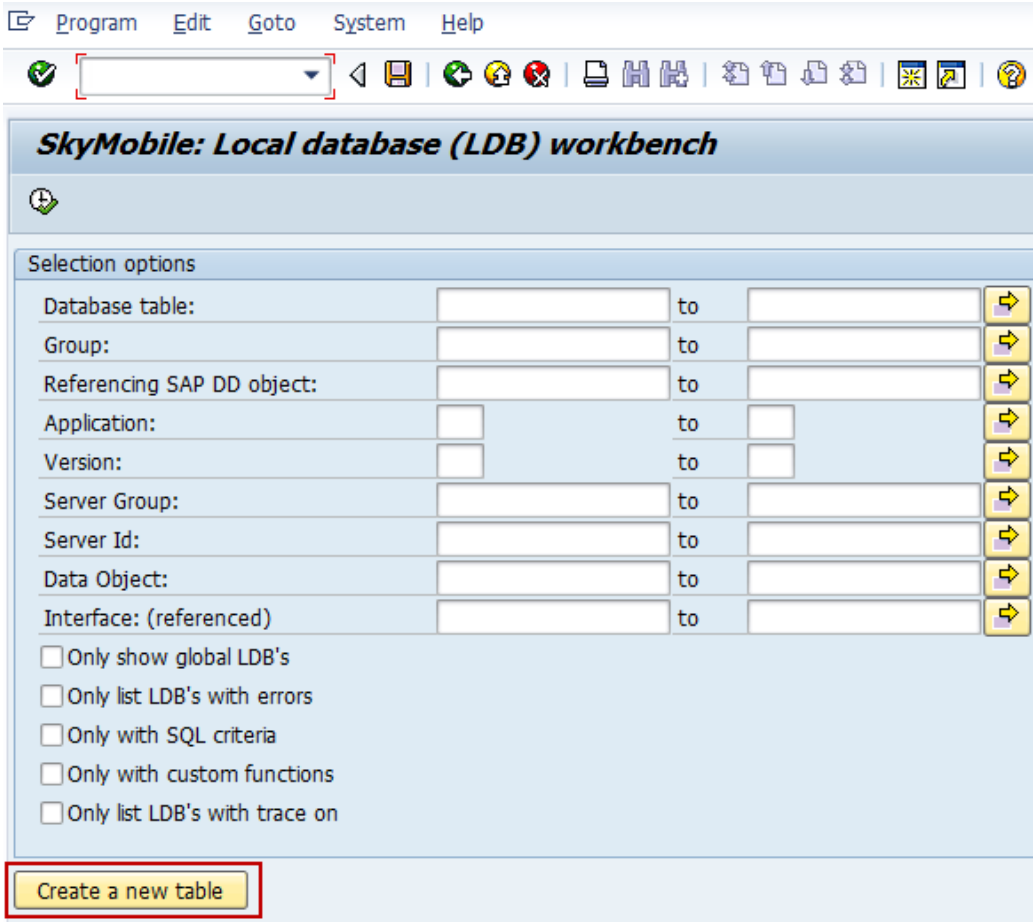
Creating an LDB Definition

In this exercise, you create a local database table (LDB) definition called SKY_CONTACTS. This LDB definition provides access to an underlying database table, and your contacts application uses to perform database actions (select, create, change and delete). This step has four parts: Creating the LDB definition, Defining the primary key(s), Setting mixed case flags and setting the timestamp and deletion fields.

Creating the LDB Header Definition

1. From the menu, choose **Utilities** -> **Local Databases** (or you can click the LDB button in the menu) to launch the LDB workbench.

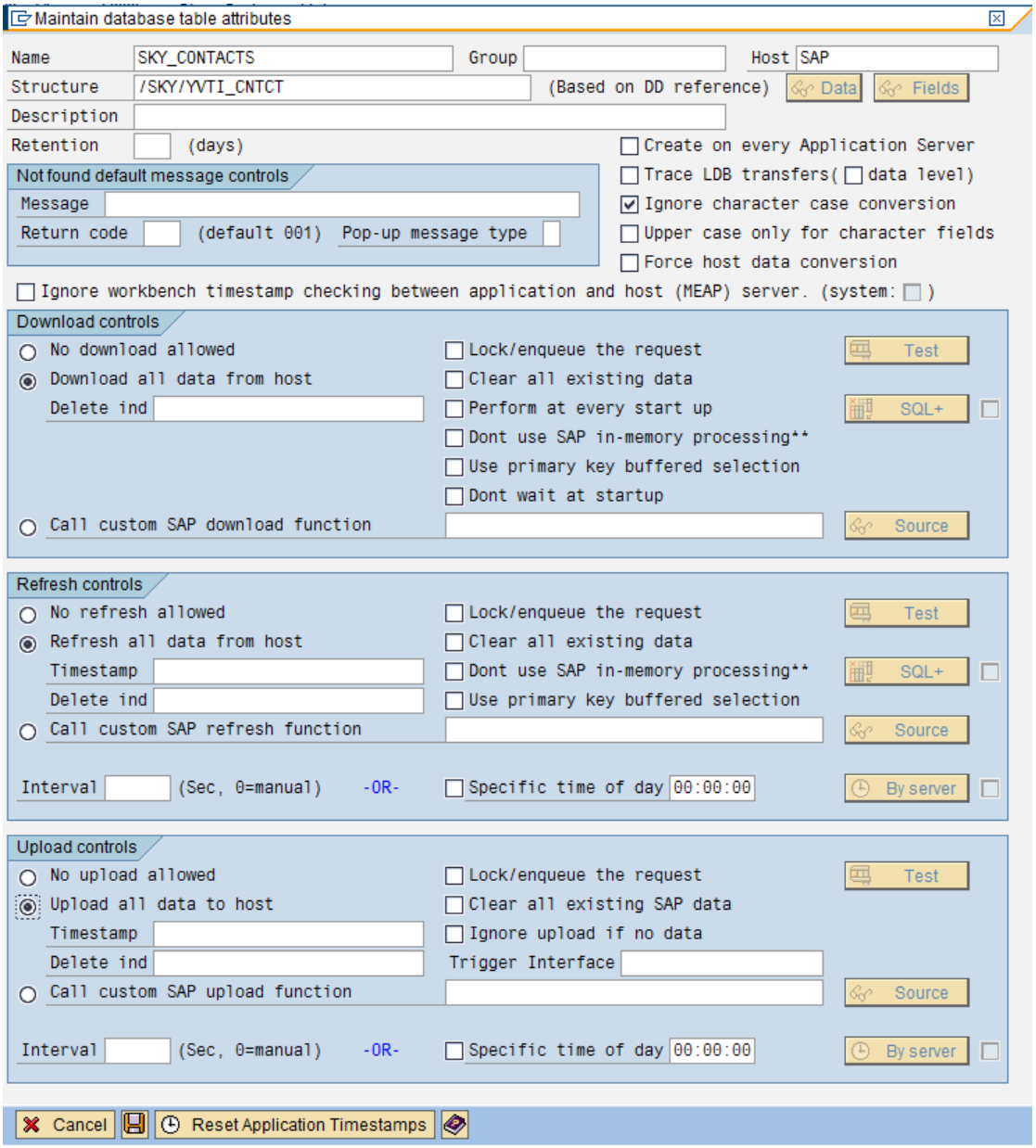
2. Click **Create a new table**.



The **Maintain database table attributes** window appears. Enter the information shown in the table:

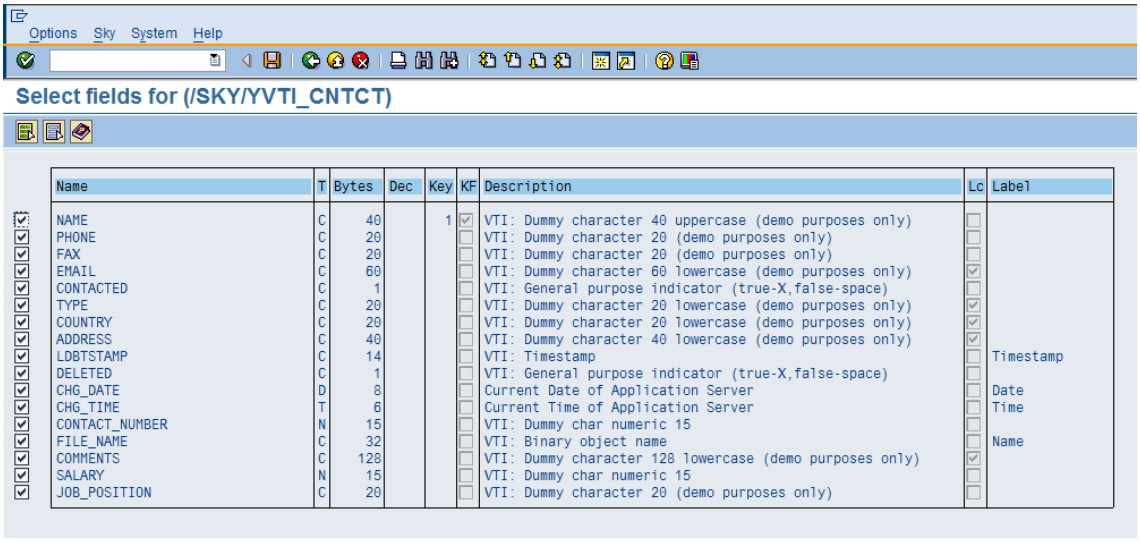
- Enter the information shown in the table below.

Field	Value
Name	SKY_CONTACTS
Structure	/SKY/YVTI_CNTCT
Download all data from host	Select
Refresh all data from host	Select
Upload all data to host	Select



> ***Note:*** The table /SKY/YVTI_CNTCT is provided as part of the SkyMobile transport. More details about this table are shown in a later step.

- You are prompted to select the fields of the underlying table to expose in the LDB. Select all fields then click **Save**.



Name	T	Bytes	Dec	Key	KF	Description	Lc	Label
NAME	C	40		1	☑	VTI: Dummy character 40 uppercase (demo purposes only)	☐	
PHONE	C	20			☐	VTI: Dummy character 20 (demo purposes only)	☐	
FAX	C	20			☐	VTI: Dummy character 20 (demo purposes only)	☐	
EMAIL	C	60			☐	VTI: Dummy character 60 lowercase (demo purposes only)	☑	
CONTACTED	C	1			☐	VTI: General purpose indicator (true-X,false-space)	☐	
TYPE	C	20			☐	VTI: Dummy character 20 lowercase (demo purposes only)	☑	
COUNTRY	C	20			☐	VTI: Dummy character 20 lowercase (demo purposes only)	☑	
ADDRESS	C	40			☐	VTI: Dummy character 40 lowercase (demo purposes only)	☑	
LDBTSTAMP	C	14			☐	VTI: Timestamp	☐	Timestamp
DELETED	C	1			☐	VTI: General purpose indicator (true-X,false-space)	☐	
CHG_DATE	D	8			☐	Current Date of Application Server	☐	Date
CHG_TIME	T	6			☐	Current Time of Application Server	☐	Time
CONTACT_NUMBER	N	15			☐	VTI: Dummy char numeric 15	☐	
FILE_NAME	C	32			☐	VTI: Binary object name	☐	Name
COMMENTS	C	128			☐	VTI: Dummy character 128 lowercase (demo purposes only)	☑	
SALARY	N	15			☐	VTI: Dummy char numeric 15	☐	
JOB_POSITION	C	20			☐	VTI: Dummy character 20 (demo purposes only)	☐	

Note:

1. You are returned to the LDB Workbench screen. Notice there is an asterisk next to your LDB definition. This indicates that the changes are not fully committed. Click the **Save** button once more.
2. You have now created an LDB definition in SkyMobile.

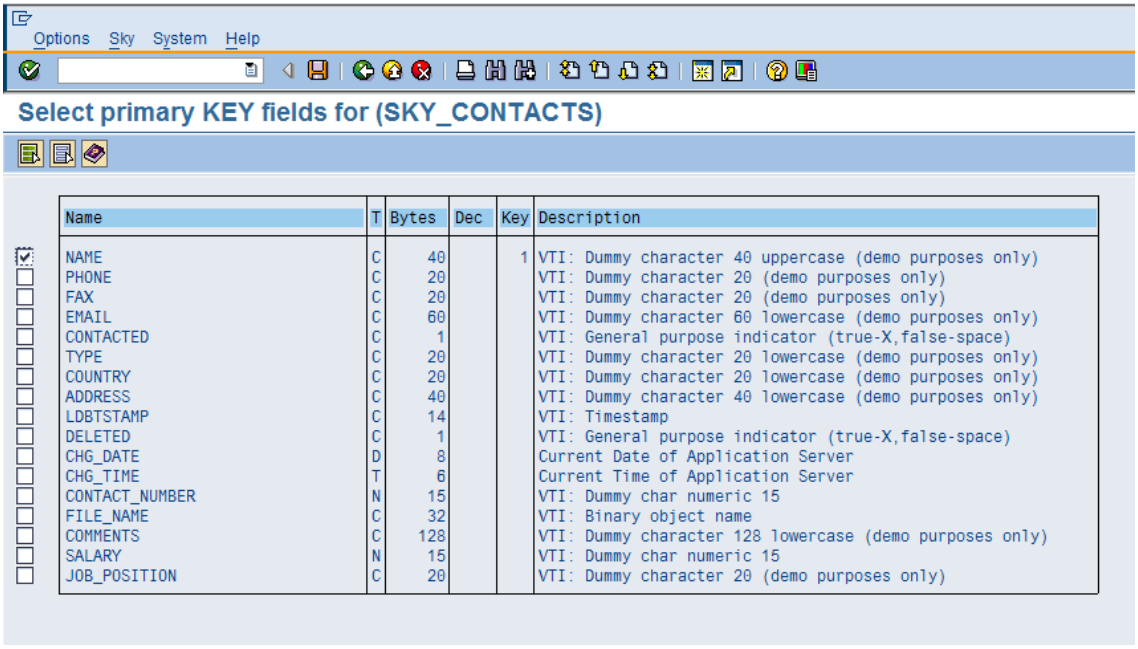
Note: The yellow highlighting of the LDB description indicates that any application does not use the LDB definition yet.

Go back to the top.

Defining the LDB Primary Key

You must define a primary key for the LDB to uniquely identify each row of data.

1. Select the LDB table SKY_CONTACTS and then click **Primary** on the toolbar icon.
2. Ensure that the **NAME** field is selected as the primary key. Click **Save**.



	Name	T	Bytes	Dec	Key	Description
☑	NAME	C	40		1	VTI: Dummy character 40 uppercase (demo purposes only)
☐	PHONE	C	20			VTI: Dummy character 20 (demo purposes only)
☐	FAX	C	20			VTI: Dummy character 20 (demo purposes only)
☐	EMAIL	C	60			VTI: Dummy character 60 lowercase (demo purposes only)
☐	CONTACTED	C	1			VTI: General purpose indicator (true-X,false-space)
☐	TYPE	C	20			VTI: Dummy character 20 lowercase (demo purposes only)
☐	COUNTRY	C	20			VTI: Dummy character 20 lowercase (demo purposes only)
☐	ADDRESS	C	40			VTI: Dummy character 40 lowercase (demo purposes only)
☐	LDBTSTAMP	C	14			VTI: Timestamp
☐	DELETED	C	1			VTI: General purpose indicator (true-X,false-space)
☐	CHG_DATE	D	8			Current Date of Application Server
☐	CHG_TIME	T	6			Current Time of Application Server
☐	CONTACT_NUMBER	N	15			VTI: Dummy char numeric 15
☐	FILE_NAME	C	32			VTI: Binary object name
☐	COMMENTS	C	128			VTI: Dummy character 128 lowercase (demo purposes only)
☐	SALARY	N	15			VTI: Dummy char numeric 15
☐	JOB_POSITION	C	20			VTI: Dummy character 20 (demo purposes only)

3. Click **Save** on the toolbar to commit the changes.

Go back to the top.

Setting the Mixed Case Flag

By default, LDB fields retrieve character table data in uppercase. In this case, you want the country name for each contact to display in mixed case. To achieve this, you need to set the mixed case flag on the **COUNTRY** field.

1. In the LDB workbench, expand the definition if collapsed so that you can see all the LDB fields
2. Double-click **COUNTRY**.

3. Ensure the Mixed case check box is selected, and then click **Save**.

> ***Note:*** When you click **Save**, the following message appears in the application status bar, "Nothing has changed, request ignored." This occurs if no value or component is modified. You may safely click **Cancel** in this instance.
>
> Repeat for the "TYPE" field. Save the LDB through the **Save** button.

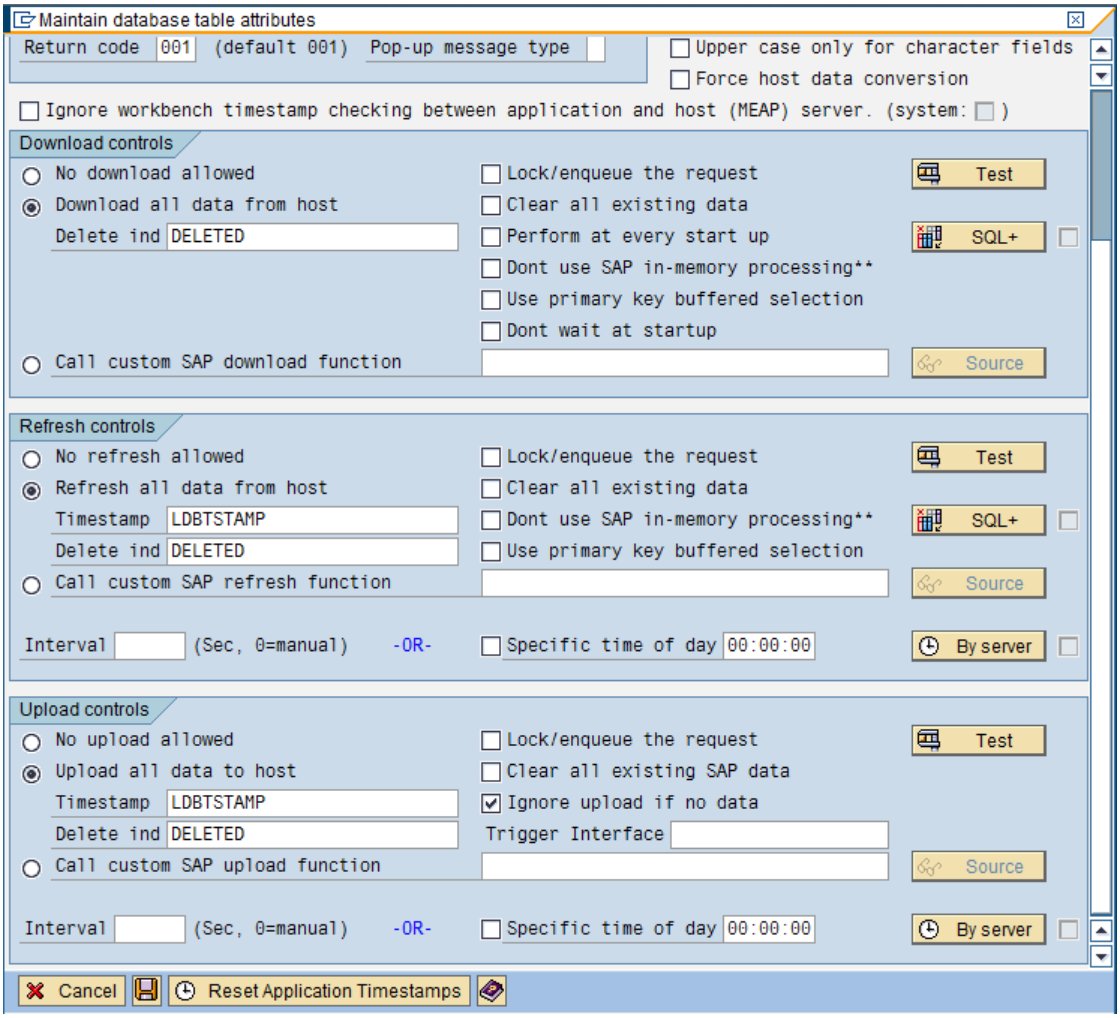
Go back to the top.

Set the Timestamp and Deletion Fields

- Double-click the SKY_CONTACTS LDB node
- The Maintain database table attributes screen appears

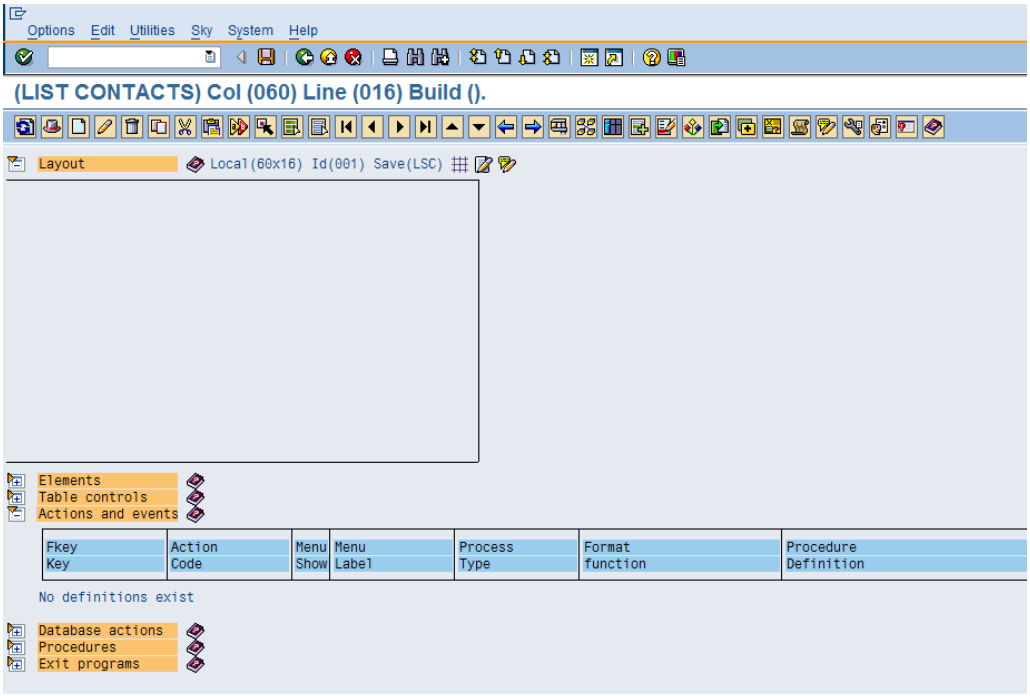
- Enter the following values, then click **Save**:

Field	Value
Download Delete ind	DELETED
Refresh Timestamp	LDBTSTAMP
Refresh Delete ind	DELETED
Upload Timestamp	LDBTSTAMP
Upload Delete ind	DELETED



Note: The SkyMobile timestamp attribute on the LDB header definition uses to detect whether any records are changed. The delete indicator is used to flag records that are logically deleted (as opposed to physically deleted). You require these two indicators to synchronise data between the SAP host and mobile devices.

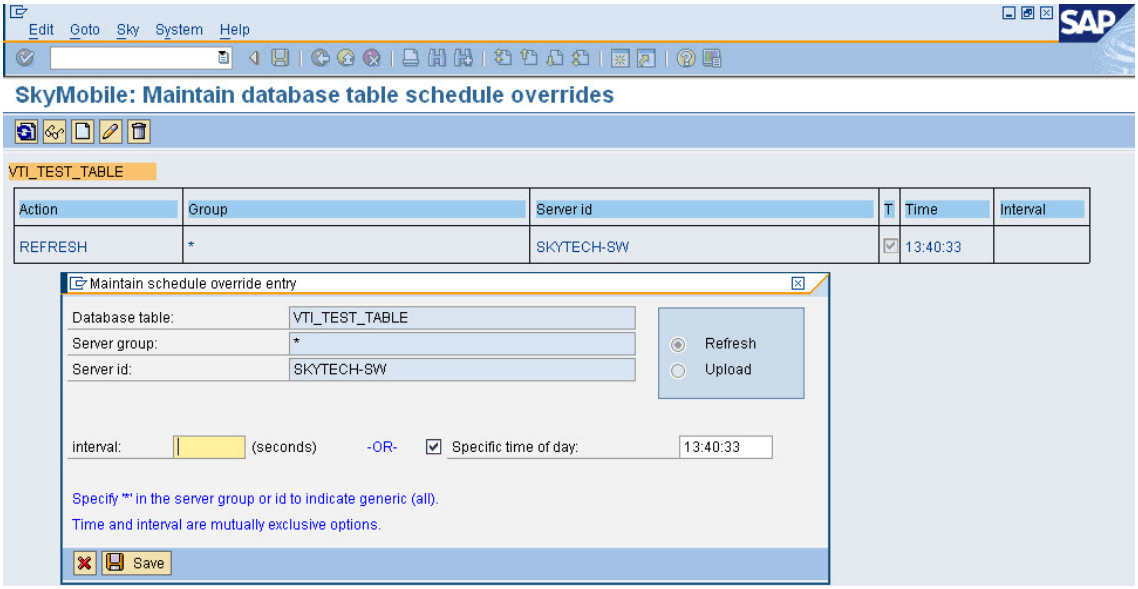
- Click Save in the LDB workbench screen to commit the LDB changes
- Click the **Back** button in the toolbar until you are back at the "LIST CONTACTS" screen.



Go back to the top.

LDB Schedule Overrides

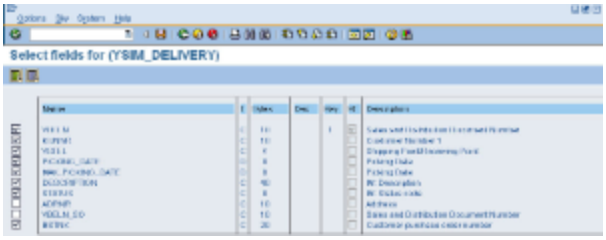
This option is selected from the main workbench utility/Application Server menu or transaction YVTS. The LDB schedule override facility enables you to configure different refresh and/or upload times or intervals for different Application Servers. By default, the configured time or interval is used for all and this can cause performance problems. This option enables you to effectively stagger data synchronisation operations so the impact on the host system is less noticeable. The utility allows you to maintain a list of override definitions.



The pop-up allows you to specify the LDB, server group/id and transfer mode that the rule is to apply to. A new time or internal is configured. When you select the LDB definition to download to the Application Server, any schedule overrides are automatically applied.

Including Fields

When you create a LDB, you are automatically prompted to create the fields. Otherwise, the LDB fields are maintained by positioning the cursor on an existing LDB definition and selecting the 'fields' icon on the application toolbar. The following field selection screen appears:

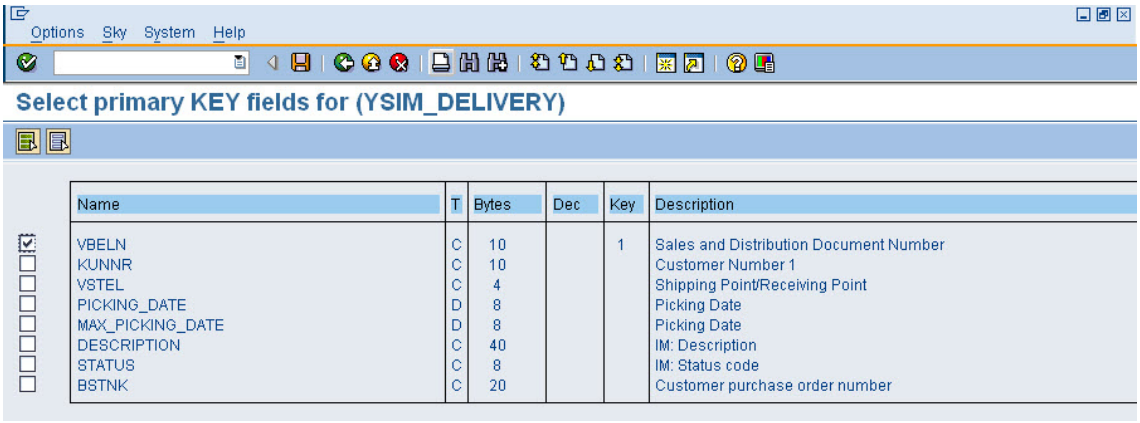


You can select or de-select the fields by checking the associated checkbox.

> ***Note:*** The field lengths appear in terms of bytes and not number of characters / digits. You are also provided application toolbar functions to select/deselect all fields.

Defining the Primary Key

You must define a primary key for the LDB that uniquely qualifies each row of data. You can maintain the primary key fields by positioning the cursor on an existing LDB definition and selecting the 'primary' icon on the application toolbar. The following field selection screen appears:



	Name	T	Bytes	Dec	Key	Description
☑	VBELN	C	10		1	Sales and Distribution Document Number
☐	KUNNR	C	10			Customer Number 1
☐	VSTEL	C	4			Shipping Point/Receiving Point
☐	PICKING_DATE	D	8			Picking Date
☐	MAX_PICKING_DATE	D	8			Picking Date
☐	DESCRIPTION	C	40			IM: Description
☐	STATUS	C	8			IM: Status code
☐	BSTNK	C	20			Customer purchase order number

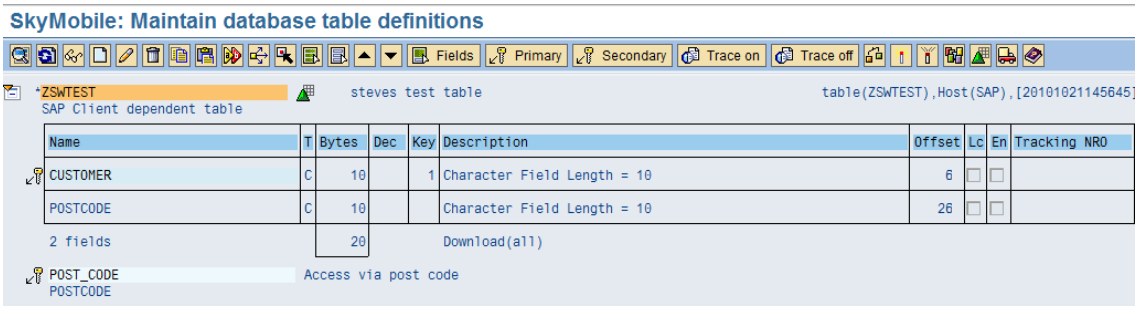
You can select or de-select the fields by checking the associated checkbox.

> ***Note:*** The field lengths appear in terms of bytes and not number of characters / digits. You are also provided the application toolbar functions to select/deselect all fields.

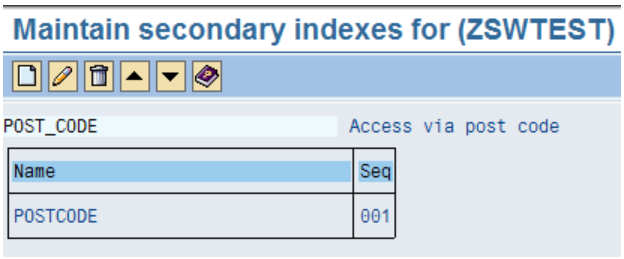
> ***Important:*** You must take care when changing the primary key of an existing database table on Application Servers. Generally, if you add fields to the end of a existing key, then there should not be a problem. However, if you are removing fields or the sequence, then you should first export the data, remove all the entries and then re-import the data after the change becomes active. Different databases handle this situation in different ways.

Defining Secondary Indices

You may optionally specify multiple secondary indices, as well as a primary key to help improve performance when accessing data. There is no limit to the number of secondary indices that you may specify. Existing secondary indices are listed after the fields.



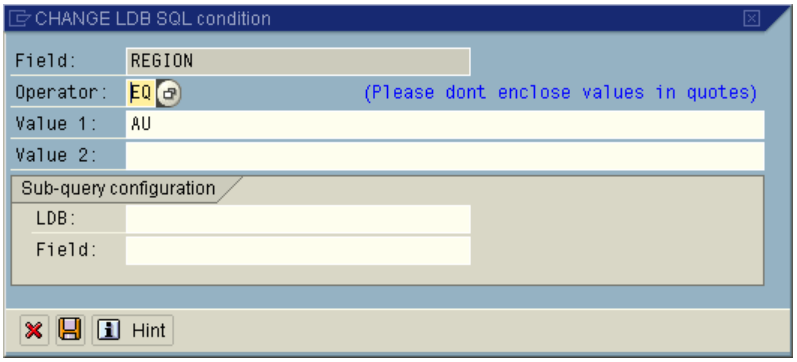
You may maintain secondary indices by positioning the cursor on an existing LDB definition and selecting the 'secondary' icon on the application toolbar or on the list. The following screen appears to list and maintain existing definitions:



From this screen, you may create secondary index definitions by using the create icon in the application toolbar. Fields are selected using a drop-down list and removed using the delete icon on the application toolbar. You position your cursor on the relevant level. You may specify the sequence of the fields by positioning the cursor on the field and clicking the up or down icons on the application tool bar.

Configurable SQL Conditions

You may define additional SQL conditions against a LDB download and refresh operation that you automatically use in the selection of data or passed onto a custom function module to process. The SQL configuration utility is invoked from the LDB definition pop-up using the SQL+ (SQL+) button. The following field condition list appears:



You maintain entries using the application toolbar or double-clicking on a row. The following pop-up appears:

This screen allows you to configure additional SQL where conditions against a specific LDB field. This is primarily to avoid writing code for simple selection criteria and to help further simplify the definition of data transfer rules. You may configure only one condition for each LDB field. Each condition comprises of the LDB field to apply it to, the operator and value to check; the basic building blocks of a where condition, for example, NAME EQ 'SMITH'. More complex options include "IN", where you may check a comma delimited list of values or the use of a sub-query "QI/QN" that derives the values to check from another LDB table and dynamically builds a IN/NOT IN list.

NOTE:

- Conditions only apply to SAP host download and refresh requests
- Every condition is treated as an AND (all must pass)

- Sub LDB queries execute the "download" operation of the specified LDB
- If sub LDB queries return no data, then the whole request is ignored (no data)

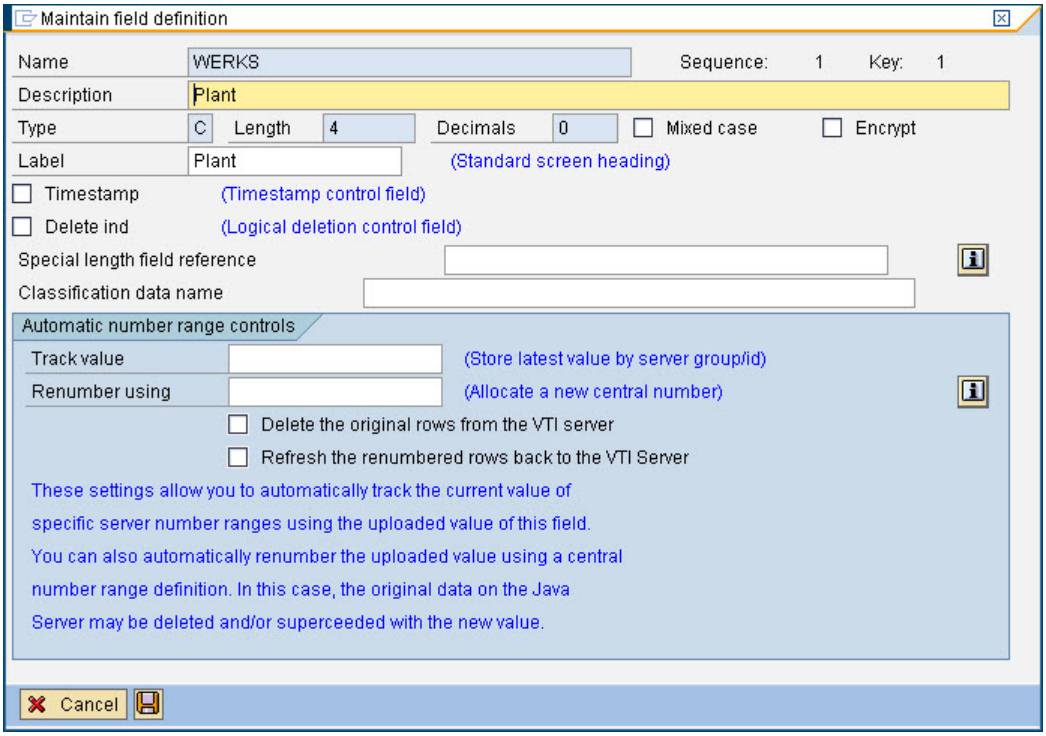
Field	Description
SQL operator	EQ - Equal to BT - Between two values NE - Not equal to NB - Not between GT - Greater than IN - Exists in the comma delimited list GE - Greater than equal to NI - Not IN LT - Less than CP - Contains Pattern (like) LE - Less than equal to NP - Does not contain pattern (not like) QI - The field selected must exist in the values returned by the sub-query (IN) QN - The opposite of QI; must not exist in the sub-query set (NOT IN).

Field	Description
Values	You may require one or more value definitions depending on the type of operator that you use. For example, for simple comparisons such as EQ,GT,LT, and NE, you require only one value; but for range tests such as BT (between) and NB (not between), you must specify two values. List operators such as IN and NI (not in) require a single value definition consisting of a comma delimited list of values, for example, "A,B". As well as static values, dynamic substitution variables are supported that are populated at run time. The supported variables are: &DATE - System date &TIME - System time &TIMESTAMP - System date/time &LTIMESTAMP - Last timestamp processed &PLATFORM - Device logical platform &SERVERID - Application Server ID &SERVERGROUP - Application Server Group &USERID - Logged on user

Field	Description
LDB Sub-queries	LDB sub-queries are used to derive values for conditions from other LDBs. You use the QI/QN operators to indicate this. As well as the LDB name for the sub-query, you need to configure the field name from which to return the data. In this way, you may build simple dependency chains. For more complex relationships, use a LDB data object definition. If the query returns no data, the request is failed. The sub-query executes the download operation of the specified LDB. You use any SQL conditions and "delete" indicator configuration defined against the sub-query LDB to dynamically generate the sub-query where statement.

Changing Field Attributes

In most cases, you copy fields from a SAP DD definition. If the LDB is not associated with a structure, you may add your own fields. Either way, you may maintain extended field attributes by double-clicking the field definition. The following screen appears:



Field	Description
Encrypt	This configuration causes the Application Server to compress and encrypt the data when it is stored. You may use this to secure sensitive data such as passwords.
Label	This is the default field label that you may automatically copy into a screen as part of a clipboard copy operation.
Timestamp/ Delete indicator	Indicates that the field is used as either a timestamp or logical deletion indicator. This configuration enables the workbench to automate the building of screen LDB operations, for example, prompt for timestamp field maps.

Field	Description
Special length field reference	This option allows you to define a "length field reference" on a field. This is to support SAP LCHAR (long character) fields that require a small integer length field (INT2) to define immediately prior to the field. This is primarily to support variable length text fields in SAP, where you must populate the length field with the string length of the LCHAR value. In SkyMobile, these field types are mainly used to support text boxes that have a variable length. The length field reference must be of type 's' (small integer INT2). The field itself must be of type character. The length field must immediately precede the LCHAR (variable) character field.
Classification data name	The LDB data classification mechanism enables all data download, refresh and upload operations to be effectively tracked by specific elements of data, for example, keep track of a specific purchase order number. Refer to LDB data classification section for more details.

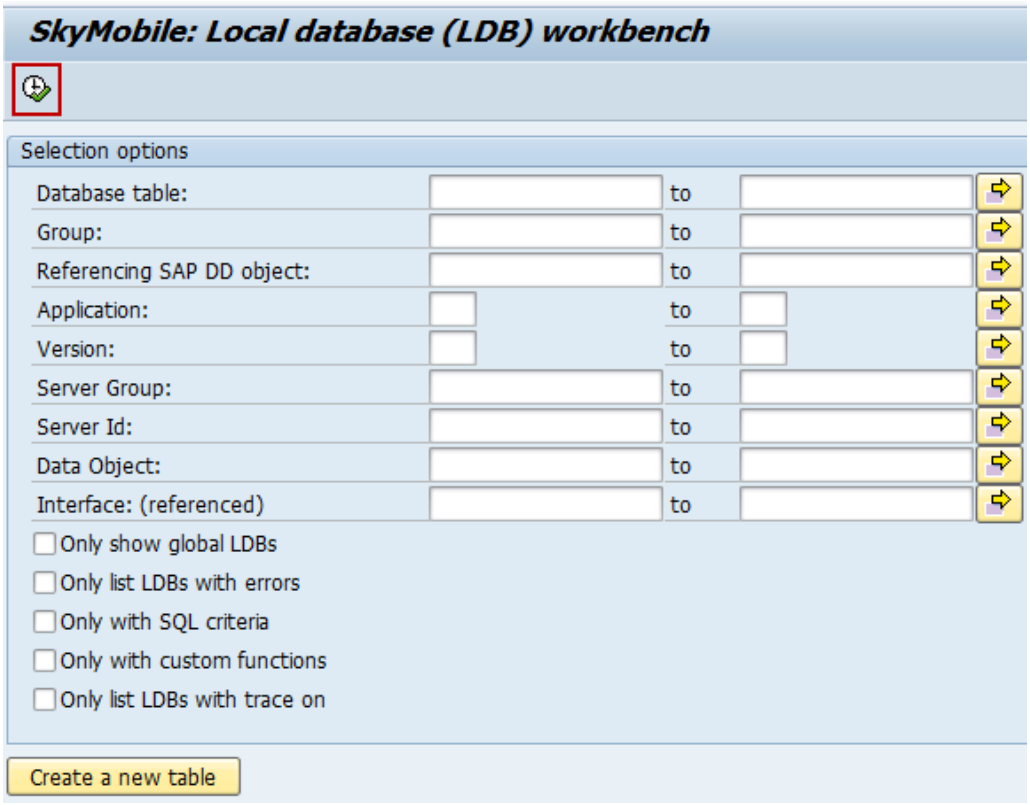
Field	Description
Automatic number range controls	You use these configurations to automatically monitor and change the field data in association with a number range object. Tracking number range values: This control saves the max value of the field on a upload request for a specific Secure Container group/id/number range combination. This is so that the last known values are saved centrally at the host and you can restored through a download/refresh if required. If this is not done, you must use the Application Server backup/restore option to make sure there is a good restore point for the number range. This option may not be suitable if there is a possibility that locally saved data may not have made it to the host. Renumbering number range values: This control is used in cases where you wish to centrally renumber a key field of an uploaded row. This is useful in cases where multiple remote Application Servers create transactions using a local number range, and you need to collect these into a central table on the host. Further attributes to this control enable you to issue delete requests back to the calling Application Server to remove the original rows from its local database; and optionally return the new renumbered rows back to the calling Secure Container. You may nor require the refresh if a scheduled LDB refresh returns the data using a last used timestamp.

Working with a Text Description for an LDB Table

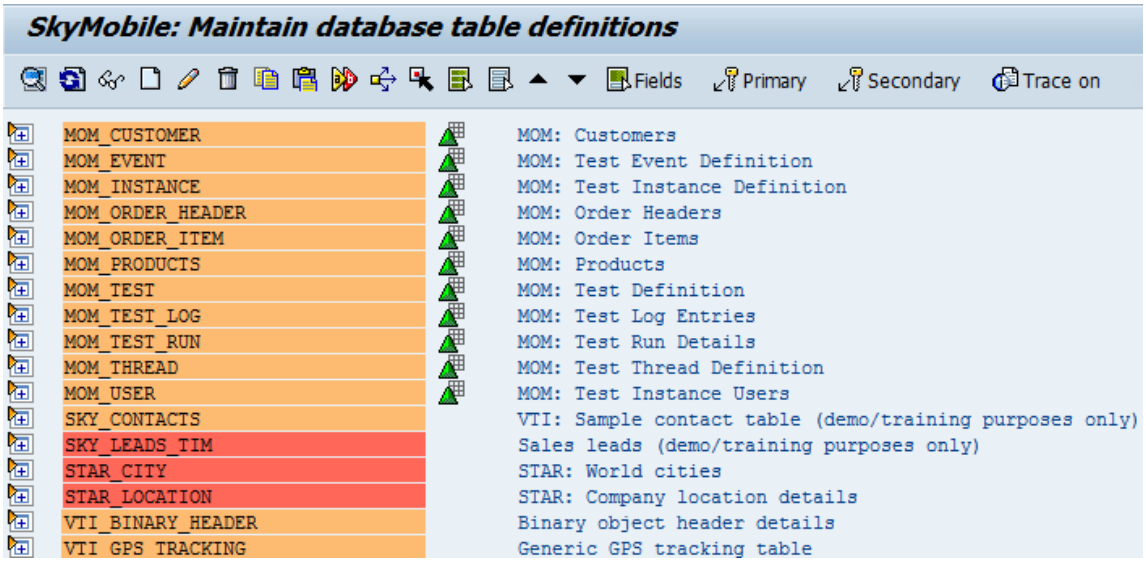
You may add extended text descriptions for Local Database Tables (LDBs) or field names.

To work with text descriptions for an LDB table, follow these steps:

1. Execute the transaction, **/SKY/YVTL (YVTL)** in SAP system.
   The **SkyMobile: Local database (LDB) Workbench** dialog appears.

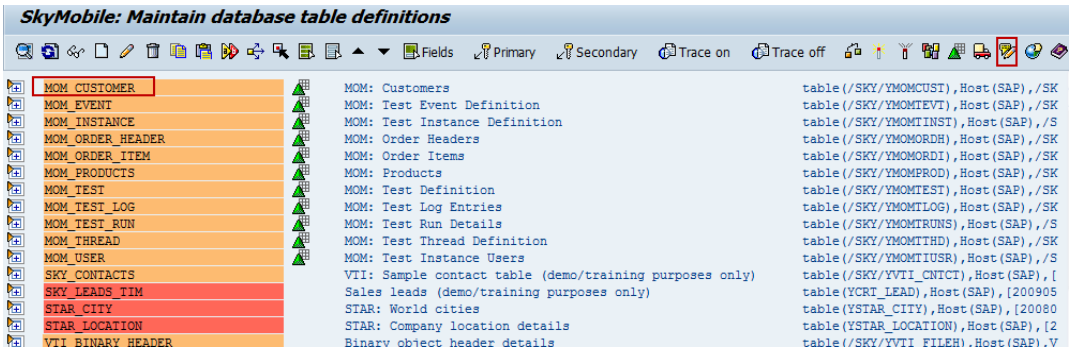


2. Click **Execute**.
   The **SkyMobile: Maintain database table definitions** dialog appears with the list of local database tables (LDBs).

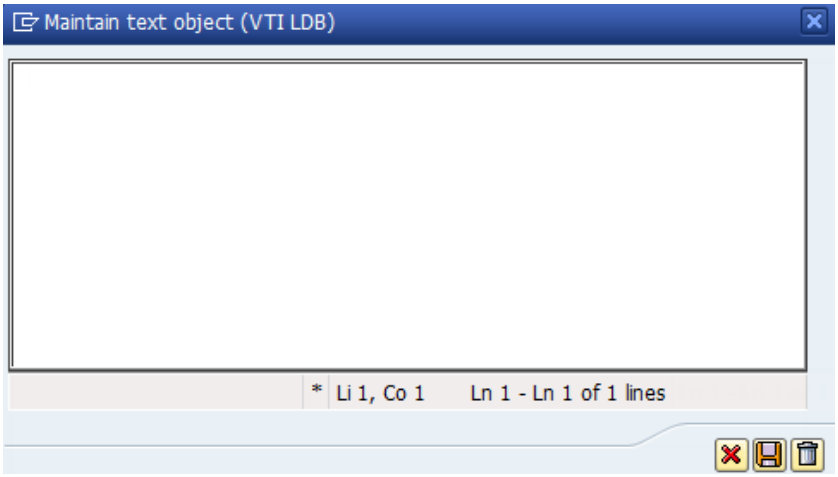


3. To create a text description for an LDB table, follow these steps:

   a. Select an LDB table from the list and click **Maintain text (Ctrl + Shift + F5)**.

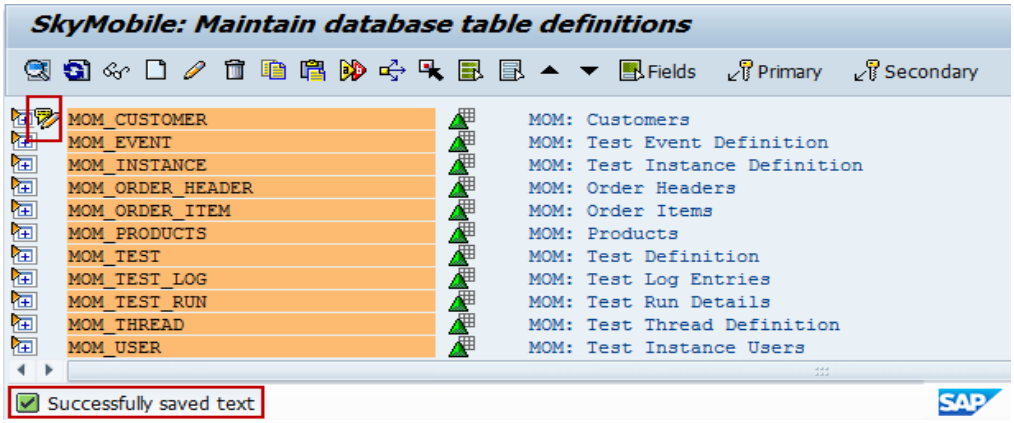


   The **Maintain text object (VTI LDB)** text editor appears.



b. Enter the text description that you want to add to the LDB table and click **Save**.

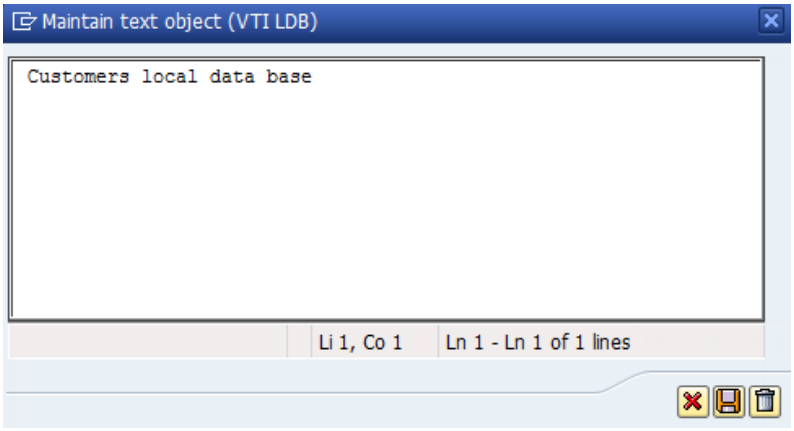
The system saves the text description.
The **Maintain text** icon appears next to the selected LDB table name. The message, "Successfully saved text" appears at the bottom of **SkyMobile: Maintain database table definitions**.



4. To change the text description of an LDB table, follow these steps:

   a. Select an LDB table name from the LDB tables that have the **Maintain Text** icon beside it.

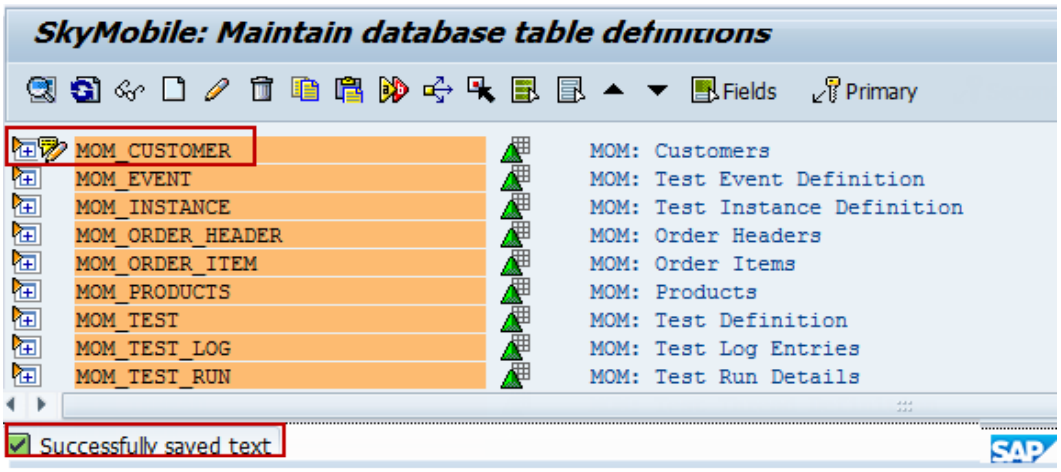
   b. Click the **Maintain text** icon.
   The **Maintain text object (VTI LDB)** dialog appears with the existing text description.



c. You may change the text description.

d. Click **Save**.

The system saves the changes. The message,"Successfully saved text", appears at the bottom of **SkyMobile: Maintain database table definitions**.

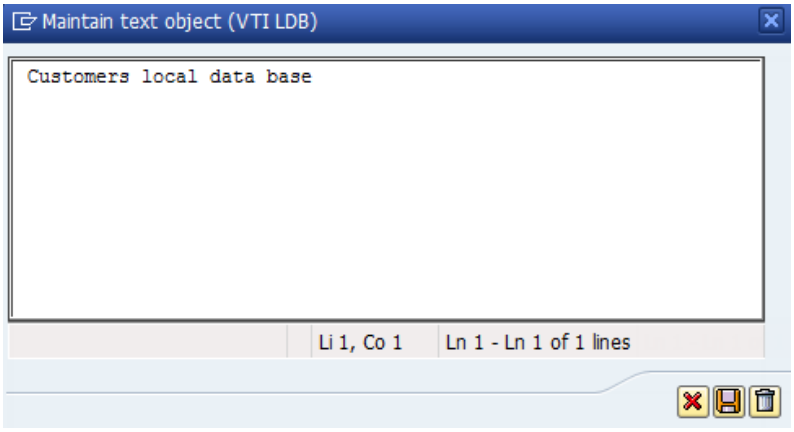


5. To delete the text description of an LDB table, follow these steps:

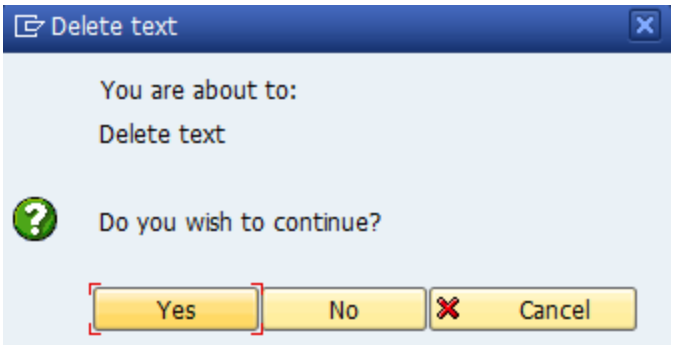
a. Select the LDB table name from the list that has the **Maintain text** icon next to it.

b. Click the **Maintain Text** icon.

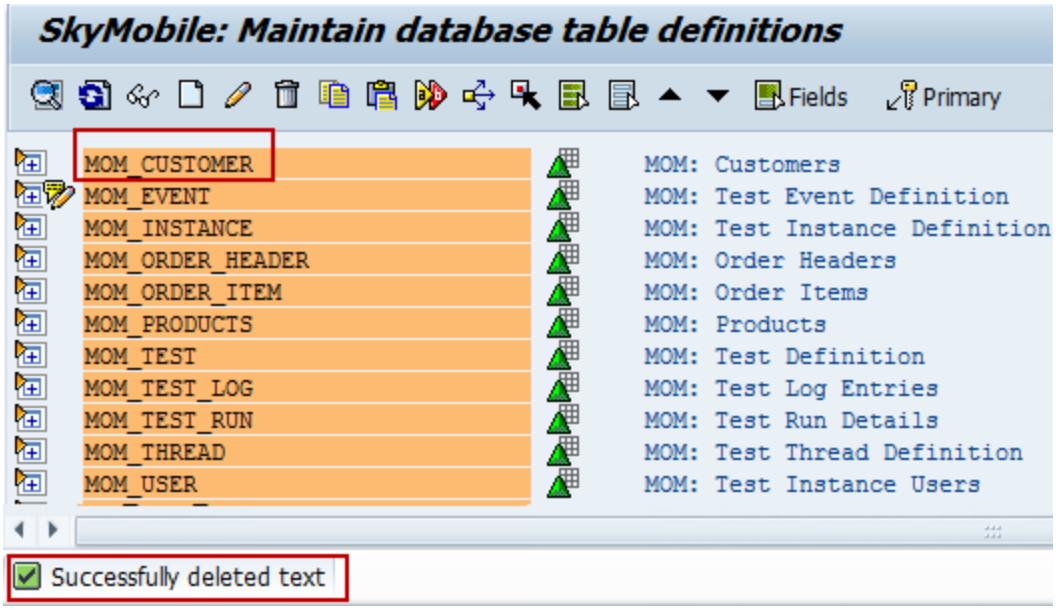
The **Maintain Text Object (VTI LDB)** dialog appears with the existing text description.



c. Click **Delete (Shift+ F1)** on the **Maintain Text Object** dialog to delete the text.
The **Delete text** dialog appears.



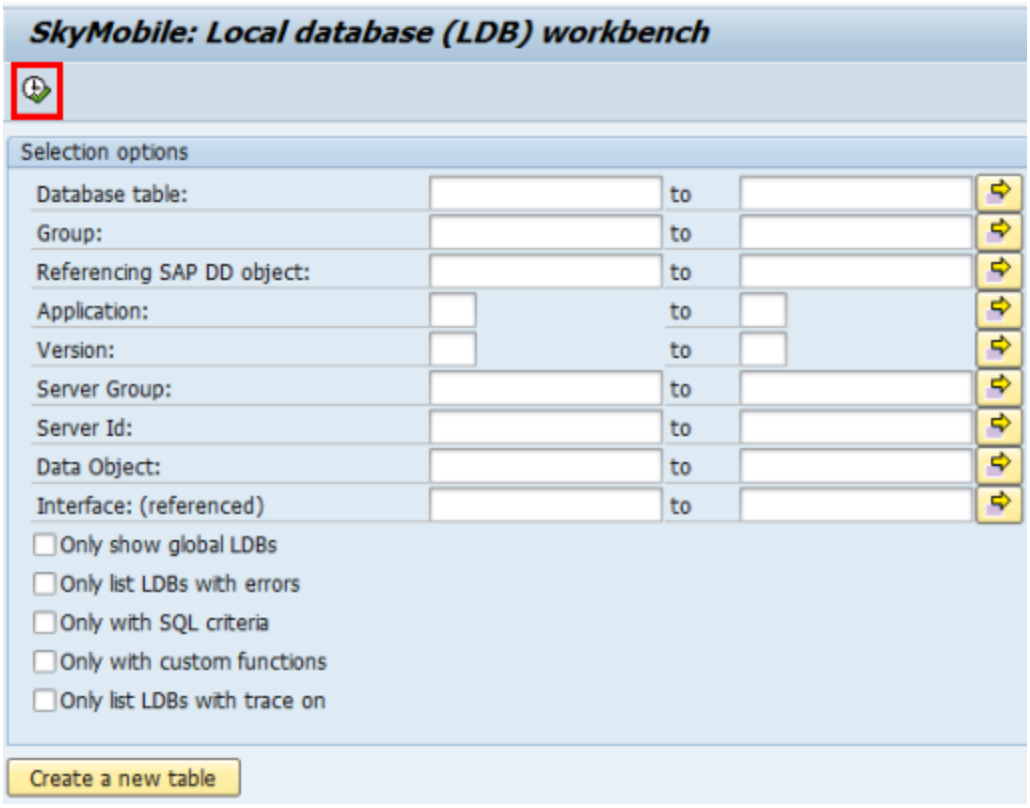
d. Click **Yes** to continue.
The **Maintain text** icon no longer appears next to the LDB name. The message,"Successfully deleted text" appears at the bottom of **SkyMobile: Maintain database table definitions**.



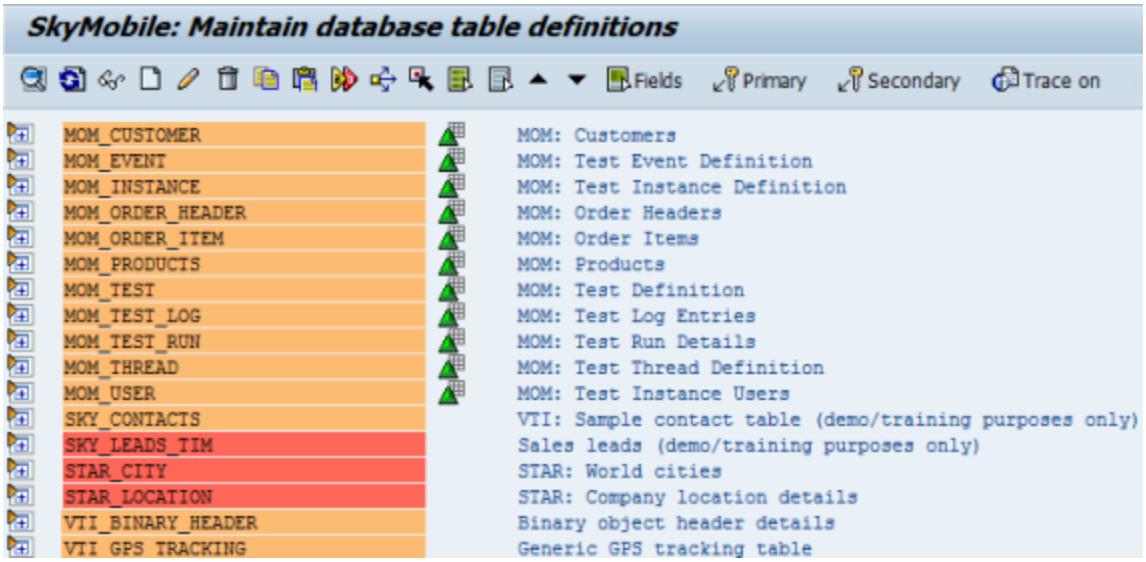
Generating a Gateway Service

To generate a gateway service for an LDB table, follow these steps:

1. Execute the transaction, **/SKY/YVTL (YVTL)** in SAP system.
   The **SkyMobile: Local database (LDB) Workbench** window appears.

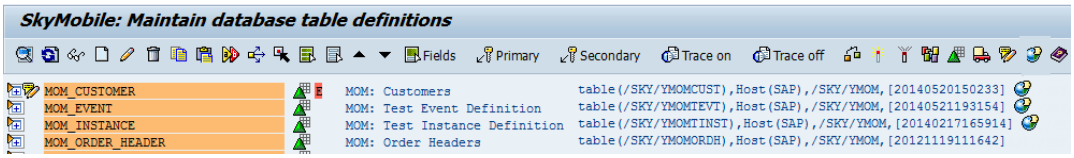


2. Click **Execute**.
   The **SkyMobile: Maintain database table definitions** window appears with the list of local database tables (LDBs).

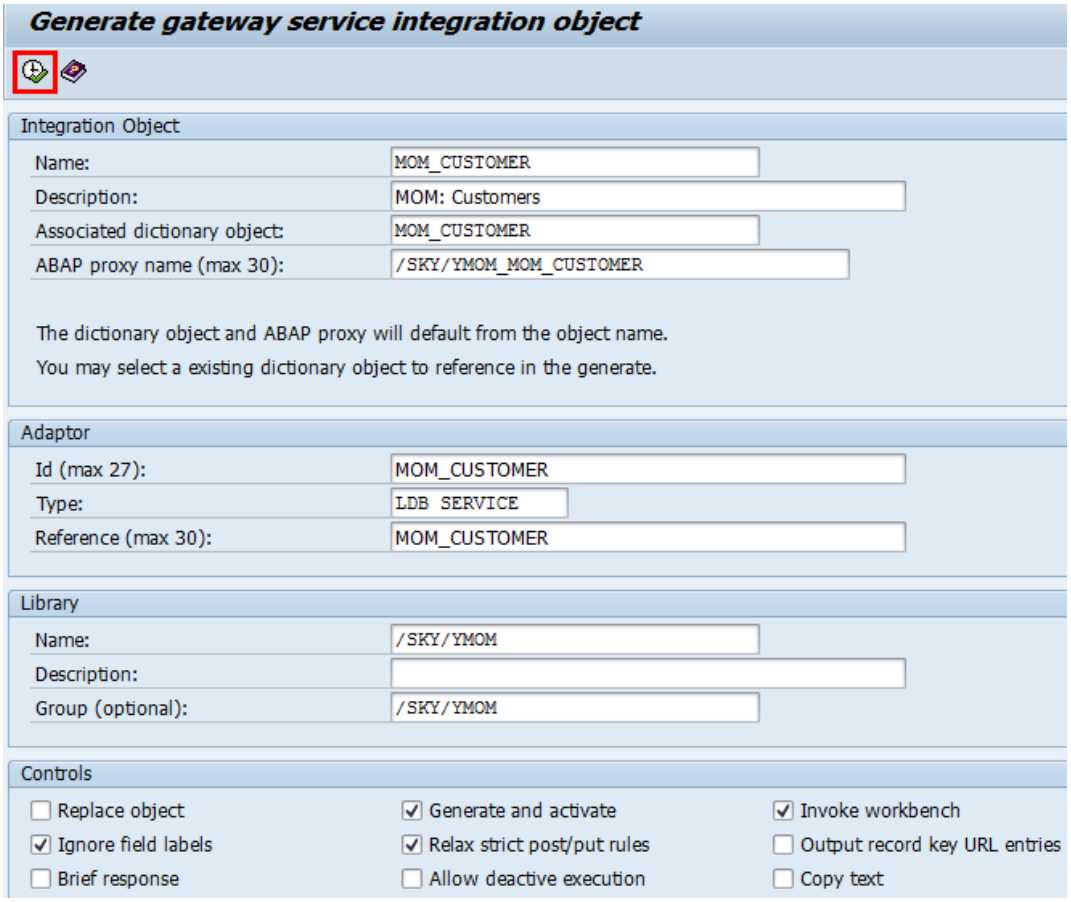


3. To generate a gateway service for an LDB table, follow these steps:

    a. Select an LDB table from the list and click the **Generate gateway Service (Ctrl + Shift + F5)** icon on the application toolbar.



    The Integration Object Workbench is called and the **Generate gateway service integration object** window appears:



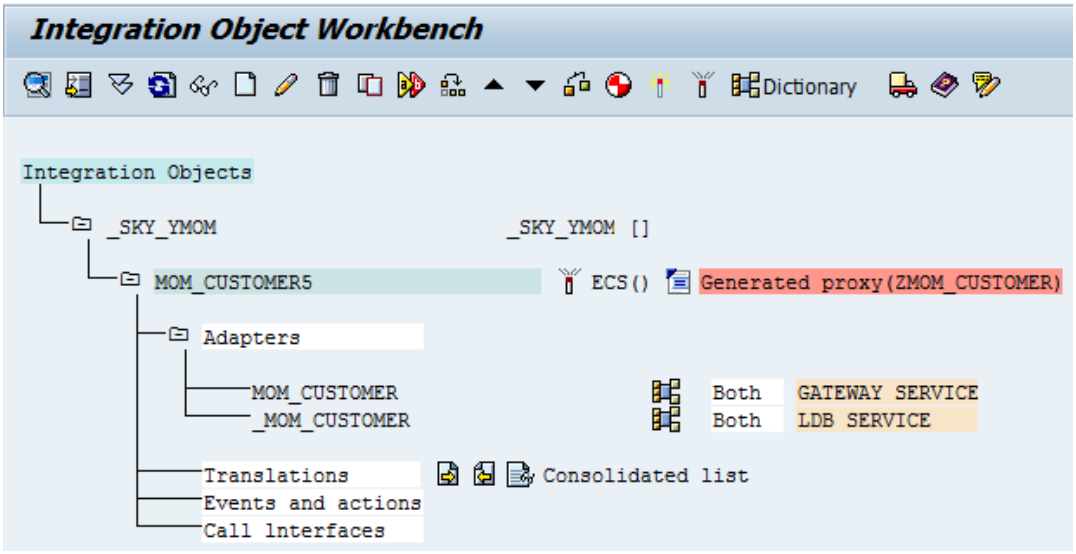
Field	Mandatory or Optional	Description
Name	Mandatory	This is a unique name of an integration object definition. It is defaulted from either the DOB or an LDB name.
Description	Optional	The text description of the IOB definition. This is defaulted from the DOB or an LDB definition.

Field	Mandatory or Optional	Description
Associated dictionary object	Optional	Each integration object has one or more associated dictionary objects. The IOB DD object defines the structures and their relationships that are to be used to map data. DD object may be shared across integration objects and are automatically generated by this utility.
ABAP Proxy	Optional	This is the name of the ABAP proxy program that is generated from the integration object definition. It should follow the usual ABAP naming convention prefixed with Z, Y or /namespace/. The name is defaulted from the object name specified above.
Adaptor		Description
ID	Optional	The name of the service adaptor, this is defaulted from the LDB name for which you want to generate the service from.
Type	Mandatory	This is the type of service adaptor used to extract from, or post transactions to, SAP, for example, LDB SERVICE, LDB REFRESH, and so on.
Reference	Mandatory	This refers to the name of the LDB that you want to generate the service from.
Library		Description

Field	Mandatory or Optional	Description
Name	Mandatory	The name of the library definition that the object belongs to. This is defaulted from the SkyMobile group.
Description	Optional	The library description. This is defaulted from the SkyMobile group.
Group	Optional	This is defaulted from the SkyMobile group. This is an optional field.
Controls		Description
Replace object	Optional	As the title implies, existing definitions are not replaced unless this option is specified.
Generate and activate	Optional	Generate the ABAP Proxy program and activate the object. This effectively enables the service to be immediately available, exposed to external clients.
Invoke workbench	Optional	After the object is generated, launch the integration object workbench.
Ignore field labels	Optional	Ignores field labels for reserved strings
Relax strict post/put rules	Optional	Ignores restrictions about Unique Records and Duplicate Records during PUT and POST operations

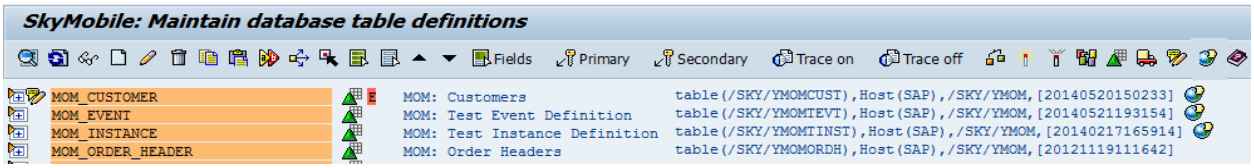
Field	Mandatory or Optional	Description
Output record key URL entries	Optional	For every LDB record in the response body, a <recordkey> element is populated that holds the full URL of the particular record.
Brief response	Optional	Ensures that no response body is returned for PUT, POST and PATCH calls
Allow deactive execution	Optional	Allow execution of service call against the DOB or LDB even if the corresponding integration object is in deactivated state
Copy text	Optional	Copies text descriptions from the DOBs, LDBs and LDB fields

b. Enter the name of the integration object in **Name** and click **Execute**.
The service is created and appears as follows:

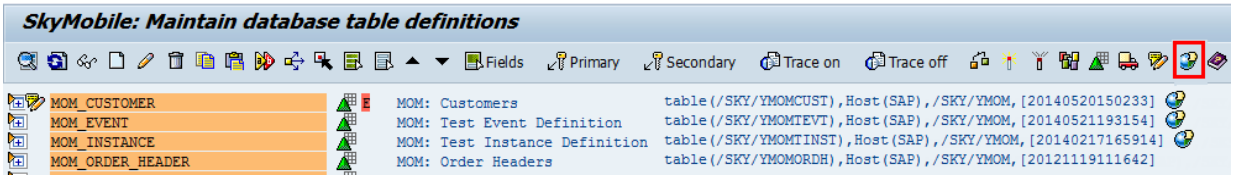


4.8.2.7 Navigating to Integration Object Workbench

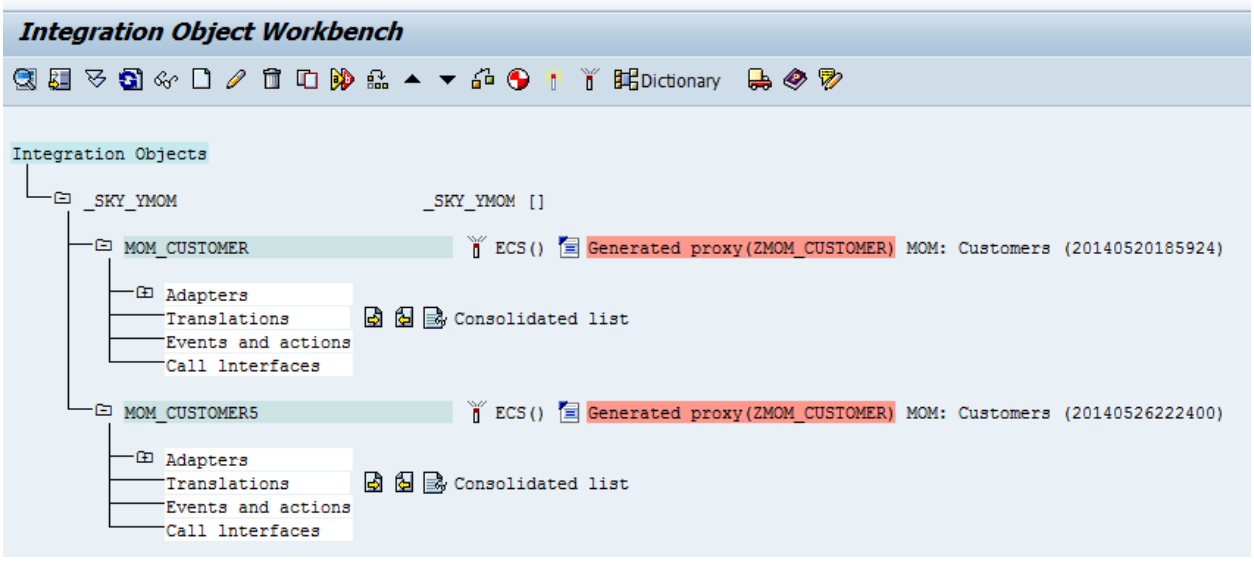
The **Generate gateway Service (Ctrl + Shift + F5)** icon appears at the LDB level when the LDB is used atleast in one integration object.



In order to see the list of gateway services of a specific LDB, click the **Generate gateway Service** icon in-line with the LDB table name.



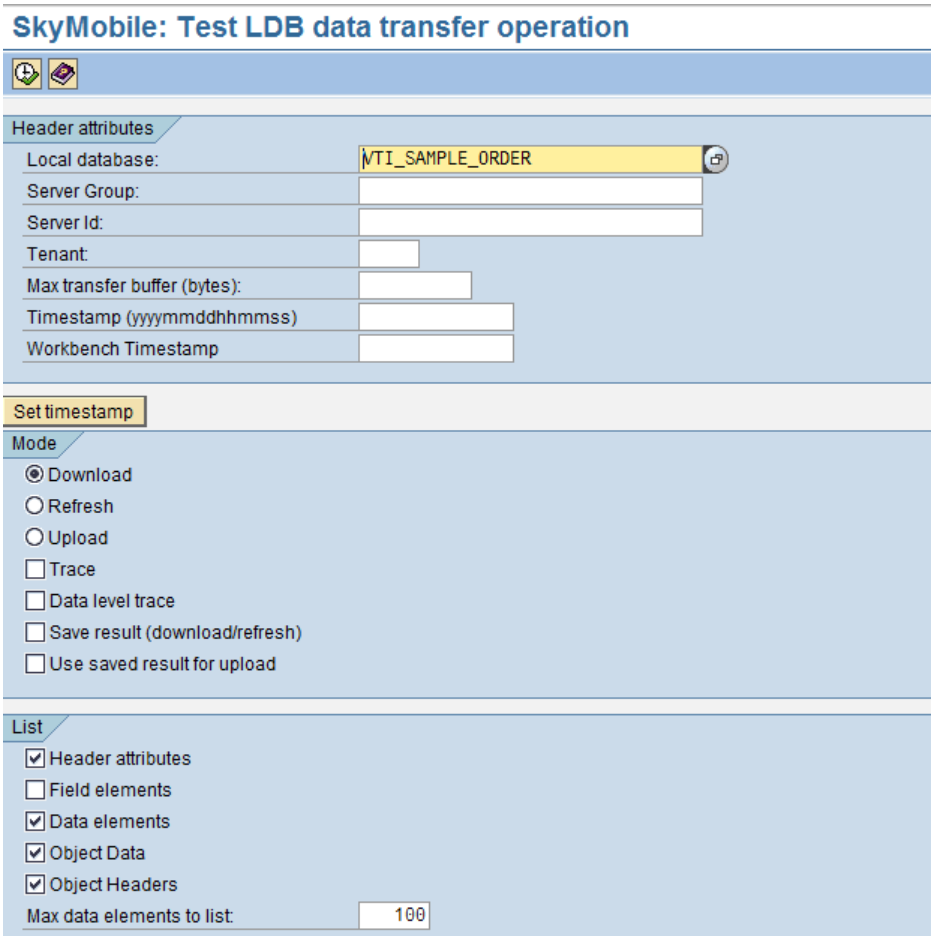
The integration object workbench is triggered and the list of integration objects where the LDB is used appears.



Testing LDB Requests

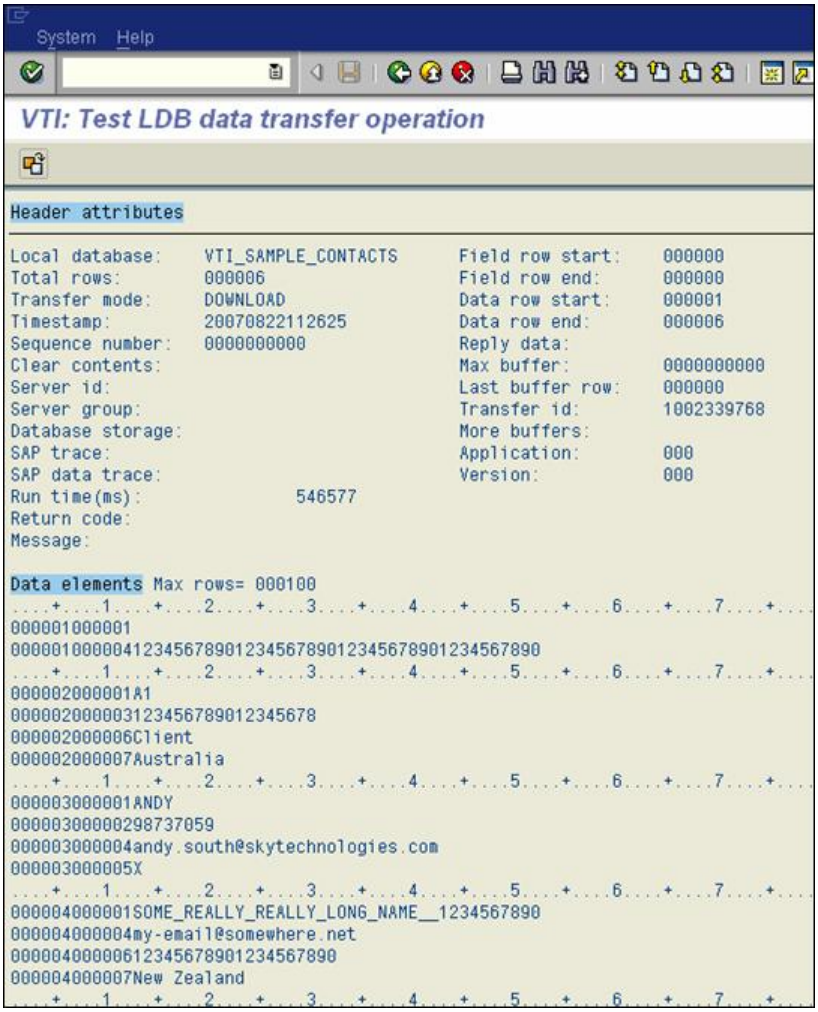
A useful facility is the ability to test download, refresh and upload transfer requests; simulate what data is processed. You may even extract data using a refresh and then import it again through an upload. This test facility is available from the LDB definition popup through the Test icon. This invokes the test selection screen where you specify call attributes, transfer mode, and list format. Remember to specify server and timestamp requirements carefully. To test an upload, you first need to perform a download or refresh and specify the "save result". Return to this screen from the list, change the mode to upload and specify "use saved result".

> ***Note:*** You may also invoke the test facility using a LDB data trace file; replay it. This is useful to debug custom functions, analyse processing, and diagnose performance. Refer to Tracing LDB requests for more information.



> ***Note:*** If your LDB relies on container connection attributes, you can simulate these by filling out the relevant details in the "Simulate heartbeat attributes" section.

Execute the test selection screen and the following list is displayed.



This is effectively a character string dump of the LDB transfer header, field list, data element list and object data lists. These are based on the following structures:

- /SKY/YVTI_TSFHD Header
- /SKY/YVTI_TSFFL Field list
- /SKY/YVTI_TSFDT Data element list

- /SKY/YVTI_OBJH Object header entries
- /SKY/YVTI_OBJD Object data entries.

> ***Note:*** You may double-click any one of the lines to view the entire structure

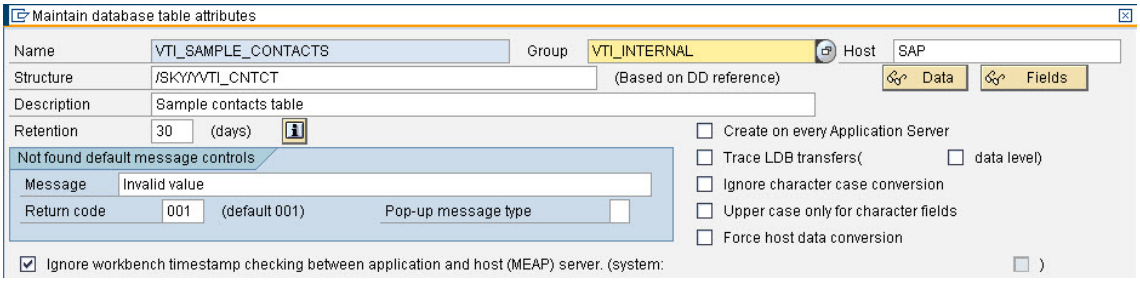
Tracing LDB Requests

The LDB manager has a facility to trace the download, refresh and upload requests and event the data being transferred. This is useful when you are either developing or supporting an application. You may set the trace facility in either the LDB definition (header) pop-up or through the trace on/off icons in the application toolbar.

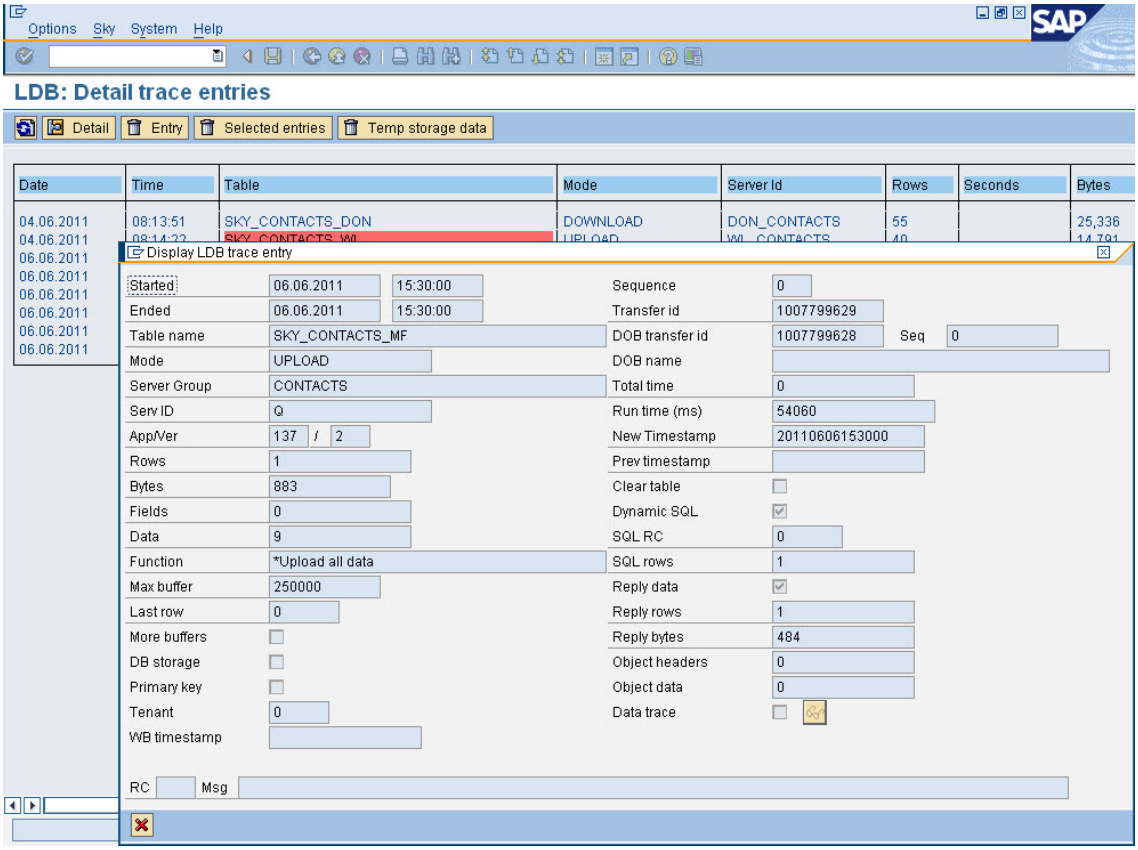
> ***Note:*** The data level trace is only available in the LDB header option. If the trace is on, a highlighted "green T" appears on the left hand side in the LDB list.

> ***Note:*** You may also automatically set the LDB trace for a specific server using the Server trace facility.

If you select a 'data level' trace, all the call input and output parameters are written to a file in the default SkyMobile working directory (configured in the default system settings using the YVTI systems management console). If a data trace is active, a green "D" appears and every call is dumped to file. You may view the data trace using the glasses button in the trace detail popup.



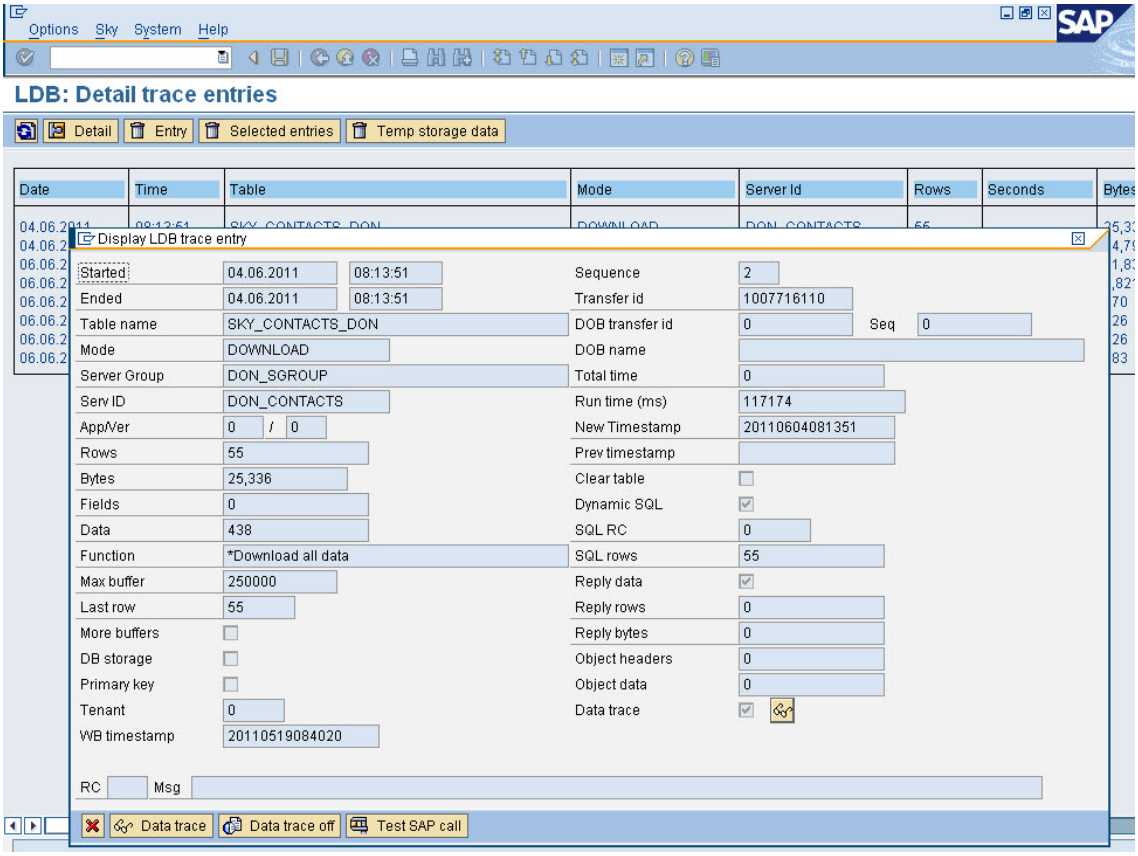
The trace list and control options are accessed through the **Utilities** >**Local Database Trace** menu option in the LDB workbench or through the **Utilities** >**SAP Host** >**LDB trace** option from the application workbench. From here, you can activate the trace for all LDBs, deactivate for all, selectively clear trace entries and list trace entries. The list has a selection screen used to filter entries, for example, server id, and date and select various different views, for example, detail, and summaries. For example, the following is an example of the detail list. You may double-click a line to view more detail and display any associated data trace (if specified).



> ***Note:*** If a data trace file is available, an option appears to invoke the SAP LDB test facility that enables you to effectively replay the request. This is useful to analyze/diagnose LDB requests and gives you the capability to debug custom code.

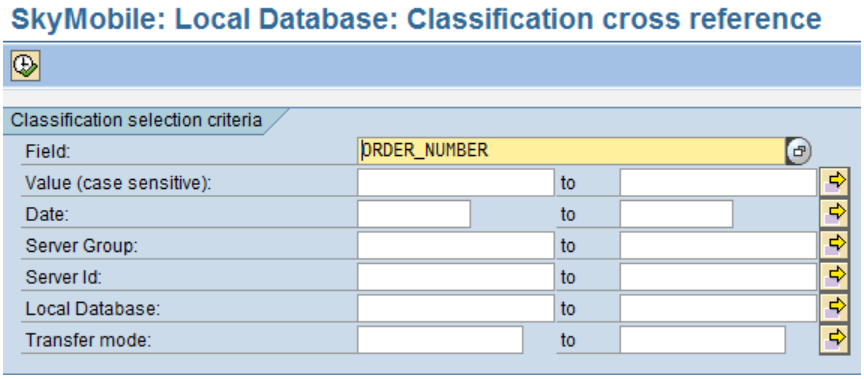
Data trace:

The data trace dumps details of all the input parameters, logic flow and output parameters to a file. This provides a significant low level of detail to analyze exactly how the request is processed. You can also use the data trace as input to the SAP to effectively replay the request. The standard data trace list allows you to expand sections and double-click structures to see the data attributes. Within a request, each level has a header explaining what it is, for example, input, flow trace, and output. Kony may request this level to solve performance problems, but it also an extremely useful tool to diagnose what LDB synchronisation requests actually do.

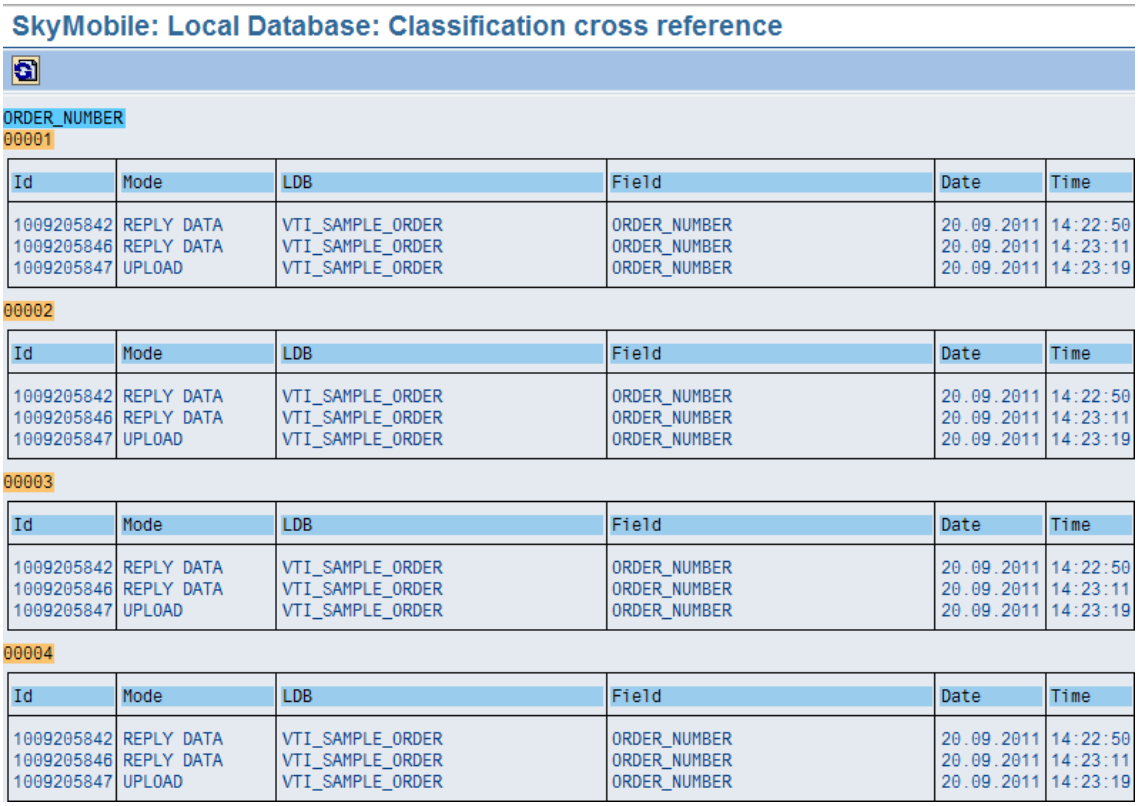


LDB Data Classification

You can nominate local database fields to be classified whenever data synchronisation occurs; track whenever a field value is transferred and to which Secure Container. The classification reference is configured by maintaining the LDB field attributes and a utility is provided to list and track the classification entries, for example, when was order 1000876 downloaded to a device.
Once you have configured the classification data against the LDB fields, you may list the entries through the classification data option on the utilities menu in the LDB workbench. A selection screen is provided to help filter the data for a list of entries.



A list of entries appears grouped by value and all the LDB operations performed against it.



SkyMobile: Local Database: Classification cross reference

ORDER_NUMBER
00001

Id	Mode	LDB	Field	Date	Time
1009205842	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:22:50
1009205846	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:11
1009205847	UPLOAD	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:19

00002

Id	Mode	LDB	Field	Date	Time
1009205842	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:22:50
1009205846	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:11
1009205847	UPLOAD	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:19

00003

Id	Mode	LDB	Field	Date	Time
1009205842	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:22:50
1009205846	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:11
1009205847	UPLOAD	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:19

00004

Id	Mode	LDB	Field	Date	Time
1009205842	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:22:50
1009205846	REPLY DATA	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:11
1009205847	UPLOAD	VTI_SAMPLE_ORDER	ORDER_NUMBER	20.09.2011	14:23:19

If you double-click a line, then any associated trace data for the unique transfer id is listed.

Data Synchronization

The configuration for downloading, refreshing and uploading local database data between SAP and the Application Server is maintained in the MEAP IDE in the LDB definition (header). Once configured, the LDB definition is downloaded to the Secure Container. You may determine this through its association with an application, its configuration to be downloaded to all servers or through an entry in the Secure Container local database cross reference. Once a local database table is created, it is initially populated using the download configuration; once off load. Subsequently, you may maintain the data automatically using periodic or manual refresh operations. Both download and refresh operations are 'pull' functions from the Container to the MEAP server. Automatic or manually invoked upload operations are 'pushed' from the Secure Container to the MEAP server. You may schedule the server refresh and upload operations to perform on an interval (every x seconds) or at a specific time of day. When a LDB definition changes in the MEAP IDE, and any associated application version timestamp is modified. The next time, the Secure Container application cache manager checks its time stamps with the MEAP, it recognizes that a change took place and issues a request to download the application and/or LDB definition. The changes are then implemented automatically. If the application version is flagged for manual synchronization, you must manually schedule the request to download using the MEAP IDE workbench (discussed later in this section). Whilst local database level synchronisation is fully supported, Kony highly recommends that you use Data Objects wherever possible for better transaction level integrity and performance.

> ***Important:*** If the LDB is part of any data object that the application uses, its synchronisation scheduling is nullified and is superseded by that of the data object; you cannot have both.

Time Stamping

Every download, refresh and upload operation is time stamped as to the last time it was successfully performed by the Secure Container. SAP set this time stamp and is simply stored by the Secure Container against the LDB definition and returned on the next call to SAP. This is so a central system time is used. The custom functions may set the timestamp and use to process changes that occurred from that point. The timestamp is automatically set for download and upload operations, irrespective of any custom function. As well as internal timestamps, you can include special timestamp fields in the LDB definition to control the automatic replication of only rows that changed.

Logical Deletion of Rows

A common problem with distributed applications is how to handle the deletion of data. You cannot physically delete the data until it is successfully replicated to all the applications in the field. The best approach is to flag the row as logically deleted and then physically delete it, once it is deleted from all the remote application databases. To do this automatically, a logical deletion flag (character 1) is added to the LDB definition and then this field is configured in the LDB download, refresh and upload operations.

The Request Structure

Each data transfer request to SAP has three components; the control header (/sky/yvti_tsfhd), the field definitions (/sky/yvti_tsffd) and the data (/sky/yvti_tsfdt, /sky/yvti_objh, /sky/yvti_objd). The header is a parameter structure, whilst the field and data definitions are internal tables. SkyMobile treats each data component like a spreadsheet cell; row, column and data coordinates, linked to a field definition in the table. This allows for a flexible mechanism to just reference and transfer specific fields and not the entire data structure. The control header contains the name of the LDB, the mode (download, refresh or upload), the Application Server id that issued the request, and the date time the last successful operation was performed. The ABAP local database programming guide has more detail on all the attributes and how they are used. A full ABAP programming API is supplied, that abstracts ABAP programmers from having to know the inner workings.

Handling Large Volumes of Data

For large volumes of data, you can configure SkyMobile to 'buffer' the transfer of data; store the extracted data set in SAP and then perform multiple transfer requests until all data is transferred. The Application Server TransferBufferSize parameter determines the transfer buffer size in its configuration. This mechanism is essential for devices with a limited memory capacity, since the Application Server must first receive the transferred data in memory in-order to uncompress and process it. It is also recommended to use the primary key traversal mechanism for high volume download and refresh operations in conjunction with a transfer buffer.

Download Operations

The downloading of data is intended as a 'one off' operation to initially load the newly created LDB table with data. As such, the the Application Server initiates the download request when it creates a new table, or if it is configured to invoke a download at every start-up or if it is manually invoked from the Application Server web status page.

Refresh Operations

As apposed to the initial download, the data refresh operation is designed to perform incremental updates on a periodic basis to help keep the local database synchronized with SAP. You may schedule Refresh operations to 'poll' SAP for changes. Every time a refresh is initiated, the previous successful time stamp is passed in the header. This is so that the data selection processes can determine the last time it ran and select data accordingly; what changed since that point in time.

Upload Operations

A data upload from the Secure Container to the MEAP may be either; automatically scheduled, invoked from a screen function or manually invoked from the Container web status page. The objective of the upload is to transfer maintenance requests from the Container to the MMEAP, for example, upload a sales order for posting. You may install a custom function to perform this task (usually to kick off an ECS Process to perform the update). In addition, you may return back a response request to the Secure Container. This 'reply data' may issue the same requests as a refresh operation, for example, may update the status of the order to 'processed'.

Tuning and Tracing Requests

You should optimze the local database operations to transfer as little data as possible. In addition, you should implement a SQL database to handle large volumes of data or a large number of related tables. You should only use the default in-built database file system for low volume, relatively low frequency tables. SkyMobile provides a MEAP LDB trace facility to monitor data transfer requests as they occur. This trace is described in detail in the system management section. The trace is useful to view how much data is transferred and how often. This helps to identify problems and potential bottlenecks.

LDB Design Considerations

Whilst application requirements differ, there are some fundamental design principles for local databases and custom functions. Designing the LDB involves pre-planning of what data is required, how often it is changed and how you identify and transfer the changes. The last point in particular, constitutes the majority of effort and the LDB functionality has some in-built functions and techniques that can assist this process. Kony recommends that you use Data Objects wherever possible for synchronisation and better performance.

Using Interim SAP Tables

It is often useful to store SAP data in interim custom tables, instead of relying on the original SAP tables themselves. This has many advantages; only specify the fields that are required, consolidate data from multiple sources, specify custom fields and implements a buffer between the mobile system and standard SAP tables.

Use a Common Function Module

Instead of creating a function module for each local database download, refresh and upload, consider coding a single function module that segregates processing based on the mode and local database; case statements on TRNSFR_MOD and LDB_NAME that are passed in the request header. In this way, all the logic is together and more easily maintained.

> ***Note:*** You may not need to code your own function module at all by configuring LDB SQL conditions. These also support nested LDB sub-queries; you can pass the result of a LDB call into the selection criteria of another. Consider these options first before resorting to custom code.

Using a Timestamps and Logical Deletion Indicators

As discussed earlier in the section, you should incorporate timestamp and logical deletion indicator fields into every LDB definition. These fields are then incorporated into the LDB replication rules and screen processing to provide effective automatic replication controls.

Using a Status Code

For more complex update operations, a status code may be necessary to identify the state of a data item, for example, created, transferred, posted, archived. This is common for header type records such as an order header. The status code technique usually involves more complex programming to select and update the value. A sub-status may also be necessary to identify pending, in-progress, complete or error states.

Handling Inter-Table Dependencies

> ***Note:*** Sky recommends that you define a data object (DOB) with all the LDB inter-table relationships to properly handle this situation. See the main section on the Data Objects Workbench for details.

In cases where a transaction spans multiple tables (for example, order header, order items), two separate refresh operations occur; one for the header and one for the items. This is not a problem if the same timestamp or sequence number technique is used for both, but does present a problem if the selection of the items is dependent on selecting the header. In this case, you may use a true/false indicator on the header to show whether the corresponding items are downloaded yet. The item refresh then selects all headers that do not have this indicator set, and subsequently sets the indicator once they are downloaded. This technique is only really applicable for a single Application Server.

Making the Key Unique for each Application Server

SkyMobile supports a number range facility that is useful when creating new unique keys, for example, order number. This number range may not necessarily be unique across multiple Application Servers. In order to make the key unique, you should also use the Application Server id to uniquely qualify the key; server-id, order number. In this way, locally created transactions are unique across all devices. You may also implement the automatic central numbering using the LDB field attribute automatic number range controls.

SAP Change Document sub-System

Most SAP master data tables have change documents. This SAP facility provides a standard mechanism whereby any change to the may be automatically intercepted and processes. The change document sub-system may implement and call the custom functions. These custom functions may update interim tables that are used in LDB refresh operations. Use of this facility is fully documented in

the SAP help.

Restricting or Partitioning Data

It is often important to segregate data depending on the server requesting it. Application Servers are identified by a unique 'server id' that is passed to SAP on every download, refresh and upload request. You should use this should as the primary mechanism to identify what data you should process; the server id is used as a key to read configuration on plant, and sales org. This becomes particularly important for mobile PDA applications, where individual data profiling is important; specific customers, products, and pricing.

Push Versus Pull Data Transfers

By default, LDB data is transferred using a 'pull' technique; the Application Server polls the SAP host at a specified time or interval. The problem with this approach is that many requests are unnecessary and if frequent across a large number of devices, can cause performance problems. Instead, you can implement a 'push' mechanism where SAP can inform the Application Server when updates are available for it to download. 'Push' requests are implemented using the heartbeat command queue and are covered in more detail in the System Management and ABAP programming guides.

Automatic Data Retention

You can fully automate the housekeeping of data by using the retention feature on the LDB header. This requires that you configure a timestamp field and optionally a delete indicator. The Kony for SAP housekeeping routine then physically deletes any LDB data where the timestamp expired after the number of configured days. If you specify a logical deletion indicator, only those rows where the indicator is set is deleted. The number of days is a business decision, but should also allow enough time for all the Application Server synchronisations to occur.

4.8.3 Data Objects (DOB)

A data object (DOB) enables you to link one or more local database (LDB) definitions together. There are two primary advantages:

- Ability to process multiple LDB synchronization requests in a single request
- Process complete transactions; all inter-related data together
- Enables the use of the Data Profile facility to automatically control the distribution of data to remote Application Servers.
- Incorporate Binary objects (BOB) into a data object definition, using classification.

So in essence, data objects help improve:

- performance by reducing the number of synchronisation events
- data integrity by synchronizing complete transactions (not table by table)
- development and support by eliminating complexity in both data synchronization and profiling

A DOB definition is made up of a header, table relationships and Binary Object relationships. All of these are managed using The data object workbench. Because data objects are not related directly to applications, you must specify complimentary references to link the DOB with applications. Be aware that DOBs are closely linked to the underlying LDB definitions; if you change a LDB definition using the workbench; the change is cascaded to the dependent DOB definitions. There are many configuration options available to help control and manage data object definitions. For details on some of these, click here.

There are basically three modes of data object synchronization:

- Download that is designed to initially populate data on the device
- Refresh, applies delta data object changes to the device
- Upload, transmits data object change from the device to the central MEAP Server

You configure the data object synchronization modes in the header definition in the workbench. Kony recommends implementing a "push" model; data object synchronizations are initiated from the host instead of each device polling the host for changes at regular intervals; event driven.

> ***Important:*** An important design consideration is that the 'root' table' in a data object, one with no parent, is the transaction boundary; all children and their children for each root parent row are retrieved irrespective of the transfer buffer size. Thus an important aspect of designing a data object definition is to understand the transaction boundaries and the expected volumes of child 'data' relationships.

4.8.3.1 DOB Hints and Tips

A data object consists of one or more LDB (local database definitions) and allows you to configure inter-dependencies (key relationships) between them. There are many benefits of using data objects; primarily you may group together related data and process as a single transaction, for example, order header, and order lines, you may also group together many table synchronization requests into a single request, for example, refresh all my master data in one call rather than for every table. A important thing to note, is that if you define a table into a data object that has periodic scheduling, for example perform every five minutes the data object scheduling takes precedence and the table level scheduling is ignored. A way to consider data object processing, is that they effectively "front end" normal LDB table processing.

You define data objects using the DOB workbench in the MEAP Server. You typically define timestamp and logical delete indicators in each of the LDB tables to enable the automatic detection of changes and housekeeping of data. It is recommended that you "push" change requests out to the device; event driven from the host rather than use a polling mechanism (interval) and use the data profiling functionality to control what data the user or server receives and can process.

Binary Objects

As well as dependent LDB database table data, you may include binary objects (files) as part of the data object. This is possible by using the binary objects classification data in the selection criteria.

There are two methods:

1. using the run time value of a table field
2. a configured classification data value (wild cards supported).

You should take care of data volumes that affect the performance of the system; watch the size and volumes of the binary object processing.

> *Important:* In DOB processing, binary objects are always transferred in their entirety. This can cause device failures due to insufficient memory if extremely large binary objects are processed. The "process large binaries asynchronously" setting causes binary objects that are larger than the DOB transfer buffer to be downloaded asynchronously; in background. Because of this, the data may not be available immediately.

"Root" Table Transaction Boundaries

The "root" table (table with no parent) defines the transaction boundary; all children rows are selected for each parent row irrespective of any max transfer buffer setting. This is a important consideration in design since it is a common error to drive the DOB selection from master data instead of the transactional data, for example, order type -> order retrieves all orders and their dependency data for each type selected. If special filtering is required, then it is recommended that you use configurable LDB SQL conditions instead.

Parent Data Conditional Processing

Apart from the keyed relationship between a child and parent table, you may also apply a single condition based on a parent field to help limit the execution of the child level, for example, only retrieve the delivery items if the delivery header status is unpicked. You can use this to significantly improve performance when extracting data and more flexibility when using "pivot" master tables with multiple different child tables.

Create on every Secure Container

You design this setting for common object definitions that you use on every container that connects to the SAP host, irrespective of what applications are used. If configured, the object and all its LDB tables are automatically maintained and be in effect for all Secure Containers.

> *Important:* You need to configure this with caution since the DOB is implemented on every Container. It is mainly used for shared internal SkyMobile objects.

Automatic Cascade Deletion

SkyMobile analyzes the refresh and upload data for deleted items. If any are found, all sub-ordinate related table data is flagged as deleted as well and applied on the container and/or back end host. This helps automate data deletion processing; you just need to delete the appropriate level and maintain the timestamp hierarchy.

> ***Note:*** Only tables that contain the entire primary key in their parent relationship are considered for processing; unique relationship.

A delete existing data option is provided on the refresh configuration that removes all data associated with a specific data object, just prior to inserting the new object. It is recommended that you configure this if you always replace the entire object; do not perform partial updates.

Automatic Data Retention (Logical Delete)

You can specify a retention option on the DOB header that allows you to specify a retention period in days for a specific data object definition. This instructs the SkyMobile housekeeping routine to physically delete objects that are older than this value from the database that is logically deleted rows whose timestamp is expired by nnn days. This processing performs a cascade delete of ALL parent and child data that have a primary key relationship with the parent; master data relationships are ignored. This feature effectively supersedes retention controls at the LDB level.

Ignore Workbench Timestamp Checking between Container and MEAP Server

By default, a data transfer is failed if the workbench timestamp is different on the application server. This setting overrides the check; allows the data to be transferred even though there may be structural differences.

Data Synchronization (Download, Refresh, Upload)

You may specify the same basic download, refresh and upload synchronization controls as those available in LDB definitions. You may also specify overrides for specific Java servers, or groups of servers.

> ***Note:*** All LDBs referenced in data objects have their download, refresh and upload synchronization controls nullified; the data object is the master.

Download only Options:

The following are the download only options:

- perform at every start up: the download is processed every time the container starts
- clear all data: all existing data and binary objects for the data object is deleted before the the new set of data is updated
- don't wait: if this is set, the server does not wait for initial downloads to complete, but continues asynchronously; letting the application load.

Refresh Protection

This option prevents data that is just uploaded from being re-selected on the next refresh; timestamp changed. The refresh operation caches and checks the primary keys of the root tables of the last upload. This is not necessary if the application uses status codes to make sure that data is not re-selected/over written. A small management overhead is incurred.

Using Profiles (Data profiling)

If you associate a profile with a transfer operation, the system looks for any selection criteria that qualifies for the profile, server group/id and data object combination; profile instance selection criteria. In the case of download/refresh operations, the criteria is used to generate dynamic SQL where criteria to use against a specific LDB table; only select data that meets the criteria. In the case of uploads, the criteria is used to determine what other servers to generate data object refreshes for; "push notification" out to other servers to refresh their data, for example, master data change.

Automatically Triggering Interface Processing

You may configure a interface to automatically start once a transfer is successfully processed. The interface is passed the data object header, a list of LDB headers and a list of the primary key values for each table. An example ABAP program /SKY/YVTI8023 and process (VTI EXAMPLE DOBUPL) is provided to demonstrate how to configure and receive the parameters from a upload. This is a useful mechanism to automatically trigger integration processing. If you leave the user blank, it defaults to the

connected user. You may specify a SAP user id, a '&' substitution variable or '*' to automatically derive it from the profile user connection (if any). You may configure a password or may specify a '&' substitution variable or '*' to derive it from the current connection details. If you use a password, then ECS user switching is invoked; the userid and password must be valid in SAP and the initiator (Sky gateway connection) must be running as a SAP service type user.

Upload Refresh Option

If configured, this option performs a refresh after a successful upload. The refresh is executed as a separate operation and the result data is appended to the upload 'reply' data. This effectively streamlines data object synchronization initiated from the device; push up data and pull down new data in one movement. If you want to ignore including the recently uploaded data, then you should use the refresh protection option in conjunction.

Tracing Data Object Requests

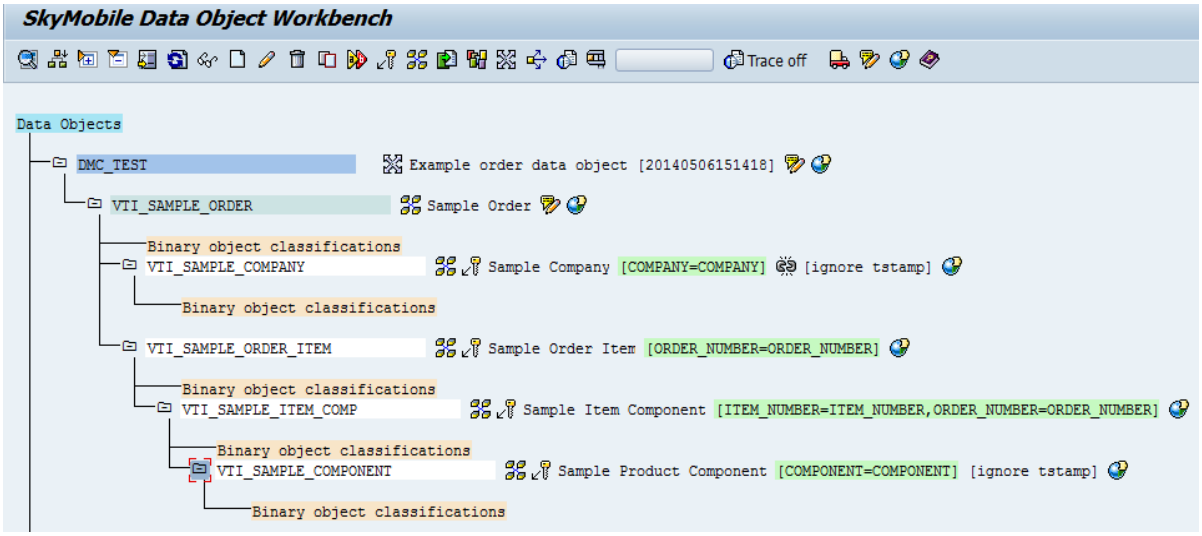
Both a summary trace and data level trace may be set on/off through the data object attribute maintenance screen. This works in the same way as the LDB trace. The data level trace dumps all the header and data areas to a file in the SkyMobile working directory (configured in systems management). All underlying LDB calls are also traced and linked to the data object trace. The trace is useful to see what data is processed and analyze the processing statistics and attributes. A trace facility is provided in the MEAP Server to list the DOB trace entries and also simulate re-processing them to diagnose problems.

Testing Data Object Requests

A test facility is provided in the MEAP Server to simulate data object requests. This is useful for development and support to see the results of a DOB transfer operation. Click the test button on the data object attribute maintenance screen. Take care when using this in a production system, as the calls are treated exactly like those from an external Secure Container.

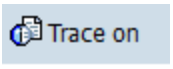
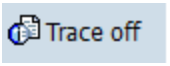
The Data Object Workbench

You invoke the Data Object Workbench in the MEAP Server using the YVTD short cut transaction code or through the  icon on the application toolbar. A selection screen first appears to allow you to filter the hierarchy list. The following hierarchy screen appears:



Field	Description
	Displays all the LDBs related to the DOB in a hierarchical format
	Expands the selected DOB hierarchy by one level
	Expands all levels of the selected DOB hierarchy
	Collapses all levels of the selected DOB hierarchy
	Positions the selected node to the top of the screen

Field	Description
	Refreshes the DOB list
	Displays a window containing the attributes of the DOB
	Displays a window to create a database table relationship
	Displays a window to maintain the database table relationship
	Deletes a DOB
	Creates a copy of selected DOB
	Renames a DOB
	Displays a pop-up list which allows to add and remove the field relationships.
	Navigates to Local Database Workbench
	Navigates to Binary Object Workbench
	Maintains complimentary data object cross reference
	Navigates to Data Profile Workbench

Field	Description
	Displays a where-used list of the selected DOB
	Lists entries of DOB transfer trace
	Tests the DOB transfer operations
	Sets the basic trace on
	Sets off the all trace options
	Exports definitions to another MEAP server or out to a file.
	Generates gateway service integration object
	Navigates to online help documentation of database object workbench

The data object definition appears as the second top level node, for example, VTI_SAMPLE_ ORDERS; the next level are the root tables (those with no parent), followed by any underlying 'child' tables and any Binary Object relationships. You maintain the definition by either double-clicking a level or positioning your cursor on the level and clicking on the application toolbar. Various hot links and short cut icons also appear that enable you to quickly branch to relevant areas for example, LDB workbench.

Typically, the data object is associated with a data profile that is used to control the data that users receive and can process. You enter the profiles to use in the DOB definition header and the association appears by the icon in the hierarchy.

The main components of a data object are as follows:

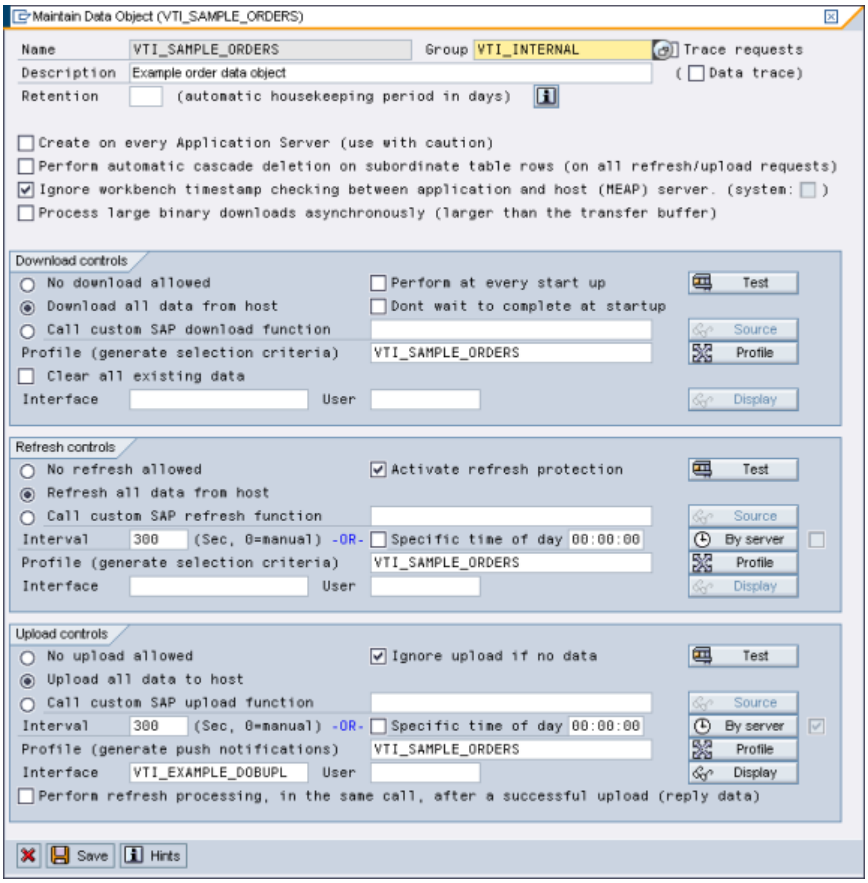
- The definition header
- The parent / child tables
- Table key relationships
- Binary object relationships

DOB Definition (header)

The DOB definition pop-up contains all the attributes that control the behavior of the data object.

> ***Important:*** To understand some of the concepts and processing options that are available for data objects, click here.

Click the create icon on the application toolbar, or position the cursor on an existing name and click the change icon. You may copy or rename the existing definitions. The following pop-up appears:



Field	Description
Retention	The housekeeping automatically deletes entire Data Objects that are flagged as logically deleted and whose timestamps are expired by the specified number of days. This option effectively automates the housekeeping of data; eliminates having to write code.
Create on every Application Server	The DOB definition is to be automatically downloaded to all Application Servers. ***Note:*** Only specify this if the definition is truly generic.
Automatic cascade deletion	This option ensures that all subordinate child table data and Binary Objects are flagged as logically deleted if a refresh or upload detects that the 'root' parent row is deleted. This prevents having to write your own code to do the same thing; you just need to flag the top level parent as deleted and Kony does the rest. Refer to Cascading DOB updates for more details.
Trace transfer requests (Data level)	Activate or deactivate the SAP DOB trace facility. The data trace option creates a trace file for every transfer request containing the internal header, field and data definitions. ***Note:*** All underlying LDB requests are traced as well; the LDB trace facility is triggered at the same level.
Ignore Workbench timestamp checking	By default, the DOB LDB definitions must match between the SAP host and the Application Server. If you find them to be inconsistent, then an error is raised and the transfer is ignored. This option suppresses the timestamp check; the application server converts the data as required. You can configure this setting system wide using the SkyMobile System Manager.

Field	Description
Process large binaries asynchronously	By default, any associated individual binary objects are transferred in their entirety irrespective of the transfer buffer size. If this option is set, any binary object that exceeds the transfer buffer is downloaded asynchronously honouring the transfer buffer. This option is required if very large binary objects are selected that have the capacity to exceed the available memory of the device and/or SkyMobile access gateway. It does mean that the binary object may not be immediately available with the data object data.
Download controls	Downloading of data is designed to initially populate new table definitions with data from SAP. You may either download all data automatically from the related LDB definitions or call a custom function to retrieve the data (see ABAP programming guide). The download may be configured to occur every time the Application Server is started by setting the "Perform at every start up" indicator and you can prevent the application start from waiting for the download to complete, using the "Don't wait to complete at start up option". If you wish to delete all existing data for the data object in the database prior to the download, specify the "Clear all existing data" option. Profile: A data profile definition may also be associated with the download operation to automatically select specific data based on rules. Refer to Data profiling for more information. Trigger Interface Use this option to automatically start a ECS interface after data has been selected by a download operation. The ECS program is passed all the primary key values of the data set that was maintained. See the ABAP programming guide for more details. This functionality is useful to perform subsequent processing in SAP after data has been selected by a mobile application e.g. change the status of an order. You may specify the SAP user id to use for the start, specify '*' to derive the user from the SkyMobile logged on user, or leave blank to use the current RFC SAP logon e.g. CPIC_VTI.

Field	Description
Refresh controls	In contrast to the one off download concept, refresh operations are intended for incremental update i.e. periodically check the SAP system for changes. You may either refresh all data automatically from the related LDB's or call a custom function to retrieve the data (see ABAP programming guide). The refresh may be scheduled to occur every xxx seconds or at a specific time of day. If no time or interval is specified (0), the refresh may be invoked manually from the Application Server Web status page. Activate refresh protection: A common problem with servers uploading updates to SAP is that the same data may be selected the next time they issue a refresh. Refresh protection keeps track of any data that is uploaded in-between refresh requests. If the same data is selected by a subsequent refresh, then it is ignored. Clear existing selected objects: This setting causes the Secure Container to delete all existing data associated with a specific data object just prior to the new definition getting inserted. Kony recommends using this option is you are totally replacing data objects and are not performing partial updates. Profile: A data profile definition may also be associated with the refresh operation to automatically select specific data based on rules. Refer to Data profiling for more information. Trigger Interface Use this option to automatically start a ECS interface after data has been selected by a refresh operation. This is identical in functionality to the same option explained in 'Download controls' above.

Field	Description
Upload controls	Upload operations are intended for incremental update i.e. periodically send changes up to SAP (push). You may either upload all data automatically into the related SAP table/view or call a custom function to update the data (see ABAP programming guide). The upload operation may be scheduled to occur every xxx seconds or at a specific time of day. If no time or interval is specified (0), the upload may be invoked manually from the Application Server Web status page. An upload can also return data in the same way as a refresh operation to create update or delete data. If ignore upload if no data has been set, the Application Server will ignore calling SAP if there is no data selected to upload. Profile: A data profile definition may also be associated with the upload operation to automatically "push" out update notifications to related remote Application Servers based on rules. Refer to Data profiling for more information. Trigger Interface: Use this option to automatically start a ECS interface after a successful upload operation. The ECS program is passed all the primary key values of the data set that was maintained. See the ABAP programming guide for more details. Trigger a refresh: You can configure a upload operation to automatically trigger refresh processing (as above) once the upload processing has successfully completed in SAP. This is useful to perform both synchronisations in one operation and can help avoid the issue of conflicting updates from both the server and the SAP host. Note that the impact "Ignore upload if no data" option should be carefully considered when using this option. It is also important to note that no transfer buffer setting is used for the refresh i.e. all the refresh data will be returned and care should be taken to avoid large volumes.
	The test facility is useful to simulate a download, refresh and/or upload operation and review the resulting data. This is particularly useful if you have implemented a custom ABAP function module and you wish to debug it online. Refer to Testing DOB requests for more information.

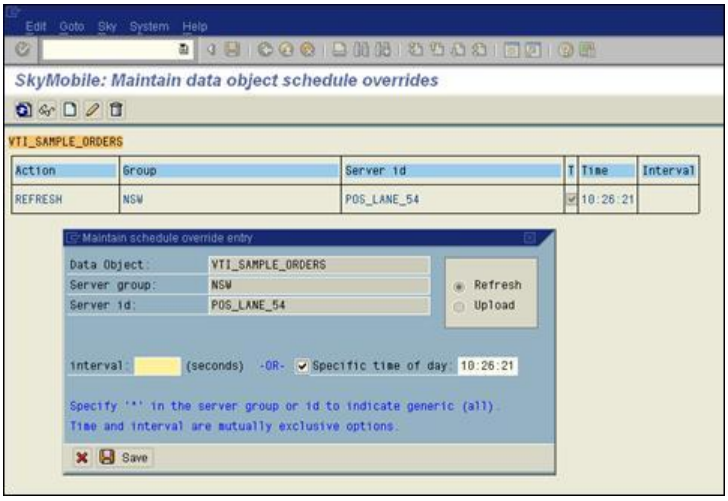
Field	Description
	This provides a quick online help reference of the options available and their behavior.
	In some cases you may wish to override the scheduling interval or time depending on the Application Server. This enables you to configure schedule overrides that take precedence over the default.

Note:

1. The data object scheduling configuration takes precedence over any of its LDBs that have theirs nullified. This means that once a LDB is part of a DOB that is downloaded to a Application Server, downloads refresh and you can trigger only uploads at the DOB level. You may however directly maintain LDB definitions from within the application and from programming exits.
2. If you configure an automatic download, refresh or upload all, then the configuration of the underlying LDBs is also checked to make sure that the option is supported. If it is not, then the user is prompted to bypass the LDB; set the no download/refresh/upload flag on the DOB table definition.
3. You may specify a LDB only once with a DOB definition.

4.8.4 DOB Schedule Overrides

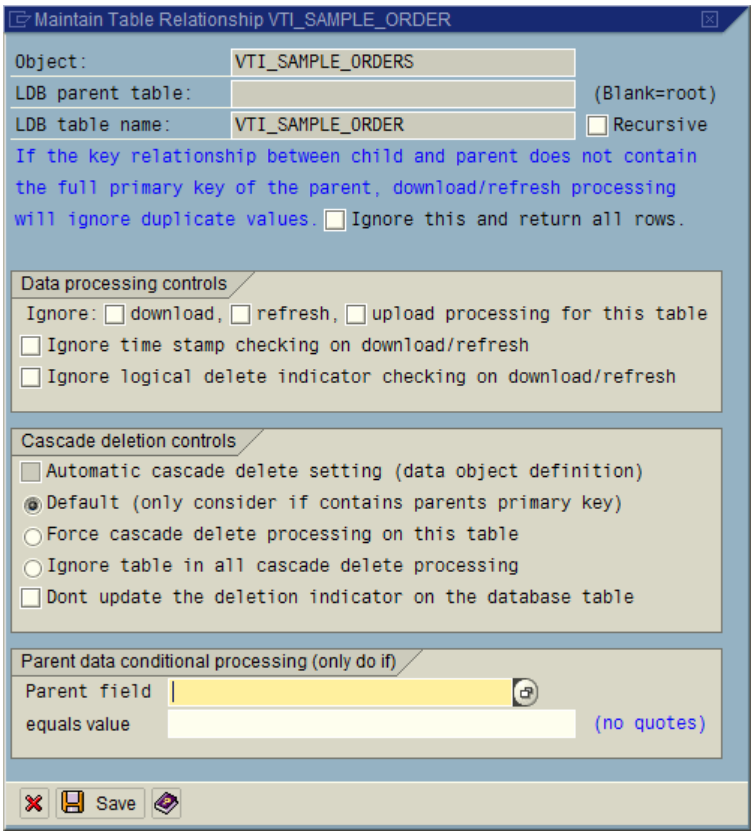
You select this option from the main workbench utility/Application Server menu. The Data Object (DOB) schedule override facility enables you to configure different refresh and/or upload times or intervals for different Application Servers. By default, the configured time or interval is used for all and this can cause performance problems. This option enables you to effectively stagger data synchronisation operations so the impact on the host system is less noticeable. The utility allows you to maintain a list of override definitions.



The pop-up allows you to specify the DOB, server group/id and transfer mode that the rule is to apply to. A new time or internal is configured. When you select the DOB definition to download to the Application Server, any schedule overrides automatically apply.

DOB Tables

You may add one or more LDB tables to a data object definition. You specify a 'root' or top level table using a blank parent name. Child tables are 'chained' together using the parent name. You position your cursor at the level you want to insert the table and click the create icon in the application toolbar; position on the DOB definition to create a 'root' table or on a existing DOB table to use as the parent.

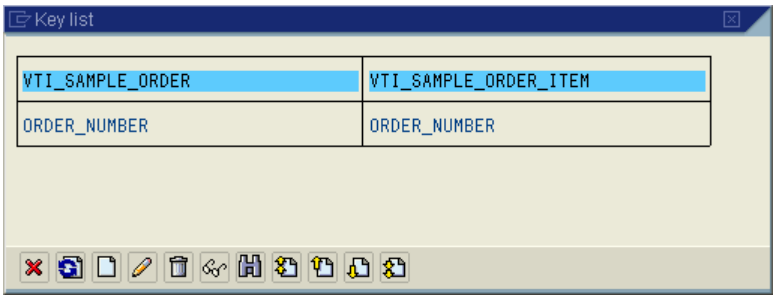


Field	Description
LDB parent table	Leave blank for 'root' table; is a top level table. Otherwise, specify an existing DOB table to form a hierarchy.
LDB table name	The name of a existing LDB to incorporate into the DOB.

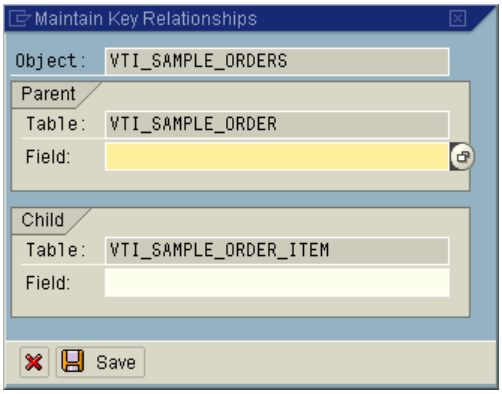
Field	Description
Ignore download/refresh/upload	These flags are used to selectively exclude the DOB table from the specified operation. There are special cases where a table is just used for reference and is excluded from data uploads; or is just a working table used just in uploads. ***Note:*** If a table is ignored, all its children are also ignored.
Recursive	
Return all rows	
Ignore time stamping	This forces any LDB timestamp checks to be ignored on refresh operations; if you want all the data irrespective of the timestamp values.
Ignore delete indicator	Ignore any LDB logical deletion indicator checking; retrieve all data irrespective of whether it is flagged for deletion or not.
Cascade controls	These options control how any cascade operations are processed. By default, a cascade operation cascades between a parent and a child if the child table contains the entire primary key of the parent; hierarchical dependency. You may override this behaviour, forcing a cascade between a parent and child table, by setting the 'Force cascade' flag on the child table - regardless of the direction of the cascade. You may ignore a table completely by setting the ignore flag, however this prevents the cascade from flowing past this table. You may also suppress, within SAP only, the updating of the deletion indicator if a deletion indicator exists.

DOB Key Relationships

Once a parent / child table relationship is formed, you are automatically prompted to add the field-to-field relationships that form the DOB table key. These relationships appear on the DOB hierarchy and you may maintain by clicking on 'key' icon. A pop-up list appears from which you add and remove the field relationships.



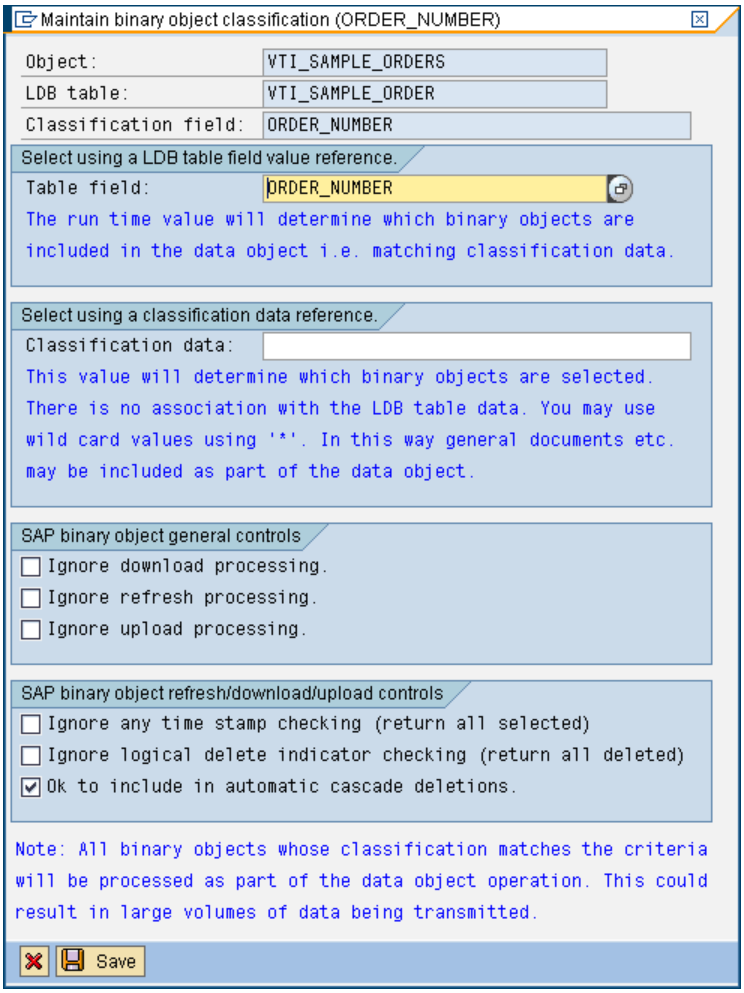
To create a key field relationship, click the create icon and the following screen appears:



Select the parent and child table fields using the drop-downs.

DOB Binary Objects

You may include the Binary Object definitions in a Data Object. This is achieved by linking the contents of a table field with the classification data of a Binary Object. Once this is done, all related binary objects are now synchronized and maintained as part of the Data Object and are no longer controlled by the Binary Manager.



Field	Description
Classification field	This is the name of a binary object classification field, used to form a key relationship between that data object table and objects in the binary repository.

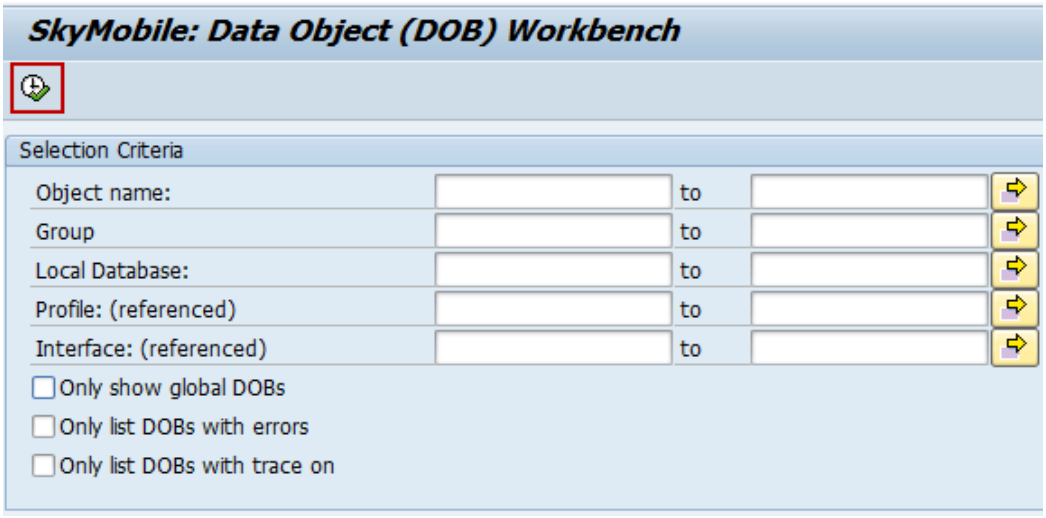
Field	Description
Table field or classification data	These two mutually exclusive options either link the binary object directly to a: • Tables field data, for example, pictures associated with a work order • Classification data value, that may be specified, that has no direct relationship, for example, general documents.
Ignore download/refresh/upload	Use these flags to selectively exclude the binary objects from the specified operation.
Ignore time stamping	This forces any timestamp checks to be ignored on refresh operations; if you want all the binary objects irrespective of the timestamp values.
Ignore delete indicator	Ignore any logical deletion indicator checking; retrieve all binary objects irrespective of whether it is flagged for deletion or not.
Ok to include in automatic cascade deletions	By default, related binary objects are not included in automatic cascade deletions unless this flag is set (See the DOB definition attribute for more details).

Working with a Text Description for a DOB

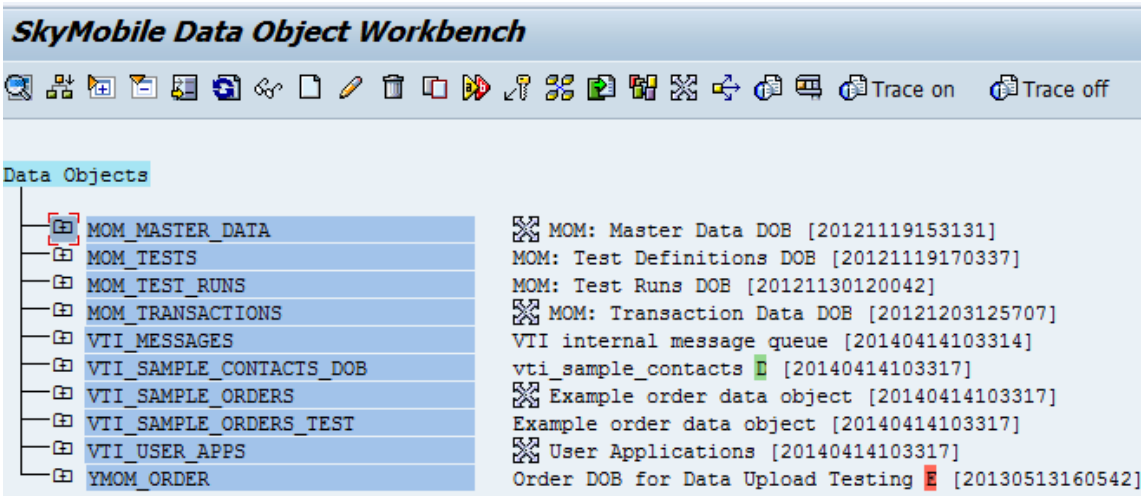
You may add extended text description for a Database Object (DOB) or a field name.

To work with text descriptions for a DOB, follow these steps:

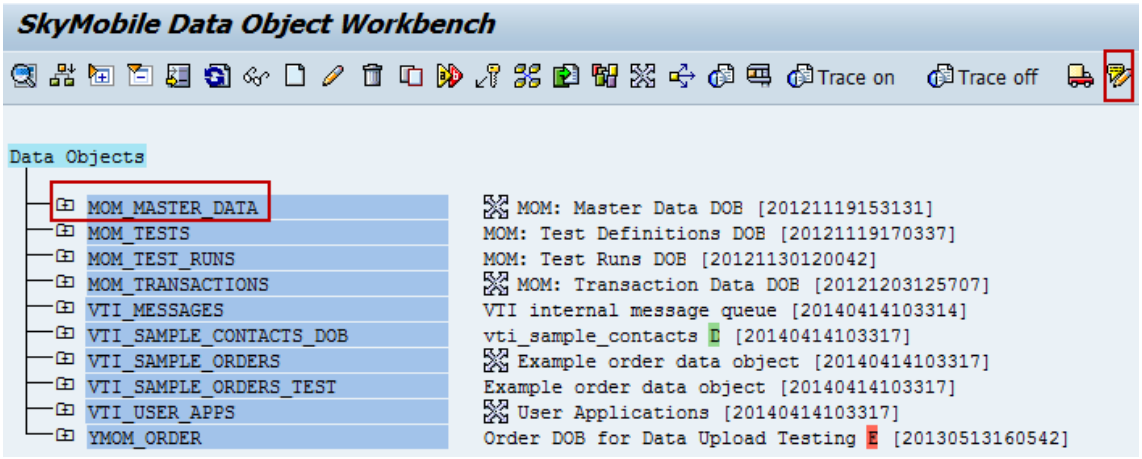
1. Execute the transaction, **/SKY/YVTD (YVTD)** in SAP system.
   The **SkyMobile: Data Object (DOB) Workbench** window appears.



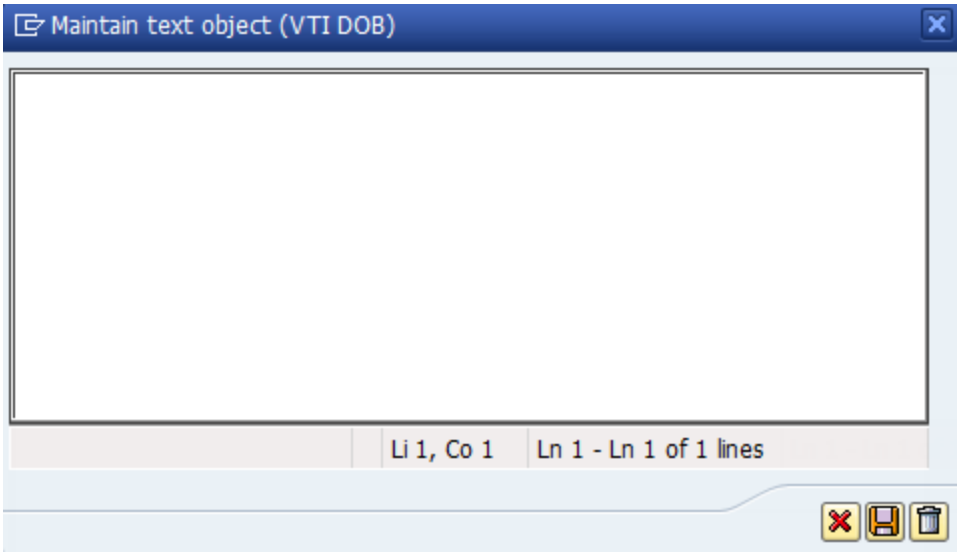
2. Click **Execute**.
   The **SkyMobile Data Object Workbench** dialog appears with the list of database objects (DOBs).



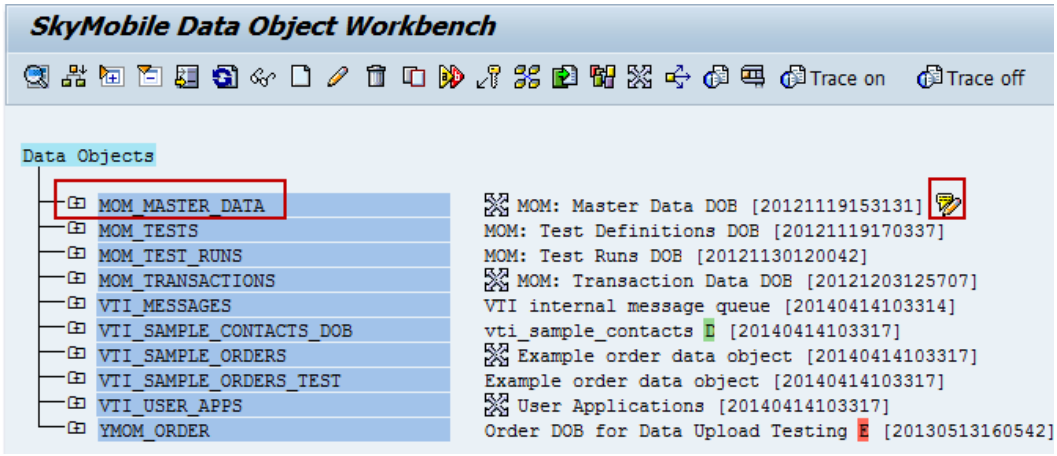
3. To create a text description for a DOB, select a DOB from the list and click **Maintain text (Ctrl + Shift + F5)**.



The **Maintain text object (VTI DOB)** text editor appears.



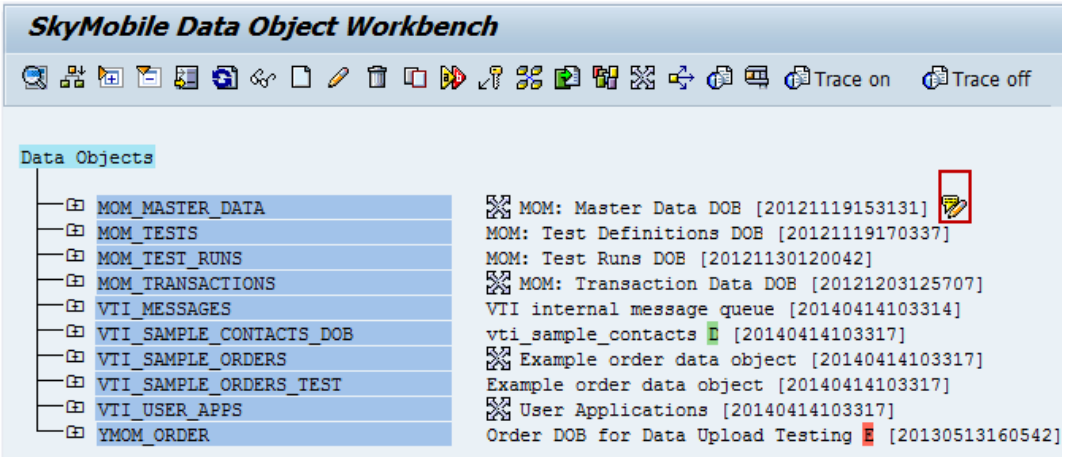
a. Enter text description that you want to add to the DOB and click **Save**.
The **Maintain text** icon appears next to the selected DOB name.



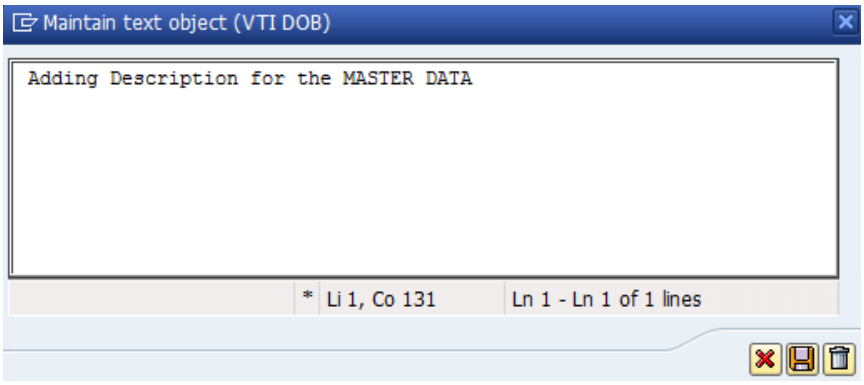
4. To change the text description for a DOB, follow these steps:

   a. Select a DOB name from the list of DOB names that have the **Maintain text** icon next to the DOB.

b. Click the **Maintain text** icon.



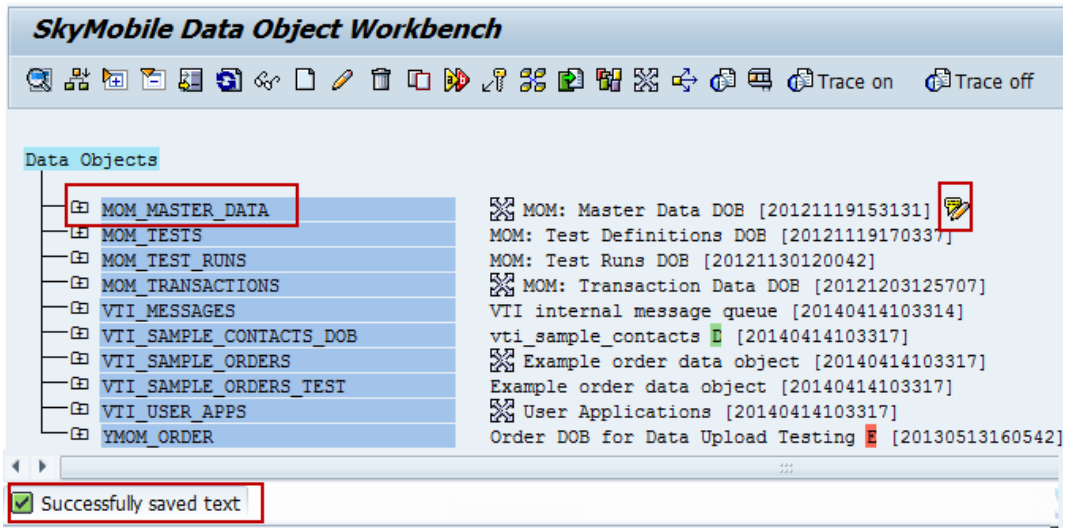
The **Maintain text object (VTI DOB)** appears with the existing text description.



c. You may change the text description, if required.

d. Click **Save**.

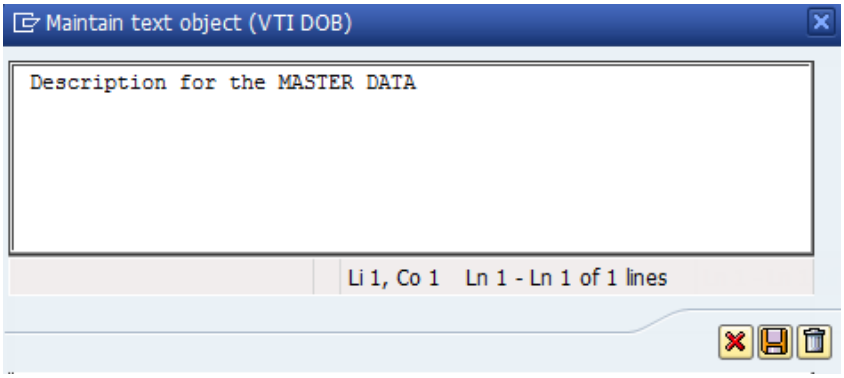
The system saves the changes. The message, "Successfully saved text" appears at the bottom of **SkyMobile Data Object Workbench**.



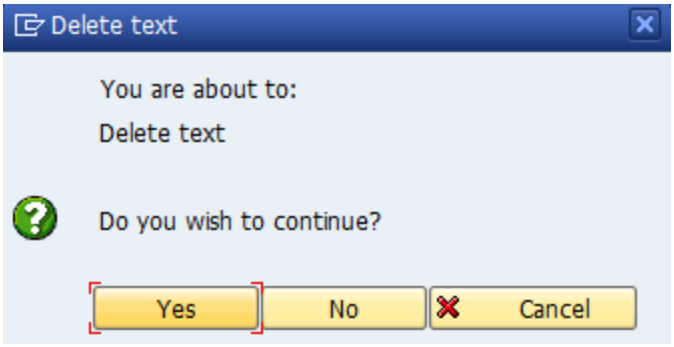
5. To delete the text description of a DOB, follow these steps:

   a. Select a DOB name from the list of DOBs that have the **Maintain text** icon next to the DOB.

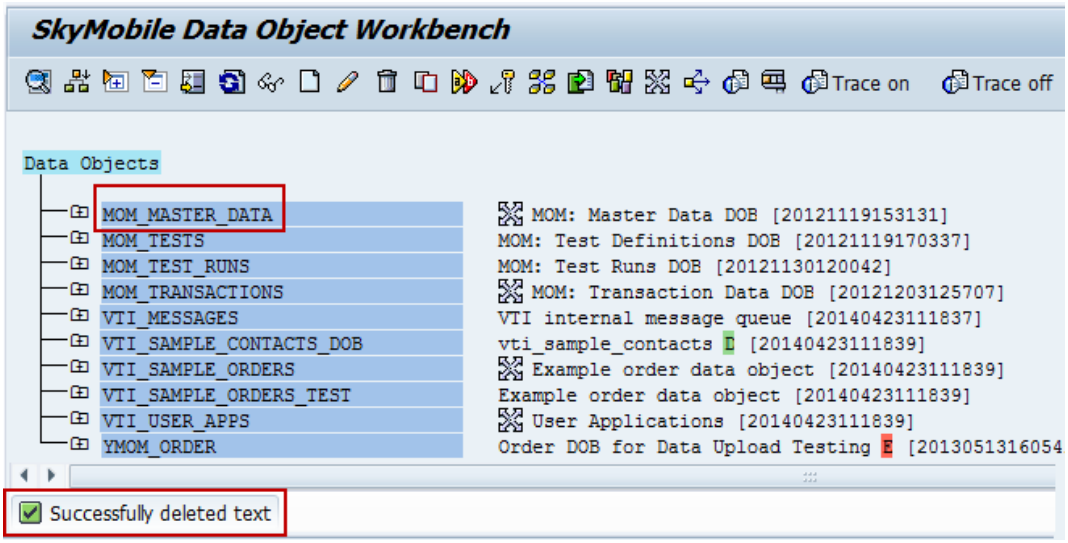
   b. Click the **Maintain Text** icon.
   The **Maintain text object (VTI DOB)** dialog appears with the existing text description.



   c. Click **Delete (Shift+ F1)** on the **Maintain text object** dialog to delete the text.
   The **Delete text** dialog appears.



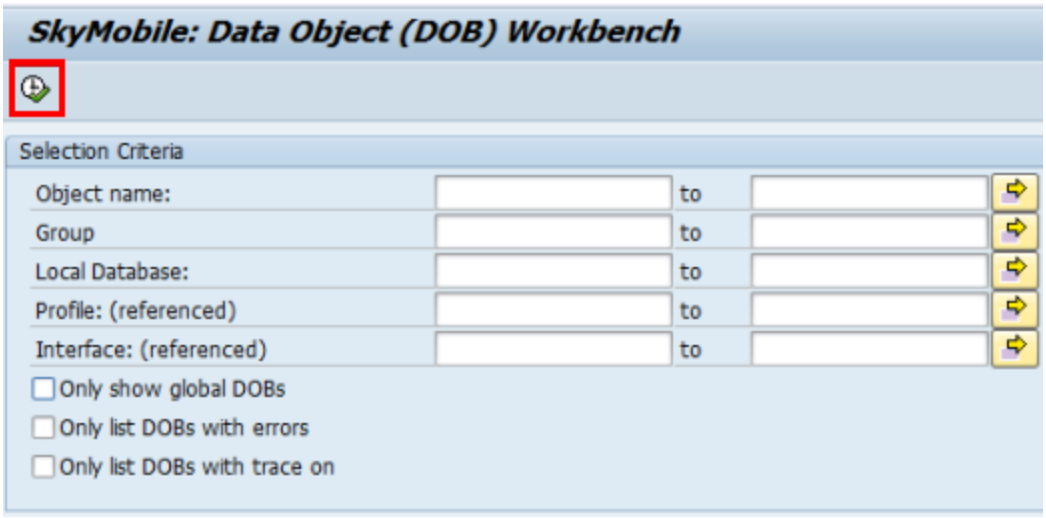
d. Click **Yes** to continue.
The **Maintain text** icon no longer appears next to the DOB name. The message,"Successfully deleted text" appears at the bottom of **SkyMobile Data Object Workbench**.



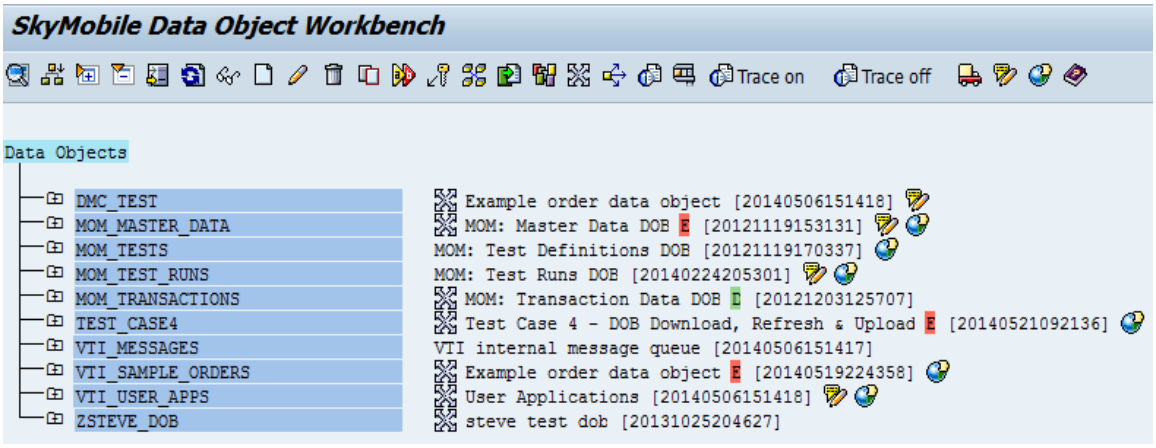
Generating a Gateway Service

To generate a gateway service for a DOB, follow these steps:

1. Execute the transaction, **/SKY/YVTD (YVTD)** in SAP system.
   The **SkyMobile: Data Object (DOB) Workbench** window appears.

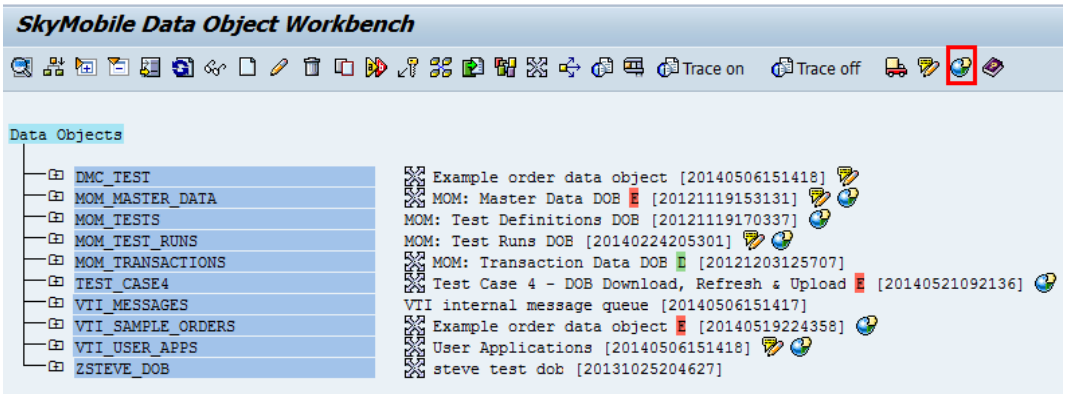


2. Click **Execute**.
   The **SkyMobile Data Object Workbench** window appears with the list of database objects (DOBs).

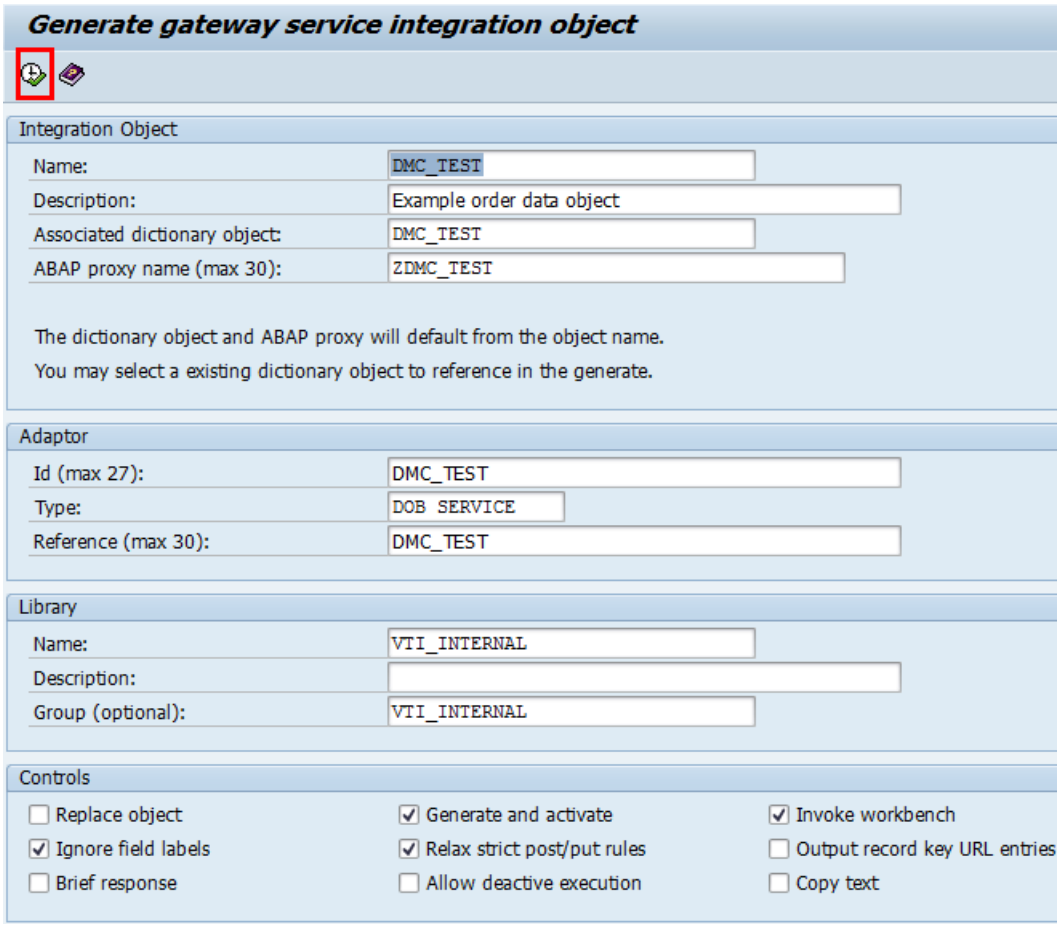


3. To generate a gateway service for a DOB, follow these steps:

    a. Select a DOB from the list and click the **Generate gateway Service (Ctrl + Shift + F5)** icon on the application toolbar.



The Integration Object Workbench is called and the **Generate gateway service integration object** window appears:



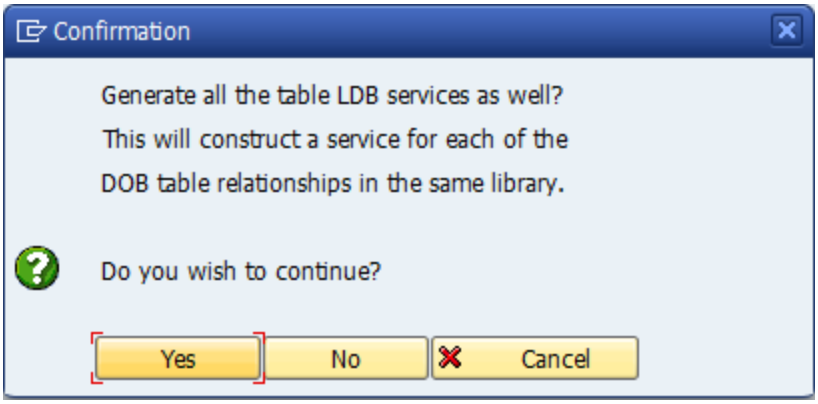
Field	Mandatory or Optional	Description
Name	Mandatory	This is a unique name of an integration object definition. It is defaulted from either the DOB or an LDB name.
Description	Optional	The text description of the IOB definition. This is defaulted from the DOB or an LDB definition.

Field	Mandatory or Optional	Description
Associated dictionary object	Optional	Each integration object has one or more associated dictionary objects. The IOB DD object defines the structures and their relationships that are to be used to map data. DD object may be shared across integration objects and are automatically generated by this utility.
ABAP Proxy	Optional	This is the name of the ABAP proxy program that is generated from the integration object definition. It should follow the usual ABAP naming convention prefixed with Z, Y or /namespace/. The name is defaulted from the object name specified above.
Adaptor		Description
ID	Optional	The name of the service adaptor, this is defaulted from the LDB name for which you want to generate the service from.
Type	Mandatory	This is the type of service adaptor used to extract from, or post transactions to, SAP, for example, LDB SERVICE, LDB REFRESH, and so on.
Reference	Mandatory	This refers to the name of the LDB that you want to generate the service from.
Library		Description

Field	Mandatory or Optional	Description
Name	Mandatory	The name of the library definition that the object belongs to. This is defaulted from the SkyMobile group.
Description	Optional	The library description. This is defaulted from the SkyMobile group.
Group	Optional	This is defaulted from the SkyMobile group. This is an optional field.
Controls		Description
Replace object	Optional	As the title implies, existing definitions are not replaced unless this option is specified.
Generate and activate	Optional	Generate the ABAP Proxy program and activate the object. This effectively enables the service to be immediately available, exposed to external clients.
Invoke workbench	Optional	After the object is generated, launch the integration object workbench.
Ignore field labels	Optional	Ignores field labels for reserved strings
Relax strict post/put rules	Optional	Ignores restrictions about Unique Records and Duplicate Records during PUT and POST operations

Field	Mandatory or Optional	Description
Output record key URL entries	Optional	For every LDB record in the response body, a <recordkey> element is populated that holds the full URL of the particular record.
Brief response	Optional	Ensures that no response body is returned for PUT, POST and PATCH calls
Allow deactive execution	Optional	Allow execution of service call against the DOB or LDB even if the corresponding integration object is in deactivated state
Copy text	Optional	Copies text descriptions from the DOBs, LDBs and LDB fields

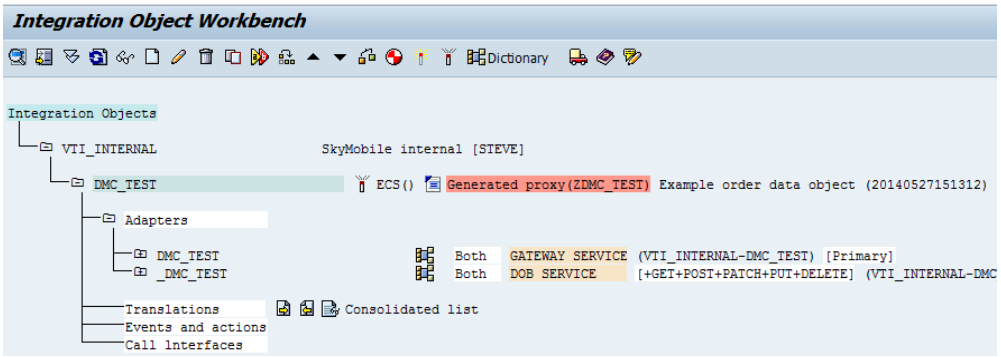
b. Enter the name of the integration object in **Name** and click **Execute**.
The **Confirmation** dialog appears.



i. Click **Yes**.
A service for the DOB and a separate service for each of the LDBs related to the

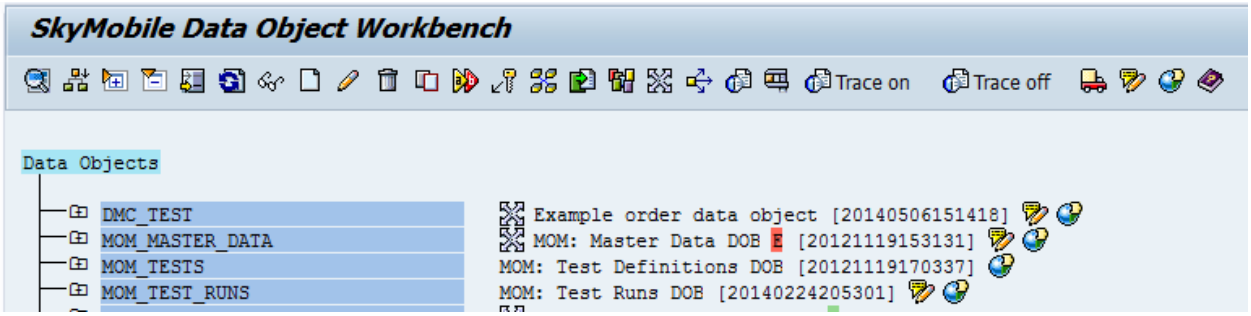
DOB are created.

ii. Click **No**.
A service only for the DOB and no service is created for the LDBs. The service is created as given below:

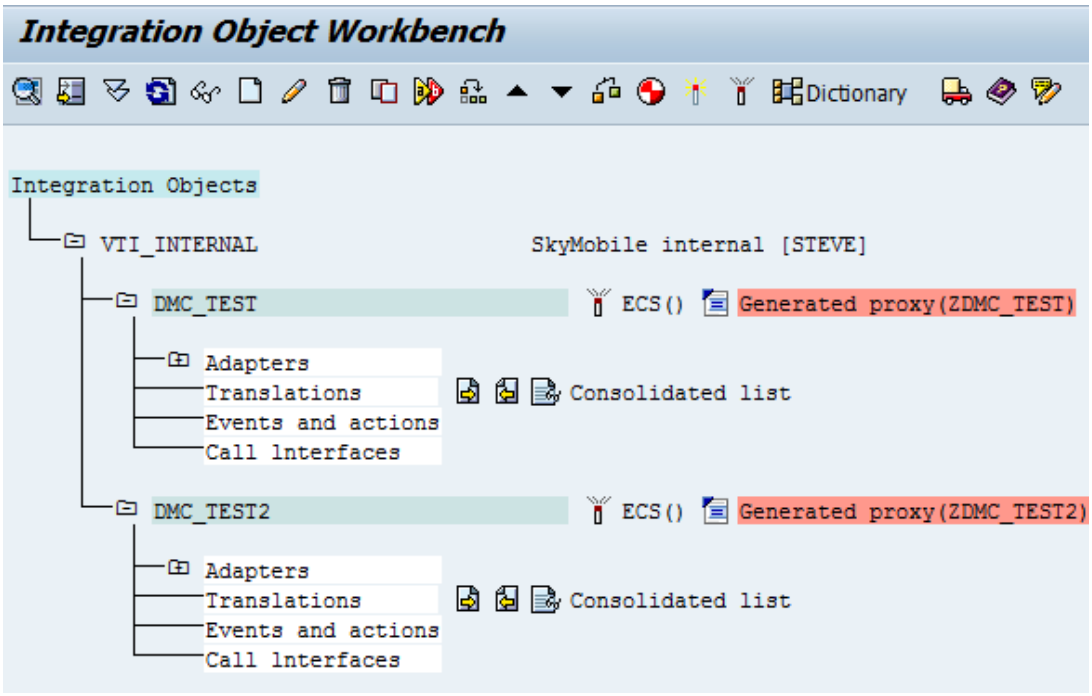


Navigating to Integration Object Workbench

The **Generate gateway Service (Ctrl + Shift + F5)** icon appears at the DOB level when an integration object exists for that DOB.

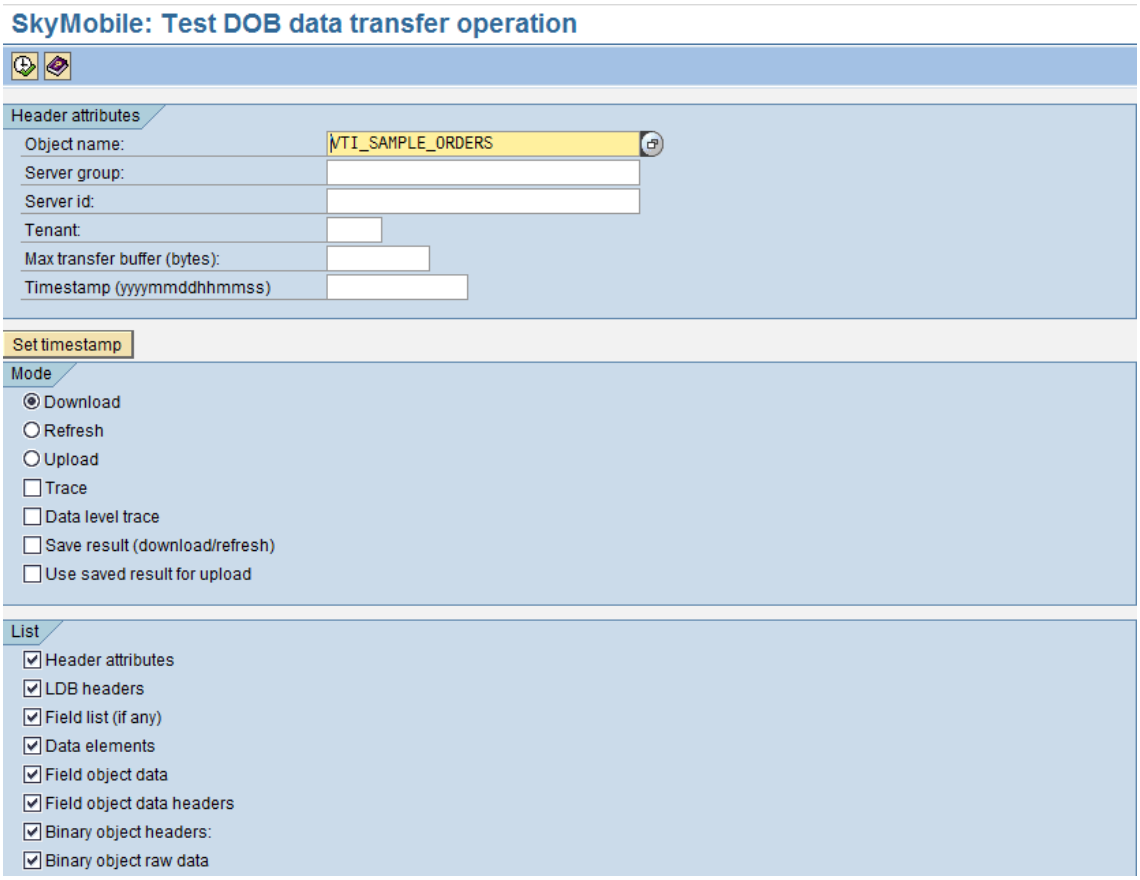


In order to see the list of gateway services of a specific DOB, click the **Generate gateway Service** icon in-line with the DOB name. The Integration Object Workbench is triggered and the list of integration objects where the DOB is used appears.

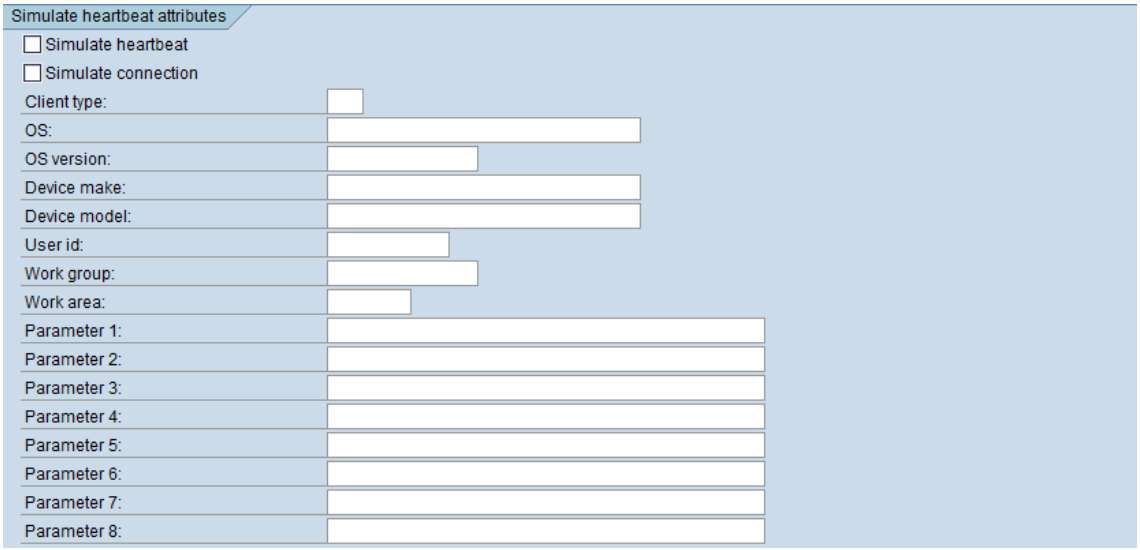


Testing DOB Requests

A useful facility is the ability to test download, refresh and upload transfer requests; simulate what data is processed. You may even extract data using a refresh and then import it again through an upload. This test facility is available from the DOB definition popup through the Test con. This invokes the test selection screen where you specify call attributes, transfer mode, and list format. Remember to specify server and timestamp requirements carefully. To test an upload, you first need to perform a download or refresh and specify the "save result". Return to this screen from the list, change the mode to upload and specify "use saved result". You may also 'replay' a data object request using the captured trace data.

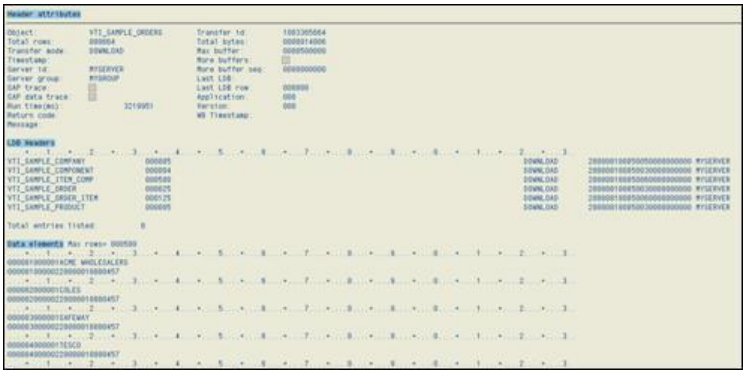


You may enter extra criteria to simulate a heartbeat connection specifying user attributes; test user level substitution and profiling. You may use the following attributes:



Execute the test selection screen and the following list appears.

> ***Note:*** The "Max data elements" to list setting, defaults to 200.



This is effectively a character string dump of the DOB transfer header, LDB headers, data element list and object data lists. These are based on the following structures:

- /SKY/YVTI_TSFHD DOB Header
- /SKY/YVTI_TSFHD LDB Headers
- /SKY/YVTI_TSFDT LDB Data element list

- /SKY/YVTI_OBJH LDB Object header entries
- /SKY/YVTI_OBJD LDB Object data entries

> ***Note:*** You may double-click on any line to view the entire structure.

Tracing DOB Requests

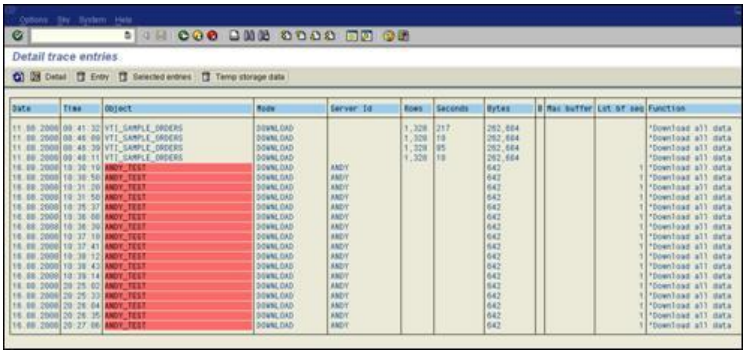
The DOB manager has a facility to trace the download, refresh and upload requests and optionally, the data transferred. This is useful when you are either develop or support an application. The trace facility is set using the DOB definition (header) pop-up. If the trace is on, a highlighted "green T" appears on the left hand side in the DOB list.

> ***Note:*** The trace may also be automatically set for a specific server using the Server trace facility. See the systems management guide for more information.

If a 'data level' trace is selected, all the call input and output parameters are written to a file in the default SkyMobile working directory (configured in the default system settings using the YVTI systems management console). If a data trace is active, a green "D"appears and every call is dumped to file. You may view the data trace using the glasses button in the trace detail popup.

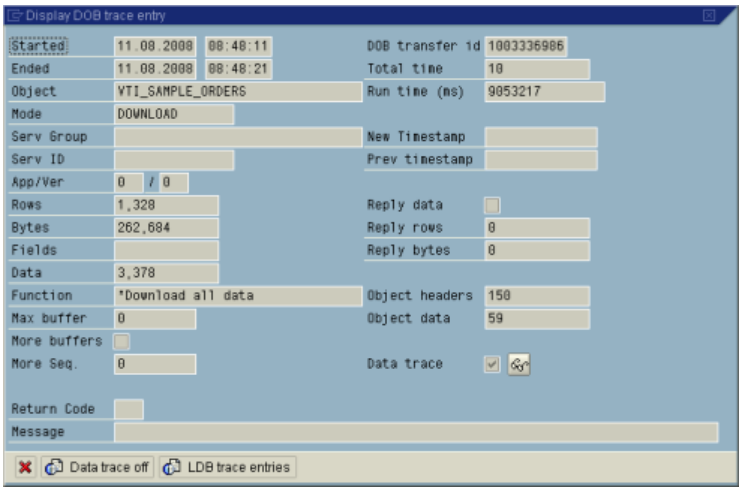
> ***Note:*** Iif you set tracing on at the DOB level, this automatically sets tracing on at run time for any underlying LDB calls. The corresponding LDB trace entries are linked to the DOB trace entry and many appear using an option from the trace detail pop-up.

You can access the trace list and control options through the "Utilities->Trace Entries" menu option in the DOB workbench or through the **Utilities->SAP Host->DOB trace** option from the application workbench. The list has a selection screen used to filter entries, for example, server id, and date and select various different views, for example, detail, summaries. For example, the following is an example of the detail list.



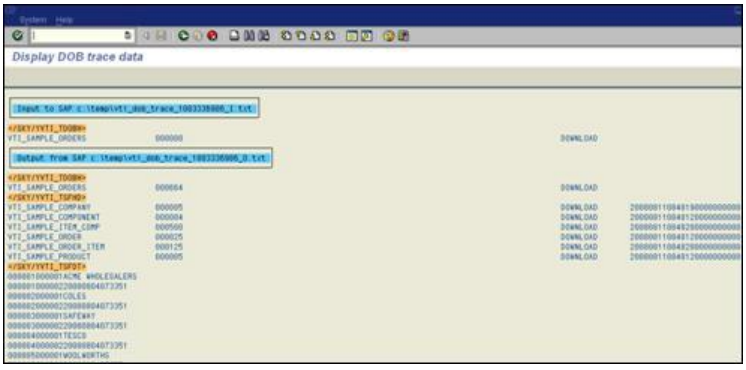
You may double-click a line to view more detail and display any associated data trace (if specified).

> ***Note:*** Note the toolbar options to view the associated LDB entries and turn the data trace off.



Data Level Traces:

If a data level trace is specified, all the low level input, processing and output details are captured into a trace file. A button with glasses appease adjacent to the 'Data trace' field. Click this to view the input, flow trace and output data sections (see below). You may also replay a captured 'data trace' request. This is useful to analyse and diagnose processing.



You may double-click any line to view the data structure.

Cascading DOB Updates

A Data Object change is only processed if the high level parent table (root table) timestamp changes and you must subsequently change all subordinate child table and binary object timestamps as well (unless ignore time stamping is explicitly specified). Therefore if you changed an item within an order, you need to update the timestamp of the header as well so that the change is picked up. This can be a tedious task if a data object has many levels.

Kony provides tools to automatically cascade both logical deletions and time stamp updates. You may specify 'automatic cascade deletion' on the data object definition or use the ABAP SDK commands: VTI_DOB_CASCASE_DELETE and VTI_DOB_CASCADE_TIMESTAMP. Refer to ABAP programming guide for more details.

There are a number of settings that you can used to influence the way that cascading behaves. These are found in the DOB workbench on the Maintain Table Relationship screens.

4.8.5 Binary Objects (BOB)

This facility is used to manage binary objects (BOB) in SkyMobile to facilitate the integration of files, such as graphics, attachments, signatures, drawings and software. A binary object is essentially an external file that an application may associate or reference, for example, graphic, attached document, and drawing object. Only certain types are allowed, as configured in the `/sky/yvti_filet` file type table populated by the check install. You find that most common file types are supported, for example, `jpg/tif/gif/bmp/doc/txt`. Once imported centrally into SAP, you may configure the binary files to automatically distribute to remote Application Servers or associated directly with applications either statically or through complimentary reference. Facilities are also provided to integrate SkyMobile binary objects with the SAP GOS (Generic Object Sub-system). See the ABAP programming guide for details.

> ***Note:*** Since the binary file definitions are stored centrally in the SAP database as raw data, you should control what and how many definitions are loaded to avoid problems with database space. Most graphics are reasonably small in size, but media type files can cause problems.

The binary object workbench provides a central repository and functionality to synchronize files with devices. APIs exist for ABAP and Java programs to interact with the manager (see the programming guides for details). A packaging tool is also provided that you may use to deploy software upgrades. For a complete description, refer to the systems management guide. All binary objects are stored in character hex encoding; stored as hex character strings such as 'F0FFF00A0D…… This is to provide a totally portable format across all UTF encoding, binary, text. However, this means that often you require more space than the original file. Each object definition is time stamped; you may refresh all remote Application Servers if a definition changes.

4.8.5.1 Logical and Physical Names

Each Binary Object is uniquely identified by a logical name (max 32 characters) and a version number (0-99), irrespective of type, and therefore careful naming conventions are required to prevent

definitions from being overridden in the central SAP repository. The physical name is what the actual file is when it is created in a directory. If not specified, the physical name defaults to the logical name. Similarly, if the extension is not embedded with the physical name, the imported file extension (if any) is used.

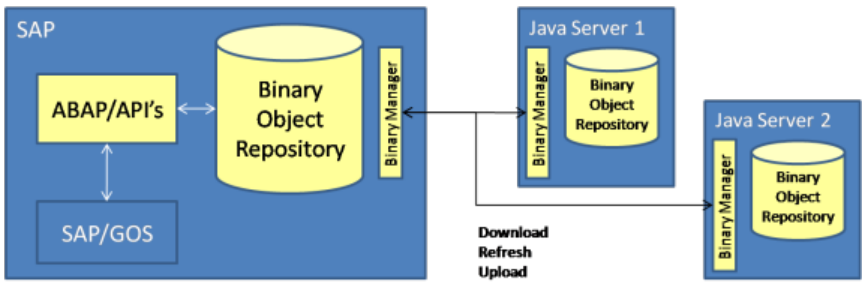
4.8.5.2 Types and Locations

Each binary object is allocated a type and location code. The type is used to categorize objects by their function, for example, graphic, sound, document, video, and executable. This is for easy identification, filtering and selection purposes. The SkyMobile check install process sets up the standard types and if not specified, the system automatically allocates one based on the file extension. The location code dictates where to store the physical file on the device. There are default system location codes, or you may specify your own custom ones and cross reference them with target directories, using location cross references in the Application Server configuration. In this way, you can dictate where you can physically store objects without specifying the actual directory paths, for example, SDCARD, DOCUMENTS, GRAPHICS, SOUNDS, and PICTURES.

4.8.5.3 Classification

Whilst an object is uniquely identified by its logical name and version, you may also allocate it an alternate reference that is called a classification. Classification data is made up of a field/value combination, for example, WORK_ORDER / 000569135. Classification is useful as an alternate business reference and you may use this in all APIs instead of the logical name and is required to link Binary Objects to Data Object definitions; a key relationship with table data.

4.8.5.4 Synchronization



In the diagram above, there is a central binary repository in SAP that is synchronized with remote binary repositories on the mobile Application Servers. This synchronization is similar in principle to data synchronization, except that there are three distinctly different modes:

- Stand alone objects:
    - directly linked to a application
    - created and linked to a specific server.
- Objects linked directly to Data Objects (through classification).

4.8.5.5 Application Level

These are binary objects that are linked explicitly with an application definition, either by referenced in screen functions or configured as complimentary objects. These objects are downloaded and refreshed as part of the application synchronization, for example, graphics used in screens.

4.8.5.6 Application Server Level

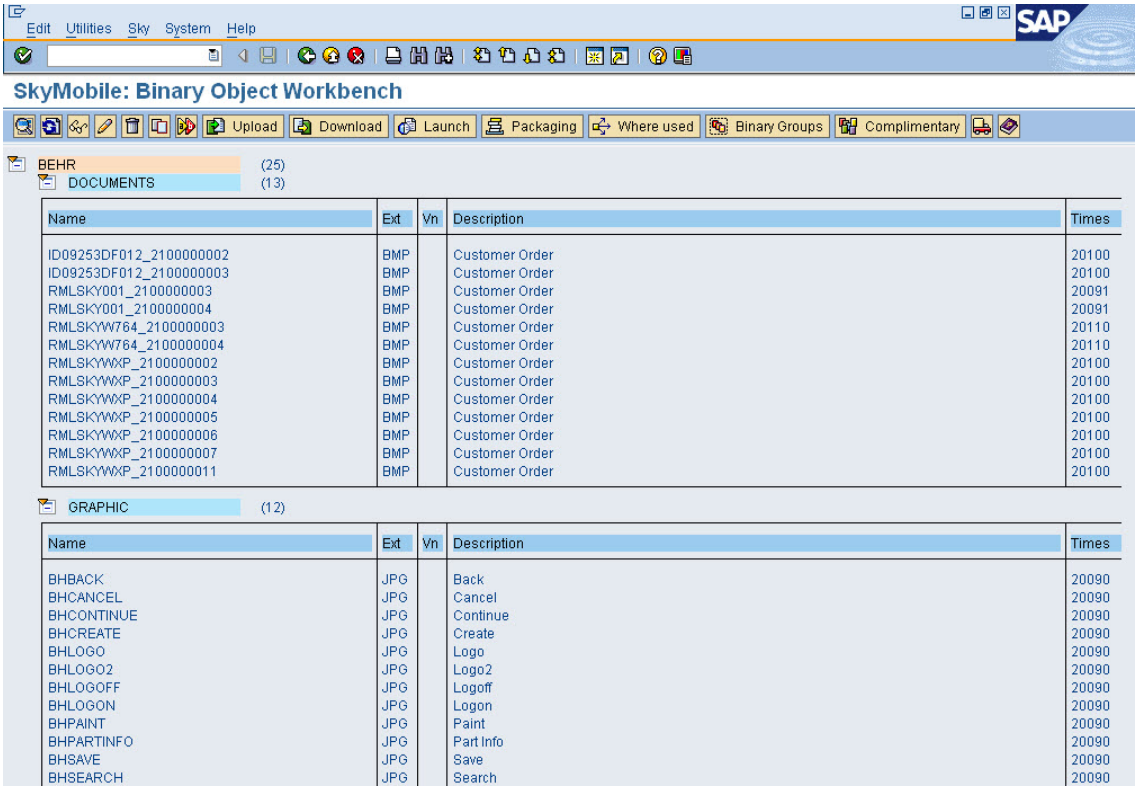
You may associate a binary object to a specific Application Server that is either scheduled for it (for example, document) or was generated by it (for example, a signature). In either case, the binary manager stores the objects centrally in SAP and automatically synchronizes any change made from either end. In this way, you may automatically restore all the binary objects that belong to a Application Server. To determine which objects go to which servers, the binary object is allocated a group that is associated with one or more 'binary group' definitions that act as filters for the download and refresh operations. An internal LDB table 'VTI_BINARY_HEADER' contains all the control information of each binary object and when the Application Server detects a change, it requests (selects) the full binary object from SAP and loads it into its local repository.

4.8.5.7 Data Object Level

One or more binary objects are linked to a data object table through a field / classification key reference. Once a data object owns a binary object, it is totally controlled by DOB synchronization and any specific individual processing by the binary manager is ignored. To do this effectively, you must properly maintain the classification data to link the object to the data object table data. You may have as many relationships as you want with tables at any level within the data object.

4.8.5.8 The Binary Object Workbench

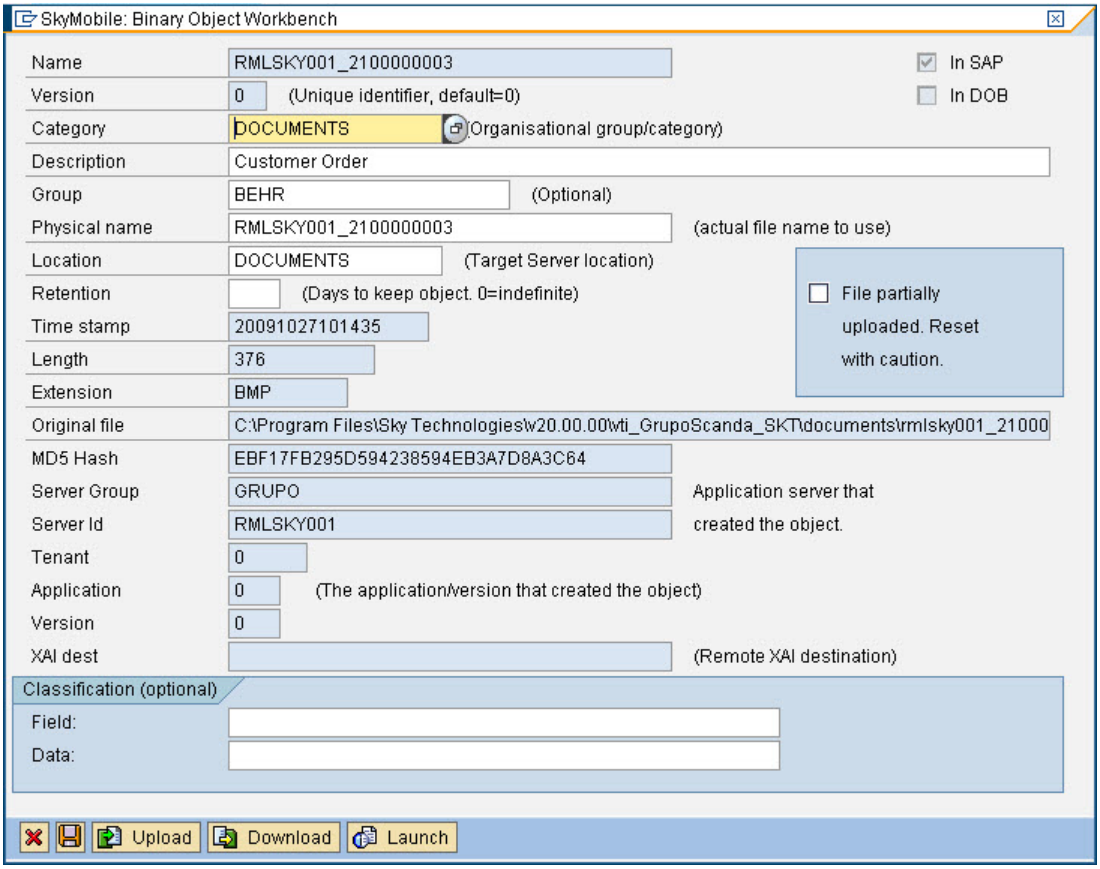
You may use a standard icon:  on the application toolbar to invoke the Binary Object Workbench manager. A selection screen enables you to filter the list that is sorted into group order. Groups are effectively used to help organize the objects into categories. Each object is allocated a type or class that further helps to organize the objects; graphic, and document.



From the Main Workbench Screen, you can:

- Upload and download files to and from the desktop
- Launch/execute files from the desktop; view images, and invoke text editors
- Change, copy and delete objects
- Package files to be downloaded to remote Application Servers

When you maintain binary objects, the following pop-up appears:



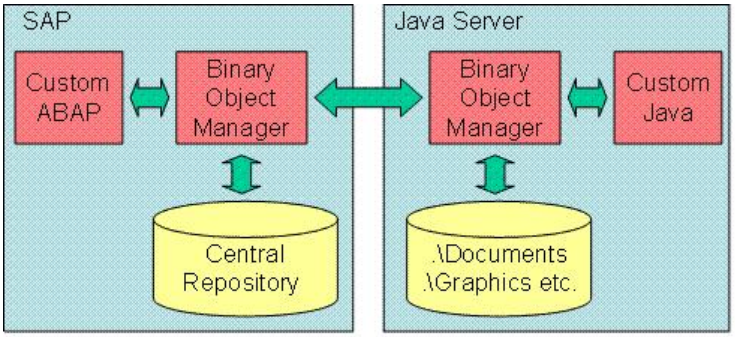
Field	Description
Name	This is the unique logical name of the object.
Version	You may use a version number to differentiate between multiple instances of the same object.
Category	The broad category that the object falls into, for example, GRAPHIC, PACKAGE, or DOCUMENTS.
Group	You may organize the object into a group. This is important for synchronizing objects with remote Application Servers.
Physical name	The actual physical name given to the object when it is written to disk.

Field	Description
Location	The default location (directory_ where the object is to be stored on a Application Server, for example, CLASSES (Custom Java classes), DOCUMENTS, ECSLIBRARY (Sky general systems library), GRAPHICS, INSTALL (the root SkyMobile installation library), SDCARD, TEMPLATES.
Retention	How many days to keep the object after it is created. Specify *0* days to keep it indefinitely.
Classification	This is an alternate reference to the object other than the logical name. It is usually used to link the object to a SAP business document such as a work order. Classification data is required to link Binary Objects to Data Object tables. You can use a classification reference in place of the logical name to reference a binary object in APIs.
MD5 hash	This is a unique identifier that the Application Server generates to determine whether a binary object is changed. Refer to The MD5 hash key for more details.

4.8.5.9 Using Groups to Organise and Synchronise Objects

If an application uses drawing objects and/or attachments, groups become very important to help organise and segregate large numbers of binary objects. For example, if a signature is captured as part of a 'proof of delivery' transaction, and the signatures are to be uploaded to the central host, how do you separate the signature bit map objects from other binary objects and also prevent them from being downloaded to every server. You may specify a group that is an optional free format value in the Application Server configuration file, against the application/version and on the drawing object definition. The preference is in order of drawing object, application/version, and then the configuration file.

If you wish to control which objects get synchronised between the host and remote Application Servers, you use binary object groups and the group cross reference utility. See the section Application Server utilities for more details. The group cross reference is used to effectively filter which objects get downloaded and refreshed to Application Servers through the internal VTI_BINARY_HEADER local database definition using the normal timestamp/delete indicator technique. If no group cross reference is configured, then no objects are distributed. You control where the object gets downloaded to on the Application Server using the type or class, for example, graphic or document places the file into the `.\graphic` or `.\document` directories.

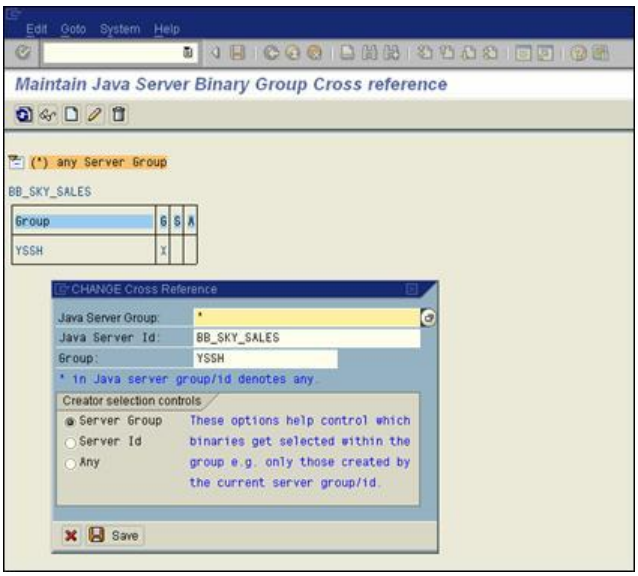


For more complex scenarios, it is also possible to programmatically change the server id and binary group reference on the VTI_BINARY_HEADER local database definition or `/sky/yvti_fileh` SAP table through ABAP or Java. There are also standard options to specify these in the ABAP binary object APIs.

> ***Note:*** Refer to the ABAP Programming Guide and the "Application Server Binary Group cross reference utility" section for more details.

Binary Group

This option is selected from the main workbench utility/Application Server menu. You may assign binary objects to a specific group. In order for binary objects, such as drawing objects, and attachments to automatically synchronize between the SAP host and specific Application Servers, you must maintain a binary group cross reference. This configuration enables you to effectively segregate the objects that is which get synchronized where?



The 'creator' selection controls govern whether only binary objects created by the specified server id and/or group are to be synchronized. For example, if you select 'Server Group' level, then any binary object created by any server within the group is replicated to all other servers within the group. If you select 'Server Id' level, then only those binary objects that the Application Server created is replicated.

The server group, server id and binary group are allocated to a binary object when it is either maintained locally by the Application Server application or through ABAP APIs. You also have the option to manipulate these settings by changing the VTI_BINARY_HEADER local database definition or `/sky/yvti_fileh` SAP table. Default group names are explicitly defaulted by specifying a value against:

- A drawing object screen element
- The application version

- The Application Server VTI configuration file [BinaryFiles] DefaultBinaryGroup option

> ***Note:*** The above order of defaulting applies.

The MD5 Hash Key

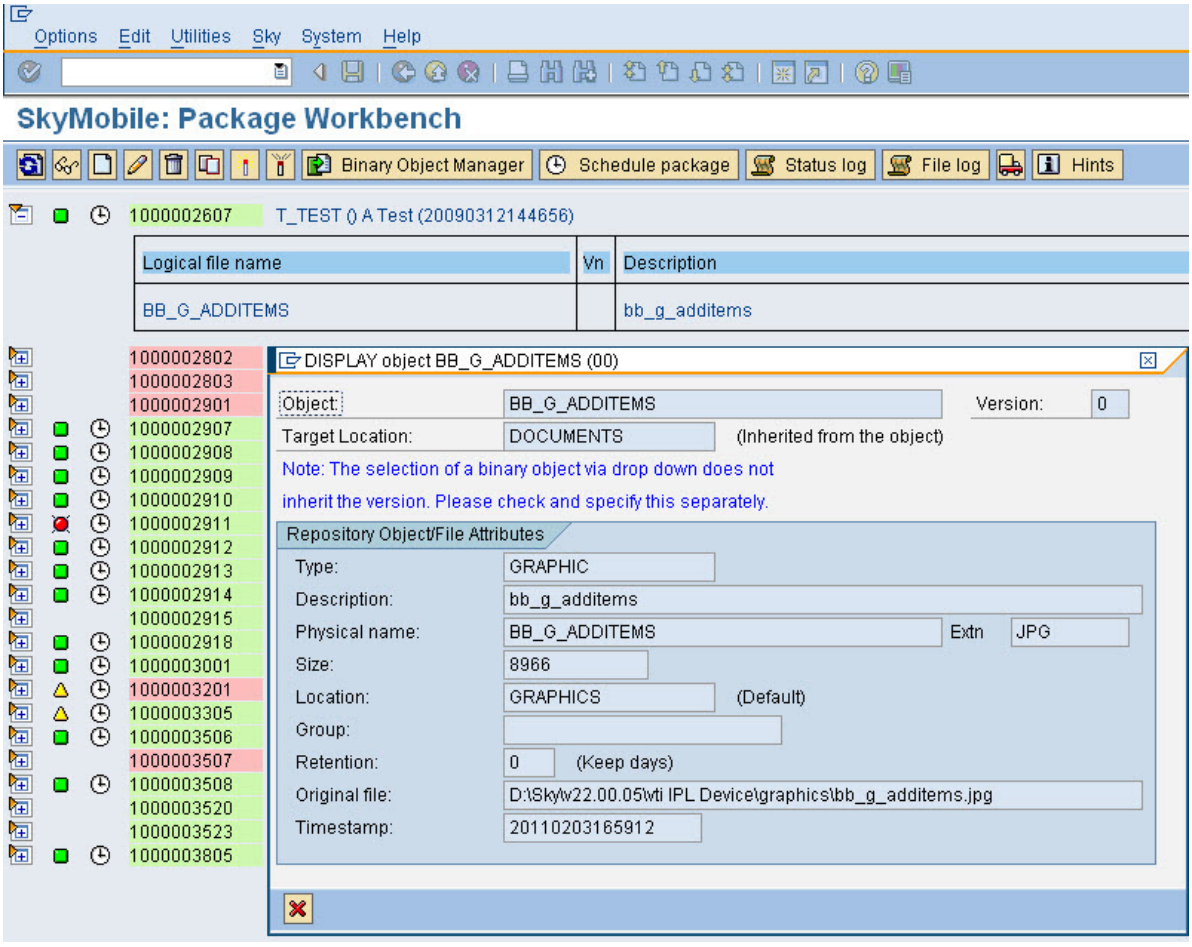
The Application Server associates a special internal MD5 hash key value with each binary object. This value is a unique calculation of the object contents and is used to determine whether the object actually changed or not. This is an important mechanism used to prevent the unnecessary transfer of binary objects that can be quite large; if the file already exists on the Application Server and the MD5 hash is the same as the host, then do not synchronize the data. The Application Server only performs all MD5 calculations.

4.8.5.10 Packaging Utility

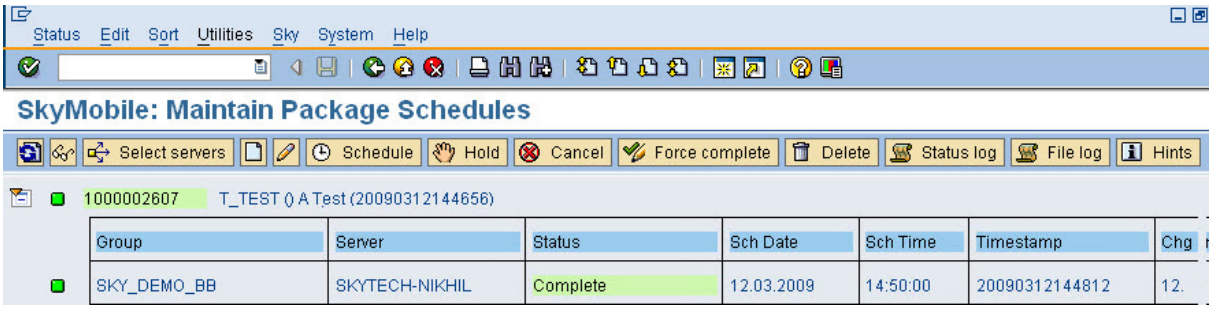
This is an extension of the binary manager that enables you to "package" up one or more binary objects and schedule them to download to one or more Secure Containers. It is used primarily to remotely implement and manage changes centrally from the SAP host. A full description of the packaging utility and how to use it is in the SkyMobile "System Management" guide. Also refer to the change management section on Binary object packages in this guide.

> ***Note:*** At this stage, the packaging facility is only available on Windows platforms.

The packaging utility allows you to create, maintain and schedule package definitions to one or more Application Servers. You add binary objects to the package, so you must first upload the objects into SAP.



Once the package is defined, you then schedule it to download and implement on Application Servers. A status is updated to give an indication of where the deployment is at.



4.8.5.11 On Demand Binary Objects

The On Demand Binary Objects functionality is designed for cases where the data content of a binary object is not synchronized with the device and instead is retrieved on request by the application when required. This prevents large amounts of "attachment data" from being stored on the device and improves the overall synchronization performance. However, the trade-off is that after making a request for data, a user must wait while the download completes or fails if no connection with the host is possible. You may either configure all the binary objects associated with a data object on demand or flag individual binary objects to be on demand the workbench or import API functions.

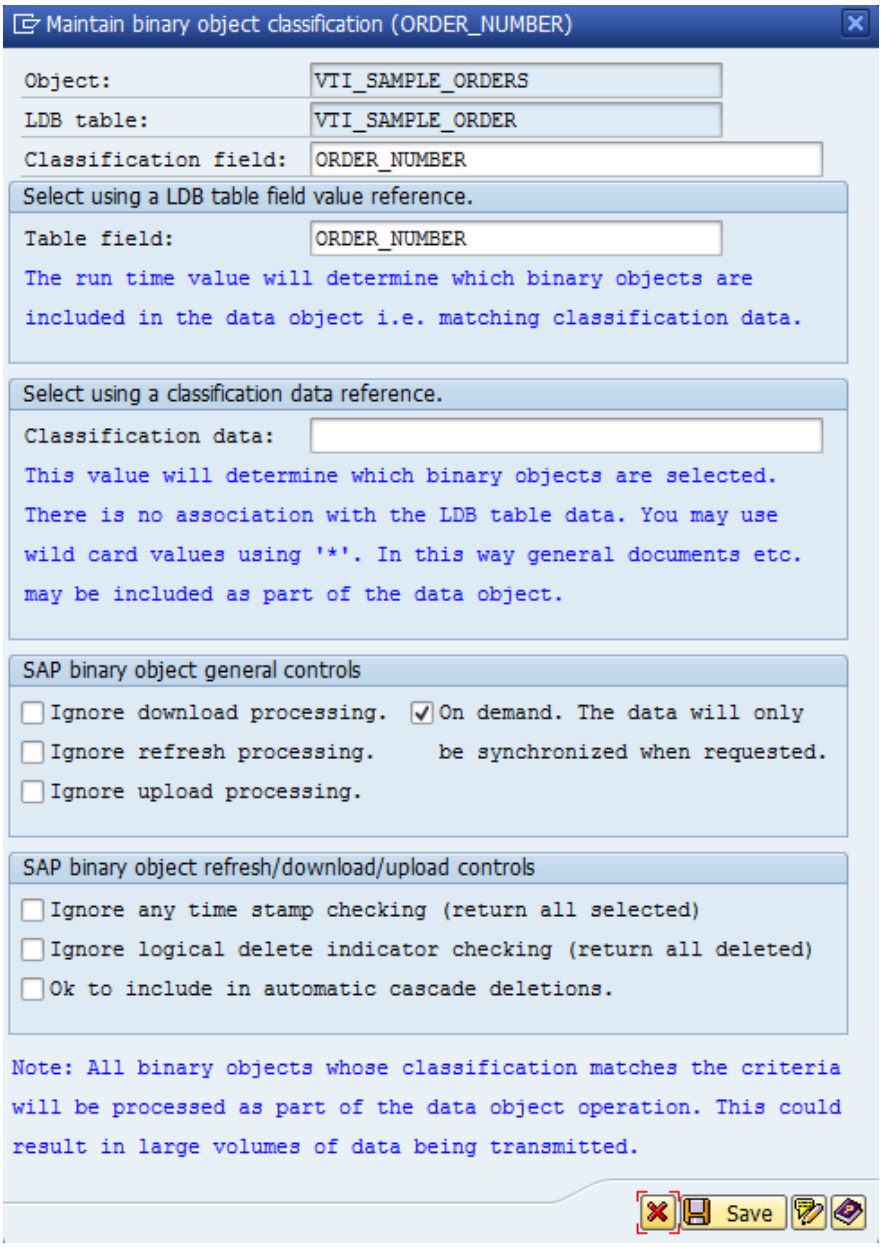
SAP Configuration

You configure the on-demand functionality in SAP, either against the data object definition, when you import the binary objects, or specifically against a binary object definition itself.

Data Object Configuration

In the data object workbench, you specify that all related binary objects are on demand using the binary object relationship pop-up (binary object general controls, **On demand** checkbox).

> ***Note:*** If **On demand** is turned off, any binary objects explicitly flagged as **On demand** are treated that way.

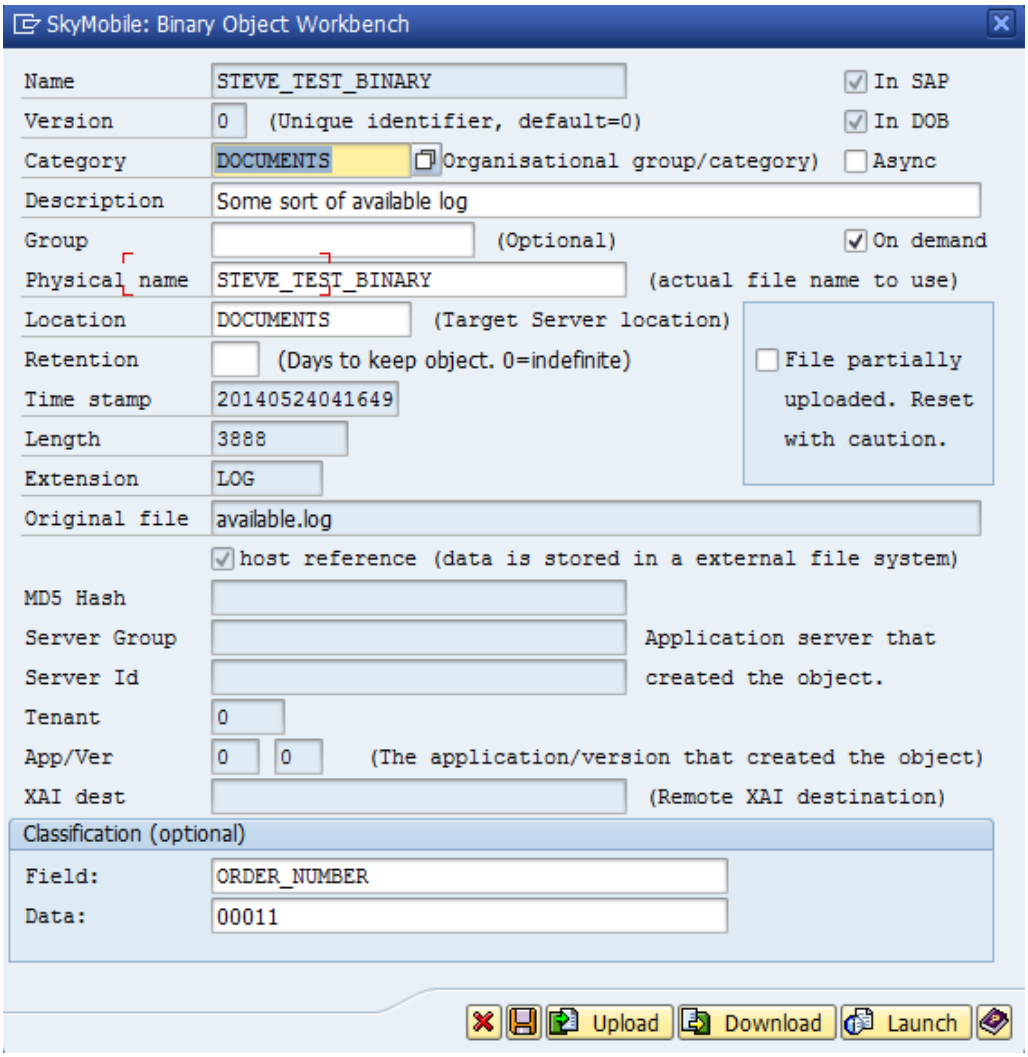


Binary Object Configuration

If you want to flag a specific binary object as **On demand**, you may either flag it at import time using the function APIs or manually through the binary object workbench. The following function APIs have an i_on_demand parameter that you must set to *true*:

- /sky/vti_import_binary_data
- /sky/vti_import_binary_file
- /sky/vti_update_binary_data

In the binary object workbench, you may maintain the **On demand** checkbox using the binary object header attributes pop-up (double-click an object).



SkySync API Processing

Normally, when accessing binary files through SkySync, the *binaryFileSelectByName()* method is used. This retrieves information about a binary file, including its pathname (that is returned to the SkySync observer as a parameter called *FILEPATH*). From here, the application can read the data of the binary file by interrogating the specified file path.

On-demand, binaries work slightly differently. If the binary file is flagged as **On demand**, only metadata about the file is initially downloaded from SAP (that is, its size, type, name, and so on). The actual file data itself is not downloaded until it is required by the application.

When using on-demand binary files, rather than using the *binaryFileSelectByName()* method, the application programmer should use the *binaryFileFetchByName()* method. This works in exactly the same way as *binaryFileSelectByName()*, accepting the same inputs and returning the same outputs. The only difference in behavior is that if the binary file is flagged as on-demand, and the associated data is not present locally, the binary file is immediately fetched from the SAP host before a result is returned to the caller.

Once fetched from the SAP host, on-demand binary data is held locally for a configurable time interval before it is automatically purged. This is controlled by a configuration item in the SkySync profile called *SERVER.BINARYFILES.ONDEMANDRETENTIONINTERVAL* that is measured in seconds. The default value for this item is *14400* seconds that equates to four hours.

The “countdown” to purging the data for each on-demand binary file is reset each time the binary file is accessed by the application, so frequently accessed data remains present locally until it is not used for the period specified in the SkySync configuration profile.

4.9 Profiling

Profiling is all about controlling the Secure Container configuration, applications and data that a user can access. In SkyMobile, there are three primary profiling functions:

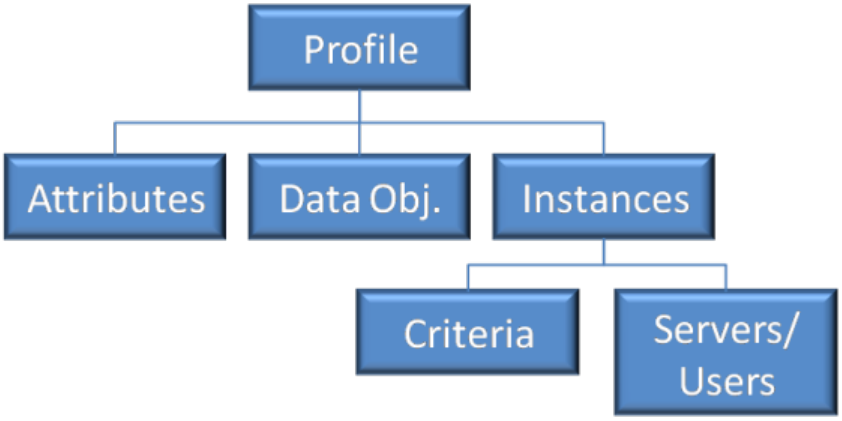
- The Server Profiler is used to automatically provision Secure Containers onto a device. The idea is that you download the SkyMobile Secure Container from a public or private application store, and then go through an automated provisioning process to either connect to the Sky Cloud or an on-premise SkyMobile MEAP server. All the required configurations are implemented automatically for a user without any manual intervention required.
- The User Profiler supports role-based provisioning, and defines what applications and data objects a user can access. Typically, a user is properly authenticated and associated directly with a device using SkyMobile Identity Management. Once the user is known, user profiling is used to control which applications and data get synchronized with the device.
- The Data Profiler controls the data objects that you may synchronize with a device/user. This consists of rules that dictate how the data is selected and maintained, as well as who should be notified/synchronized when changes occur.

4.9.1 Key Topics

Server Profiling
User Profiling
Data Profiling

4.9.2 Data Profiling

A data profile is used with data objects to effectively control the distribution of data to Secure Containers. A data profile is basically a set of rules that are applied to data elements and a list of containers (servers) that are affected. A single profile definition may have many different instances of criteria and associated servers and users, so that you may configure multiple concise sets of rules. You may invoke a profile from any data object (DOB) download, refresh and upload operation as well as from ABAP extract programs through SAP Add-in ABAP SDK. Profiles are associated with applications through data object or complimentary object references. The primary benefit of data profiles is to help limit or eliminate the need to write code to manage the distribution of data. You maintain profile definitions using a workbench.



A data profile consists of:

- Attributes (fields) that conditions (rules) are configured against
- A list of associated data objects to consider for data synchronization
- One or more instance definitions, each having a list of criteria (rules) that qualifies the instance and a list of one or more servers/users associated with the instance. An instance is effectively a unique set of rules and associated actions, for example, if the sales area is 'VICTORIA' then push the updates out to the 'VIC' server group, if a data refresh request for work orders is received from server group 'CREW5', then only select work orders where the high level functional location is 'REGION5'.

You can configure either specific Server group/ID criteria or Work group/User id level associations. For Work Group / User Id to work effectively, you must pre-populate these details into the session header using a procedure or Java/ABAP custom program exit, for example, through a logon screen. These details are then linked to the Client/Server connection details that are subsequently passed onto to SAP through the heartbeat request. A cross reference of current connections is automatically maintained in SAP that the profiler uses to derive what Application Servers are associated with a Work Group / User ID and vice versa.

> ***Note:*** The Container Server Group/Id criteria takes precedence over Work Group/User ID criteria. Kony recommends that you use user level profiling with identity management for maximum control ad ease of configuration.

4.9.2.1 Overview of the Data Profiler Process

Data profiles are optionally associated with Data Object (DOB) download, refresh an upload configuration. This means that every time the SAP host processes a DOB synchronisation request, it invokes the associated profiles and apply its rules on data selection and notification. It is important to note that data profiling is bi-directional; filter criteria is applied for data downloads and refreshes to a remote server; and associated servers are notified of any relevant data changes on a upload from a specific server.

You may also invoke the profiler independently from a ABAP program using the ABAP SDK; when data is maintained on the SAP host, invoke the SkyMobile profiler with a set of key attributes to notify (push) subscribing servers/users that something changed and they should issue a data object refresh.

Action	Description
DOB Download Refresh	Using the calling Application Server group/id or User/Work group, the profiler looks for a matching instance in the profile. Once found, it generates dynamic SQL where conditions that are used in the data selection criteria. In this way, selection criteria may be varied according to the caller, for example, sales order changes for a particular sales region.
DOB Upload	The purpose of the upload profile is to notify other servers/users that data is changed ; push out changes to other related servers. The profiler does this by scanning the uploaded data set and checking for instances that qualify; matches the instance selection criteria. DOB refresh heartbeat commands are then generated for all the server /user list combinations of the selected instances. This notifies these servers to issue DOB refresh calls to retrieve changes; you can effectively synchronise any data changes that affect a group of remote Application Servers in a timely fashion without complex code or constant polling.

Action	Description
ABAP Extract	Profile APIs are provided in the ABAP SDK to enable ABAP programs to invoke the profiler. This is designed primarily for data extract programs/functions that retrieve/update host data and then need to "push" this out to remote Application Servers/Users. In this case, the ABAP program provides a list of attributes and invokes the profiler specifying the profile to use. The profiler then selects instances that match the criteria and generates DOB refresh heartbeat commands for all the server/user list combinations of the selected instances.

4.9.2.2 The Data Profile Workbench

You invoke the Data Profile Workbench through the ? Unknown Attachment icon on the application toolbar. A selection screen first appears to allow you to filter the hierarchy list.

? Unknown Attachment

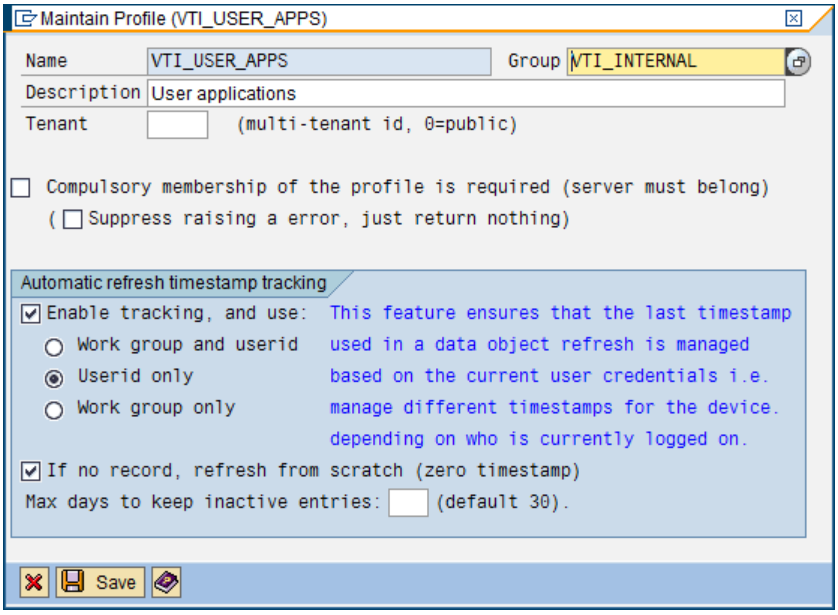
The data profile definition appears as the second top level node, for example, VTI_SAMPLE_ORDERS; the next level are the attributes, data objects and instances. You maintain the definition by either double-clicking a level or positioning your cursor on the level and clicking the application toolbar. Various hot links and short cut icons also appear that enable you to quickly branch to relevant areas, for example, DOB workbench.

A profile is made up of:

- The definition (header)
  - Attributes
  - Data Object references
  - Instances
    - Selection criteria
    - Server level selection
    - User level selection

Creating and Maintaining Profile Definitions

Position your cursor on the profiles node and click the create icon on the application toolbar. The following pop up appears.



Field	Description
Name	Each profile has a unique name.
Group	Assign the profile to a logical SkyMobile group.
Description	Descriptive text.
Tenant	A unique tenant assigned to the profile (Multi-tenant support)
Compulsory membership	If a profile is flagged as mandatory, a match must occur or else an error is raised; you must select an instance. There is an option to suppress the error; ignore the request and return nothing back to the Application Server.

Field	Description
Automatic refresh timestamp tracking	This option is used in situations where multiple users use the same Secure Container (device); the last data refresh operations (timestamps) are kept and automatically restored when a user switch occurs. This means that the data synchronization points are properly retained for each user. By default, a unique entry is kept for the server group/id and work group/user combination, but there are options to ignore the work group or user ID levels if they change frequently of are not applicable. You may also configure that the timestamp starts from scratch (zero) for new user/work group combinations.

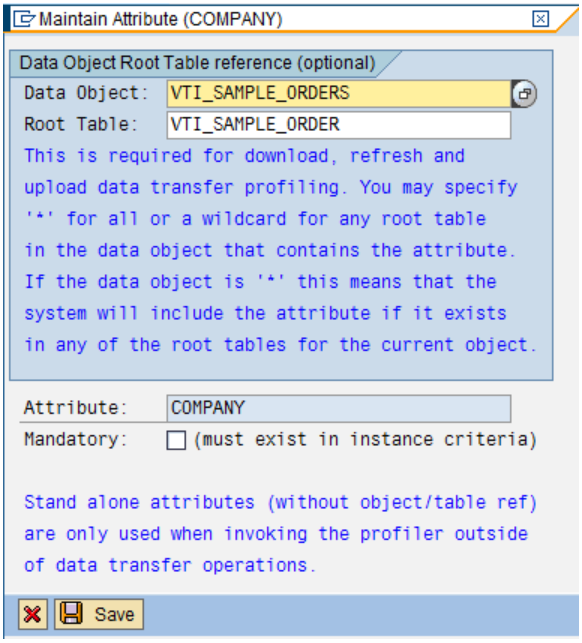
Once a profile definition is created, you may now specify the associated attributes, data objects and instances. You may also copy and rename at the profile level.

Data Profile Attributes

A profile attribute is essentially a field or variable to apply a logical condition against. All attributes are declared at the profile level. You may associate an attribute with a data object LDB fields or declared stand alone. For data object download/refresh/upload operations, the attributes must refer to a data object table (LDB) in question so that you can work out relevant column/type information. For ABAP SDK usage (extract program), you may use attributes without a LDB reference. If the attribute refers directly to a data object table, you must configure the profile data object first.

> ***Note:*** You can apply special generic references at the data object and LDB root table levels that are only evaluated at run time, for example, if a you place an asterisk in the root table, then all root tables containing the attribute are considered; if you place a '*' in the data object, then any data object with root tables containing the object is considered. These options enable you to use generic profiles across data objects; the attribute is not tied to a specific data object root table combination.

Position your cursor on the "Attributes" node of the profile definition and click the create icon on the application toolbar. The following pop up appears.



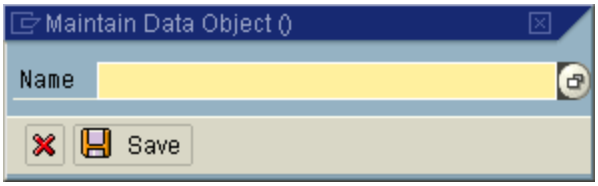
Field	Description
Data Object Root Table reference	This indicates that the attribute directly refers to a data object LDB table field. This is mandatory if the attribute is used in instance selection criteria for data object download/refresh/upload operations. This is not necessary if the attribute is just used for instance selection criteria for ABAP extract programs using the SkyMobile SDK profile commands. If you configure a data object of '*' then the current data object is processed. If you configure a root table of '*', then all root tables of the data object that contain the attribute (field) are considered in the profile processing.
Attribute name	A unique name within the profile definition.
Mandatory	If checked, you must specify the attribute value by the calling process for all instances that refer to the attribute in their selection criteria, for example, if sales area is flagged as mandatory and is used in instance selection criteria, you must specify it by any processes that invoke the profile; DOB download/refresh/upload operations, and ABAP extract programs.

Data Profile Data Objects

This is a list of data objects associated with the profile, for example, if an instance is selected for a DOB upload then refresh commands are issued for all the data objects in the list. You must declare the data objects before you can associate any directly associated attributes with them.

> ***Important:*** If the profile is designed to be generic, that is, you can use across multiple data objects, then define your attributes with a asterisk in the data object field and then there is no need to enter specific data objects here.

Position your cursor on the "Data Objects" node of the profile definition and click the create icon on the application toolbar. The following pop up appears:

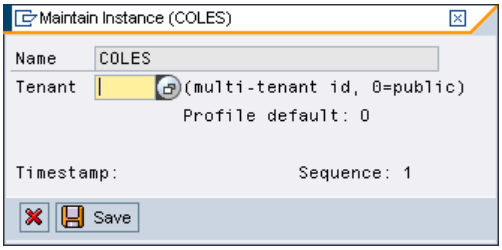


Field	Description
Name	The name of an existing data object definition.

Data Profile Instances

Instances are the heart of the profile engine. Each instance of a profile consists of its selection criteria and the target servers/users. Instances enable you to configure multiple different sets of rules against a single profile definition. When you invoke a profile, all the instances are scanned for matching criteria using either the attributes or server/user list, depending on the type of operation performed. For example, for a DOB download or refresh, the instance is selected using the server/user list and the associated attributes are used to generate SQL where statements. For a DOB upload or ABAP extract, the attribute list is used to derive a list of servers/users to issue DOB refresh commands to.

Position tour cursor on the "Instances" node of the profile definition and click the create icon on the application toolbar. The following pop up appears:



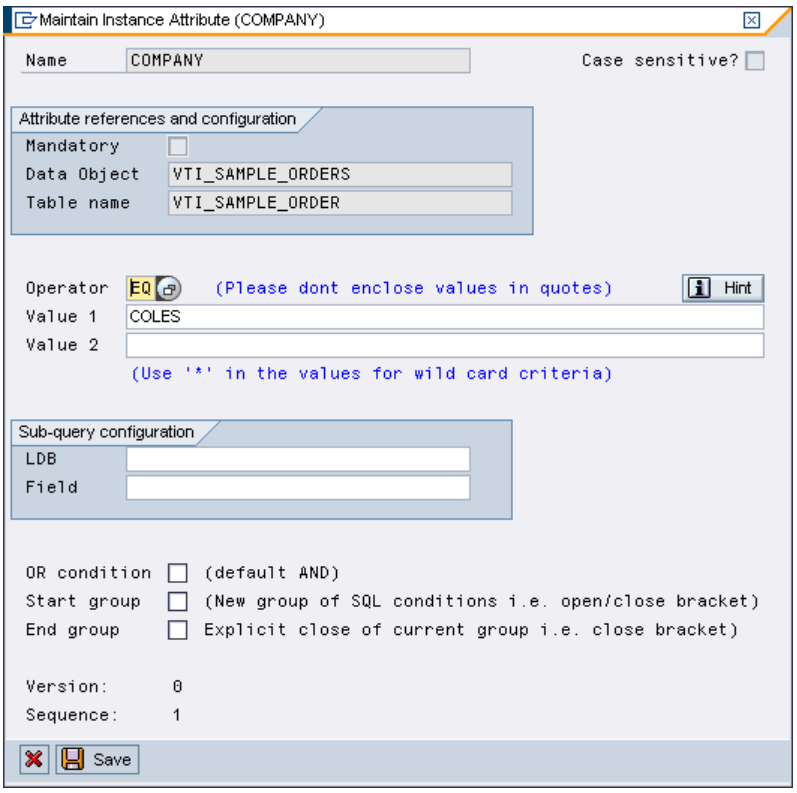
Field	Description
Name	The unique name of the instance definition.
Tenant	A unique tenant assigned to the profile (Multi-tenant support)

Once you defined the instance node, you may now define its selection criteria and the list of associated Servers and Users.

Data Profile Instance Criteria

Each instance has selection criteria that is basically a list of attributes and logical expressions that are either used to select the instance or to generate dynamic SQL where conditions, depending on the type of operation performed. An instance attribute must refer to an existing "profile" attribute and consists of the attribute name, a logical operator, one or more values or a LDB sub-query reference. A drop-down of available operators is available. A LDB sub-query (QI/QN) returns a list of values to compare against from another LDB table.

Position tour cursor on the Instance "Selection Criteria" node of the instance definition and click the create icon on the application toolbar. The following pop up appears:



Field	Description
Name	Name of a profile attribute to use.

Field	Description
Operator	The logical operator to apply, for example, EQ, GT, IN, and CP. You may use a drop-down list of all available options.
Value 1/2	The value(s) to compare against. Most operators may only require one value, but some such as "between" require two (range).
Sub-query configuration	For QI and QN operators, specify the local database (LDB) and one of its fields that you use to retrieve a list of values. This is effectively an IN condition with the list of values populated dynamically from a LDB table.
OR conditions and expression grouping.	These options are used to define an OR condition and the start and end of a group of selection criteria. By default, each condition is a AND.

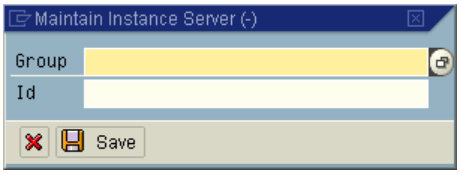
Note: Selection if the attribute is flagged as mandatory, you must specify it by the calling process, for example, DOB operation or ABAP extract program using the SDK.

Important: If you use the attribute in data object download/refresh/upload operations, it must refer to the data object and refer to one of its "root" tables. Use of attributes with no reference or attributes for another data object causes the instance to ignore.

Data Profile Instance Servers

The instance target server list may consist of specific or generic Secure Container server group and id combinations that you use to either; issue DOB refresh commands to or select the instance depending on the type of operation performed. For example, if a DOB download/refresh is performed, then the instance with the most specific server group/id criteria is selected; For DOB upload and ABAP program extracts, the instance is selected using the attribute selection criteria and the server list is used to issue DOB refresh commands for all data objects associated with the profile.

Position tour cursor on the Instance "Server Level Selection" node of the instance definition and click the **Create** icon on the application toolbar. The following pop-up appears:



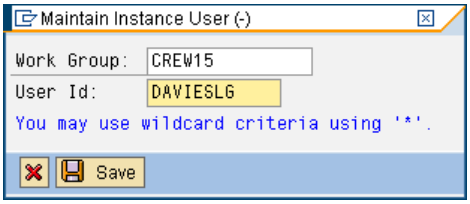
Field	Description
Server Group	A specific or generic container server group. A '*' is used as the wildcard operator.
Server Id	A specific or generic container server id. A '*' is used as the wildcard operator.

Data Profile Instance Users

The instance target user list may consist of specific or generic work group and user id combinations that are used to either; issue DOB refresh commands to or select the instance depending on the type of operation performed. For example, if you perform a DOB download/refresh, then the instance with the most specific work group/user id criteria is selected; For DOB upload and ABAP program extracts, the instance is selected using the attribute selection criteria and the user list is used to issue DOB refresh commands for all data objects associated with the profile.

> ***Note:*** To use work group / user id level selection lists, you must populate the work group and user id attributes in the session header (usually through an exit on a logon screen); this is then used to maintain an internal server/user cross reference that the profiler uses.

Position your cursor on the Instance "User level selection" node of the instance definition and click the create icon on the application toolbar. The following pop up appears:



Field	Description
Work Group	A specific or generic work group. A '*' is used as the wildcard operator.
User ID	A specific or generic user id. A '*' is used as the wildcard operator.

> ***Note:*** Server level selections take precedence over user level selections; if a server match is found, then all user level entries are ignored.

Data Profile Instance Utility

A utility (function module) is provided to dynamically maintain data profile instance definitions from a custom ABAP program. You may require to automate the adding of criteria, or cater for mass import/maintenance scenarios. You can create the function module /SKY/VTI_PROFILE_ INSTANCE_UTIL to create, change and delete instances. This is fully documented with examples in the ABAP programming guide.

4.9.3 Server Profiling

The Server Profiler allows you to automatically provision all the configuration, applications, and data profiles that a remote SkyMobile Secure Container requires. Its primary objective is to provide an install and run capability without having to perform any local configuration on the device. It also enables to maintain all definitions centrally and 'pushed out' to the remote devices without them having to come in to the office. The following sections document how the provisioning process works and how to use the Server Profile workbench to maintain definitions. For more information on implementing specific provisioning scenarios, refer to the System Administration section.

> ***Note:*** The SkyMobile check install process automatically creates a SKY_SAMPLE_DEVICE profile that you use to demonstrate how to implement a cross platform profile. This example profile utilizes the Sky sample logical platforms that are also automatically created for you. You can copy this profile and use it immediately to deploy applications to all the major platforms.

> ***Note:*** Typically, profiles incorporate an encryption object that you use to define what level and method of network security to apply. If you plan to provision secure containers that require encryption, refer to the network encryption section for background information on how SkyMobile encryption is implemented.

In many cases, you use the server profiler in conjunction with the user profiler to perform "role based provisioning"; users only get what applications and data they are authorized to. The best way to look at it is that server profiling is all about the secure container runtime environment on the device and user profiling is about roles, applications and data. The user profiler is optional; you can maintain applications and data profiling at the server profile level. If you do this way, the applications and data profiling applies to all users. If you want to get user specific, you must use the user profiler and define roles.

> ***Note:*** Sky recommends that you only define the application console at the server profile level and use the user profiler to manage all on demand applications and data profiles through roles.

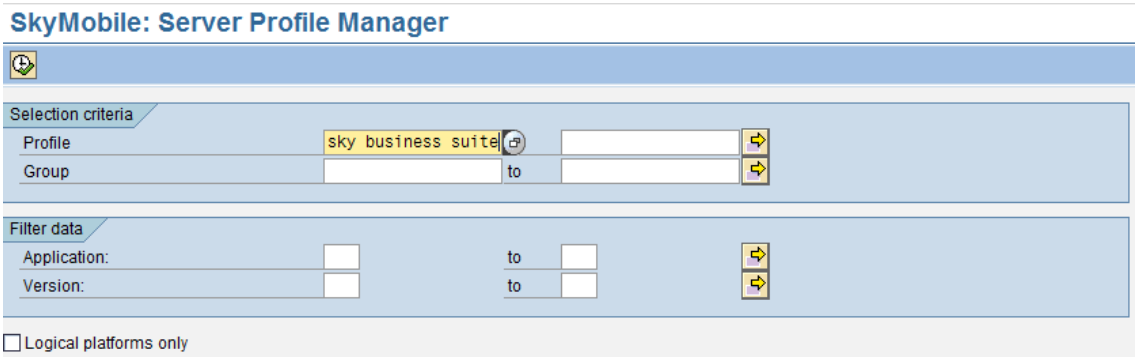
You require the following basic steps to initially provision a server:

1. Configure a profile definition using the Server Profiler Workbench.
2. Download and install the Secure Container.
3. Start the Secure Container and follow the provisioning process.

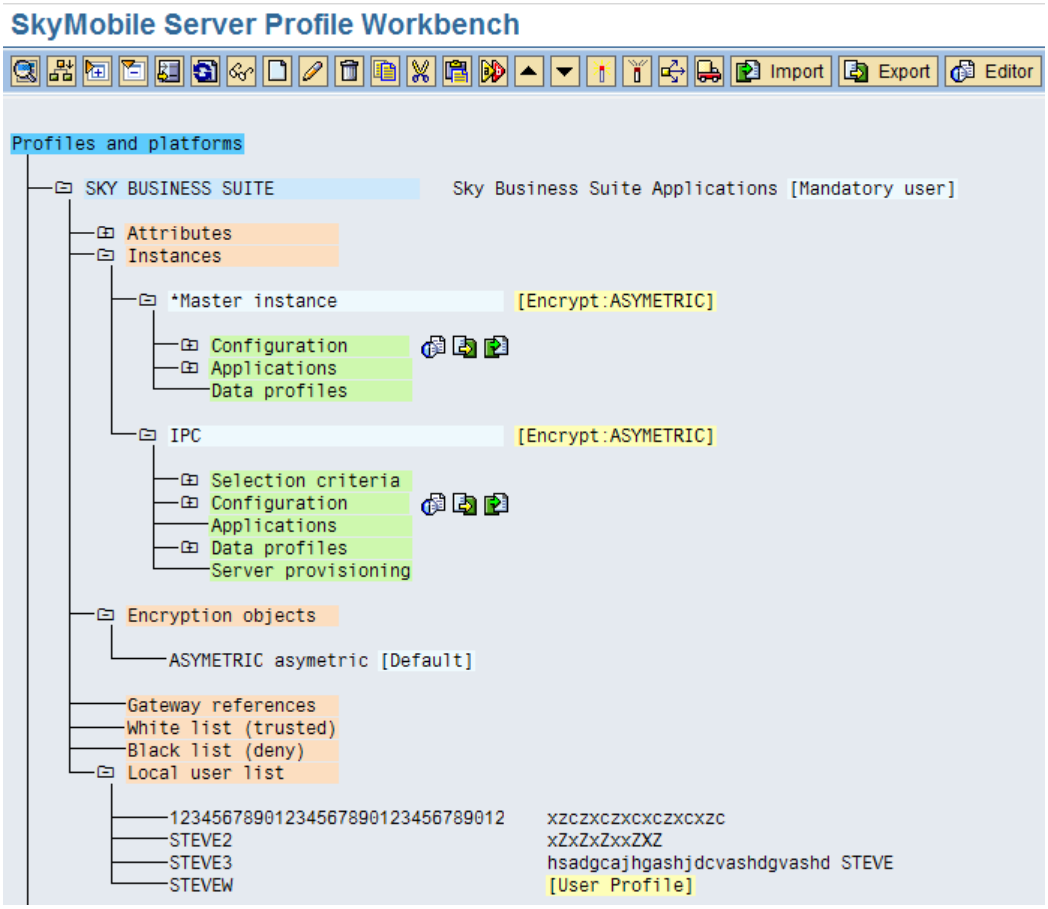
After the initial provision, you can automatically or manually "push out" subsequent changes to the profile to the remote servers.

4.9.3.1 The Server Profiler Workbench

You invoke the Server Profile Workbench through the icon on the application toolbar. A selection screen first appears to allow you to filter the hierarchy list.



Once the selection criteria is specified and executed, the following hierarchy screen appears:



The server profile definition appears as the second top level node, for example, SKY BUSINESS SUITE; the next level are the attributes and instances. You maintain the definition by either double-clicking a level or positioning your cursor on the level and clicking the application toolbar. Various hot links and short cut icons are appear that enable you to quickly branch to relevant areas, for example, export configuration.

A profile is made up of:

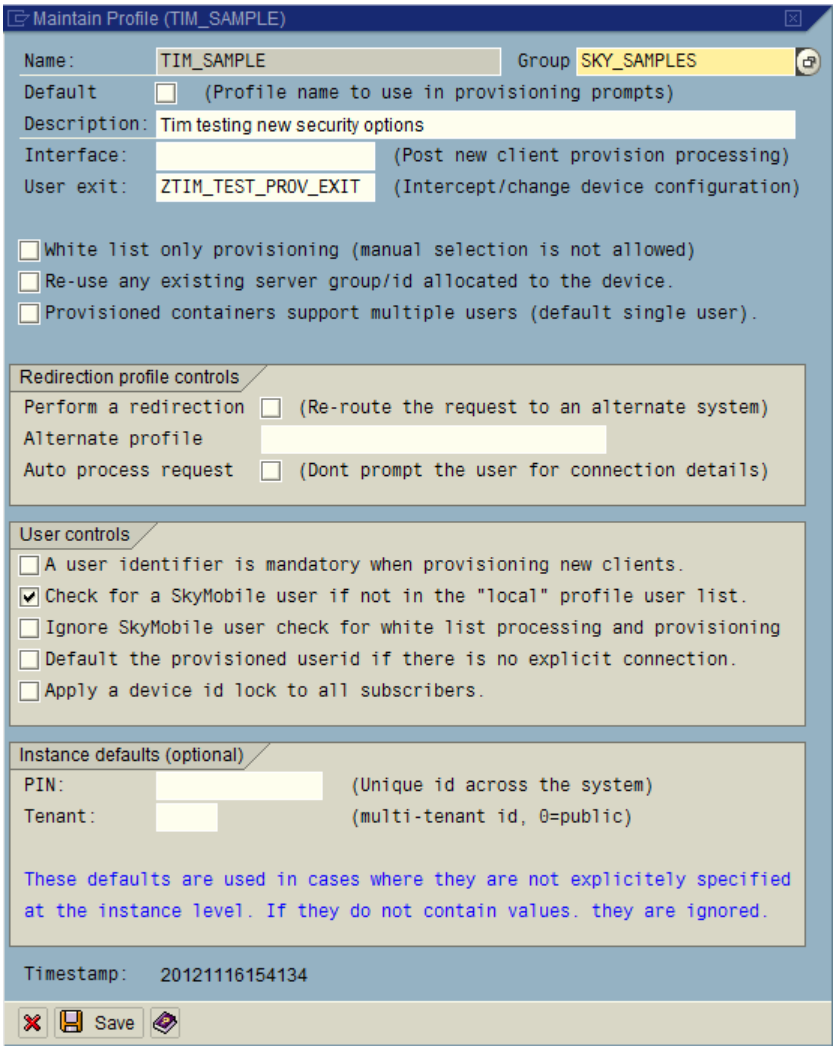
- The definition(header)
    - Attributes
    - Instances
        - Selection criteria
        - Configuration
        - Applications
        - Data Profiles
        - Server Provisioning (list and controls)
    - Encryption objects
    - Gateway references
    - White lists
    - Black lists
    - Server profile users

> ***Important:*** Some hierarchy entries are filtered to help reduce the number of entries that appear; improve performance. These are the white list and provisioned server nodes. The filter defaults to a max of 50 entries, and you may change this and other criteria at any time to extend or restrict the list by clicking the filter icon. To assist users, the total number of provisioned servers appear against the profile and instance nodes. If this count contains a "+", this indicates that the number of possible entries exceeded the filter; it is a restricted view.

4.9.3.2 The Server Profile Definition

A server profile is used to effectively control the provisioning of Secure Container run time environments. A single profile may have one or more "instances" of criteria so that you may configure multiple concise sets of rules. For example, a profile called "CRM" may have multiple instances to handle different configurations depending on the operating system of the device. By using server profiles, you can effectively manage your mobile application distribution and configuration from a central point. Each profile is given a unique name within the system. You must define a list of applicable attributes (fields) to use in instance selection criteria at the profile level. You may flag a profile as the 'default profile', meaning that its name is automatically filled in on any server provisioning prompts. If you flag the profile as a "redirection" profile, then it is used to nominate another SkyMobile MEAP Server that is to perform the provisioning. A gateway reference to the alternate system and a optional encryption object are the only attributes allowed. Instances are not applicable.

To create a new profile, position your cursor on the profiles node and click the create icon on the application toolbar. The following pop up appears:



Field	Description
Name	Each profile is given a unique name.
Group	Assign the profile to a logical SkyMobile group.
Default	If selected, the Profile name is used as the default (pre-populated) in the profile name prompt on the Secure Container provisioning screen.
Description	Descriptive text.

Field	Description
Interface	Whenever a new server is provisioned, you may trigger a ECS interface to perform additional functions (optional). The server provisioning details are to the interface through structure `/sky/yvti_svpsvr`.
User Exit	Whenever a server is provisioned, you may manipulate the configuration options before they are passed back to the device. This is useful where you need to generate individual device configurations(for example, encryption keys).
Redirection profile	A redirection profile has no instances or criteria of its own, but used purely to re-direct provisioning to another MEAP Server. This is mainly used to re-direct provisioning requests away from the Sky Cloud to a Customers on-premise MEAP Server. When provisioning is redirected, the process is starts from scratch; looks for white/black lists in the customers system.
White list only	If this is set, then white list matching is mandatory for this profile; a unique device ID match must take place before you can successfully provision the Secure Container.
Re-use group/ID	If the Secure Container is provisioned before, then its unique group/ID is re-used if it is ever re-provisioned.
User controls	This section controls how you should validate and manage a User ID, if specified as part of the provisioning process. You can make it mandatory to specify a valid user ID (checked against either the profile user list or the SkyMobile central user table). You may also default the provisioned user against all server requests against the host. This enables applications that have no logon to be associated with a user, kiosk style. You may also apply a device lock, meaning that the Secure Container do not start if you copy it to another device.See the extended user control setting documentation below for more information.
Instance defaults	You use these values if you do not specify defaults at the profile level. • A PIN is a secret pass code that is used to secure access to the profile. • The Tenant is a unique five digit number used to identify a unique tenant within a multi-tenant application. Click here for more information.

Once a profile definition is created, you may now specify the associated attributes, and instances. You may also copy and rename at the profile level.

User Control Settings

The following are the user control settings:

Server Profile User List

The profile user list gives the option to provision a server against a specific user ID. You may either use a "local" independent user list associated with the profile, or invoke the Sky user profiler functionality (see the "also check the Sky user profiler" section below). This option is only really valid if a single user associates and uses the server. If its a multi-user device, then you should use "user level provisioning" instead ; not at server provision time, but rather at user authentication (identity management) time. The "local" profile user list is available as a alternative if you choose not to use the Sky user "role based" profiling option, or to support other uses, for example, CRM business partner, sales division, and region to drive dynamic substitution criteria in data profile generation (substitution variables (&P1, &P2.).

> ***Note:*** You can use a combination of the "local" profile user list and user profiler to control which users may be provisioned against the server profile. To do this, specify Use the **Sky user profile as the master definition** on the local user definition and disable the **Check for a SkyMobile user** option at the profile level.

A user identifier is mandatory when provisioning new clients
This option forces a valid user ID to enter at provision time.

Check for a SkyMobile user if not in the "local" profile user list
If the userid value does not exist in the "local" profile user list, check against the SkyMobile user database (user profiling).

> ***Note:*** Sky recommends using the SkyMobile user database that is maintained using the user profile.

Ignore SkyMobile user checking for white list processing and provisioning

You may associate a user ID against a white list entry. This is mainly used to automatically default a user against a unique device; avoid having to log on. By default, this user must exist in either the "local" profile user list, or in the central SkyMobile user table. This option suppresses all checking and is usually used in cases where the user IDs are managed outside of SkyMobile control.

Default the provisioned user ID if there is no explicit connection

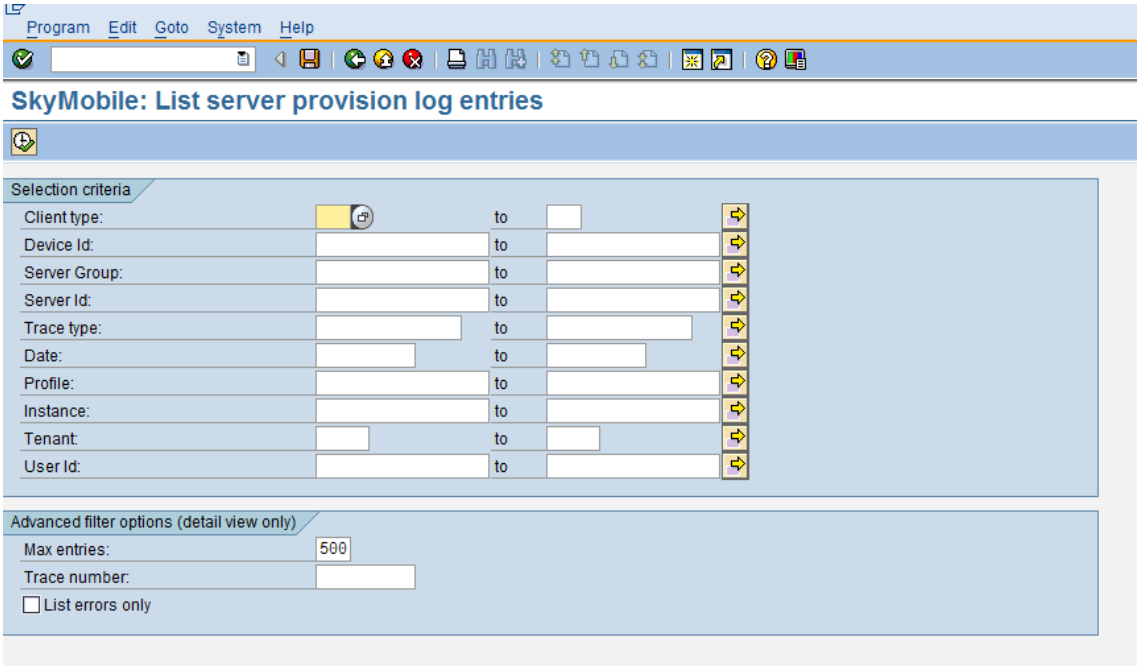
Once a Secure Container is provisioned against a specific user ID, automatically default it in the connection details of a heartbeat and in other calls to the MEAP Server if no user ID is specified. This option is useful in cases where there is no user authentication on the device and you need to derive the associated user ID, for example, kiosk mode.

Apply a device lock

This configuration ensures that you can start the Add-in Secure Container on a specific device; is valid only for a certain device ID. Once you configure this, you cannot copy and start the "secure container" on any other device.

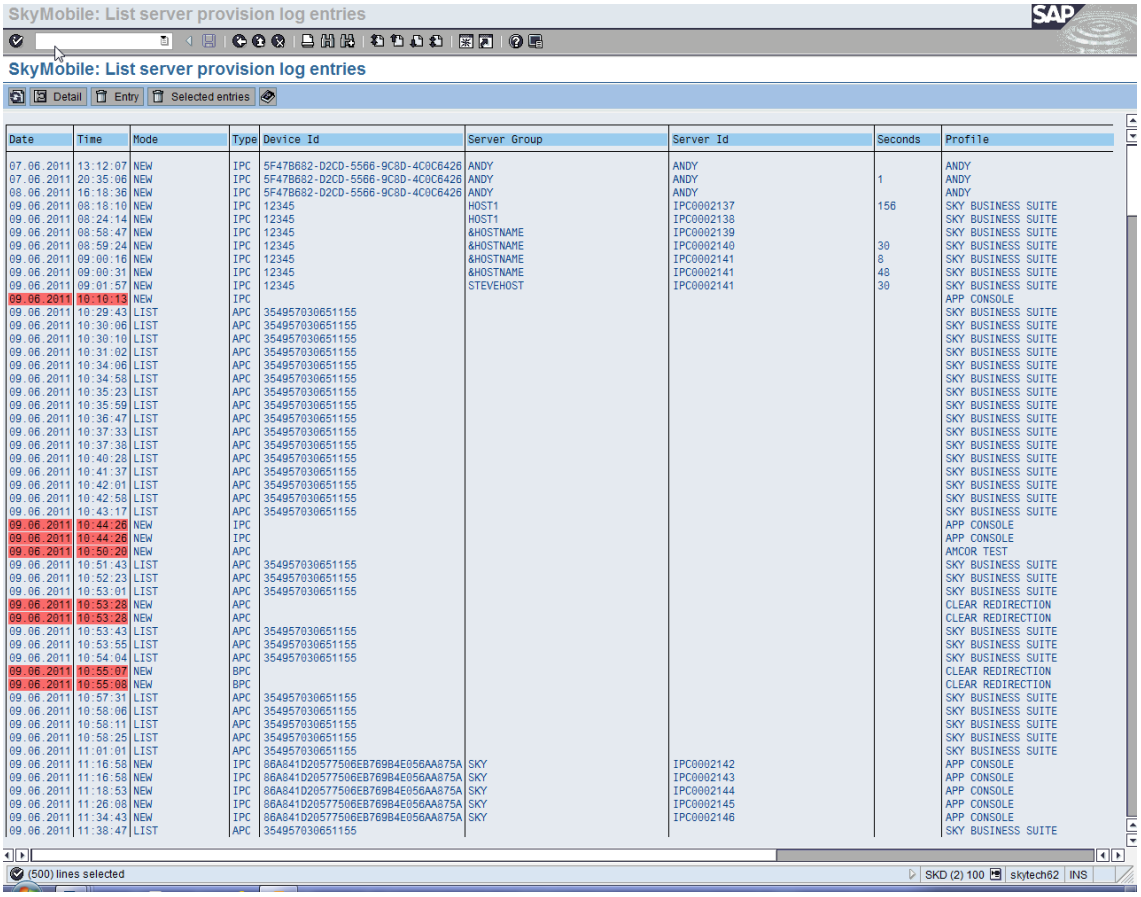
4.9.3.3 Server Profile Log Entries

All server provisioning requests get logged. You can access the log entries from the server profiler workbench by clicking the log button in the menu bar. A selection screen allowing you to filter the log entries by a range of criteria appears:

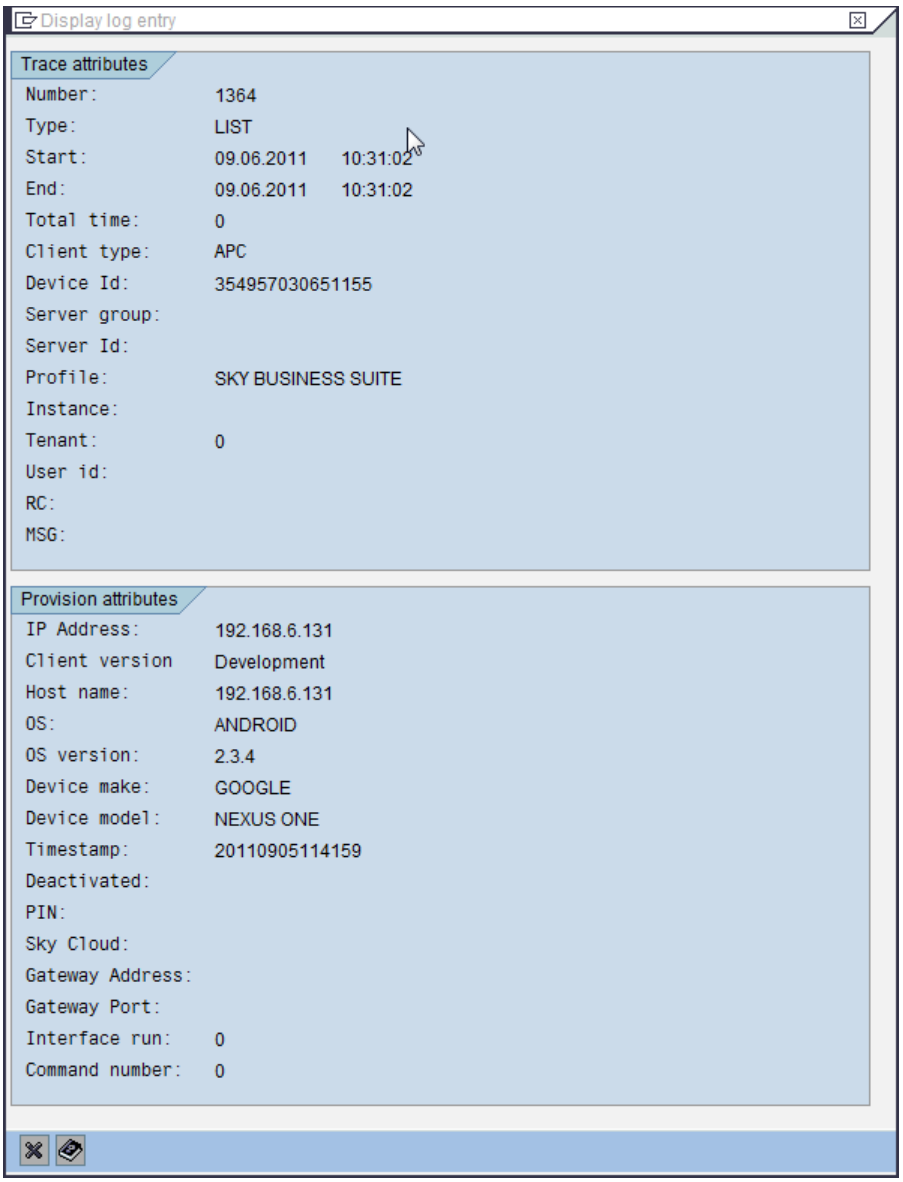


A summary screen of log entries matching the criteria appears. Double-clicking a log entry displays full details of the provisioning request.

> ***Note:*** Provisioning requests highlighted red indicate that an error (for example, invalid profile name provided, and device blacklisted) occurred during the provisioning process. Double-click the log entry to display the specific error message.

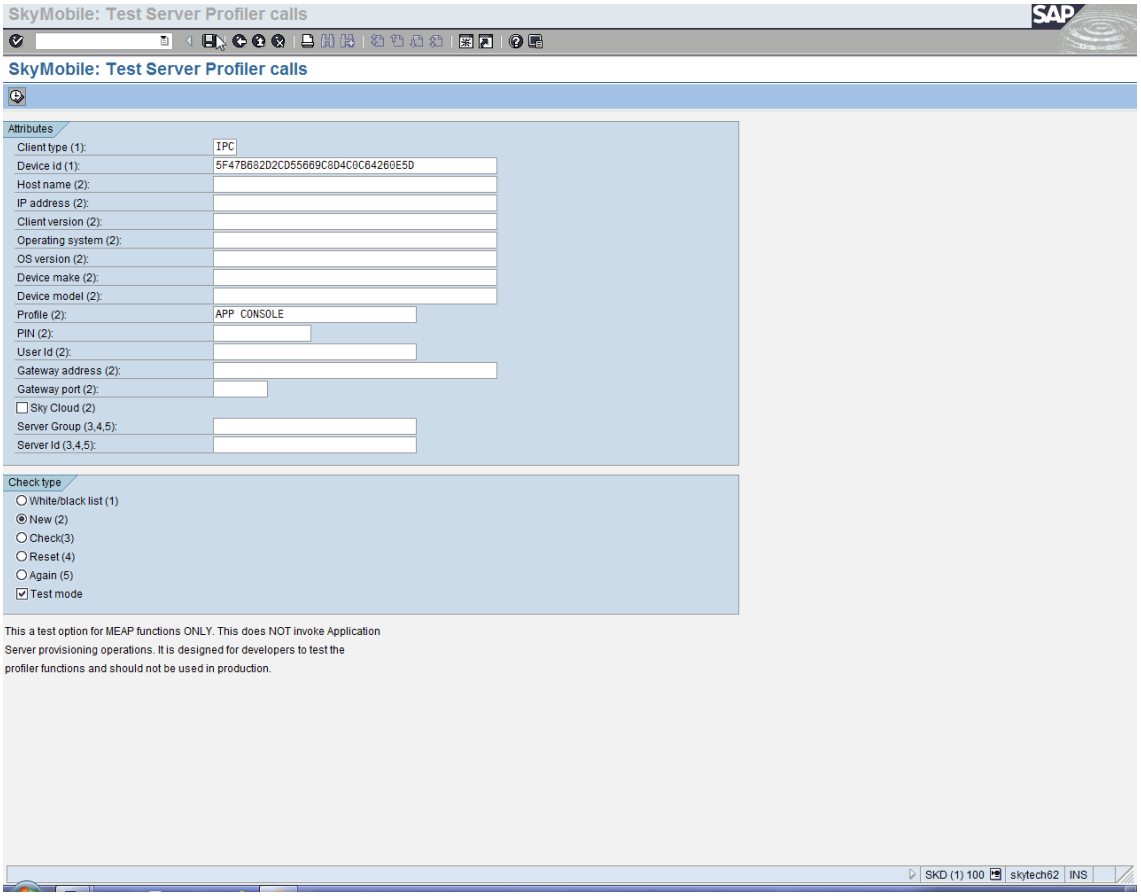


Date	Time	Mode	Type	Device Id	Server Group	Server Id	Seconds	Profile
07.06.2011	13:12:07	NEW	IPC	5F47B682-D2CD-5566-9C8D-4C0C6426	ANDY	ANDY		ANDY
07.06.2011	20:35:06	NEW	IPC	5F47B682-D2CD-5566-9C8D-4C0C6426	ANDY	ANDY	1	ANDY
08.06.2011	16:18:36	NEW	IPC	5F47B682-D2CD-5566-9C8D-4C0C6426	ANDY	ANDY		ANDY
09.06.2011	08:18:10	NEW	IPC	12345	HOST1	IPC0002137	156	SKY BUSINESS SUITE
09.06.2011	08:24:14	NEW	IPC	12345	HOST1	IPC0002138		SKY BUSINESS SUITE
09.06.2011	08:58:47	NEW	IPC	12345	&HOSTNAME	IPC0002139		SKY BUSINESS SUITE
09.06.2011	08:59:24	NEW	IPC	12345	&HOSTNAME	IPC0002140	30	SKY BUSINESS SUITE
09.06.2011	09:00:16	NEW	IPC	12345	&HOSTNAME	IPC0002141	8	SKY BUSINESS SUITE
09.06.2011	09:00:31	NEW	IPC	12345	&HOSTNAME	IPC0002141	48	SKY BUSINESS SUITE
09.06.2011	09:01:57	NEW	IPC	12345	STEVEHOST	IPC0002141	30	SKY BUSINESS SUITE
09.06.2011	10:10:13	NEW	IPC					APP CONSOLE
09.06.2011	10:29:43	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:30:06	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:30:10	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:31:02	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:34:06	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:34:58	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:35:23	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:35:59	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:36:47	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:37:33	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:37:38	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:40:28	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:41:37	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:42:01	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:42:58	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:43:17	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:44:26	NEW	IPC					APP CONSOLE
09.06.2011	10:44:26	NEW	IPC					APP CONSOLE
09.06.2011	10:50:20	NEW	APC					AMCOR TEST
09.06.2011	10:51:43	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:52:23	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:53:01	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:53:28	NEW	APC					CLEAR REDIRECTION
09.06.2011	10:53:28	NEW	APC					CLEAR REDIRECTION
09.06.2011	10:53:43	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:53:55	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:54:04	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:55:07	NEW	BPC					CLEAR REDIRECTION
09.06.2011	10:55:08	NEW	BPC					CLEAR REDIRECTION
09.06.2011	10:57:31	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:58:06	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:58:11	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	10:58:25	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	11:01:01	LIST	APC	354957030651155				SKY BUSINESS SUITE
09.06.2011	11:16:58	NEW	IPC	86A841D20577506EB769B4E056AA875A	SKY	IPC0002142		APP CONSOLE
09.06.2011	11:16:58	NEW	IPC	86A841D20577506EB769B4E056AA875A	SKY	IPC0002143		APP CONSOLE
09.06.2011	11:18:53	NEW	IPC	86A841D20577506EB769B4E056AA875A	SKY	IPC0002144		APP CONSOLE
09.06.2011	11:26:06	NEW	IPC	86A841D20577506EB769B4E056AA875A	SKY	IPC0002145		APP CONSOLE
09.06.2011	11:34:43	NEW	IPC	86A841D20577506EB769B4E056AA875A	SKY	IPC0002146		APP CONSOLE
09.06.2011	11:38:47	LIST	APC	354957030651155				SKY BUSINESS SUITE

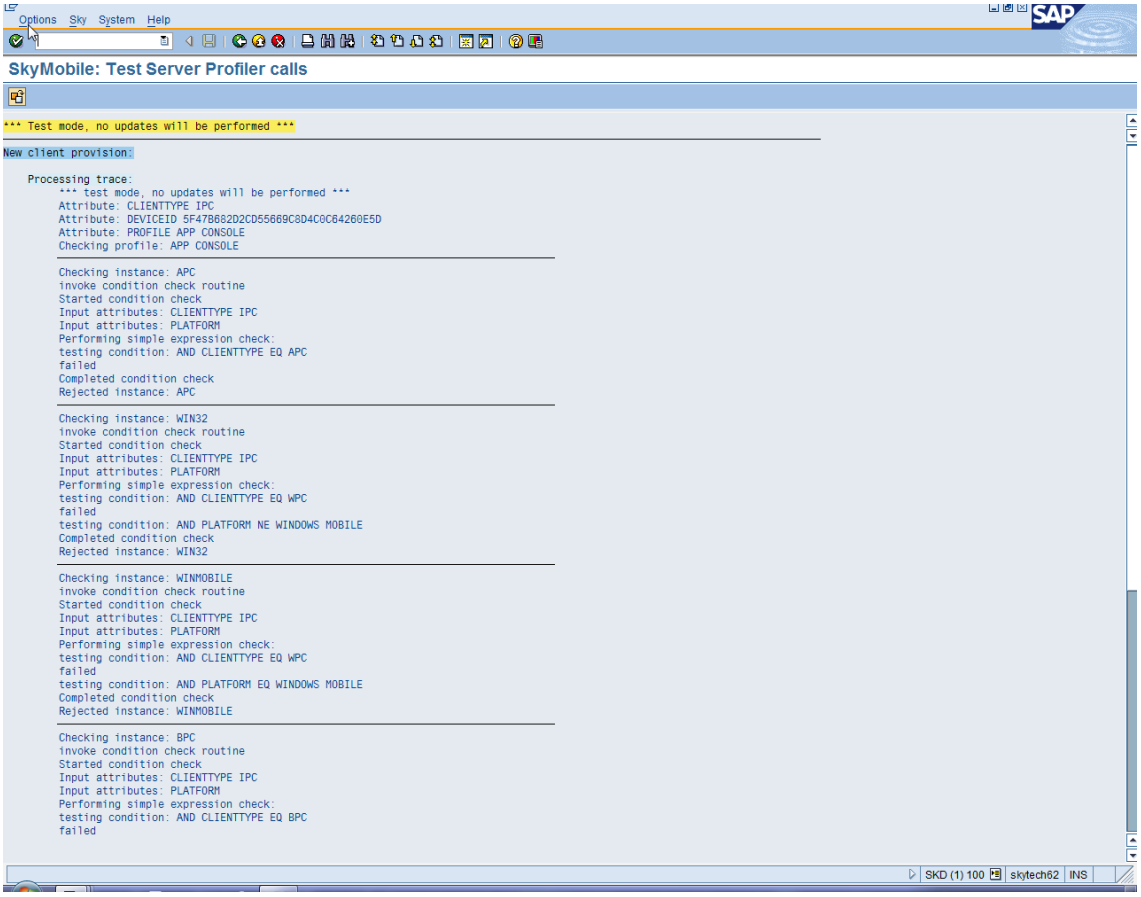


4.9.3.4 Server Profile Test Facility

A test facility is provided in the server profiler workbench. The test facility allows you to simulate a provisioning request by providing inputs for the provisioning technical attributes such as client type, operating system and version, and Device ID.



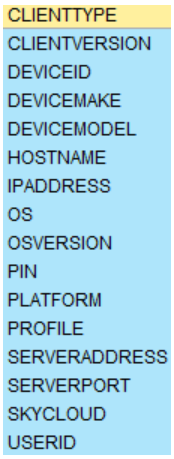
When you submit the test, it evaluates the technical attributes provided and indicate which if any profiles are matched and the resulting configuration that is provisioned.



> ***Note:*** By default the provisioning test facility submits a new provisioning request test. You can change the provisioning message type through the radio buttons at the bottom of the test screen.

4.9.3.5 Server Profile Attributes

A profile attribute is essentially a field or variable used in criteria to select a server profile instance. All attributes are declared at the profile level. Position your cursor on the "Attributes" node of the profile definition and click the create icon on the application toolbar. The following pop-up appears:



Field	Description
Attribute name	A technical attribute name that is available (sent up) by the Application Server, for example, OS, and DEVICEID. A drop-down lists what is available.
Mandatory	If selected, you must specify the attribute value in the instance selection criteria, or else it is ignored in the selection process.

You may select and use the following technical attributes for instance selection:

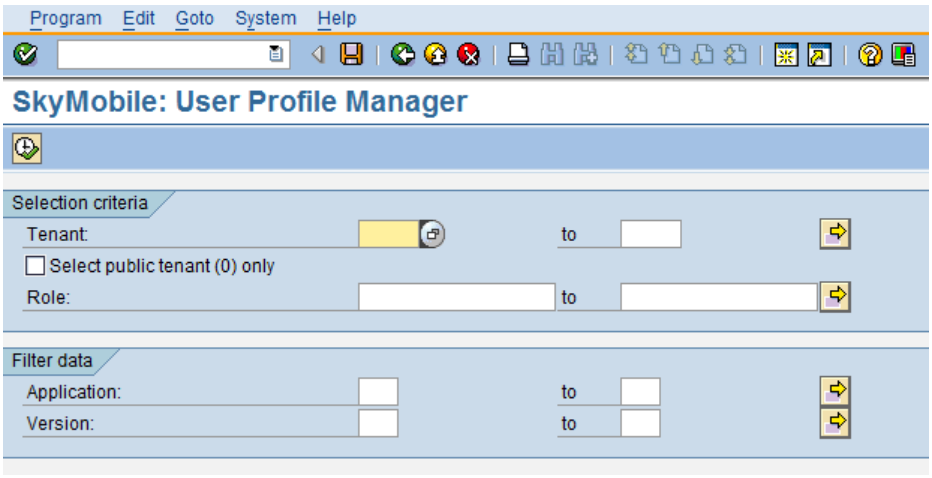
Attribute	Description
CLIENTTYPE	SkyMobile client type (for example, APC, BPC, IPC, WPC)
CLIENTVERSION	Version of the SkyMobile client (for example, v23.00.10)
DEVICEID	Unique Device Identifier

Attribute	Description
DEVICEMAKE	Make of device (for example, HTC, Apple)
DEVICEMODEL	Model of device (for example, iPhone)
HOSTNAME	Hostname of the device
IPADDRESS	IP address of the device
OS	Operating system
OSVERSION	Version of the operating system
PIN	PIN that the user provides during provisioning
PLATFORM	Logical platform derived from technical attributes
PROFILE	Profile name that the user provides during provisioning
USERID	User ID value that the user provides during provisioning

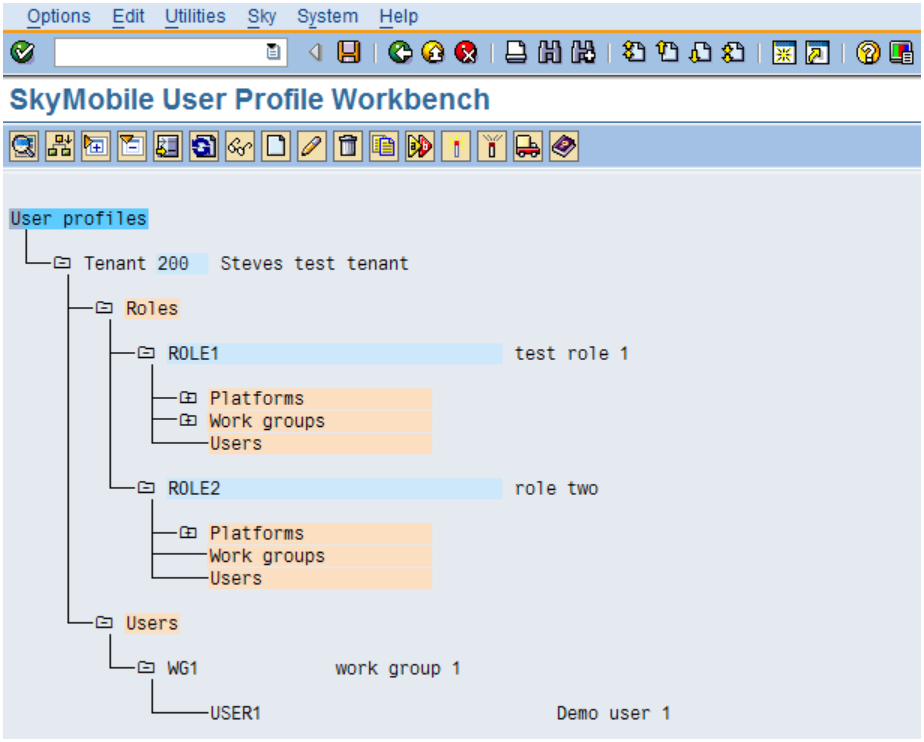
Note: Sky recommends selecting profile instances using a combination of client type and logical platform.

4.9.3.6 The User Profiler Workbench

You invoke the User Profile Workbench through the Utility or Sky profiling menu options; or you may use the YVT6 short cut transaction code. A selection screen first appears to allow you to filter the hierarchy list by tenant, and role.



Once the selection criteria is specified and executed, the following hierarchy screen appears:



The hierarchy display shows all the roles and users associated with a specific tenant. For most on-premise implementations, tenant 0 is used. You maintain entries by position your cursor on the relevant node and clicking the create icon on the application toolbar. To change, double-click the entry. You can only maintain role definitions if they are deactivated.

The maintenance hierarchy is:

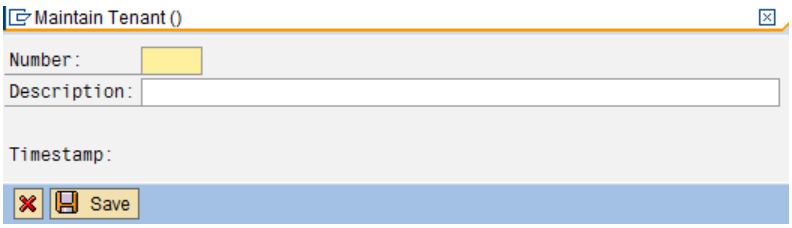
- Tenant
    - Role
        - Platform
            - Applications
            - Data profiles
        - Work groups
        - Users
    - Work groups
        - Users
    - Profile user groups

Once you created your user profile and user definitions, you may test the outcome; what is synchronized using the test tool.

User Profile Tenant

You may use tenants to segregate users, applications and data within a SAP add-in environment. In most cases with a on-premise Customer implementation, the default public tenant 0 is used. The user profiler may be used to maintain additional tenants as required and assign roles and users to them.

To create a new tenant, position your cursor on the user profiles node and click on the create icon on the application tool bar. The following pop-up screen appears:

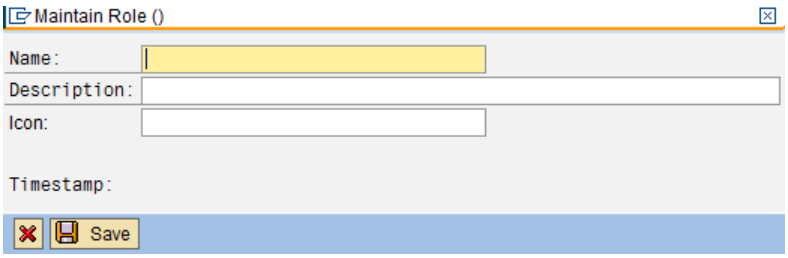


Field	Description
Number	A five digit unique tenant number
Description	Descriptive text of the tenant

User Profile Role

A role is unique within a tenant and contains the applications that users are assigned and the data profile definitions to generate. You may maintain only role definitions when they are deactivated, which is the default when they are created/ When a role is activated, a runtime object is generated that the application console uses (and possibly custom applications) to determine what applications are available for the current user. Once you create a role, you may then assign entire work groups, profile user groups or individual users to it. In this way, you may assign multiple roles to a user.

To create a new role, position your cursor on the tenant node and click the create icon on the application tool bar. The following pop-up screen appears:



Field	Description
Name	The unique name of the role within the tenant
Description	Descriptive text of the role
Icon	The binary object icon used to identify the role that appears on the application console screens. You require this only if the underlying applications are to appear under a role in the console, for example, a parent folder type concept. If you are going to configure the applications as "stand alone"; have their icons appear on the main console screen, then you don't require a role icon.

Once you created the role, you may now create its associated platforms, profile user groups work groups, profile user groups and/or users.

User Profile Role Platform

You can assign multiple logical platforms to a role. This is to support users who require applications across a range of device types, for example, smartphone and tablet. When you create a role, a default master platform is automatically created. The master platform is generic and its definitions are copied to any other platforms within the role. You maintain the logical platforms through the server profiler.

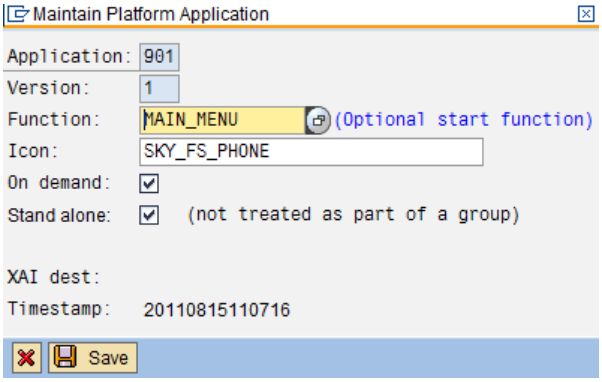
To create a new platform, position your cursor on the roles platforms node and click the **Create** icon on the application tool bar. The following pop-up screen appears:

Field	Description
Platform	The name of an existing logical platform definition. You can go to the logical platform utility using the button in the tool bar.

Once you create the role platform, you may now create its associated applications and/or data profile criteria.

4.9.3.7 User Profile Role Platform Application

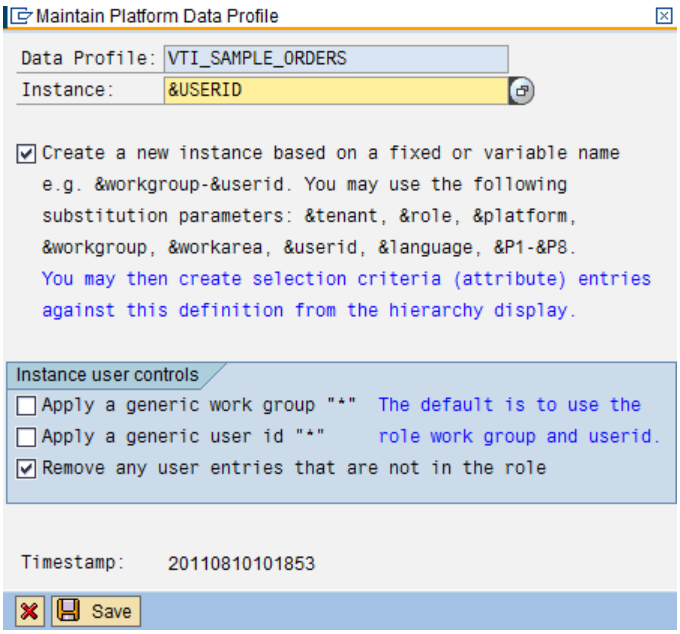
You can add one or more applications to a role platform definition. To create a new application reference, position your cursor on the platforms applications node and click the **Create** icon on the application tool bar. The following pop-up screen appears:



Field	Description
Application	The three digit application area identifier
Version	The three digit version identifier
Function	An option function within the version to start from. If this is left blank, the initial function is used.
Icon	A binary object graphic you use to identify the application in the application console. You must load this into the binary object repository
On demand	The application is not automatically loaded when the Secure Container initializes, but you may select from the application console
Stand alone	By default, all the applications appear under a "role" folder/icon. If you select this check box, the application always lists on the primary screen

4.9.3.8 User Profile Role Platform Data Profile

You can configure data profile instance definitions to automatically generate for all users of a role. These are essentially used to profile what data is distributed to which users. This feature can help avoid having to manually configure profile entries. Position your cursor on the Data Profiles node and select the create icon in the tool bar and the following pop-up appears:

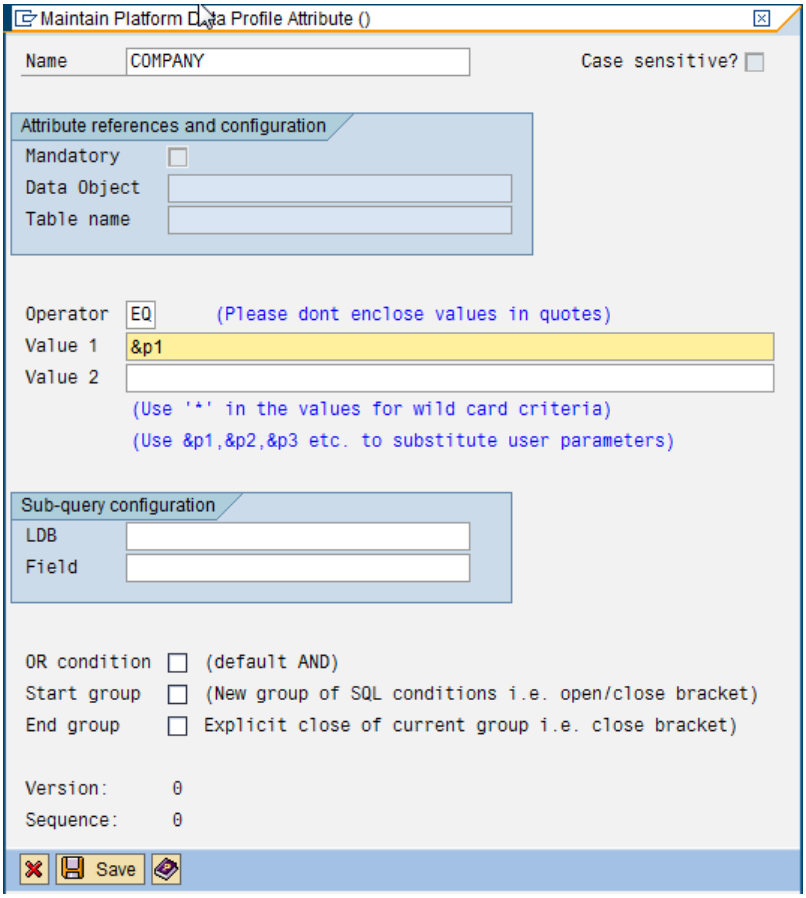


Field	Description
Data Profile	An existing Data Profile definition.
Instance	An existing Data Profile Instance definition, or a template name to use if the create new instance option is selected (see below).
Create new instance	This instructs the provisioning process to automatically create a new Data Profile instance using the template name specified in the Instance field. You may use substitution variables to generate a unique name at run time, for example, &clienttype-&deviceid. If you leave this blank, then a new instance name is generated.

Field	Description
Instance user controls	These are options that you use to manage how the data profile user entries are generated and whether to automatically housekeep them: • Apply a generic work group - generate a generic '*' work group instead of the current work group value assigned to the role user. • Apply a generic user id - generate a '*' userid instead of using the current role userid value • Remove any user entries that are not in the role - this option looks for user assignments against the data profile instance that are included in the role work groups and/or users. You use this to automatically housekeep data profiles that you use specifically with user profiling.

4.9.3.9 User Profile Role Platform Data Profile Attribute

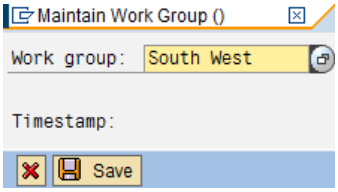
You can configure the selection criteria for a data profile instance. This screen operates in essentially the same way as the data profiler criteria screen except that it has access to the user profile parameters (p1 - p8).



User Profile Role Work Group

Add one or more work groups to the role. A work group can contain many users and is a convenient way to assign and maintain groups of users. A limitation of using work groups is that a user can only be assigned to a single work group. A much more flexible approach is to use profile user groups [profile user groups] which has no limitations i.e. enable many-to-many relationships.

To assign a work group to a row, position your cursor on the roles work groups node and click on the create icon on the application tool bar. The following pop-up screen appears:

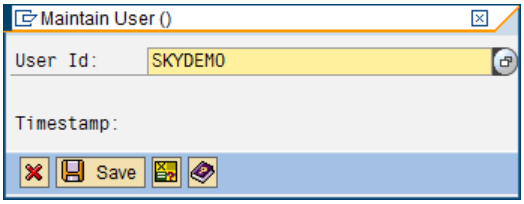


Field	Description
Work Group	The name of a existing work group definition in the user profiler.

User Profile Role User

Assigns a specific user to a role. Kony recommends that you consider using profile user groups for maximum flexibility and ease of maintaining user role assignments.

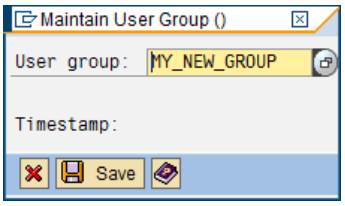
To assign an existing user to the role, position your cursor on the roles users node and click the **Create** icon on the application tool bar. The following pop-up screen appears:



Field	Description
User Id	The name of an existing userID defined in the user profiler.

User Profile Role User Group

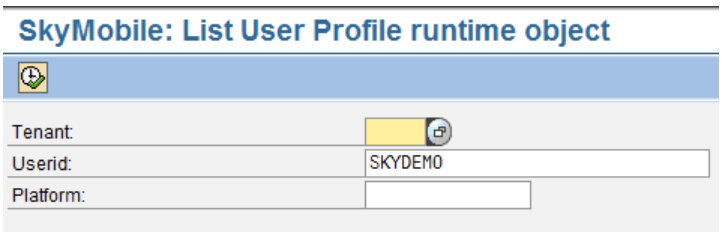
Apart from assigning individual users and work groups to a role, Sky also provides the ability to assign profile user groups to a role. This is by far the most flexible way of associating users with roles as it has no limitations; you may assign users to multiple user groups. To add a profile user group to a role, position your cursor on the "Profile user groups" node and click the **Create** icon on the application tool bar. The following pop-up appears:



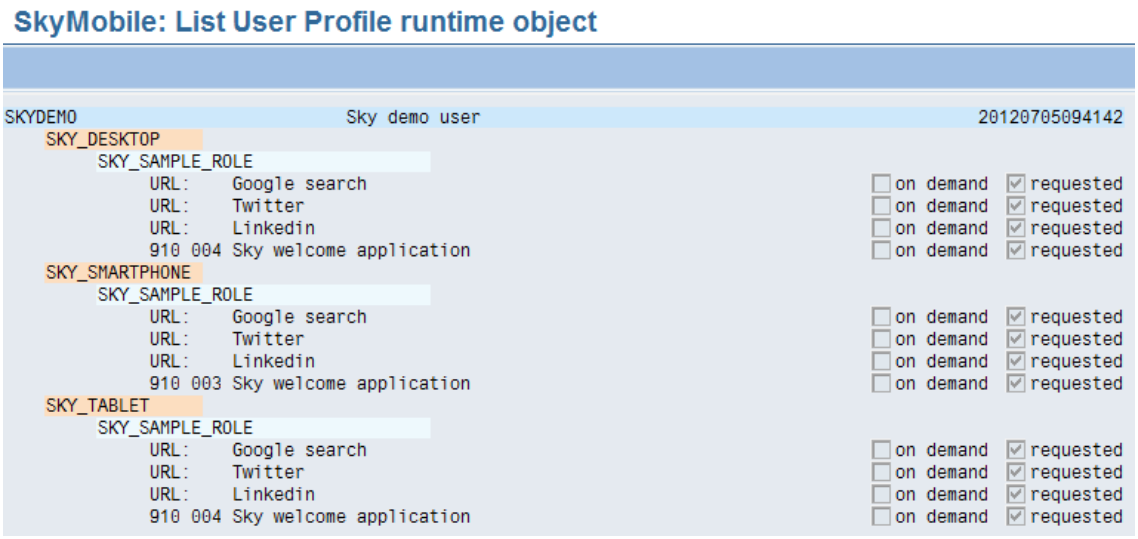
When the role is activated, then role run time objects are generated for all the associated users.

User Profile Role Runtime Objects

When you activate a role, or add a user to a assigned work group or profile user group, the user profiler automatically maintains a summary of the roles and associated applications for each of the assigned users. This is known as the role runtime object and is synchronised with the device via data object VTI_USER_APPS. To view the runtime object definition for a user, you may position your cursor on a user, and role and either select the "user runtime object" option off the edit menu, or click the icon in the application tool bar. The following selection screen appears:



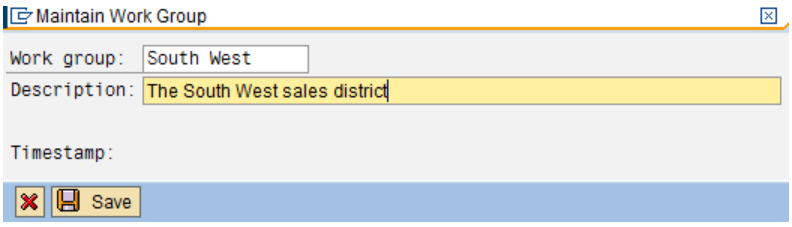
Select the user that you want to list the details for and click on the execute icon. A list of all the users platforms and associated applications appear. This is useful to help diagnose problems if a user is not receiving applications.



As per the above example the list is organised in a user, platform and role hierarchy. The requested flag indicates that the application or URL link is "live" and should be available.

User Profile Work Groups

You must assign all users in the user profiler to a unique work group. First, you must create the work group definition by positioning your cursor on the users node and click the **Create** icon on the application tool bar. The following pop-up screen appears:

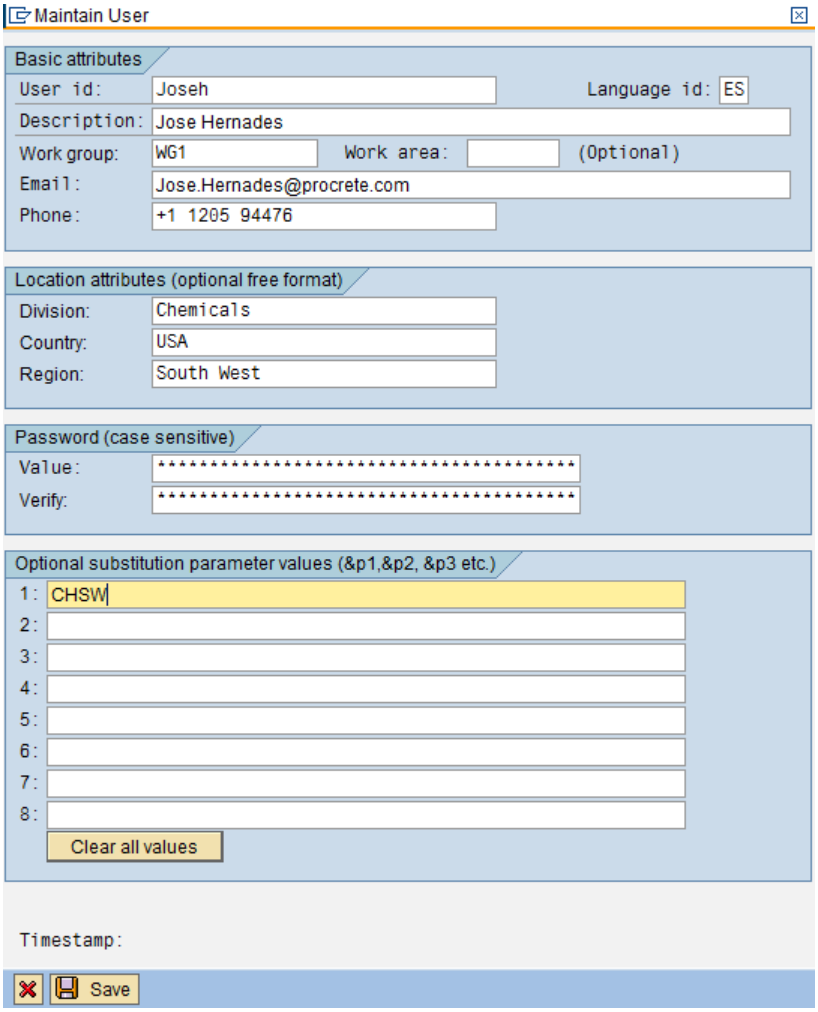


Field	Description
Work Group	A unique work group name within the tenant
Description	Descriptive text of the work groups function

Once you created the work group, you may now create its associated users.

User Profile Work Group User

Add a unique user to the tenant and assign it to the current work group. To create a new userid, position your cursor on the work group and click on the create icon on the application tool bar. The following pop-up screen appears:



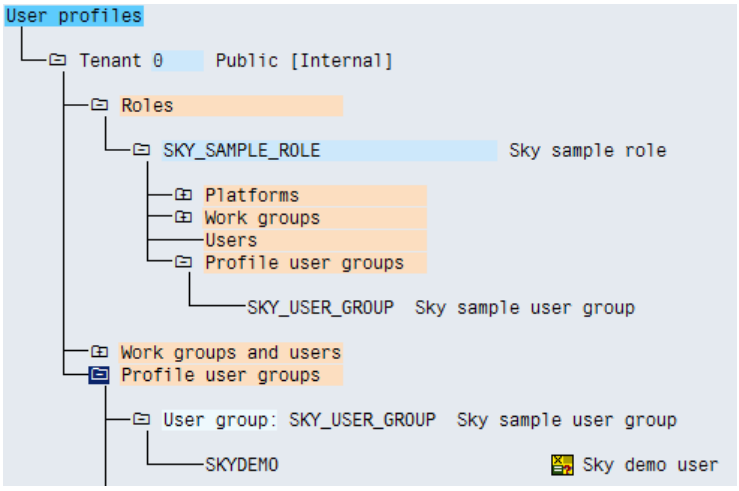
Field	Description
User Id	A unique user ID within the tenant
Language	The primary language code. If this is left blank, the default system language is used

Field	Description
Description	Descriptive text of the user
Work Group	The work group that you want to assign the user
Work Area	A optional work area value, for example, a factory or store code
Email	The users contact email
Phone	The users contact phone number
Location attributes	This is free format information that you may be use to describe the users organisation and location
Password	The users password. You really require this is only if you use SkyMobile user authentication
Optional parameter values	You may enter up to eight parameter values that you may use in SkyMobile applications and data selection criteria, for example, employee number, and organisational unit. These equate to the &P1-&P& variable substitution values in screen field defaults, and data profile criteria.

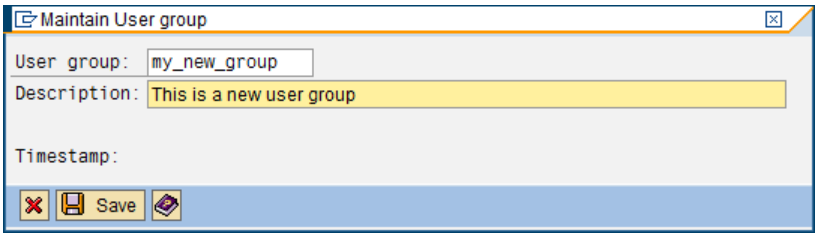
As well as manual entry, you may also import lists of users using the /SKY/VTI_IMPORT_PROVISION_LIST function.

User Profile User Groups

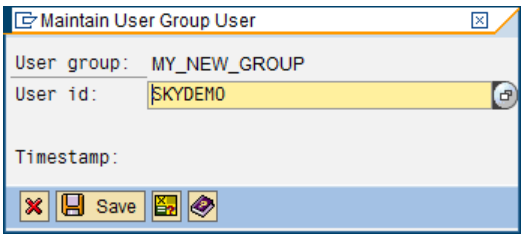
The limitation of a work group is that a user may only be assigned to a single work group. This is limiting when assigning users to roles where many to many relationships are required. To provide this flexibility, the user profiler has "user groups" which is simply a cross reference with no limitation. So instead of assigning individual users or work groups to a role, you may assign one or more users groups. You may assign a user to as many groups as you want. In the following screen shot, you assign the user SKYDEMO to user group SKY_USER_GROUP that is assigned to role SKY_SAMPLE_ROLE.



To create a new profile user group, position your cursor on the "Profile user groups" node and then click the create icon on the application tool bar. The following pop-up appears:



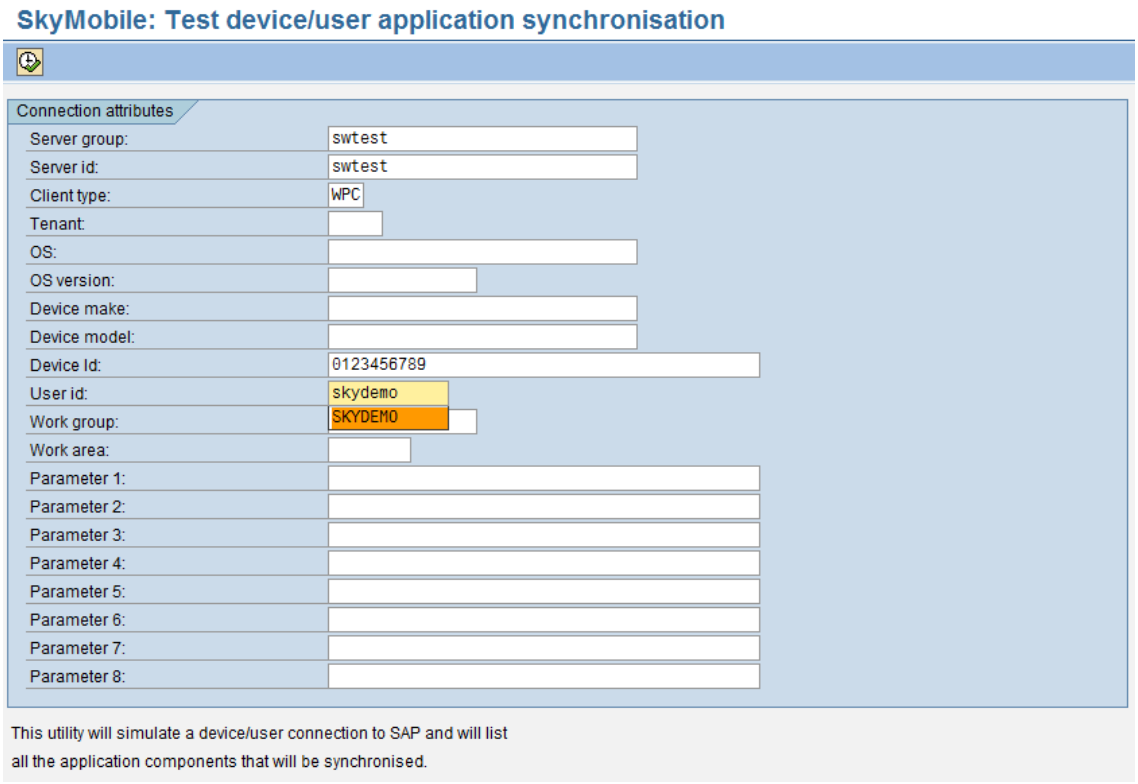
Once the new group is created, you may then assign users to it by positioning your cursor on the group and clicking the create icon on the application tool bar.



When you click **Save**, the user role run time objects are automatically maintained for all the roles that the user group is assigned to. The same happens when you then add the profile user group to a role and activate it.

User Profile Test Tool

The test tool allows you to simulate what applications, data objects, LDB, binary objects are synchronized with a device. This is useful to both simulate provisioning models and diagnose problems where the user is not receiving the correct applications. You launch the test tool through the **Test user/device application sync** option on the Utilities menu. The following selection screen appears:



Here you specify the Secure Container connection attributes as passed up in a heartbeat request. The test tool then simulates a heartbeat and application synchronisation. The result is listed as a report of the processing ; what logical platform was assigned, the application summary, data objects, and binary objects.

SkyMobile: Test device/user application synchronisation

```
Hearbeat established:
    Logical platform:   SKY_SMARTPHONE

Retrieved application summary (select timestamps)

910 003 20120702130543
    Binary Objects:
         SKY_GOOGLE_SEARCH_ICON
         SKY_LINKEDIN_ICON
         SKY_TWITTER_ICON
         SKY_WELCOME_LAUNCH_LOGO
         SKY_WELCOME_LOGO
         SKY_WELCOME_LOGOFF
```

4.9.3.10 De-Provisioning Users using the /sky/vti_import_users Function

You may use deprovision users utility to totally remove a user and their run time information from the system. You may do this at either the user or user group relationship levels by setting the **deprovision** flag to *true*. You do not have to perform an “i_regenerate_roles” in this case.

> ***Note:*** De-provision is different from a delete. A delete only removes the relationship and is thus mutually exclusive.

Configure the **deprovision** flag either on the "it_user_list" or on the "it_user_group_user" tables, that is deprovision the entire user or just from a user group. Do not use in conjunction with the “i_delete_user_list” and “i_delete_user_groups" options. In addition to maintain the meta-data, the **de-provision** option removes any provisioned servers and associated session key information from the system. In addition to de-provisioning specific users, you can also de-provision a user from a group that has the effect of removing the servers associated with the roles, for example, if a user moves from sales into manufacturing, they no longer have access to the CRM application.

Example call 1 (specific users):

```
It_user_list-userid = ‘jsmith’.
It_user_list-deprovision = ‘X’.
append it_user_list.

CALL FUNCTION '/SKY/VTI_IMPORT_USERS'
  IMPORTING
    E_RC                    = l_rc
    E_line                  = l_line
    E_MSG                   = l_msg
  TABLES
    IT_USER_LIST            = it_user_list.

if l_rc ne 0.
```

```
   " handle error.
endif.
```

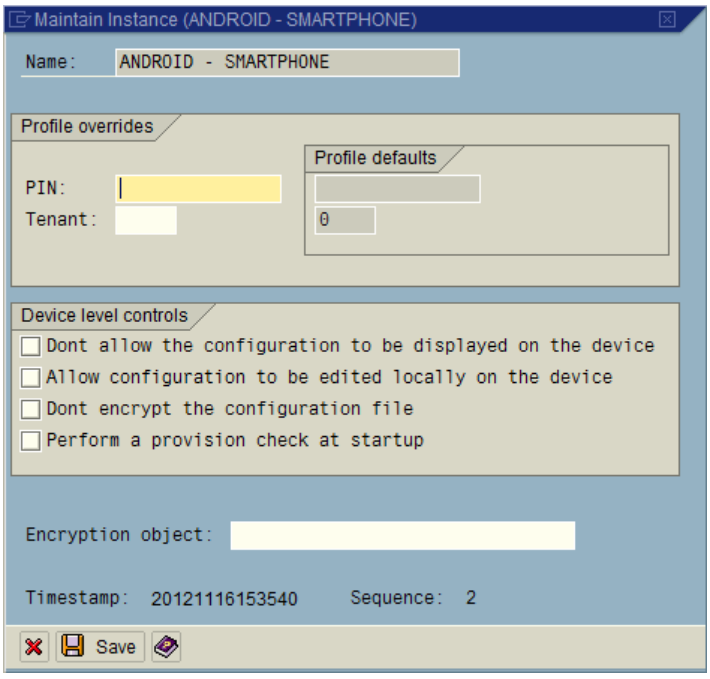
Example call 2 (remove user from group):

```
It_user_group_user-user_group = 'sales'.
It_user_group_user-userid = 'jsmith'.
It_user_group_user-deprovision = 'X'.
append it_user_group_user.

CALL FUNCTION '/SKY/VTI_IMPORT_USERS'
  IMPORTING
    E_RC                    = l_rc
    E_line                  = l_line
    E_MSG                   = l_msg
  TABLES
    IT_USER_GROUP_USER      = it_user_group_user.

if l_rc ne 0.
   " handle error.
endif.
```

4.9.3.11 Server Profile Instances

Instances are the heart of the profile engine. Each instance of a profile consists of its selection criteria, applications, data profiles to generate, configuration, and so on. Instances enable you to configure multiple different sets of rules and configurations against a single profile definition. When a profile is invoked, all the instances are scanned for matching criteria using the technical attributes passed up by the calling Application Server. For example, separate instances could be used to deploy different versions of the application if the device type is a phone or a tablet. A Master Instance may be specified against the profile that contains all the common definitions to be applied for all selected instances, unless explicitly overridden at the instance level. This prevents duplication of configuration, applications, and so on.

Position tour cursor on the Instances node of the profile definition and click on the create icon on the application toolbar. The following pop up will be displayed.



Field	Description
Name	The unique name of the instance definition.

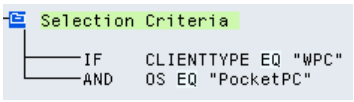
Field	Description
Profile overrides	These values can be specified globally at the profile level, or specifically at the instance level. The PIN is a secret pass code used to verify access to the profile. The tenant is a unique five digit number used for multi-tenant applications. Refer to Multi-tenant support for more information.
Device level controls	These options allow: • prevent the configuration can be viewed locally on the device (default is not enabled) • the configuration to be edited locally on the device (default is protected) • Un-encrypt the configuration data locally on the device • Always perform a provision check at Application Server start up.
Encryption object	Associate a profile encryption object with the instance. This is used to specify the encryption mode, type, key files and options to use to communicate with the Access Gateway.

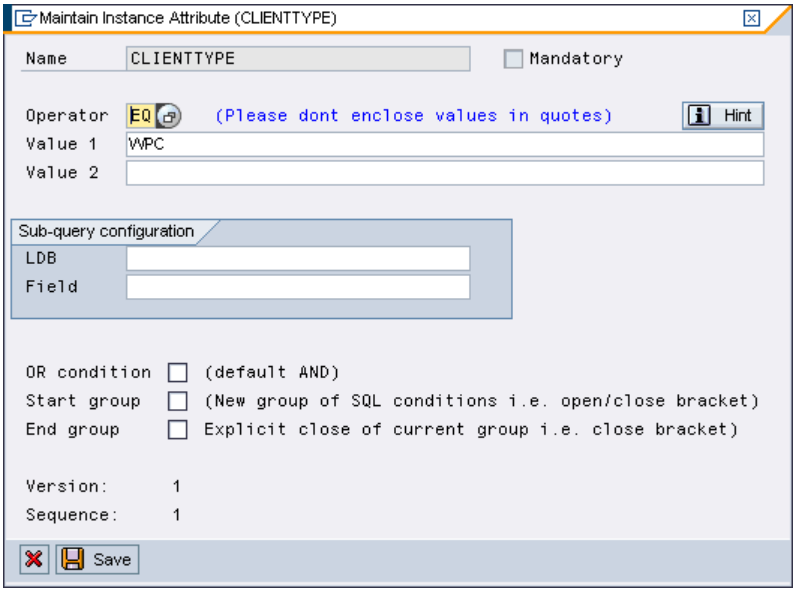
Once you have defined the instance node, you may now define its selection criteria and the list of associated Applications, and Configuration.

Server Profile Instance Criteria

Each instance has selection criteria which is basically a list of attributes and logical expressions that are used to select the instance. An instance attribute must refer to an existing "profile" attribute and consists of the attribute name, a logical operator, one or more values or a LDB sub-query reference. A drop down of available operators is available. A LDB sub-query (QI/QN) returns a list of values to compare against from another LDB table. Complex expressions are also supported using the OR and start/end group options.

Position tour cursor on the Instance "Selection Criteria" node of the instance definition and click on the create icon on the application toolbar. The following pop up will be displayed.





Field	Description
Name	Name of a profile attribute to use.
Operator	The logical operator to apply, for example, EQ, GT, IN, CP. A drop down list of all available options may be used.

Field	Description
Value 1/2	The value(s) to compare against. Most operators may only require one value, but some such as "between" require two (range).
Sub-query configuration	For QI and QN operators, specify the local database (LDB) and one of its fields that will be used to retrieve a list of values. This is effectively an IN condition with the list of values being populated dynamically from a LDB table.
OR conditions and expression grouping	These options are used to define an OR condition and the start and end of a group of selection criteria. By default, each condition is a AND.

Note: Note that selection if the attribute is flagged as mandatory, it must be specified by the calling Application Server.

Note: Sky recommends using a combination of client type (CLIENTTYPE) and logical platform (PLATFORM) to select instances to apply platform specific configuration etc. over and above the 'MASTER' instance.

Server Profile Instance Configuration

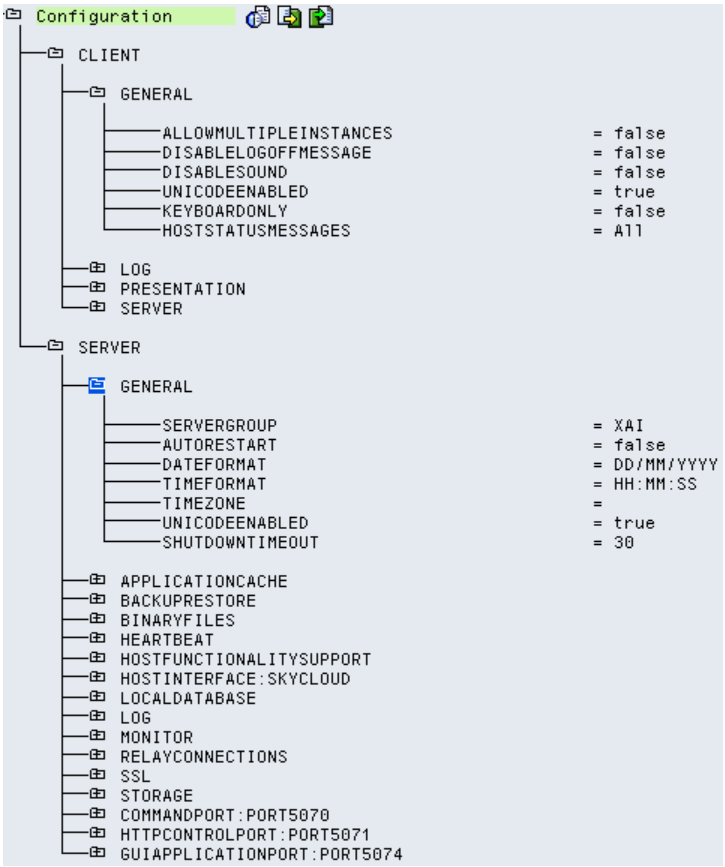
You can use the server profile instance technical configuration for the Application Server, dictates the behavior and control information. The format of a configuration entry is:

Configuration Entry

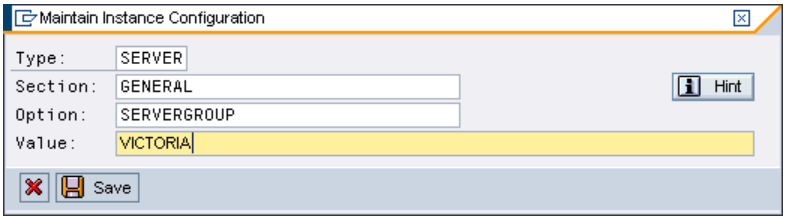
```
"{type}.{section}.{option} = {value}"
```

This is represented in this format in the hierarchy. As well as maintaining each configuration entry individually, you can also Export and Import configuration definitions to and from the PC. This makes it easier for mass editing.

> ***Note:*** If you specify configuration at the master instance level, any option defines at the instance level will override the master.



To create a new configuration option click on the relevant node or level and select the create icon in the application toolbar. If you wish to change a configuration option, just double-click it. The following pop-up appears:



Field	Description
Type	The configuration type, for example, Global, Client, Server. Use the drop down to display a list.
Section	Use the drop down to select a available section within the type, or enter your own custom section.
Option	The configuration option within the section.
Value	The configuration value to assign to the option. You may use special substitution parameters in the configuration value that are resolved at provisioning time (see below).

Substitution parameters

A range of substitution parameters are available to be used in the configuration:

Parameter	Description
&clienttype	Device client type (for example, APC, BPC, IPC, WPC)
&clientversion	Device client version (for example, v23.00.10)
&date	Current MEAP server date
&devicemake	Device make (for example, HTC, Apple)

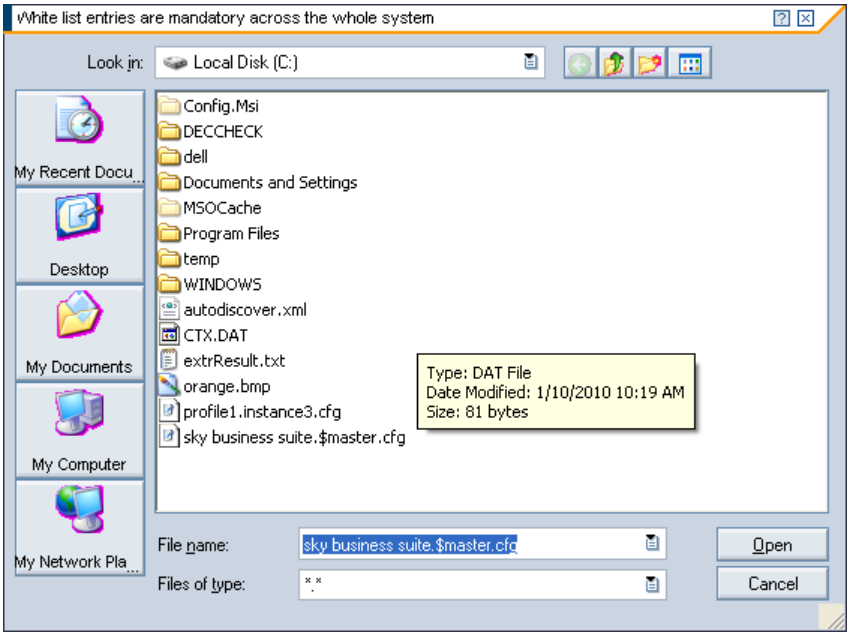
Parameter	Description
&devicemodel	Device model
&gatewayhost	Default Gateway server hostname
&gatewayport	Default Gateway server gateway port number
&hostname	Device hostname
&ipaddress	Device IP address
&mandt	MEAP server client
&meaphost	MEAP server hostname
&meapip	MEAP server IP address
&os	Device operating system
&osversion	Device operating system version
&profile	Profile name passed up by the device
&sysid	MEAP server SID
&sysno	MEAP server system number
&tenant	Tenant number of device
&time	Current MEAP server time
&timestamp	Char14 timestamp based on current MEAP server date and time
&user	User ID from provisioning screen or identity management

Importing and Exporting Configuration

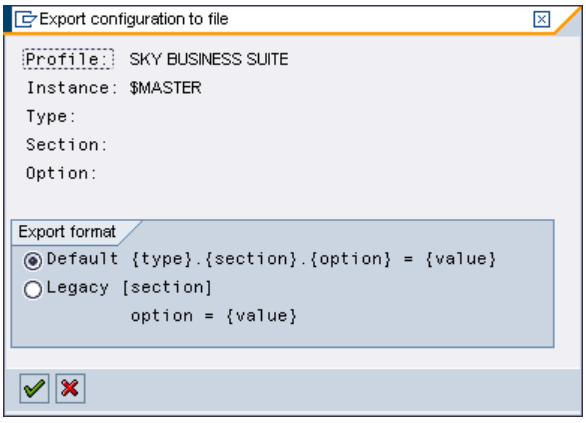
As well as entering configuration options and values using the hierarchy and pop-up, mass editing or importing of legacy configurations can be achieved using the import and export utilities. This enables you to edit a configuration file locally on the PC. Two formats as supported, the legacy "[Section header], followed by options" and the current "{type}.{section}.{option} = {value}' format.

4.9.3.12 Export

The export configuration process, first prompts for the target file name and directory.



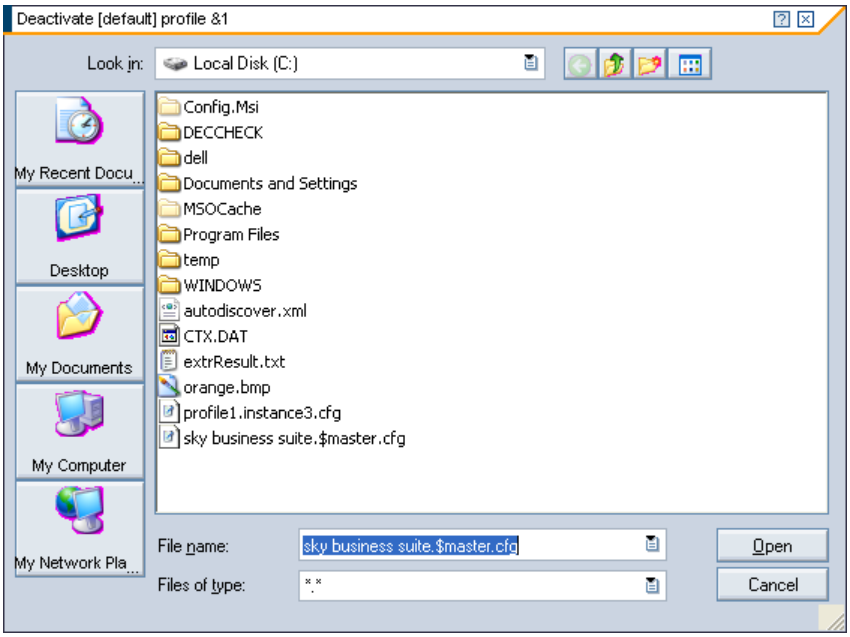
Once the target file is specified, you will be prompted for the format. Sky recommends using the default format, but the legacy may be used for earlier Windows versions of SkyMobile.



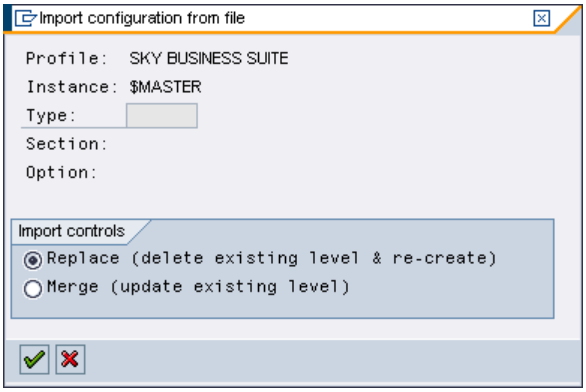
Select the 'green tick' and a file that contains the configuration is created on the machine.

4.9.3.13 Import

The import configuration process, first prompts for the source file name and directory. The source file must be in the SkyMobile default or legacy Windows format, or else the import is rejected.



Once the source file has been specified, you are prompted to either replace all existing values at the level specified (all, type, section, option) or merge in the configuration with the existing configuration.



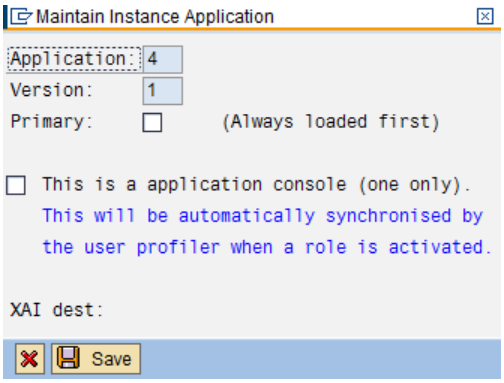
Select the 'green tick' and the configuration is imported.

Server Profile Instance Applications

This is a list of SkyMobile Application Versions that are associated with the Profile Instance; to download and maintain on the device. The primary application, or the lowest appid/verid value, is started first when the Secure Container initializes. If the top level Application contains explicit references to other 'sub-applications through events or procedure start commands, it is only necessary to specify the high level application as the dependant applications are downloaded automatically.

> ***Note:*** You can define globally the list of applications at the profile level by specifying them against the "master" instance definition.

Position your cursor on the Applications node and select the create icon in the toolbar and the following pop-up appears:



Field	Description
Application	The unique SkyMobile three digits APPID associated with an application definition.
Version	The unique SkyMobile three digits VERID within the application definition.
Primary	Indicates that this Application is to start when the Secure Container first starts.

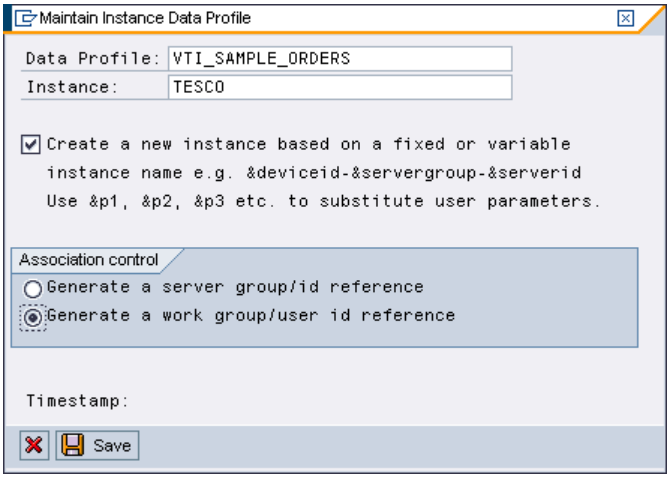
Field	Description
Console	Select this check box if the application acts as a "application console". This means its definition is automatically re-synchronised whenever a associated user profile role is activated to ensure that it has all the latest application and role icons to display. Select this check box if you use the Sky Application Console.

When you provision a Secure Container, the Application/Server cross reference is automatically maintained.

Server Profile Instance Data Profiles

When you first provision an Application Server, you typically want to generate the associated Data Profiling rules to filter the data for that specific server/user. You can associate one or more Data Profile definitions to generate with the Instance. You can define a global list at the profile level against the "master" instance. You have two basic options, add the server or user reference to an existing Data Profile Instance or create a new Data Profile Instance using the supplied Instance "template" name; unique name generated at provisioning time.

Position your cursor on the Data Profiles node and select the **Create** icon in the toolbar and the following pop-up appears:

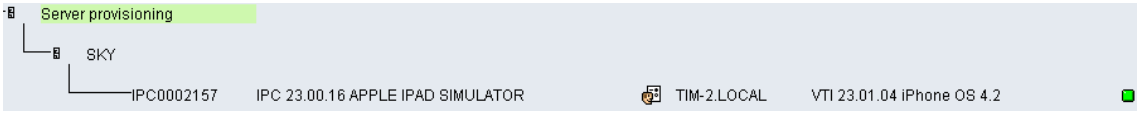


Field	Description
Data Profile	An existing Data Profile definition.
Instance	An existing Data Profile Instance definition, or a template name to use if the create new instance option is selected (see below).
Create new instance	This instructs the provisioning process to automatically create a new Data Profile instance using the template name specified in the Instance field. You may use substitution variables to generate a unique name at run time, for example, &clienttype-&deviceid.

Server Profile Servers

Server profile servers is a list of Secure Containers that are provisioned using the profile instance. The Server Group and ID orders the list in a hierarchy. You can provision only a container against a single profile and instance at a time. This list is used to indicate:

- Whether the container needs to be re-provisioned (highlighted in yellow and a provision icon appears next to the server ID)
- A green or red LED indicates whether the server is currently connected and provides a hot spot to easily navigate to the YECJ server connection display
- The organization data associated with the server
- The provisioned attributes (double-click in the line)



You can perform a provision, or by either positioning on the server line in the hierarchy display, or by selecting the option in the application toolbar of the detail pop-up (see below). These options allow you to update the provisioned information, reset the configuration on the device or totally reset (re-provision) the device. The following icons indicate the options:

Check	Provision check (this is done automatically if you configure against the instance)
Reset	Provision reset
New	Provision new

Double-clicking the line displays the provisioning attributes.

Server Provision Check

If the timestamp of the profile instance is greater than the timestamp of the instance server definition, then the server is a candidate for a provision check. You can initiate this at the profile, instance, server group and individual server level. The process pushes out a PROVISIONCHECK heartbeat command to the selected servers that initiates a call back to the server profiler to return a list of delta configuration changes and application cross reference.

A provision check is performed automatically whenever the Application Server starts if specified at the instance level.

Server Provision Reset

This option removes the existing configuration from the device and request a complete new set of configuration from the server profiler. It also regenerates the server/application cross reference and triggers a application check. All existing data and the same server group/id are maintained. You must restart the server for the new configuration to take effect. The process pushes out a PROVISIONRESET heartbeat command to the selected servers that initiates a call back to the server profiler.

Server Provision New

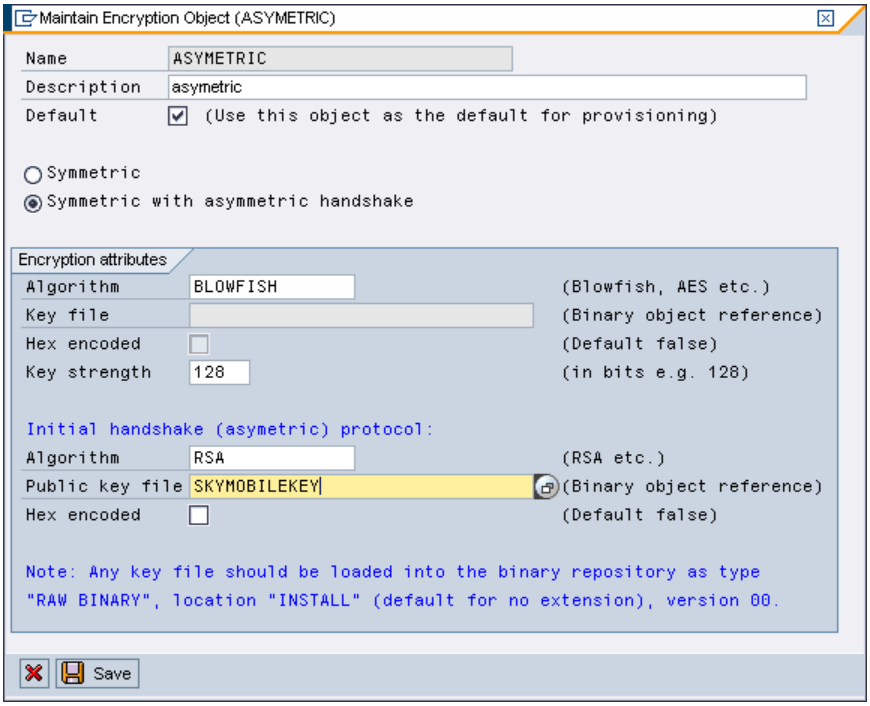
This option effectively removes all existing configuration and data from the server and re-provisions it from scratch, including the server group/id. The same profile/instance and device id is used as before. A PROVISIONNEW heartbeat command is pushed out to the selected servers that calls back the server profiler that passes back the new configuration and application cross reference. Once this is successfully received, it then removes all data and initiates a server restart.

4.9.3.14 Encryption Objects

You use Encryption objects to specify how to implement encryption between the Application Server and the Access Gateway as part of the initial provisioning process. They are associated either against an instance (or master instance), against a "white list" entry or flagged as default; always used where nothing is specified. This is required if the SkyMobile Access Gateway target port (s) are encrypted; all communication from the application server needs need's to use the same encryption mode and attributes. Encryption is used to implement a totally secure connection between the remote Application Server (secure container) and the back end host system, often over a public internet connection.

For more information, refer to the management section on implementing network encryption.

Position your cursor on the Encryption Objects node and select the **Create** icon on the toolbar. The following pop-up appears:



Field	Description
Name	A unique name of the encryption object within the Profile.

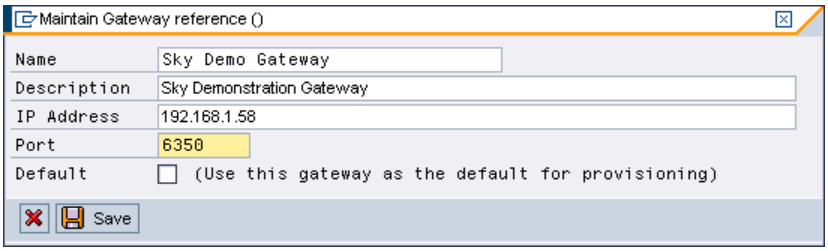
Field	Description
Description	Text further describing the object and its function.
Default	Use the object where none is specified.
Implementation (type)	The type of encryption to implement; symmetric or a hybrid of symmetric with an asymmetric handshake (recommended).
Attributes	The encryption attributes, depending on the implementation type: • The algorithm to use • The key strength to use (in bits) • The key file to use. This must be generated using Kony EKG utility and imported into the SkyMobile Binary Repository. • Whether the key file is hex encoded or not.

> ***Important:*** Kony recommends using the above configurations.

Refer to the management guide for more information on encryption and using the EKG key file generator.

4.9.3.15 Gateway References

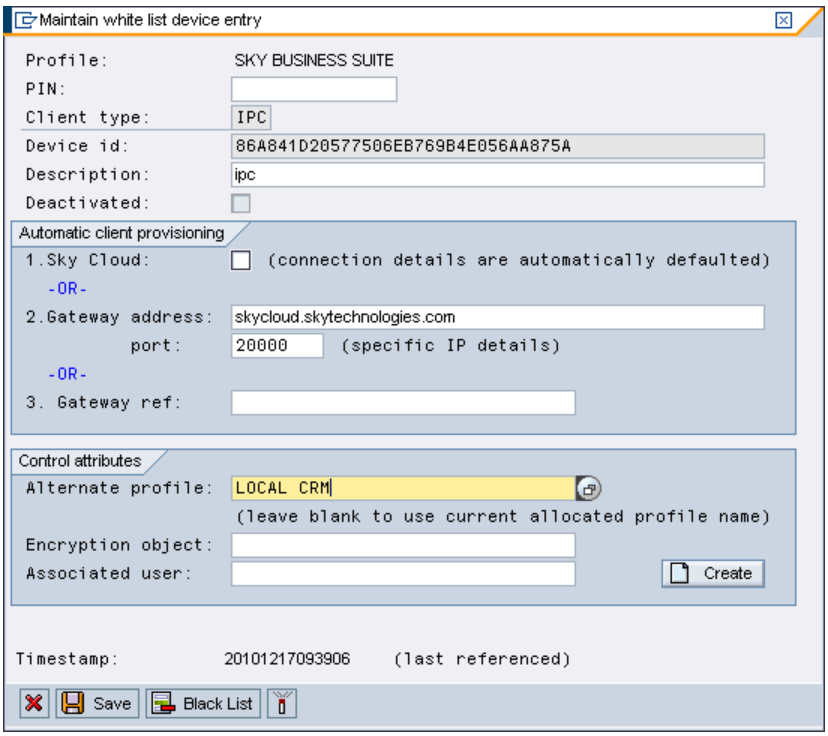
You use a gateway reference to abstract the technical SkyMobile Access Gateway attributes away from "White list" entries; defined in one place. Position your cursor on the Gateway References node and select the create icon in the toolbar. The following pop-up appears:



Field	Description
Name	A unique name of the gateway reference within the profile.
Description	Text further describing the reference.
IP Address	The TCP/IP address or name of the SkyMobile access gateway.
Port	The port number of the SkyMobile access gateway.
Default	Use this gateway reference if you provide none.

4.9.3.16 White Lists

A white list entry is used to define a unique "trusted device" to the system. This is defined in the Sky Cloud to fully automate the provisioning process; the user is not prompted for any information and is launched directly into the application after the Application Server is configured. Each white list entry is unique across the whole system, using the client type and device ID. As well as viewing and maintaining white list entries using the hierarchy, you may use a more productive list mode.

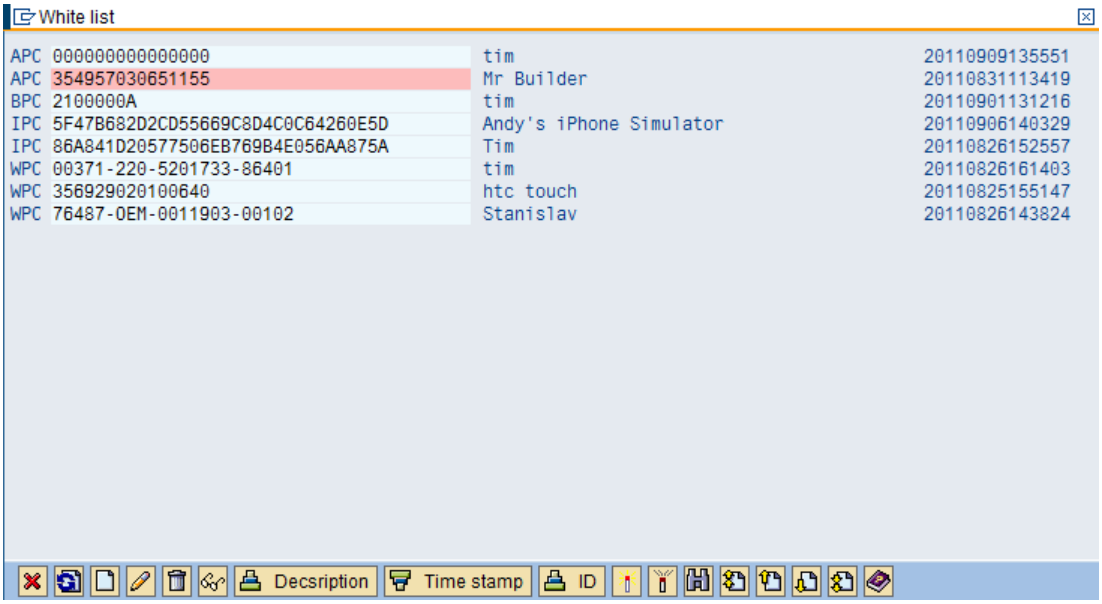


Field	Description
Device id	The unique device ID. This is different for each device type.
Client type	The SkyMobile client type: WPC (Windows), APC (Android), IPC (IOS), BPC (BlackBerry).
Description	Text further describing the entry.

Field	Description
Profile	The profile name to use (you can override this using the alternate profile option).
PIN	The PIN pass code to use.
Description	Text further describing the entry.
Automatic client provisioning	Specifies the provision connection option to use: • Provision to the Sky Cloud • Provision to a specific gateway using either a specific address/port or a profile gateway reference
Control attributes	You may specify an alternate profile to use for the provisioning request, an associated encryption object and an associated profile userid to associated with the request. The userid is useful to associate a user directly with the unique device identifier and the MEAP Server can automatically default if the userid is unknown by activating the "Default the provisioned userid if there is no explicit connection" option on the profile. In this way, no explicit user authentication is necessary.

White List Mode

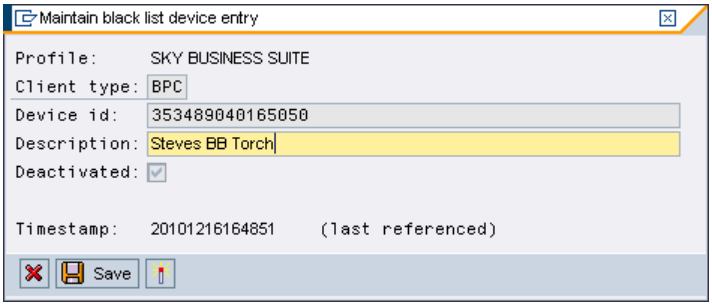
You have the option to display and maintain a profiles white list entries in a list rather than the hierarchy. This is useful when maintaining multiple entries with added sort and find capabilities.The list filter still applies to restrict the overall list. To invoke the list, click the icon on the White List node and the following pop-up appears:



As well as the standard maintenance functions, you can sort and find entries as required.

4.9.3.17 Black Lists

As opposed to a white list that is used for automatic provisioning, you use a black list to deny access to the system and reset the device if it tries to connect. A reset removes all configuration and data from the Application Server.

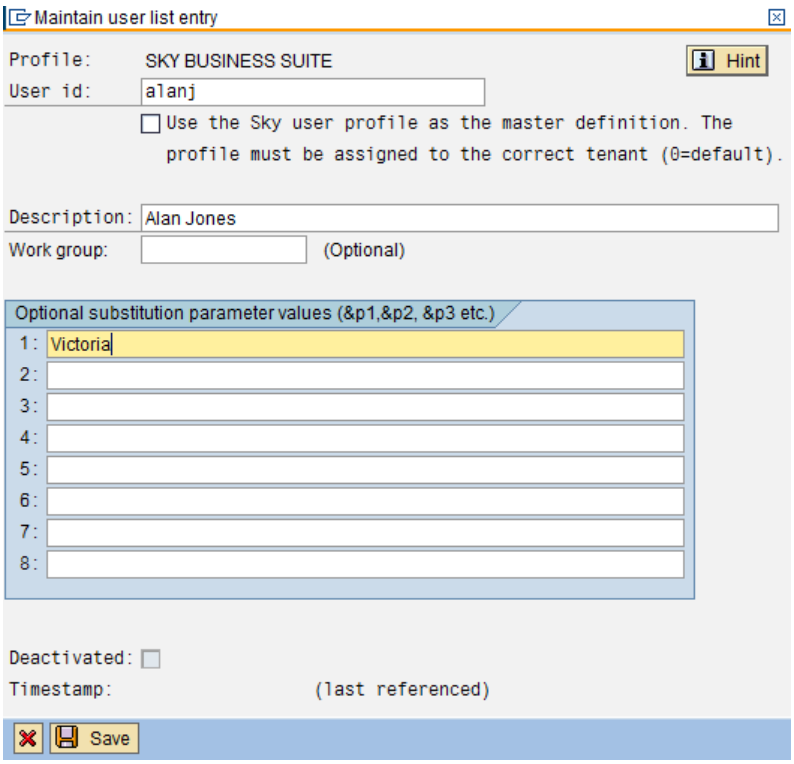


Field	Description
Device id	The unique device id. This is different for each device type.
Client type	The SkyMobile client type: WPC (Windows), APC (Android), IPC (IOS), BPC (BlackBerry).
Description	Text further describing the entry.

Important: Blacklisting the device causes the Application Server to reset; all configurations and data are wiped.

4.9.3.18 Server Profile Users

A Profile user list is used to identify a particular user at provisioning time; link a user with the Application Server. You may also configure that a user is mandatory and whether to check the user table as well on the profile definition level. Position your cursor on the Users node and select create from the toolbar and the following pop-up appears:



Field	Description
User ID	A unique name within the profile

Field	Description
Use Sky User profile as master	This indicates that the Sky User Profiler definition is the master definition and you use this entry purely to restrict the users that are associated with the profile. ***Note:*** Sky recommends that you use the user profiler always as the master user definition, and local profile users only to restrict which users can access the profile definition.
Description	Text further describing the definition
Work Group	A optional work group that the user is allocated to
Parameter values	Up to eight parameter values, used for dynamic substitution

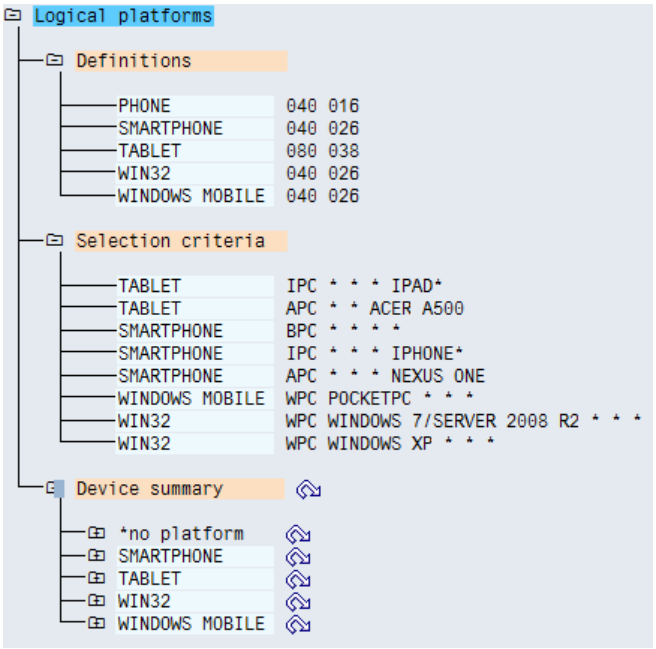
Note: This definition is used primarily to associate a unique user with an Secure Container at provision time, so that you may generate user level data profiling. It is also used for kiosk style applications that don't necessarily have a logon screen, but still need to have an associated user.

4.9.3.19 Logical Platforms

The logical platform facility enables you to abstract the technical device OS, make, and model away from profile selection criteria and use a common platform name instead, for example, SMARTPHONE or TABLET irrespective of the device type. For instance, a Company may deploy iPad and Android tablets, and iPhone and BlackBerry smart phones, so instead of specifying all the different possible device selection criteria on every server profile, you create a TABLET and SMARTPHONE logical platform, assigning the technical selection criteria and then use the logical platform for server profile instance and user profile role selections.

> ***Note:*** Kony highly recommends using logical platforms and they are mandatory for user profiling with roles. Kony check-install process automatically creates sample logical platforms to cover all the major mobile platforms. You can activate and use these immediately.

All logical platform definitions are maintained using the functionality in the server profile workbench (see the node at the bottom of the hierarchy). You can restrict the display to just logical platforms by selecting the **logical platforms only** check box on the selection screen.



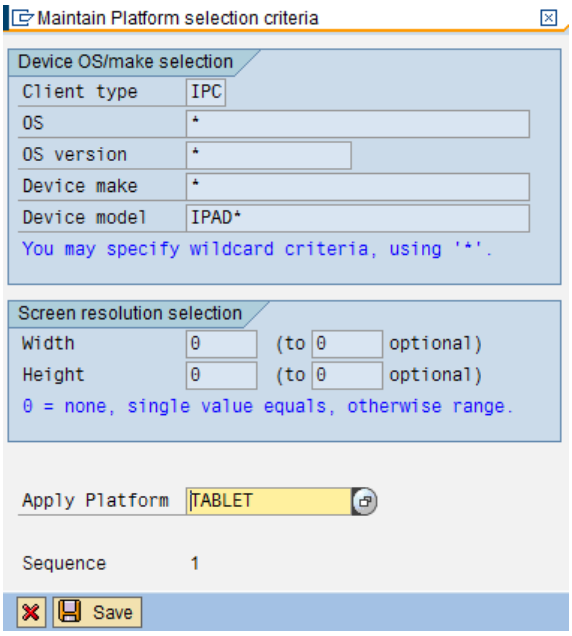
The logical platform list is made up of three main sections:

Section	Description
Definitions	These are the logical platform names and recommended screen dimensions.
Selection criteria	The technical device attributes used to select the logical platform. These are assigned in sequence from top to bottom.
Device Summary	This is a automatically summarized list of all devices that are connected to Sky mobile. It is broken into assigned logical platforms (if any).

To maintain any of these entries, just position your cursor and click the create, change, delete icons in the application tool bar. At any stage, you can force the logical platform summary to be re-evaluated by clicking the icon adjacent to the hierarchy entry.

Maintaining Selection Criteria

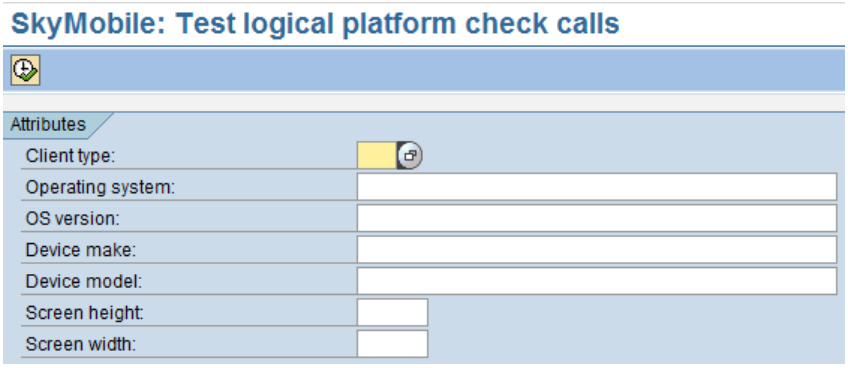
You use the selection criteria to assign technical device attributes to a logical platform definition using client type, OS, device make/model and resolution. It is evaluated in sequence and you can toggle this using the up/down arrows in the application tool bar. The following pop-up appears:



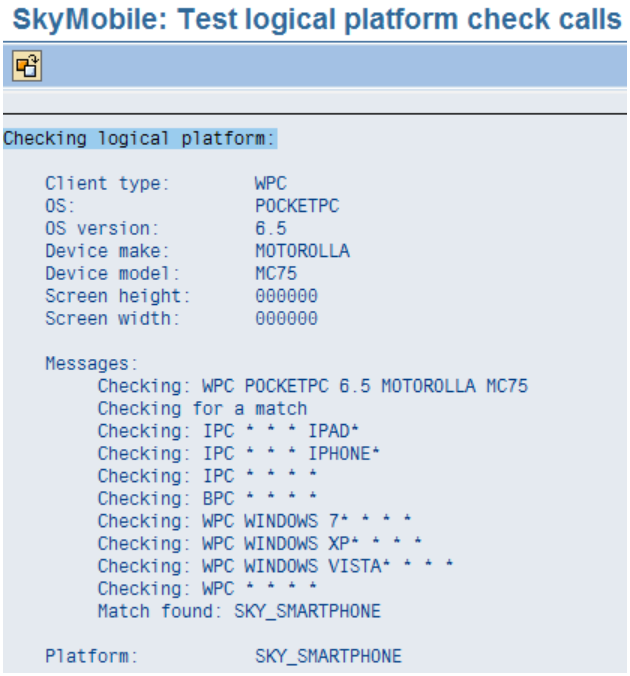
You enter specific values or wild card entries using '*' in the attributes. When you connect or provision a Secure Container, it is assigned a logical platform by scanning the criteria top to bottom. Once a logical platform is assigned, it is retained unless the criteria is reset.

Logical Platform Test Facility

You may use this utility to test the selection criteria used to determine the logical platform. Position your cursor on the logical platform node and click the Test icon on the application tool bar. The following selection screen appears:

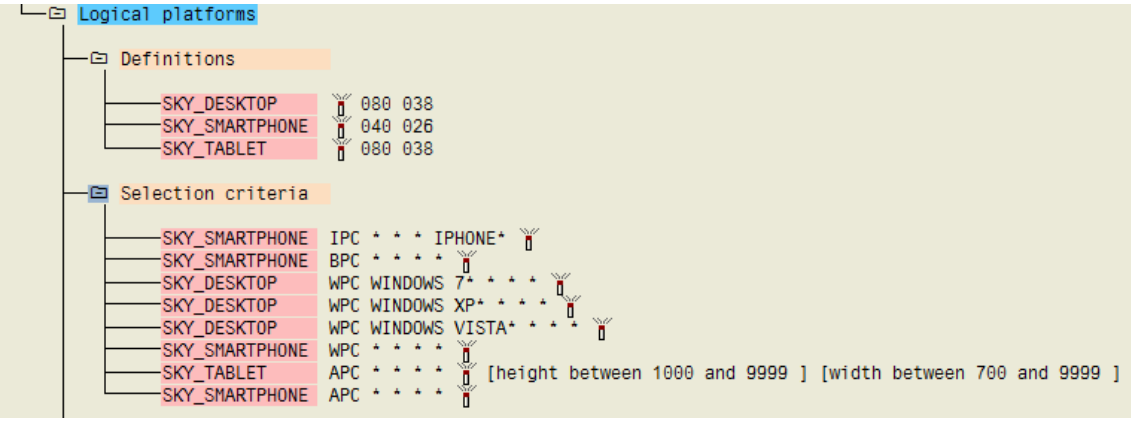


Enter the criteria and execute, and a report of all the processing appears. This option is useful to make sure the criteria is valid before applying it in a production system.



Sample Logical Platforms

The SkyMobile check installation process automatically creates logical platform definitions and selection criteria for all the major mobile platforms. Initially, these are deactivated, but you can activate to put to use immediately. You may also copy the definitions.



> ***Note:*** The selection criteria covers IPC (IOS), BPC (BlackBerry), WPC (Windows) and APC (Android). The selection criteria is carefully ordered to ensure that the correct platform selection is performed, for example, if it is not a WINDOWS 7/XP/VISTA desktop or server, then it must be windows mobile.

4.9.3.20 Unique Device Identifiers

You use the following attributes as unique device identifiers.

Platform	Device Identifier
IOS (iPhone, iPad, iPod iTouch)	• UDID stripped of hyphens
Android	• IMEI (If available) • Mac address of the WIFI (if available) • Mac address of the bluetooth adapter (if available) • Unique internal device ID
BlackBerry	• IMEI (If available. Not available on CDMA) • MEID (CDMA only) • Unique internal device id
Windows Mobile	• IMEI
Windows Desktop	• <Computer name>.<Microsoft product id>

IMEI Number (Wikipedia):

The International Mobile Equipment Identity or is a number, usually unique, to identify GSM, WCDMA, and iDEN, mobile phones as well as some satellite phones.

It is usually found printed inside the battery compartment of the phone. You can also display it on the screen of the phone by entering *#06# into the keypad on most phones. The GSM network uses the IMEI number to identify valid devices and therefore you can use for stopping a stolen phone from accessing the network in that country. For example, if a mobile phone is stolen, the owner can call his or her network provider and instruct them to "blacklist" the phone using its IMEI number. This renders the phone useless on that network and sometimes other networks too, whether or not the SIM of the phone changes. You use the IMEI is only for identifying the device and has no permanent or semi-permanent relation to the subscriber. Instead, the transmission of an IMSI number identifies the subscriber that is stored on a SIM card that you can (in theory) transfer to any handset. However, many network and security features are enabled by knowing the current device that the subscriber uses.

iOS UDID:

Every iPhone, iPod touch and iPad has a unique identifier number associated with it, known as a UDID (Unique Device ID). Your UDID is a 40-digit sequence of letters and numbers that looks like this: 0e83ff56a12a9cf0c7290cbb08ab6752181fb54b. To find your iPhone's UDID, just plug it into your computer and wait until iTunes recognize it. Select your phone from the Devices list in iTunes and click the **Summary** tab. To see your UDID, click the word "serial number" beside the picture of the iPhone. You should see the word "identifier" and an alphanumeric string – this is your UDID. Alternatively, you can install and run the "Ad Hoc Helper" application. It creates an email with your UDID. Send the email to nobody@example.com.

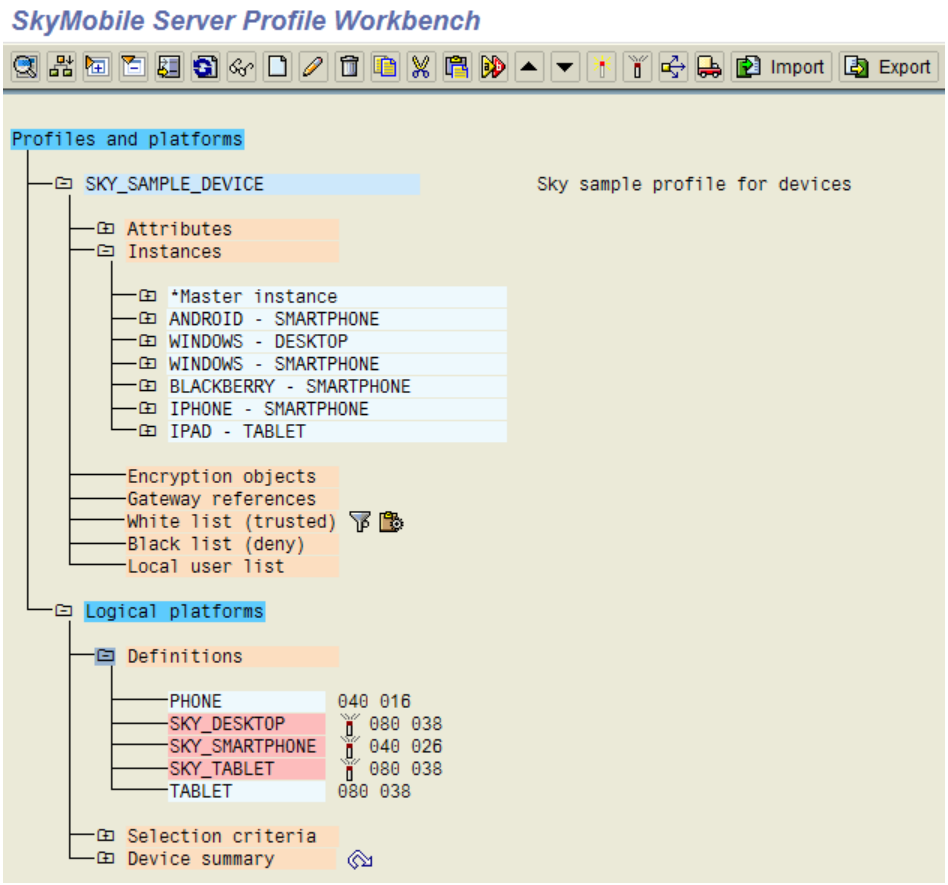
4.9.3.21 Sample Device Profile

The SkyMobile check install process automatically creates an example profile that demonstrates configurations and instance selection for all the major platforms. The idea is that you may copy this and then use it as a basis to deploy your own profile. The profile uses the Sky logical platform feature and samples of these are also automatically created, for example, SKY_TABLET, and SKY_SMARTPHONE.

> ***Important:*** Do not use the SKY_SAMPLE_DEVICE profile directly as it is re-implemented for each new release and you may lose your changes.

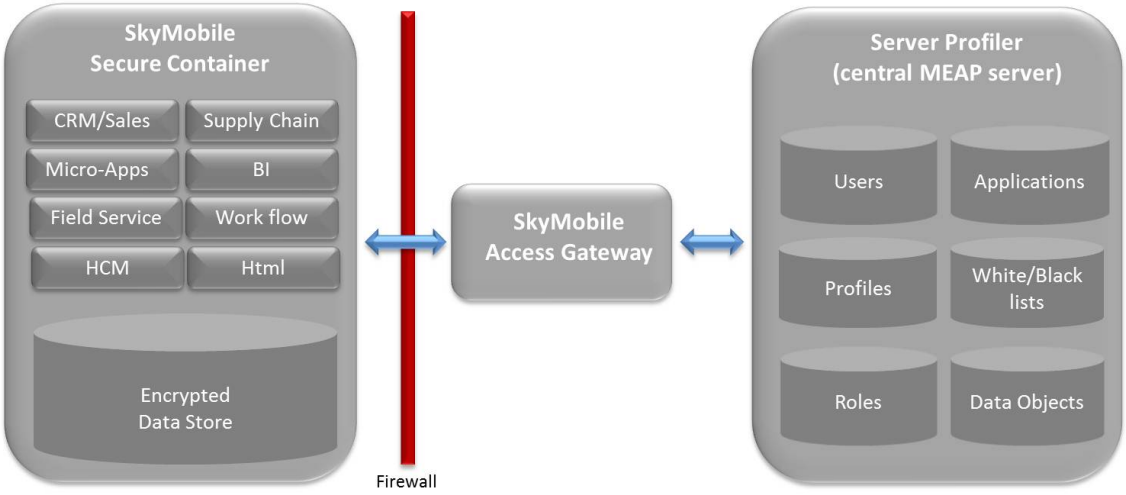
The following screen shot shows the sample device profile with instances for all the major device platforms. You can literally copy this profile using the workbench and use as is.

> ***Note:*** You need to activate the SKY* logical platform definitions and selection criteria.

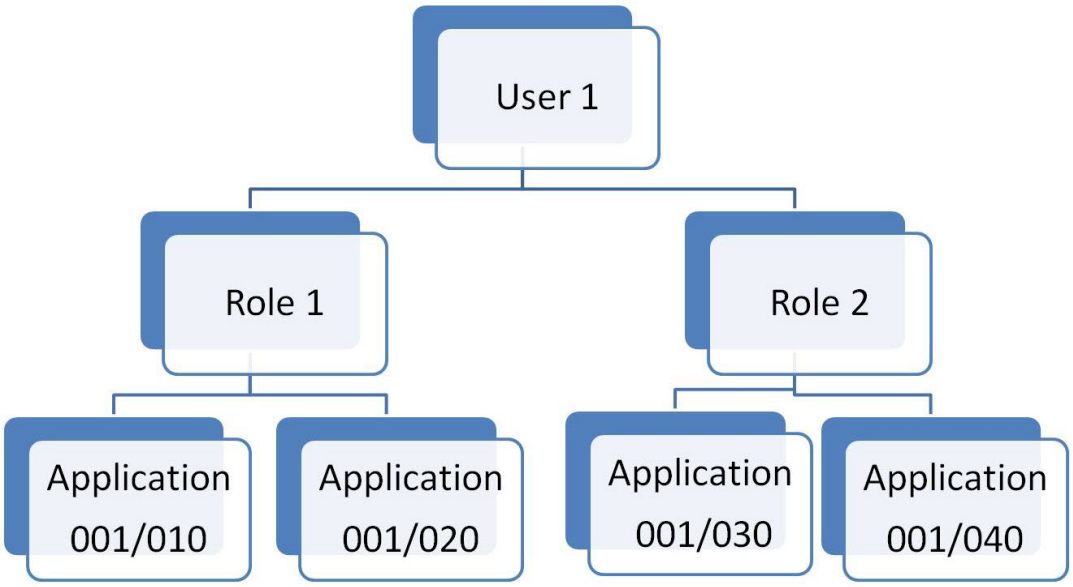


4.9.3.22 User Profiler

User profiling is all about "role based provisioning"; defining exactly what applications and data a specific user (or group of users) gets. In the user profile workbench, you maintain user and role definitions. Applications and data profile definitions are then assigned to a specific role. When the role is activated in the workbench, a runtime object is generated that is automatically synchronized with the Secure Container on the device once it is successfully provisioned and the user is identified. The Sky application console may use this data object, basically a application launch pad, or a custom applications to determine what applications are requested or not. User profiling is designed to use in conjunction with server profiling, identity management and the application console to fully automate on-demand application and data synchronization.



The basic premise is that users are assigned to roles that then have logical platforms and applications assigned to them. It is really important to understand how logical platforms are defined and are then assigned as part of the container provisioning process. You need to assign different platforms under a role so that you can manage cases where users have different or multiple devices and may require different application versions and/or data profiling.



The user is identified and linked to a device either by Sky Identity Management (recommended) or a custom logon screen. The current user is passed up to the host through the heartbeat connection. The logical platform is derived and assigned when the Secure Container issues its first heartbeat request to SAP. Both the user and platform information is used to control which applications and data objects are required to be synchronized and what application console meta data is to be downloaded through the user runtime object.

Using Identity Management and the Application Console

The basic process using the application console and identity management is as follows:

- Define the server profile for the Secure Container:
  - Add the identity management configuration
  - Assign the Sky application console (Smart phone - 910/001 , Tablet - 910/002) as the default application and tick the "this is a application console" check box. The Sky application console
- Define the user profile definitions:
  - Maintain the roles, platforms, applications and data profiling as required
  - Maintain the work groups and users and associate them with the roles

- When the role is activated, a run time data object is generated for each user.

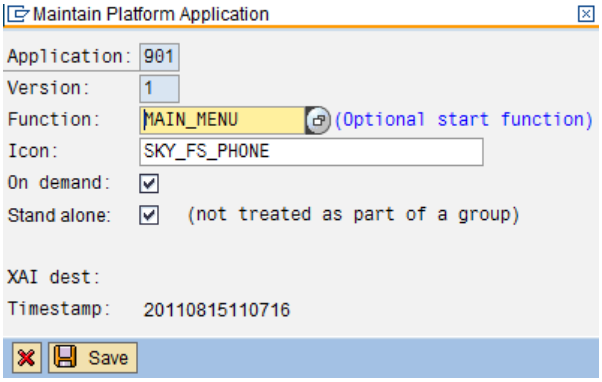
- Download and start the Secure Container:
  - The user is authenticated (through identity management) before the container starts
  - The users applications and data are synchronized automatically with the device
  - The Sky application console is invoked displaying all the applications available (users may request some on demand)

Logical Platforms and the Master Platform

The Logical platform definitions are maintained in the server profiler workbench. You use them as a further level under the role to distinguish what applications and data profiling apply to specific device platforms, for example, a smart phone versus a tablet. This enables 'role based provisioning' across different device types for a specific user. For example, a sales representative may have a smart phone and a tablet device and have different applications for each. If you are not interested in segregation by platform, you may just use the '*master platform' definition to define all common objects.

User Profile Role Platform Application

You can add one or more applications to a role platform definition. To create a new application reference, position your cursor on the platforms applications node and click the **Create** icon on the application tool bar. The following pop-up screen appears:



Field	Description
Application	The three digit application area identifier
Version	The three digit version identifier
Function	An option function within the version to start from. If this is left blank, the initial function is used.
Icon	A binary object graphic you use to identify the application in the application console. You must load this into the binary object repository
On demand	The application is not automatically loaded when the Secure Container initializes, but you may select from the application console
Stand alone	By default, all the applications appear under a "role" folder/icon. If you select this check box, the application always lists on the primary screen

4.10 Design Considerations

This section covers some specific design principles and 'how to' examples.

4.10.1 Key Topics

Client Capabilities
Unicode Support
Multi-Language Support
Multi-Tenant Support
Look and Feel
Sounds
Integration Options

4.10.2 Modular Application Design

The following section describes how Kony for SAP supports the concept of modular application design and how effectively you can implement.

Overview

What is modular application design?

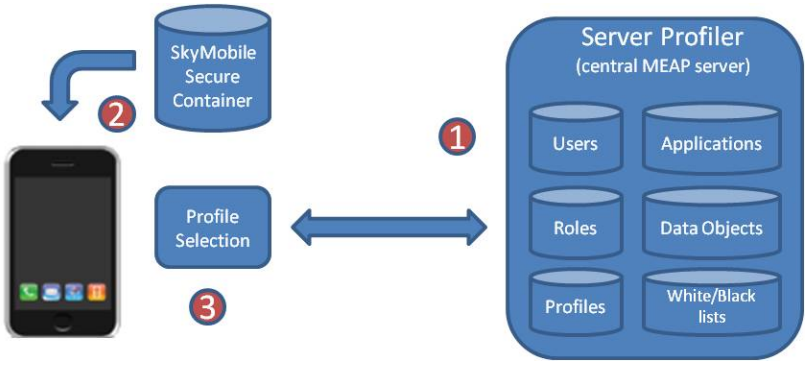
Technical implementation

The application console

FAQ

4.10.2.1 Overview

Companies expect to be able to securely deploy multiple mobile applications to their Employees, Customers and Vendors based on their business roles. The SkyMobile framework provides a native "secure container" that you can deploy onto all major mobile devices to manage mobile applications on-demand, and effectively profile and securely store all data on the device. Everything runs and is stored within the SkyMobile "secure container" and thus, Enterprise applications and data can securely co-habit with private applications and data on the same device. The following diagram provides a summary of the SkyMobile automatic provisioning process:



1. The users, roles and provisioning requirements (configuration, applications, and data profiles) are defined centrally using the SkyMobile Server Profiler and are associated with a unique provisioning profile.

2. The SkyMobile Application Server “secure container” is downloaded from the major app stores, provisioned using device management software, or manually implemented using device specific workbenches and methods.

3. When the SkyMobile container is first started on the device, a provisioning profile is associated with the user, either through a trusted “white list” entry (unique device type/id) or by entering a profile and pass code when prompted. Once selected, SkyMobile automatically configures the device for use and synchronizes all the applications and data objects that the user requires.

Though you may still deploy one large application with built-in role security checks, SkyMobile also supports the deployment of multiple separate applications (modules), associated with distinct roles, that you can download and execute “on demand”. You need to think of deploying a “modules based on role” approach, rather than “main menu down” through in terms of its impact on application deployment, the overall data model, data synchronisation and usability. This paper explores SkyMobile modular application design concepts and some considerations that you should take into account.

4.10.2.2 What is Modular Application Design?

Modules are logical groupings of business functionality based on user, customer or vendor roles. Instead of grouping all the functionality into one large application and then permitting and restricting access based on the user permissions; SkyMobile provides the flexibility to define and deploy multiple applications (applets); each catering for a specific business role. The primary advantage of this is to use the SkyMobile framework to securely provision and manage multiple Enterprise mobile applications within the “secure container” on the device; you only get and see what you are authorised to use.

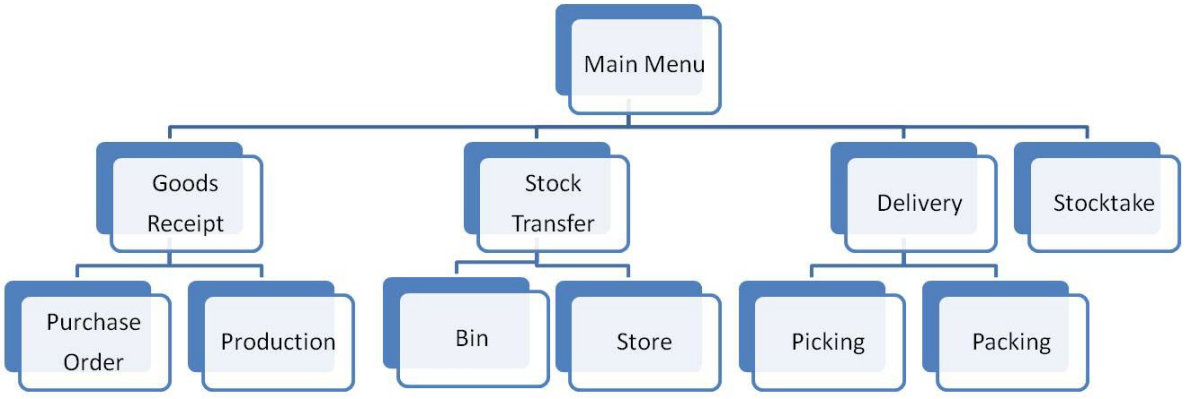
It is obvious right away, that you must take technical inter-screen and data dependencies into account as to the level of granularity of the modular design, for example, breaking the functionality right down into individual transactions probably adds more complexity than benefit. A must seek correct balance in the design phase considering the role, overall business process and technical implementation; logic and data model.

Example:

A basic Warehouse Management application with distinct roles to:

- Goods receipt a production order into stock
- Goods receipt product off a production line into stock
- Transfer stock within a store (bin to bin)
- Transfer stock between two stores
- Pick and pack deliveries
- Perform a stock take.

In the first case, you implement all the functionality into a single application and the user is restricted to what they can do (or see) by security permission checks built into the application logic. The screen hierarchy may look something like this:



In the second case, the functionality is broken into six separate applications (modules) based on the distinct business roles. You may still imbed “cross calls” between modules, for example, invoke stock transfers within delivery picking and packing. If the application function does not exist, the user is informed through a pop-up message.

What is important to note is that both designs cater for the same business roles and functionality, but the difference is in how the applications are presented and managed. In the first case, the entire application is deployed out to the device along with the user permissions. In the second case, just deploy the application that they are authorized to and launch it from the SkyMobile application console (or custom main menu). In either case, you deploy the entire Warehouse Management data model and only the data objects that they are authorised to are synchronized and stored on the device.

Benefits

Both “single integrated” and modular application designs are valid. A modular design can be more flexible when you deploy functionality to distinct roles (especially external customers and vendors); and when sharing common functionality between different application areas. For example: A stock inquiry can be common to Warehouse Management, Field Service and Sales; and a leave request can be common to all Employees. A single integrated application may be preferred if there is a high level of integrated logic, business process work flows and/or dynamic run time context switches.

4.10.2.3 Technical Implementation

This section provides a high level description of the functionality and recommendations used to implement role based modules. For a more in-depth discussion, consult the SkyMobile online help.

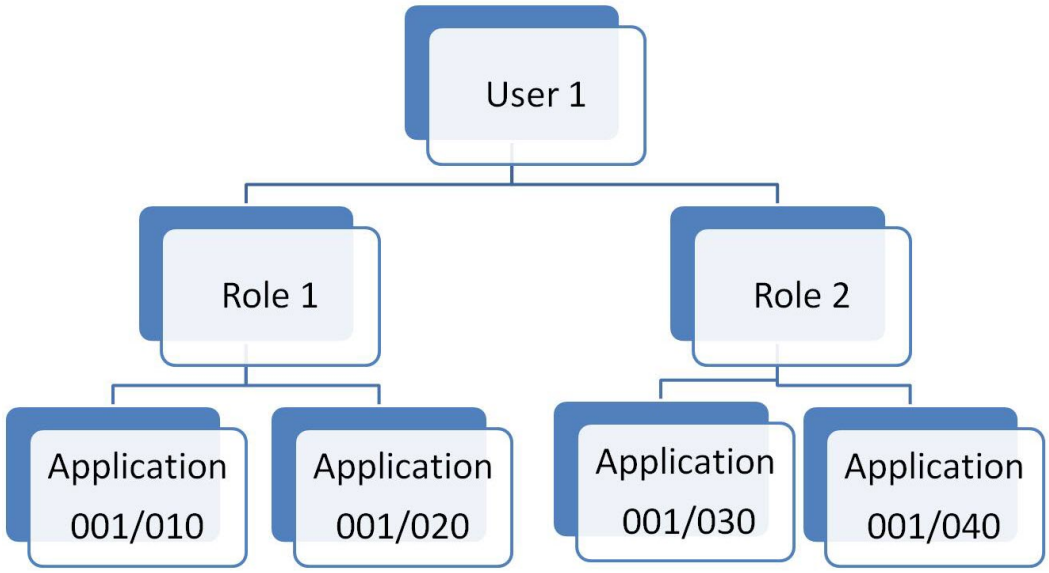
Application versions:

In terms of SkyMobile technical terminology, a module equates to a specific application / version combination. For example, 001/010; and you can have up to 900 thousand of these. Sky recommends that you assign a unique application number to a specific application area, for example, Warehouse Management, and then assign each module a unique version number in increments of 10, allowing for variations, for example, tablet versus phone, differences in the level of functionality, and user interfaces.

APPID	Description	VERID	Description
001	WM	020	Goods receipt production
		030	Stock transfer (bin to bin)
		040	Stock transfer (inter-store)
		050	Delivery picking and packing
		050	Delivery picking and packing
		060	Stock take

Roles and Users:

Once the application versions (modules) are defined, they are assigned to roles that are then associated with a user, either directly or by association with a provisioning profile.

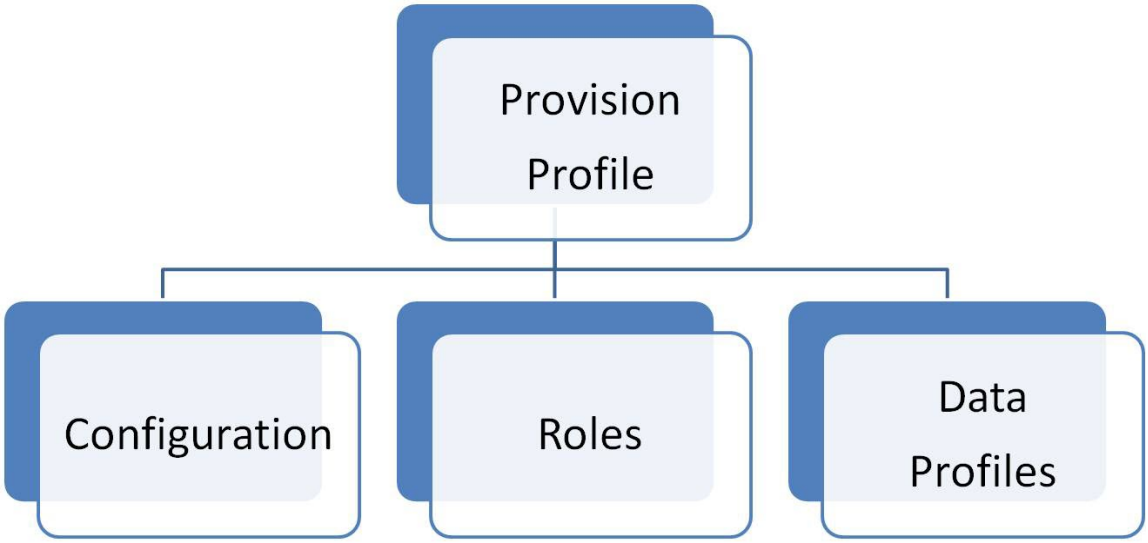


Note: You can populate the SkyMobile user table from an external active directory, and HR employee meta-data; used not as the master, but as a place holder.

Provisioning Profiles:

A provisioning profile links the entire configuration, role and data object requirements to implement onto a device. As described in the preamble, a provision profile is assigned the first time the SkyMobile Application Server (secure container) is first started; either automatically through a “white list” entry, or manually through a prompt. The following processing takes place:

- Data profiling rules are generated (data synchronization controls).
- All configuration, applications, data definitions and initial data loads are implemented and synchronized with the Application Server (secure container) on the device.
- Any subsequent changes are either automatically or manually synchronized.



There is flexibility in the Server Profiler functionality to cater for many different scenarios:

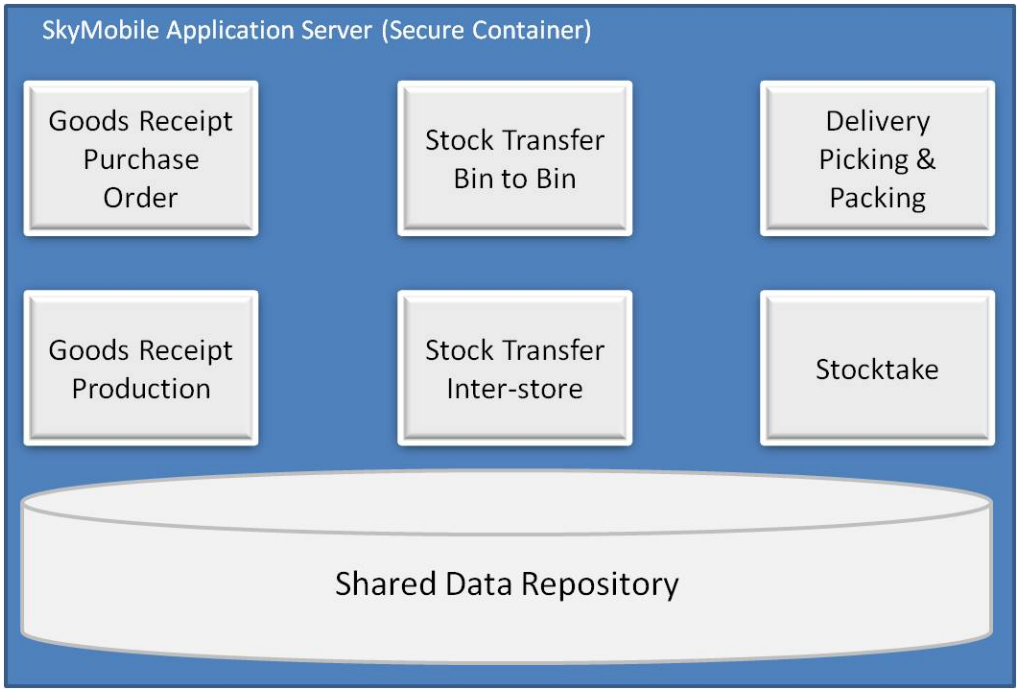
- You may associate applications or roles directly with a provisioning profile; at the Application Server level with no association with a specific user.
- Multiple users may share the same device; their provisioning profile is determined at user authentication time (identity management) and each user only sees the applications they can use. Because it is a shared database, you need to build data segregation into the data model and applications.
- You may configure data profiling rules at either the role or provision profile levels.

Code free applications:

An Important concept of the SkyMobile framework is the ability to configure “code free” applications; everything is configured and no programming is required. The Application Server downloads the application definitions as meta-data and interpret at run-time (secure container) on the device. This means that you need not manage program executables, thus making provisioning much easier and flexible. It also means that you can make changes to applications and data definitions dynamically at run time while the user is active.

Managing the Data Model:

Assuming that the primary motivation is to break the functionality down into logical business roles; the first question you usually ask is: “what about the data model?” In most cases, you can integrate the data model across all the functionality and can rarely separate along with each module (shared master data). To avoid complexity, Sky recommends that the entire data model is managed at the application area level and that you then use data profiling to determine what data gets synchronized (all users get the entire data model regardless, and only receive the data they are authorized to). This is made possible through SkyMobile flexible data object architecture and data profiler. Storing all the definitions on the device does not incur extra processing overhead, and does not consume much storage.



To make this easier and avoid having to assign all the data objects to every application version, you may now assign common data objects, just once, at the application level and these get inherited by every associated version (module). You may share the same data objects across application versions and duplication does not cause conflicts. You may also assign data profiling configuration and rules at the role level in the application profiler or in the server profiler. These get generated automatically when the user is first provisioned onto the Application Server (container) on the device.

Note:

1. In terms of application design, the same principles apply, that is, design the data model first. Getting it right is essential to the success of the project.
2. Although Sky recommends deploying the entire data model, you still have the option to deploy only the data models associated with specific modules.

4.10.2.4 The Application Console

Kony provides a default Application Console application that you may use as a launch pad for applications. The console provides the framework for “mobile applications on demand”; you can add and remove applications from an authorized list. The user may also perform some basic customization (if allowed), such as change the background image, assign different colours, manage a favourites list, and toggle the list view. You can fix all or part of the user interface, for example, the company logo and standard colour scheme. The application console also provides extra functionality to easily navigate between modules using the standard navigation bar that is present on every screen.

The user logs onto the console, using standard Identity Management functionality. You use the user id to determine the available applications and groups. The mobile applications are organized within their application group (if any). The first console screen shows the group and all applications that are not explicitly assigned to a group. You can easily expand this to display the related applications.

Note:

- Using the Kony Add-in application console is optional; you can configure your own main menu or equivalent “launcher” application.
- If the application requires additional control attributes, for example, current store, you must prompt these for when that specific application starts. You can use the shared memory to store and manage common attributes between modules.

For more information on the application console, click here.

4.10.2.5 FAQ

Do modules make change management more difficult?

By breaking down application areas into modular units, you can make management and deployment easier and faster. You may also deploy functionality in a phased manner.

Do modules make my data profiling more difficult?

No. There is no link between data profiling and applications. It is done at the server or user levels. So if you have one application or multiple applications, there is no impact on the number or how complex your data profiles are.

Do modules impact performance?

No. There is no difference between a single application and having multiple applications. The application memory cache on the device determines how many screen function definitions you may store in memory before swapped out based on an aging algorithm.

Can multiple users share the same SkyMobile application server on the device?

Yes. Each user only sees the applications that they can use. However, the data model is shared and you need to incorporate this into the overall application design, for example, each user is only shown the orders they are assigned and not all the orders on the device. Though it is possible to deploy multiple SkyMobile Application Servers (containers) on the device, this can lead to additional provisioning and management complexities. In most cases, a single server suffices.

Is testing more complex?

This depends on the granularity of the modules; the more modules you have spanning a business process, the more complex the testing of the whole process can become. In other words, if you download a single application (containing everything), you can test the roles independently, without having to re-provision the device for each different role.

Do I need device management to manage the applications?

No. The SkyMobile “secure container" handles internally all application synchronization and management on the device and so there is no need to use device management to control the applications. You may use device management to deploy releases of the SkyMobile “secure container” itself.

4.10.3 Client Capability

The tables below highlight points of difference between the various client platforms. In an environment where a heterogeneous mixture of devices is present, you should take this into account during application design.

Field Types

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Buttons	✓	✓	✓	✓	✓	✓
Charts	✓	✓	✓	✓	✗(3)	✗
Checkboxes	✓	✓	✓	✓	✓	✓
Date Fields	✓	✓	✓	✓	✓	✓
Drawing Areas	✓	✗(3)	✓	✓	✗(3)	✗
Email Addresses	✓	✓	✓	✓	✗(3)	✗
Graphics	✓	✓	✓	✓	✓	✗
Hotlinks	✓	✓	✓	✓	✓	✓
Input Text Fields	✓	✓	✓	⚠(5)	✓	✓
List Boxes	✓	✓	✓	✓	✓	✓
Output Text Fields	✓	✓	✓	⚠(5)	✓	✓
Password Fields	✓	✓	✓	✓	✓	✓
Phone Numbers	✓	✓	✓	✓	✗(3)	✗

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Radio Buttons	✔	✔	✔	✔	✔	✔
Shade Areas	✔	✔	✔	✔	✔	✔
Text Boxes	✔	✔	✔	✔	✔	✔
Text Labels	✔	✔	✔	✔	✔	✔
Time Fields	✔	✔	✔	✔	✘(3)	✔
Toggle Fields	✔	✘(3)	✔	✔	✘(3)	✔
Web Views	✘(3)	✘(3)	✔	✔	✘(3)	✘

Input Aids

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Alphanumeric Keypad	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘
Calculator	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘
Date	✔	✔	✔	✔	✔	✘
Duration	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘
File Selector	✔	✘(3)	✘ (3)	✘(3)	✔	✘
Numeric Keypad	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
QWERTY Keypad	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘
Rapid Text Entry	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘
Time	✔	✔	✔	✔	✘(3)	✘

Navigation Options

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Back Button	✔	✔	✔	✔	✘	✘
Client Scroll Support	✔	✔	✔	✔	⚠(7)	✘
Menu	✔	✔	✔	✔	⚠	✘
Screen Table Row Function Keys	✔	✔	✔	✔	⚠	✘(3)
Table Paging Controls	✔	✘(3)	✘ (3)	✘(3)	✘(3)	✘(3)

Other Functionality

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
About Box	✔	✔	✔	✔	✘(3)	✘
Client Connection Information	✔	✔	✔	✔	✘(3)	✘
Client Idle Timeout	✔	✔	✔	✔	✘(3)	✔

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Client File Upload Capability	✗(3)	✗(3)	✗ (3)	✗(3)	✓	✗
Client Performance Tracing	✓	✗(3)	✗ (3)	✗(3)	✗(3)	✗
Configuration Editor	✗	✓	✓	✓	✗	✗
Customizable Icons	✓	✓	✓	✓	✗(3)	✗
Field Border Colors	✓	✓	✓	✗(3)	✓	✗
Field Input Masking	✗(3)	✓	✓	✓	✗(3)	✗
Gradients	✓	✓	✓	✓	✗(3)	✗
HTTPS Client Certificates	✗(3)	✗(3)	✓	✗(3)	✗(3)	✗
Log Viewer (with Email)	✗(3)	✓	✓	✓	✗	✗
Message Viewer	✓	✗(3)	✗ (3)	✗(3)	✗(3)	✗
Orientation lock	⚠	⚠	✓	✓	✗	✗
Packaging Support	✓	✗	✗	✓	✗	✗
Partial Screen Updates	✓	✗(3)	✗ (3)	✗(3)	✗(3)	✓
Rounded Corners	⚠(8)	✓	✓	✓	✓	✗

Component	Windows	BlackBerry	iOS	Android	HTML	VT220
Screen Background Images	✓	✓	✓	✓	✓	✗
Screen Transitions	✗(3)	✗(3)	✓	✓	✗	✗
Self-Test	✓	✓	✓	✓	✗(3)	✗
Shade Area Background Images	✓	✓	✓	✓	✓	✗
Shade Area Groups	✓	✓	✓	✓	✓	✗
Shade Area Function Keys	✓	✓	✓	✓	✓	✗(3)
Sound Support	✓	✓	✓	✓	✓	✗
Status Bar Color Customization	✓	✓	✓	✓	✗(3)	✗(3)
Visible Length /Maximum Length	✓	✓	✓	✓	✓	✗(3)

4.10.3.1 Legend

✓ = Supported; ⚠ = Partial; ✗ = Not Supported

4.10.3.2 Notes

1. No colours
2. No date picker
3. Still to do
4. No cursor
5. No colors
6. No time picker
7. Need to investigate
8. Not on date or time fields

4.10.4 Unicode and Non-Unicode Support

As of the ECC releases, SAP provides a Unicode compliant kernel (620+). Customers have the option to then nominate whether they wish to operate in either full Unicode or non-Unicode modes. http://www.unicode.org is the official Unicode web site containing every thing that you wish to know about Unicode and why it is so important to global software vendors and customers. By simple definition, Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. The Unicode Standard is adopted by such industry leaders as Apple, HP, IBM, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others. The emergence of the Unicode Standard, and the availability of tools supporting it, is among the most significant recent global software technology trends. In other words, it is a language independent global data encoding standard.

All SkyMobile components are fully Unicode and/or non-Unicode compliant, meaning that full double byte Chinese; and Kanji mobile applications are supported. Sky already deployed mobile applications in Chinese for Customers and has demonstration capability for both Windows Mobile and Blackberry devices. Because of the ABAP syntax constraints and subtle differences in functionality, Sky provides two versions of its SAP add-in software; ECC for SAP Kernel release 620+ and R/3 for 4.x Kernel releases. This is to provide backward compatibility for customers, until they upgrade to the Unicode compliant releases. There are some limitations in functionality with the R/3 4.x release; are only possible in a SAP ECC environment.

The SkyMobile Access Gateway and Secure Container is fully Unicode compatible and provides language resource files for all messages, log information, and web page displays. The same Access Gateway can also inter-operate with the non-Unicode versions of SAP. You configure that you wish the Secure Container to operate in Unicode mode in its configuration file (`skymobile.cfg`). See the installation guide for more details. To use Unicode with SkyMobile, you must:

- Be on SAP kernel release 620+
- Have installed SAP with Unicode turned on
- Install the appropriate language packs for the Application Server OS
- Configure the Access Gateway or Secure Container with Unicode turned on

> ***Note:*** The SkyMobile Access Gateway and Secure Container is inter-operable with both Unicode and non-Unicode modes; it can run in non-Unicode mode against a SAP Unicode encoded environment and vice versa.

4.10.5 Multi-Language Support

Kony for SAP supports multi-language screen definitions and messages and the dynamic switching between them. This means that you can set up the same application definition so that the user language selection automatically switches all the text labels, graphics, and messages to a different language without having to define multiple application definitions.

You define language elements at the screen function level and you invoke the editor using the  icon in the toolbar from either the main hierarchy (right hand side of each function) or from the screen function builder toolbar. When you define an application, the default language is blank. You may use this option then to provide alternate values for a variety of languages. Multi-language support is provided for screen function labels, fields, default event texts and application messages.

The most important design considerations when implementing multi-language applications are:

- Allow enough visual display room to cater for the longest display
- Use application messages wherever possible in procedures and exits
- Allow enough time in the project for your translation (you can export the SkyMobile language elements to csv/excel format for external editing and then re-import). You can do this screen by screen or all screens in a application version.
- Do not translate the values of internal work variables, for example, codes and status fields.

4.10.5.1 Language Considerations

Many of Kony customers have operations in different countries and require their applications to be in different languages. The Unicode support in SkyMobile makes this possible. In most cases, the dynamic language support suffices, for example, Spanish, but for some double byte languages there are special design and implementation considerations. For example, the following screens support double byte Chinese characters:

Considerations above and beyond the usual mobile application requirements

- What application components need to be translated (screen names, text fields, menu items, pop-up messages, and buttons)?
- What devices does the mobile application deploys onto and are specific version of the device &/or its operating system required to display the foreign language?
- Does the application require entry of text in the foreign language and if so how is this achieved? For example, use of CE-STAR
- Who provides support for the application (and devices running them) and what language(s) are these individuals fluent in?

For Example, Implementing Support for Chinese

- SAP system must run in Unicode (ECC47+).
- SAP Chinese (ZH) language pack imported

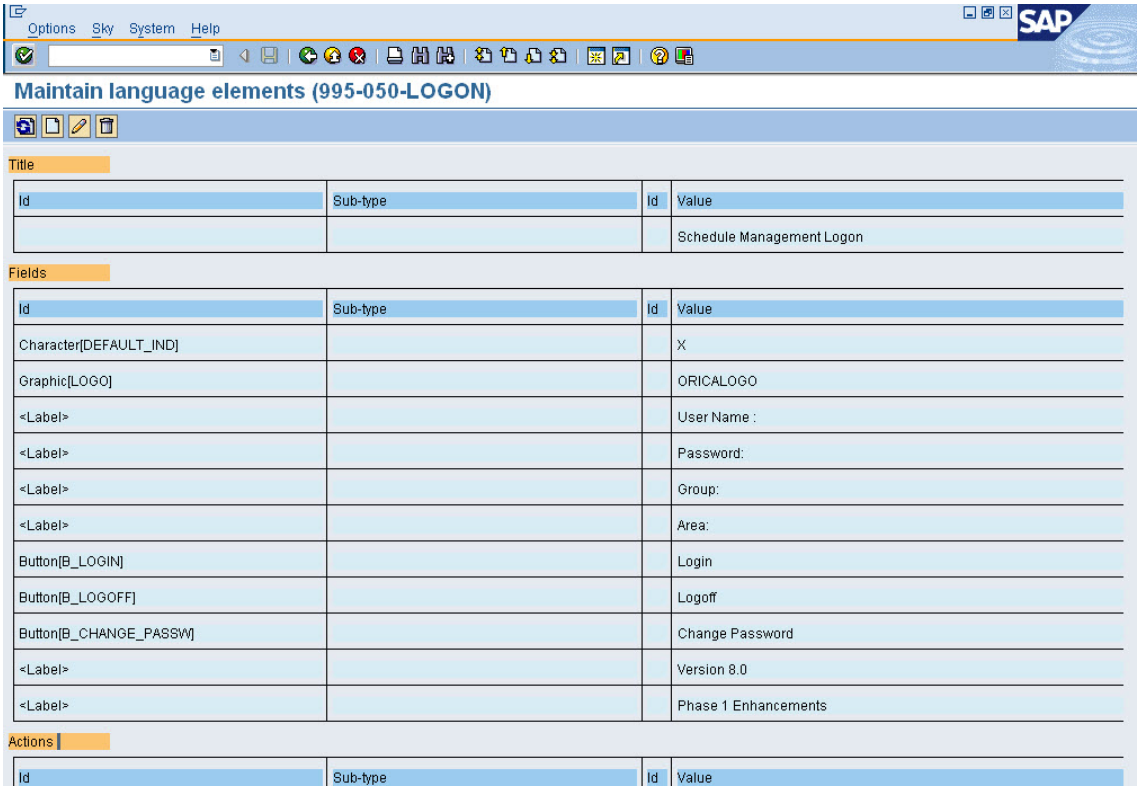
- SkyMobile Chinese properties files (VTI and WPC)
- Chinese (PRC) language pack for Windows
- SAPGUI configured for Unicode and Multi byte language input
- Language support for the mobile device (for example, CE-STAR)

Further Information:

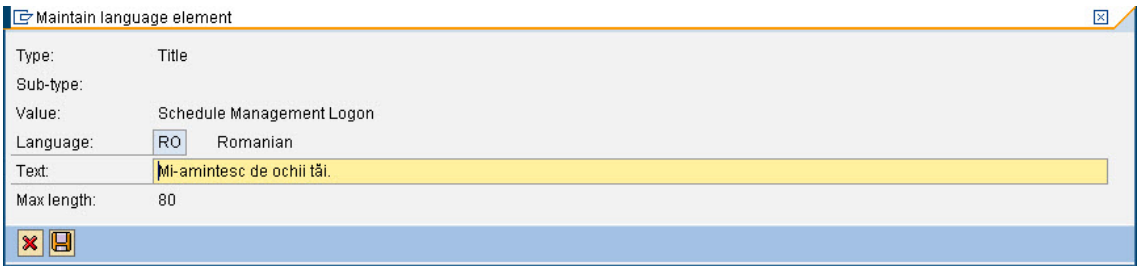
Installing Chinese Language support on Windows:	http://www.chinese-tools.com/resources/windows-xp.html

4.10.5.2 Language Element Editor

The editor is for one or more screen functions and you can filter the list based on language, field, and element type through an initial selection screen. The language elements are broken down into sections and object types, for example, the title, fields, actions, and local database covering all the applicable areas. As well as standard fields, you may also specify graphics, button text, and links. The main thing to remember is that the target area must be large enough to contain the visual display length of all the language elements. You may create and maintain language elements directly against the screen elements using the tool bar options, or you may export the selected list to a CSV (comma delimited) file that you may edit externally and then re-import.

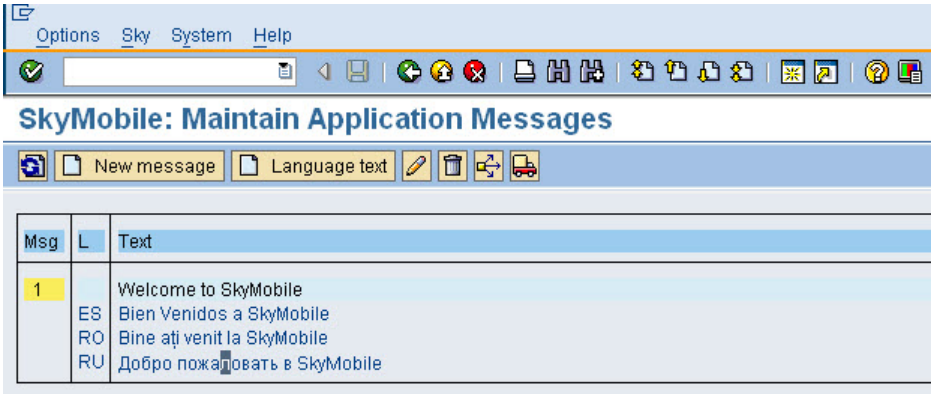


To maintain the language elements, simply position the cursor and select create, change or delete from the toolbar. You may simply change existing values by double-clicking them. You use the following pop-up to maintain the values, displaying the original value and maximum length of the element.

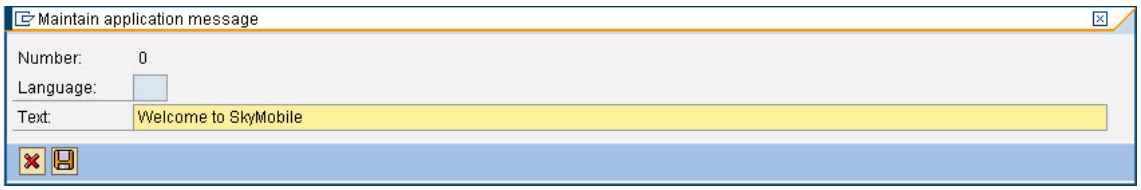


4.10.5.3 Language Message Editor

You may launch the Application message editor through the procedure editor (double-clicking a message number) or through the utilities menu on the main screen function builder screen. All the messages associated with the application version appear. From here, you may maintain the message definition and any associated language text.



To maintain the message texts, simply position the cursor and select create, change or delete from the toolbar. You may simply change existing values by double-clicking them. You use the following pop-up to maintain the values:



4.10.6 Multi-Tenant Support

SkyMobile supports the concept of tenants to segregate application data; you may share the same application definition between multiple logical tenants identified by a unique five digit number (Tenant ID). This option is primarily used for SaaS (Cloud based - Software as a Service) implementations. You maintain tenant entries using the User Profiler. In the case of most on-premise implementations, the default public tenant (0) is used. As well as at the application level, you can also use tenants to segregate data at the user level; assign a unique tenant to each user.

4.10.7 Look and Feel

You may use the following techniques to improve the look and feel of SkyMobile applications.

4.10.7.1 Colours and Gradients

You may assign different foreground and background colours most screen fields and objects through the use of styles. However sub-screens, tables, shade areas, drawing objects, charts and the screen functions themselves may have their background defined as either a colour, a gradient, or an image. You can use the shade areas in particular to group and highlight one or more fields. You define colours through the workbench colour definition manager and you can define with an optional gradient colour that is used when a gradient is specified. Gradients are linear and can be vertical, horizontal or diagonal in either direction.

4.10.7.2 Field Styles

The difference between a colour and a style is that a style incorporates a background and foreground colour scheme, bold, italic, and underline. The concept is to define standard styles that you use to provide a consistent user interface, for example, column heading, and text descriptions. Each style is given a unique number for easy reference. You may dynamically alter styles by exit programs (see the ABAP and Java programming guides). This is useful to emphasize data values, for example, status CRITICAL highlighted in red.

4.10.7.3 Screen Orientation

By default, smart phone platforms such as IOS and Android perform portrait and landscape screen orientation automatically when the device is tilted or rotated. You may however, specify the orientation at the application version and screen function levels or dynamically change it using a procedure command at run time. This may be useful to fix the display irrespective of how the device is held, for example, fix the orientation to landscape for a better report list view.

4.10.7.4 Transitions

Screen transitions control the visual behaviour when migrating from one screen to another within a application. This concept was introduced by the new IOS and Android smart phone platforms, for example, fly in the next screen from left-to-right of the current screen. SkyMobile allows you to manipulate or even turn off screen transition processing at the application version or function levels and even dynamically by procedure command.

4.10.7.5 Using Graphics

"Active" graphic objects are supported; you can press or click them; and you may dynamically specify them at run time through a database table or exit program. Because of this flexibility, graphics are extremely useful as navigation and input aids; a picture tells a thousand words.

For maximum effect, you can design a consistent suite of graphics, for example, company logo, back, next, up, down, save, and select. Usually you need a professional graphics package such as Paint shop Pro or you can create basic graphics using Microsoft PowerPoint and other basic drawing tools. You must load graphics into the Binary Object Manager (see the section in this manual).

As you are probably aware, there are many aspects to designing and using graphical images as per the most appropriate type, aspect ratio, and scaling factors. Your company may have already invested in standard graphics for marketing, web sites and other applications that you may leverage from. You may also have access to a specialist graphical artist who can assist. Another approach is to purchase or download free graphics from the internet.

4.10.8 Sounds

You may configure sounds to play when certain events occur, such as when:

- The application first starts (start up)
- The application shuts down (logoff)
- A screen first appears (formatted)
- An error (or no error) occurs on a screen operation
- A screen event occurs, such as clicking on a button.

You may specify the default sounds to use at the application version level. These defaults are then inherited whenever a new screen function or event is created. Sounds are stored as SkyMobile binary objects in the same way as graphics and so you may import your own custom sounds to use with the application using the Binary Object Workbench. You configure the screen start sound on the screen function attributes and start, success and error sounds on the screen event definition.

Sounds can be a very effective way to improve the user interface by helping to highlight when events occur, for example, an error, or even to deploy interactive voice help.

4.10.9 Integration

There are a range of integration options available to communicate with serial devices, trigger SAP transactions, synchronize database data, and manage attachments, external and third party applications. The aim of this section is to provide a summary of some of the major options available and where you can find out more details.

Area	Description
Data Synchronisation	Sky recommends that wherever possible: • Use data objects. Refer to the main section on Data objects. Data objects: ◦ Guarantee that the entire "transaction" is synchronized. ◦ Enable Data profiling to distribute data to the correct servers. ◦ Automatically trigger SkyConnect ECS interfaces when data is uploaded. • Use data profiling. Refer to the section on Data profiling. • Use heartbeat "push" commands from SAP to trigger refresh synchronization on an 'as required' basis instead of repeatedly polling the SAP host to pull any updates down. Refer to the ABAP programming guide or Data profiling for more details. • Use the primary key traversal LDB mechanism for all medium to high volume data download and refresh operations.

Area	Description
ECS interfaces	You may trigger ECS interfaces from data object (DOB) and local database (LDB) uploads, from procedures, or ABAP exits. Either way, Sky recommends that you use ECS to manage all interfaces. An important consideration when starting interfaces is what SAP user they run under. ECS has some strict rules on what userid is used and when. You may also dynamically switch the SAP user under certain conditions. You may dynamically switch SAP users at run time to ensure that the transaction is executed under the correct SAP user. ECS interface performance is particularly important to avoid bottlenecks and SAP resource issues.
SAP cross system services	If you need to integrate multiple SkyMobile applications across distributed SAP systems, for example, ERP <-> CRM, then you may want to consider using the SkyMobile XAI SAP support that enables you to nominate a "master" SAP MEAP that then automatically re-routes requests to other remote SAP systems.
Executing SAP business transactions directly	You may invoke SAP interface transactions directly from ABAP screen exits in a real time scenario (for example, BAPI call), but Sky recommends that you perform as much validation as required and then trigger SkyConnect ECS interfaces to execute asynchronously. This approach frees up the mobile user to continue with other tasks, is much more conducive to support offline processing and provides more management controls.
Using Attachments	The Binary Object Repository supports a full range of options to store various file types, manage the synchronization with Application Servers and interface with standard SAP sub-systems such as GOS (Generic Object Sub-system). Refer to Binary Objects in this guide and the procedure and ABAP programming guides for more information. As well as the built-in binary procedure commands, the Standard Object Framework (SOF) has options available to launch and load attachments from screen functions. You may either use these standard mechanisms or code your own using the SkyMobile Java SDK. Refer to the main section on Standard Object Framework for more details.

Area	Description
Devices and printers	Refer to SkyMobile Java programming guide for the options available to integrate with serially connected devices, sockets interfaces, and printers.
File Polling	The SkyMobile Application Server is capable of polling for and processing input files. See the SkyMobile Java programming guide for the options available.
External system and 3rd party applications	The SkyMobile XAI (external application integration) functionality enables bi-directional interfacing between SkyMobile and external applications, for example, the launching of a SkyMobile screen from a email link and positioning on a specific transaction; launching of a GIS viewer from a SkyMobile application screen. Refer to the SkyMobile XAI guide for more information. Standard SOF objects are also provided to seamlessly integrate SkyMobile applications with native applications such as email, office tools, contacts, and calendar. See the section on SOF workbench for more details.

4.10.9.1 Interface Performance

Throughput of SAP ECS interfaces is vital to prevent bottlenecks and ride "transaction storms". More information is available in the ECS interfacing documentation. The following are some design recommendations:

- Keep your interfaces as modular (multi-phase) as possible, the shorter and sharper the better
- Utilize ECS background server processing as much as possible for short sharp transactions with queue priority. Kick off long running interfaces in job mode. Also investigate using the parallel processing option to maximize concurrency (there are SAP resource considerations for using this).
- Tune your ABAP programs to be as efficient as possible, particularly SQL accesses. Consider using a persistent cache with a timestamp for data that is always read, but rarely changes.

4.10.9.2 SAP User Switching at Run Time

ECS supports the dynamic switching of SAP users; execute the transaction under another user. For example, you may receive transactions in SAP under the SkyMobile Access Gateway logon userid, for example, CPIC_VTI or SERVICE_VTI or directed to execute under a ECS background server. If you wish to change the user to another, or the current logged on user, you must use the ECS switch user functionality. This may be done automatically on data object uploads by configuring the userid and password fields to '*' that automatically defaults the user to the current logged on user or you may use '&' substitution to dynamically substitute run time values, for exampel, &P1. You may also use procedure commands or the VTI/ECS SKD commands in ABAP to achieve the same thing.

> ***Important:*** In order to switch SAP users at run time, the "current: user must be configured as a SAP service type user; the SkyMobile Access Gateway or ECS back ground server user.

4.10.10 Multi-Device and Multi-User Scenarios

By default, SkyMobile expects a one-to-one relationship with a user and device and automatically maintains the applications and database on the device accordingly. If you have cases where multiple users share a device or a user has multiple devices, then you need to consider the following configuration options:

4.10.10.1 Server Profile Multi-User Option

On the server profile header definition, there is a Provisioned containers support multiple users (default single user) option. You need to check this if multiple users share the same device so that SkyMobile automatically removes applications and databases tables because they are deemed are no longer used.

4.10.10.2 User Timestamp Tracking

If multiple users share the same device and they have different data requirements, then you should enable "User level timestamp tracking" functionality on data profile header. There are several options that govern what you should track, but essentially, the users last recognized timestamp is automatically defaulted in SAP.

4.10.10.3 Single User with Multiple Devices

SkyMobile should handle situations where users have multiple devices, for example, smart phone and a tablet without any special configurations. The considerations here come down to how data is synchronized, for example, does the same data get replicated to both devices, or does just the latest information get refreshed. These are more application design considerations rather than SkyMobile restrictions. Kony recommends that you fully use the server, data and user profiling options when provisioning applications and data. These options provide in-built support for multi-device/user options.

4.11 Shared Objects, Tools and SDKs

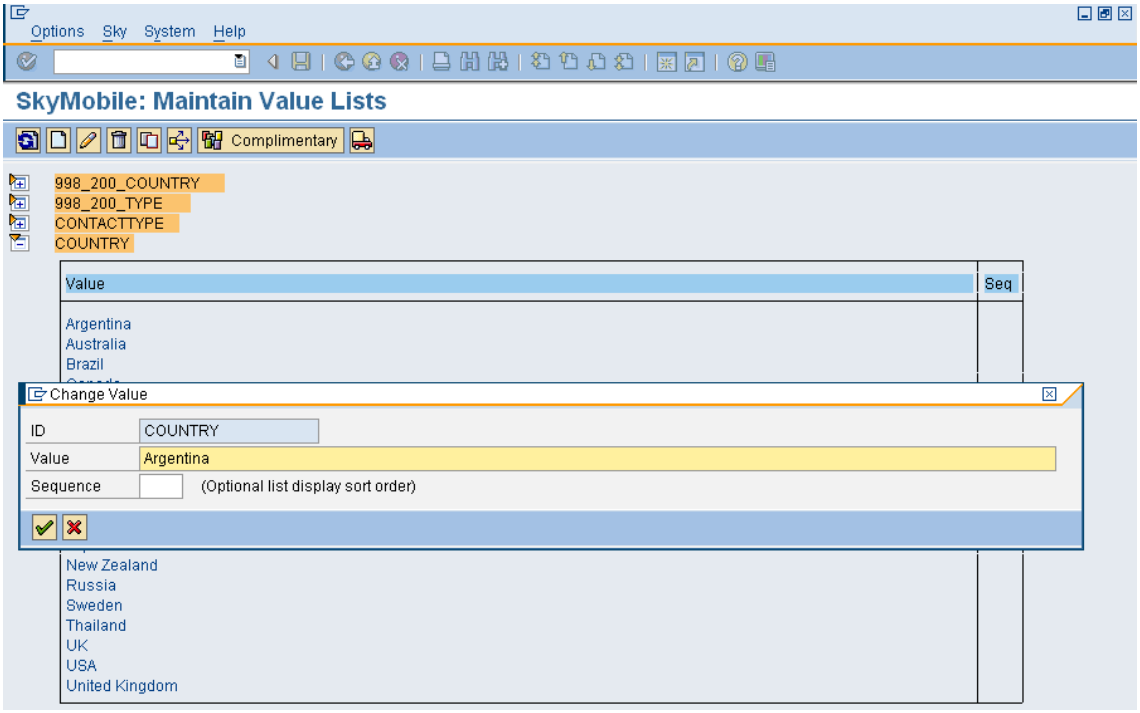
As well as the standard workbench definitions, there are other object types that you may associate with one or more applications, for example, number ranges, value lists, colours, styles, parameter values, and programming exits. This section describes all the development tools available to implement and manage a variety of shared object types and definitions.

4.11.1 Key Topics

Value Lists
Number Ranges
Colour Definitions
Styles
Parameter Values
Complimentary Objects
Generic Fields
Clipboard Manager
Standard Object Framework
ABAP Programming Guide
Java Programming Guide

4.11.2 Value Lists

You may define a static range of values to use in screen field value checking. You may create and maintain value lists from the utilities menu in the workbench or screen painter or through transaction YVTV. You use the standard icon from the application toolbar to invoke the value list manager.



You use create, change, delete and copy icons on the application toolbar to maintain value list definitions. Position your cursor on the level or name you require and click the icon. For instance, to create a new value list, position the cursor in the header section and click the create icon. To create a value, position the cursor on the relevant value list and click the create icon. You may order values using the optional sequence number.

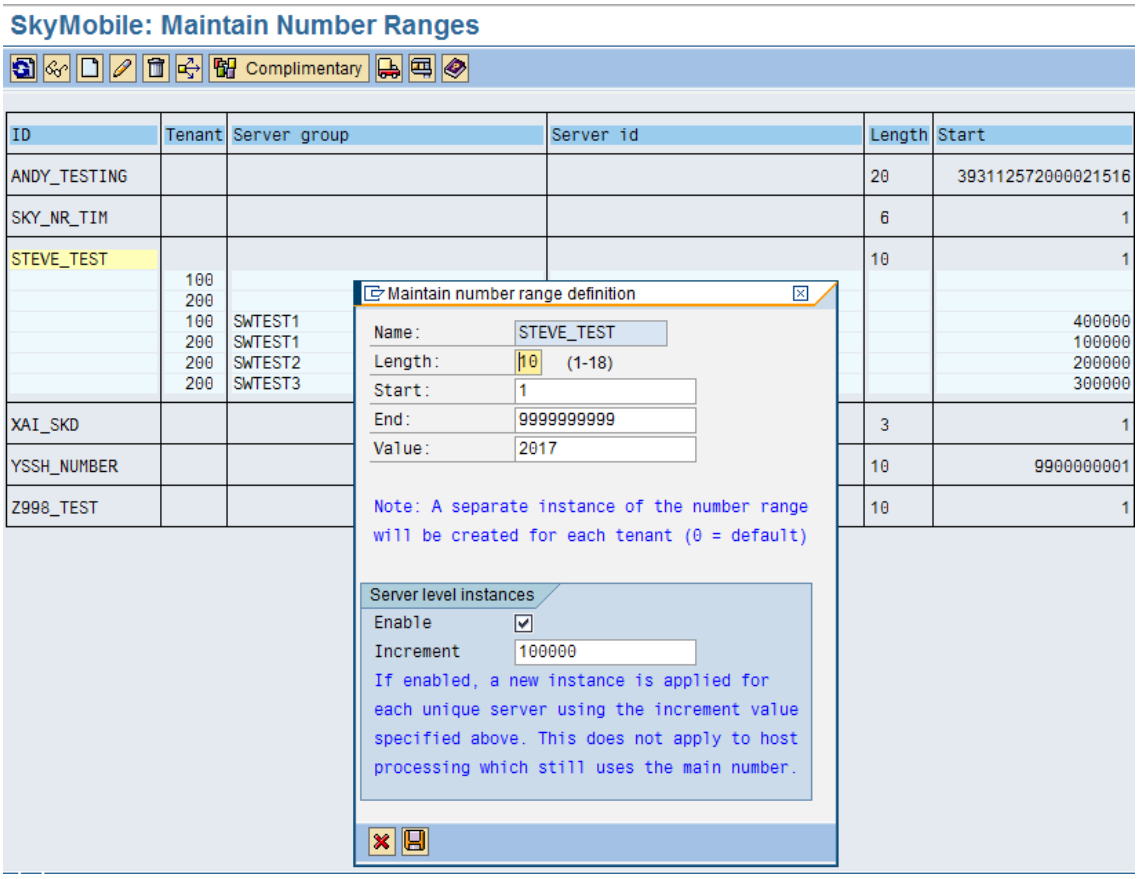
Value list entries use a timestamp and logical deletion indicator to replicate changes as per a normal LDB definition (see "VTI_VALUE_LIST"). By default, a value list refresh is configured to run every two hours. You may customise this as required.

> ***Important:*** Value lists are stored in a local database definition VTI_VALUE_LIST. If you are use the default in-built 'file system' database, and you have values that exceed 40 characters in length, you can get insert errors from the Application Server because of the excessive file name length generated. Avoid this scenario by either using a database or values up to 40 characters.

You may transport value list definitions to remote SAP systems as components of an application using the transport facility from the workbench.

4.11.3 Number Ranges

A number range object is used to automatically retrieve the next unique number, for example, a order number. It is only used in local database (LDB) operations, procedures and programming exits. You can have number ranges that are unique within each SkyMobile tenant or within each server instance. Number range definitions are maintained from the utilities menu or through transaction YVTN. A selection screen allows you to restrict the display. The main screen lists the selected number range definition and allows you to maintain them and even transport them across environments.



Use Create, change, delete and copy icons on the application toolbar to maintain number range definitions. Position your cursor on the level or name you require and click on the icon. For instance, to create a new number range, position the cursor in the header section and click the create icon. To

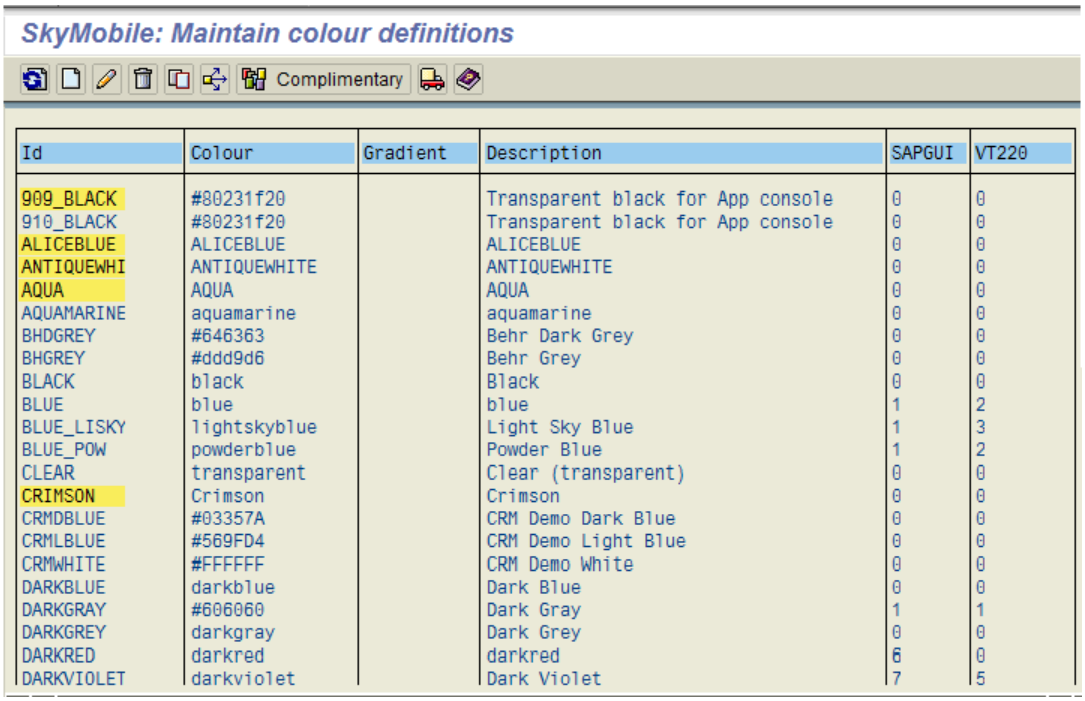
change an existing number range, position the cursor on the name and click the change icon. Number ranges are only implemented by Secure Containers on the device if they don't yet exist, therefore if you change a definition, you have to either re-provision or force a download of VTI_NUMBER_ RANGE.

Option	Description
Name	The unique name of the number range across the whole system
Length	The maximum length of the number from 1 to 18 digits
Start	The minimum start value
End	The maximum end value
Value	The current value
Server level instances	If enabled, a unique number range is automatically generated for every unique server group/id that accesses the system. In this way, you can guarantee unique local numbers. You simply configure the increment to apply in each case, for example, in blocks of 1,000,000,000.

You may transport number range definitions to remote SAP systems as components of a SkyMobile application using the transport facility from the workbench. There are controls to define number range controls at the LDB field attribute level. You may use these to store the last known value or centrally renumber a value uploaded from a Application Server. In addition, Procedures, ABAP and Java exit programs may manipulate number ranges directly. See the relevant programming guides for details.

4.11.4 Colour Definitions

You use colour definitions to create and maintain all colour definitions in the system. You may configure your own custom colour definitions. You may use a standard icon: on the application toolbar to invoke the style manager. Alternatively, you can run transaction YVTH. All the available styles then appear. Colours that an application unreferences are highlighted.



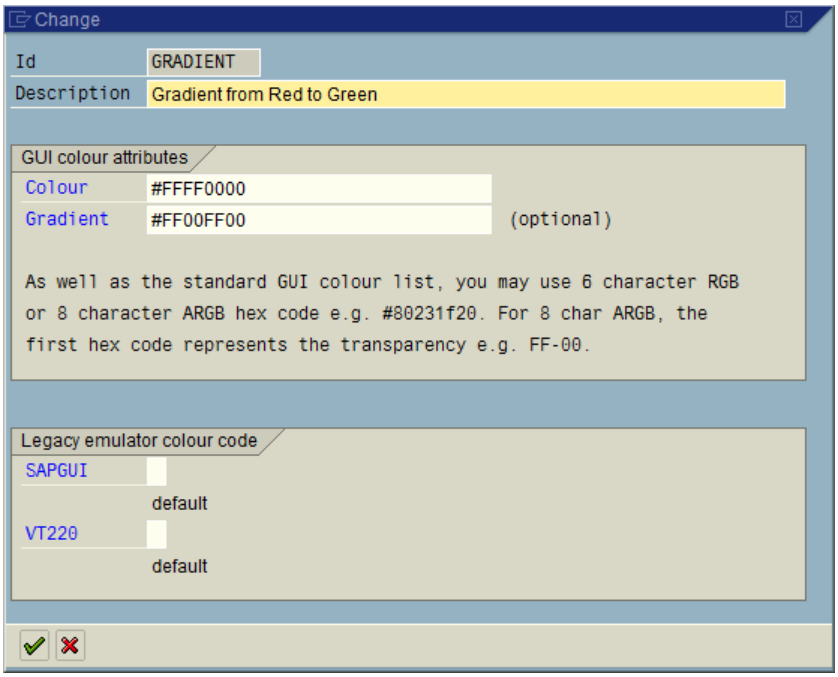
Id	Colour	Gradient	Description	SAPGUI	VT220
909_BLACK	#80231f20		Transparent black for App console	0	0
910_BLACK	#80231f20		Transparent black for App console	0	0
ALICEBLUE	ALICEBLUE		ALICEBLUE	0	0
ANTIQUEWHI	ANTIQUEWHITE		ANTIQUEWHITE	0	0
AQUA	AQUA		AQUA	0	0
AQUAMARINE	aquamarine		aquamarine	0	0
BHDGREY	#646363		Behr Dark Grey	0	0
BHGREY	#ddd9d6		Behr Grey	0	0
BLACK	black		Black	0	0
BLUE	blue		blue	1	2
BLUE_LISKY	lightskyblue		Light Sky Blue	1	3
BLUE_POW	powderblue		Powder Blue	1	2
CLEAR	transparent		Clear (transparent)	0	0
CRIMSON	Crimson		Crimson	0	0
CRMDBLUE	#03357A		CRM Demo Dark Blue	0	0
CRMLBLUE	#569FD4		CRM Demo Light Blue	0	0
CRMWHITE	#FFFFFF		CRM Demo White	0	0
DARKBLUE	darkblue		Dark Blue	0	0
DARKGRAY	#606060		Dark Gray	1	1
DARKGREY	darkgray		Dark Grey	0	0
DARKRED	darkred		darkred	6	0
DARKVIOLET	darkviolet		Dark Violet	7	5

You may create, change, delete or copy colours using the application toolbar. The pop-up then prompts you for the colour attributes. You may specify colours in one of the following ways:

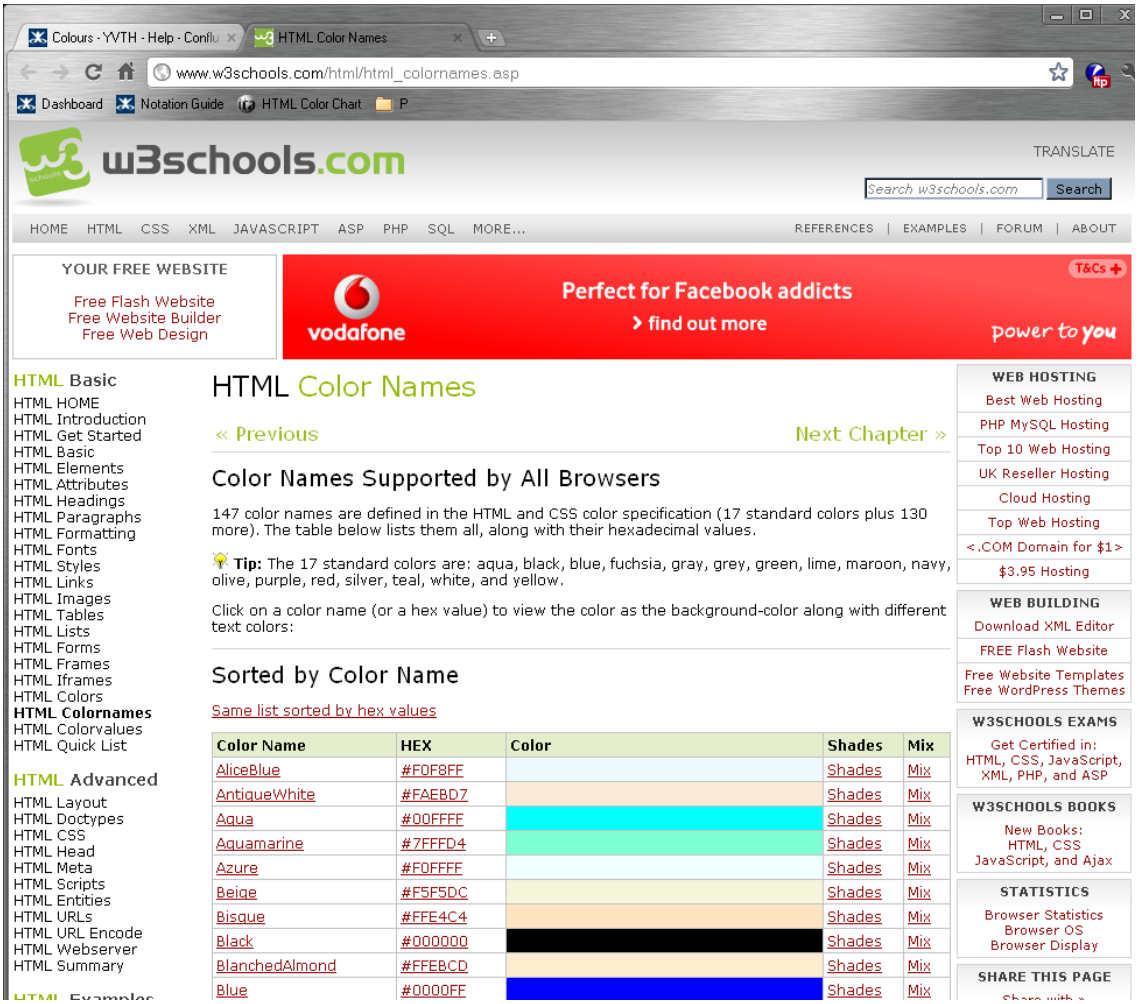
1. As an ANSI colour name; "red", "blue" or "forestgreen".
2. As an RGB colour value. You can specify any colour using its red, green and blue components. You must specify each component in hex and enter in the following format: #RRGGBB.
3. As an ARGB colour value. Similar to RGB except an additionally transparency component (alpha) is specified where 00 is transparent and FF is completely opaque: #AARRGGBB.

You may also specify a second colour to use when generating gradients. This is optional, however if you do not specify, then no gradient appears.

Separate entries for the different user interface modes appear, where applicable, for example, VT220 has a restricted colour pallet in comparison to the Windows presentation client (WPC). Use the relevant drop-downs to display what options are available. Leave attributes blank to use the system defaults.

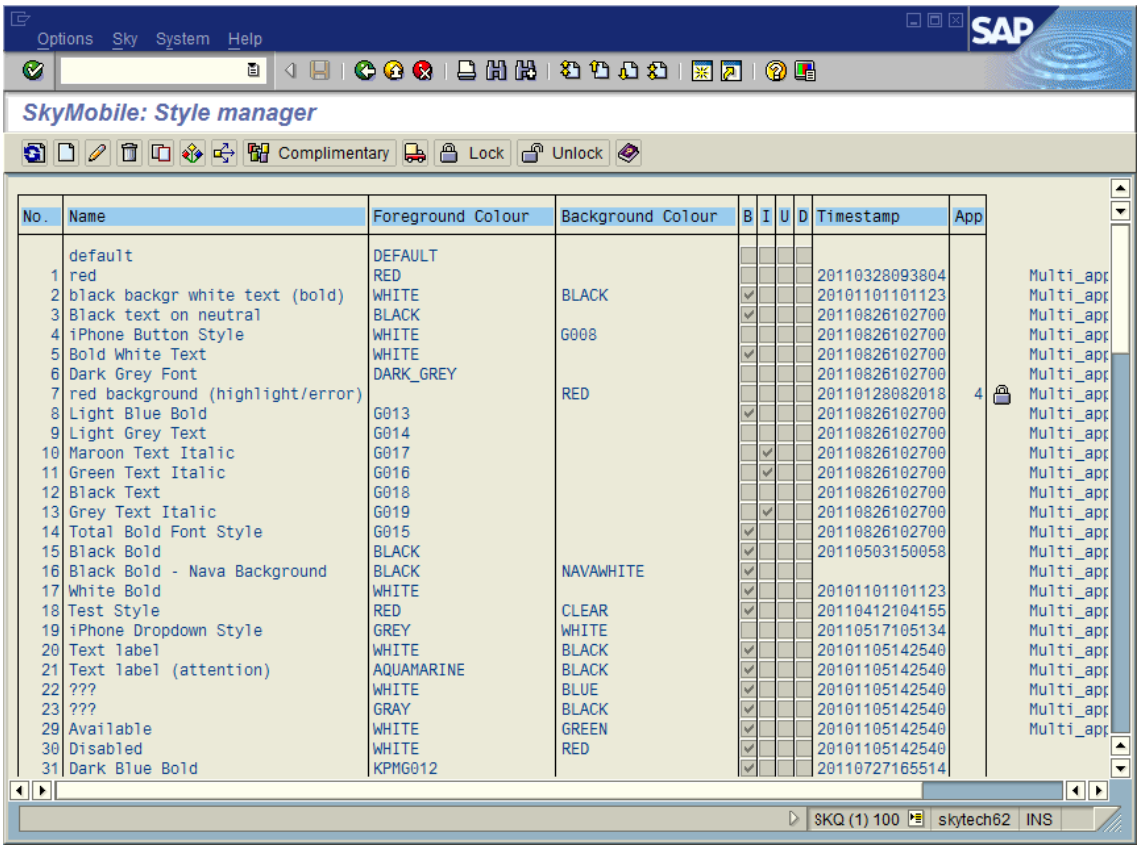


> ***Note:*** The Check Installation procedure automatically creates all colour codes (refer to the installation section). There are many pallets available on the internet, for example, http://www.w3schools.com/html/html_colornames.asp.



4.11.5 Styles

You may use styles may to change the look and feel of data on screens, for example, foreground/background colour, bold, and underline. You may effectively use them to ensure consistency on screens, for example, heading, and column heading. You may configure styles for field definitions and other screen elements such as radio buttons. You may invoke the style manager from multiple points; directly from the workbench, from the screen painter, from the field definition screen or through transaction YVTF. You may use a standard icon: on the application toolbar to invoke the style manager. All the available styles then appear.

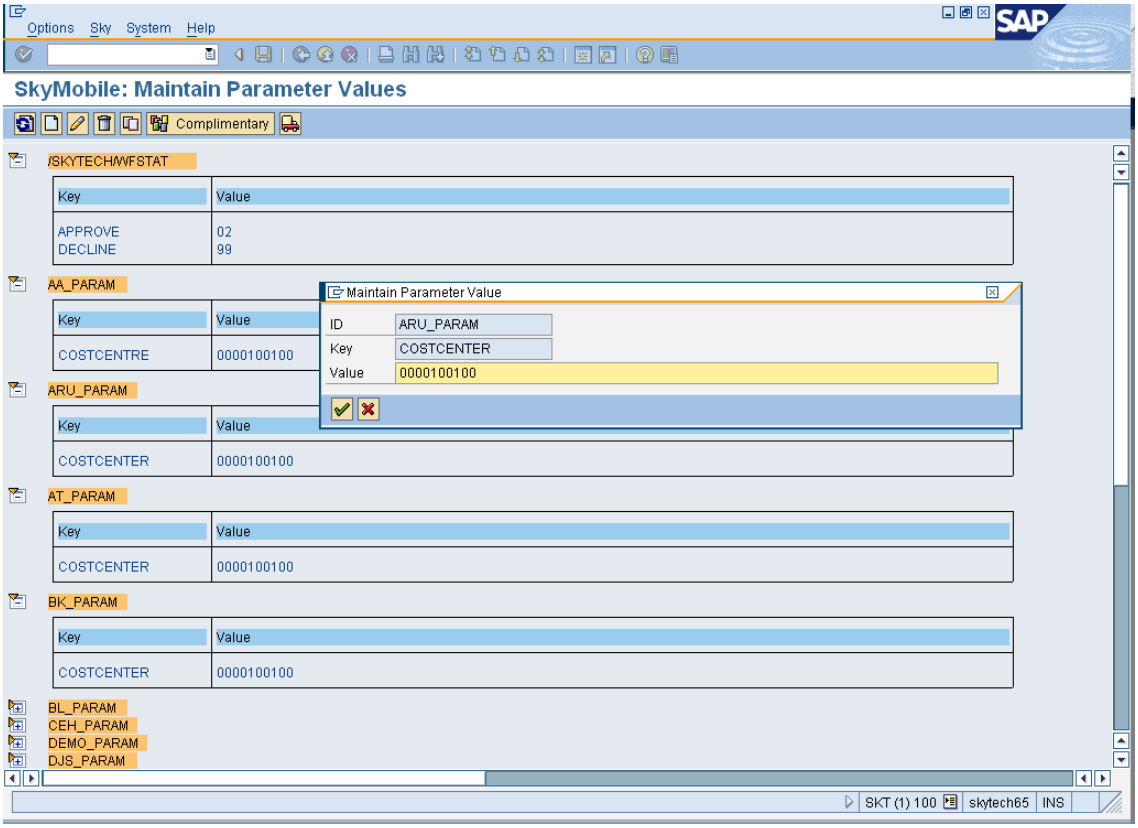


No.	Name	Foreground Colour	Background Colour	B	I	U	D	Timestamp	App	
	default	DEFAULT								
1	red	RED						20110328093804		Multi_app
2	black backgr white text (bold)	WHITE	BLACK	✓				20101101101123		Multi_app
3	Black text on neutral	BLACK		✓				20110826102700		Multi_app
4	iPhone Button Style	WHITE	G008					20110826102700		Multi_app
5	Bold White Text	WHITE		✓				20110826102700		Multi_app
6	Dark Grey Font	DARK_GREY						20110826102700		Multi_app
7	red background (highlight/error)		RED					20110128082018	4	Multi_app
8	Light Blue Bold	G013		✓				20110826102700		Multi_app
9	Light Grey Text	G014						20110826102700		Multi_app
10	Maroon Text Italic	G017			✓			20110826102700		Multi_app
11	Green Text Italic	G016			✓			20110826102700		Multi_app
12	Black Text	G018						20110826102700		Multi_app
13	Grey Text Italic	G019			✓			20110826102700		Multi_app
14	Total Bold Font Style	G015		✓				20110826102700		Multi_app
15	Black Bold	BLACK		✓				20110503150058		Multi_app
16	Black Bold - Nava Background	BLACK	NAVAWHITE	✓						Multi_app
17	White Bold	WHITE		✓				20101101101123		Multi_app
18	Test Style	RED	CLEAR	✓				20110412104155		Multi_app
19	iPhone Dropdown Style	GREY	WHITE					20110517105134		Multi_app
20	Text label	WHITE	BLACK	✓				20101105142540		Multi_app
21	Text label (attention)	AQUAMARINE	BLACK	✓				20101105142540		Multi_app
22	???	WHITE	BLUE	✓				20101105142540		Multi_app
23	???	GRAY	BLACK	✓				20101105142540		Multi_app
29	Available	WHITE	GREEN	✓				20101105142540		Multi_app
30	Disabled	WHITE	RED	✓				20101105142540		
31	Dark Blue Bold	KPMG012		✓				20110727165514		

You may create, change, delete or copy styles using the application toolbar. The pop-up then prompts you for the style attributes. You may then specify colour and basic font controls. Styles are very useful to improve the user interface by highlighting and differentiating areas on the screen. For example:

4.11.6 Parameter Values

Do not get confused with run time field parameter ids. Parameter values provide a flexible utility to configure values that you can easily reference directly in screens and/or in programming exits. This is primarily to avoid hard coding data and enable systems to be more easily configurable. You use a standard icon Parameter Manager on the application toolbar to indicate the parameter value manager. Alternatively, you can call it through transaction YVT2.

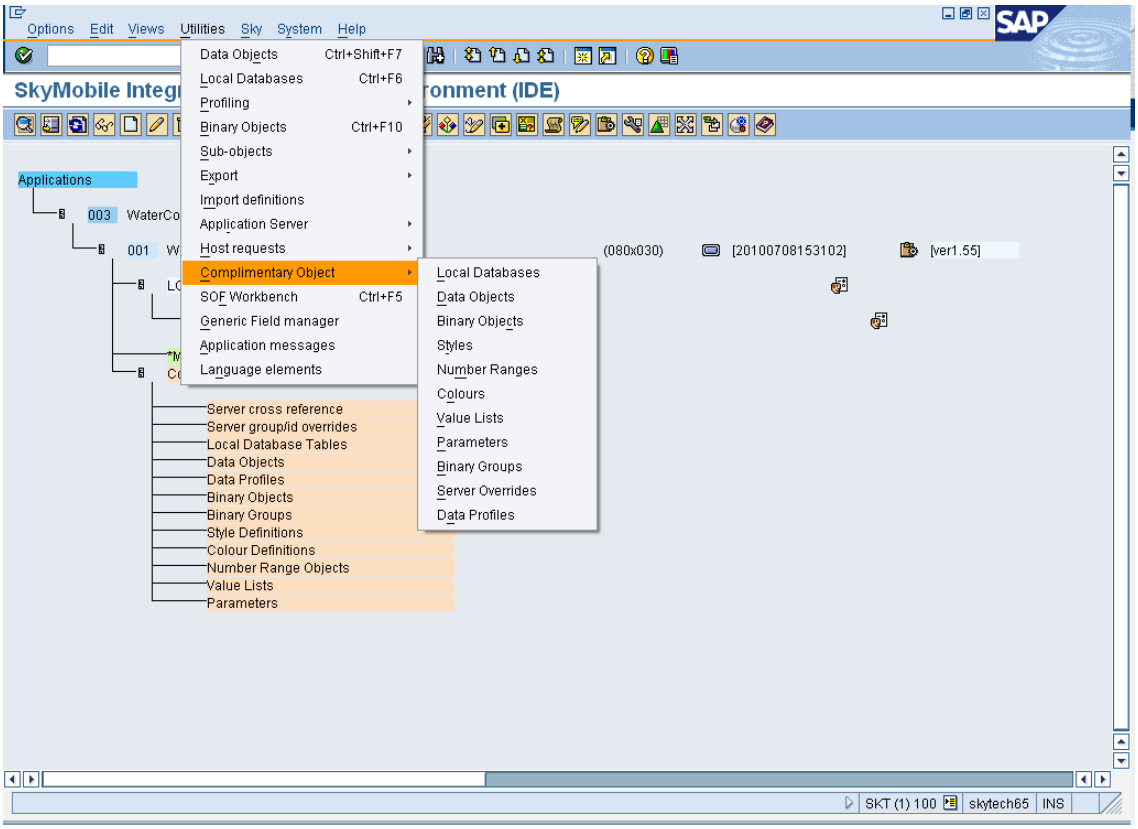


Each parameter value has a high level id, for example, sales configuration. It then has one or more key/value pairs associated with it. You may default screen fields using the '=(aaaa.bbbb)' convention where "aaaa" is the parameter id and "bbbb" is the key.

> ***Note:*** Once created, you may default parameter values into screens using the '=(section.key)' convention, or through procedure get/set parameter commands.

4.11.7 Complimentary Objects

The workbench only knows to download those LDBs, binaries, and styles that are statically referenced in screen functions. If you require additional objects, for example, dynamically assigned or external objects, you must assign them as a complimentary object to the application version. You select these options from the main workbench **Utility->Complimentary Objects** menu or directly from the **Complimentary** nodes on the workbench hierarchy. You automatically maintain the cross references for rename, copy or delete.



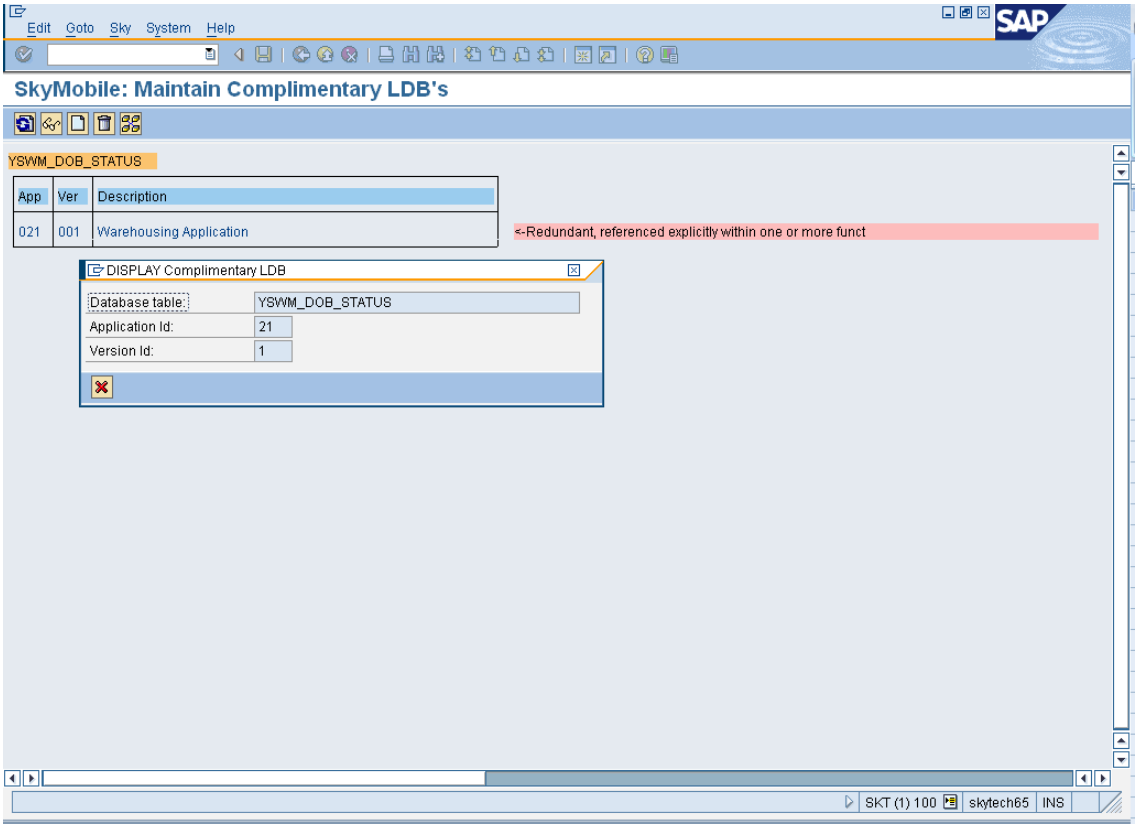
4.11.7.1 Types of Complimentary Objects

- LDB database tables
- Data objects (DOBs)
- Data profiles

- Binary objects
- Binary groups
- Styles
- Colours
- Number ranges
- Value lists
- Parameter values

4.11.7.2 Complimentary LDBs

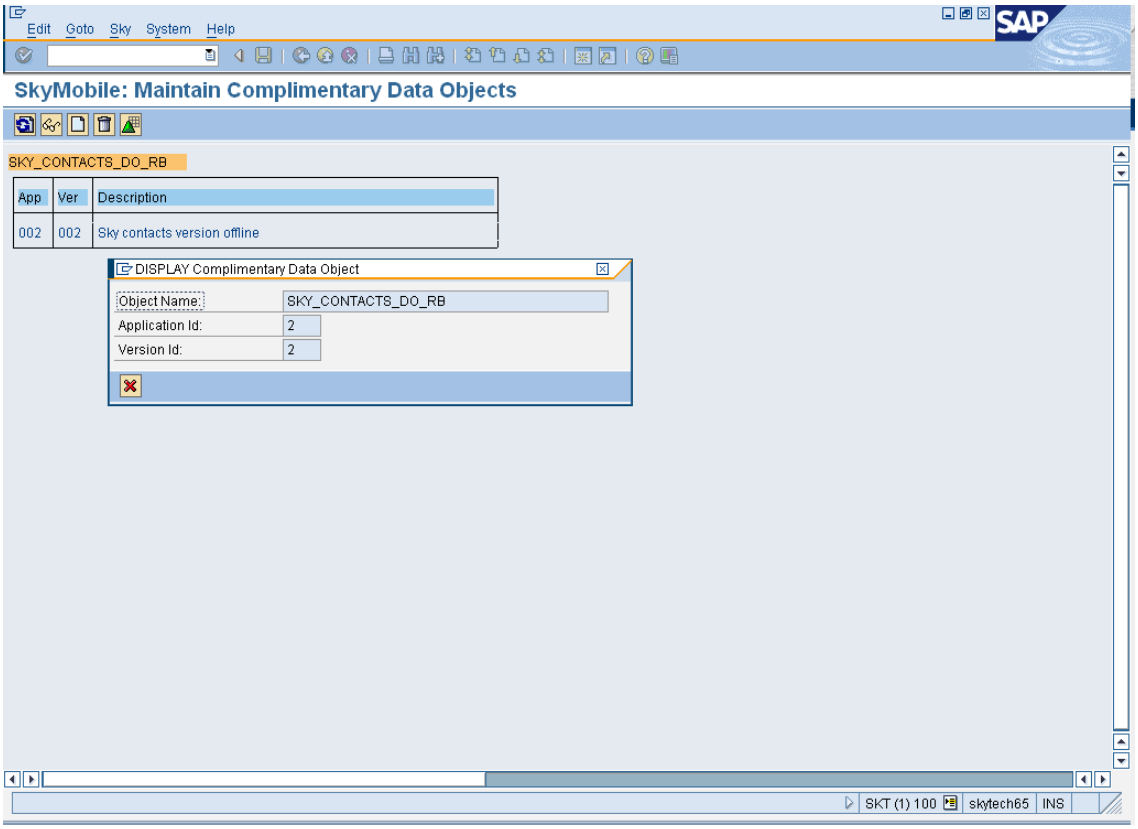
You use this option to specify additional database table definitions that you may download to a specific Application Server. By default, Application Servers automatically download all database tables used in screen functions. Sometimes additional definitions,for example, you need to download work tables as well, but are not directly referenced in screen functions. You do this by specifying a cross reference of application versions and database table names. You may use a selection screen to restrict the cross reference display.



You may create and remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option. The workbench highlights entries that are directly referenced in screens and thus do not need to directly cross reference.

4.11.7.3 Complimentary DOBs

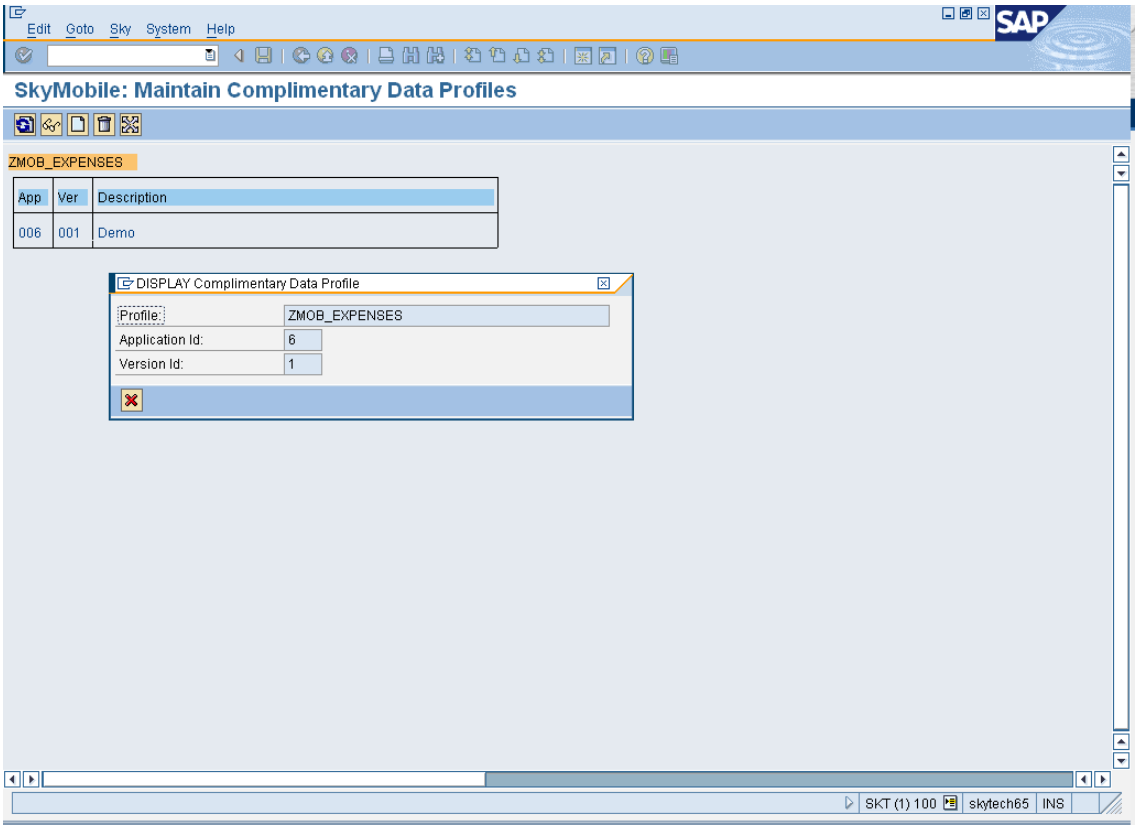
This option is used to specify data objects that are associated with a specific application, so that the definitions are downloaded to subscribing Application Servers. A selection screen may be used to restrict the cross reference display. Data objects are not directly associated with applications in the same way that LDB definitions are, so complimentary references are essential to include them in the application schema.



Entries may be created and removed by positioning your cursor on the appropriate level and clicking the application toolbar option.

4.11.7.4 Complimentary Data Profiles

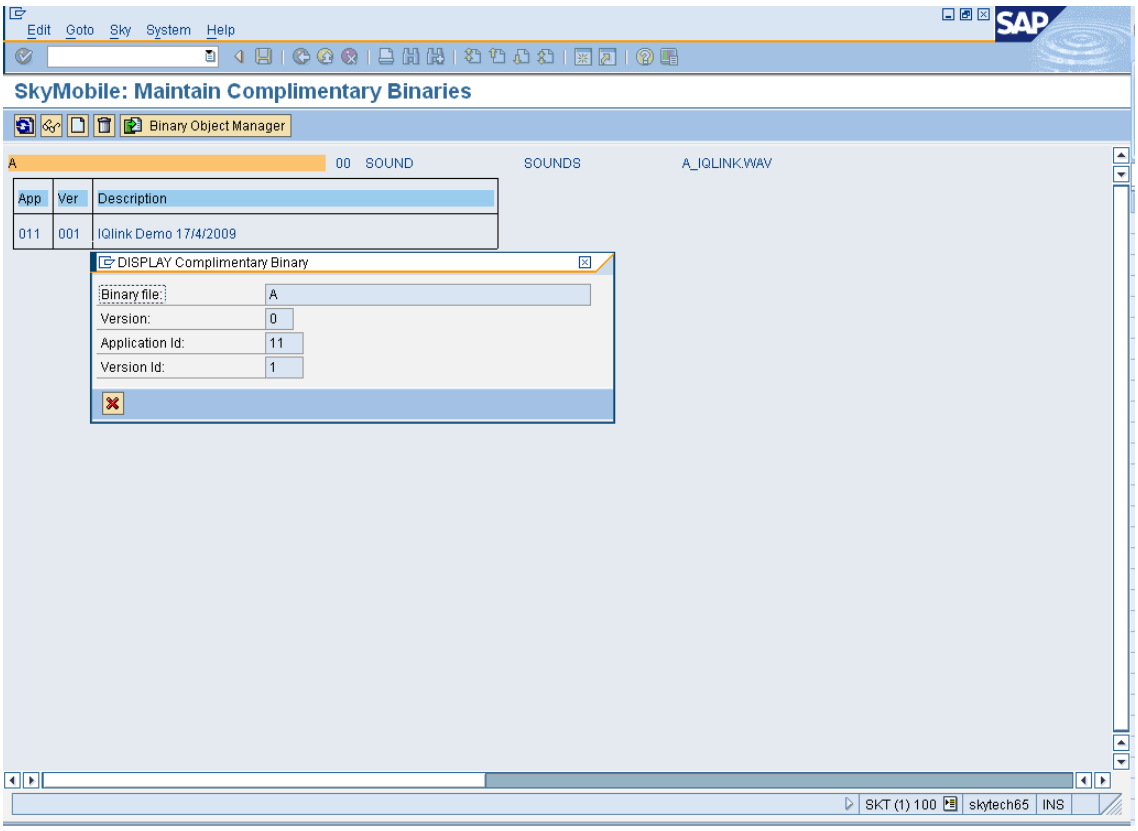
You use this option to specify data profiles that are associated with a specific application, so that you download the definitions to subscribing Application Servers. You may use a selection screen to restrict the cross reference display. Data profiles are not directly associated with applications in the same way that LDB definitions are, so complimentary references are essential to include them in the application schema.



You may create and remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option.

4.11.7.5 Complimentary Binary Objects

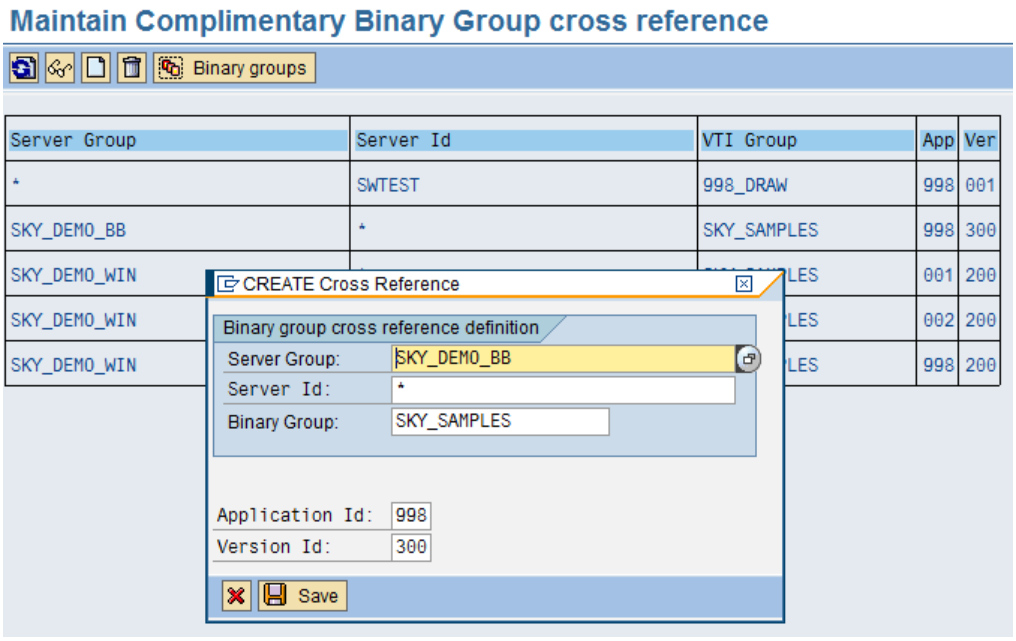
You use this option to specify binary object definitions that are cross referenced to a specific application version. By default, the Secure Container automatically downloads all binaries that are directly referenced in functions and procedures. Sometimes you need to download additional binaries that are not directly referenced, for example, in an external programming exit. You may use a selection screen to restrict the cross reference display.



You may create or remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option. The workbench highlights entries that are directly referenced in functions and procedures and thus do not need an entry here.

4.11.7.6 Complimentary Binary Groups

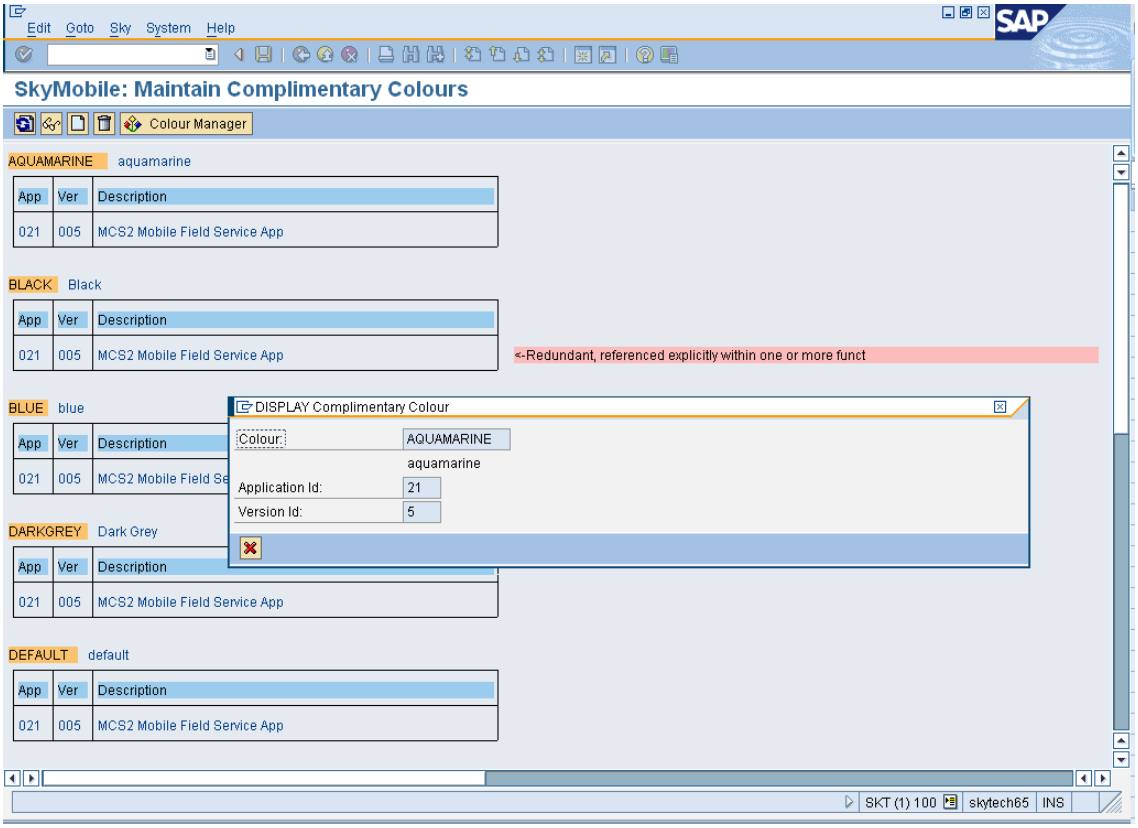
You use this option to specify binary group definitions that are cross referenced to a specific application version. You may use a selection screen to restrict the cross reference display.



You may create and remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option.

4.11.7.7 Complimentary Colours

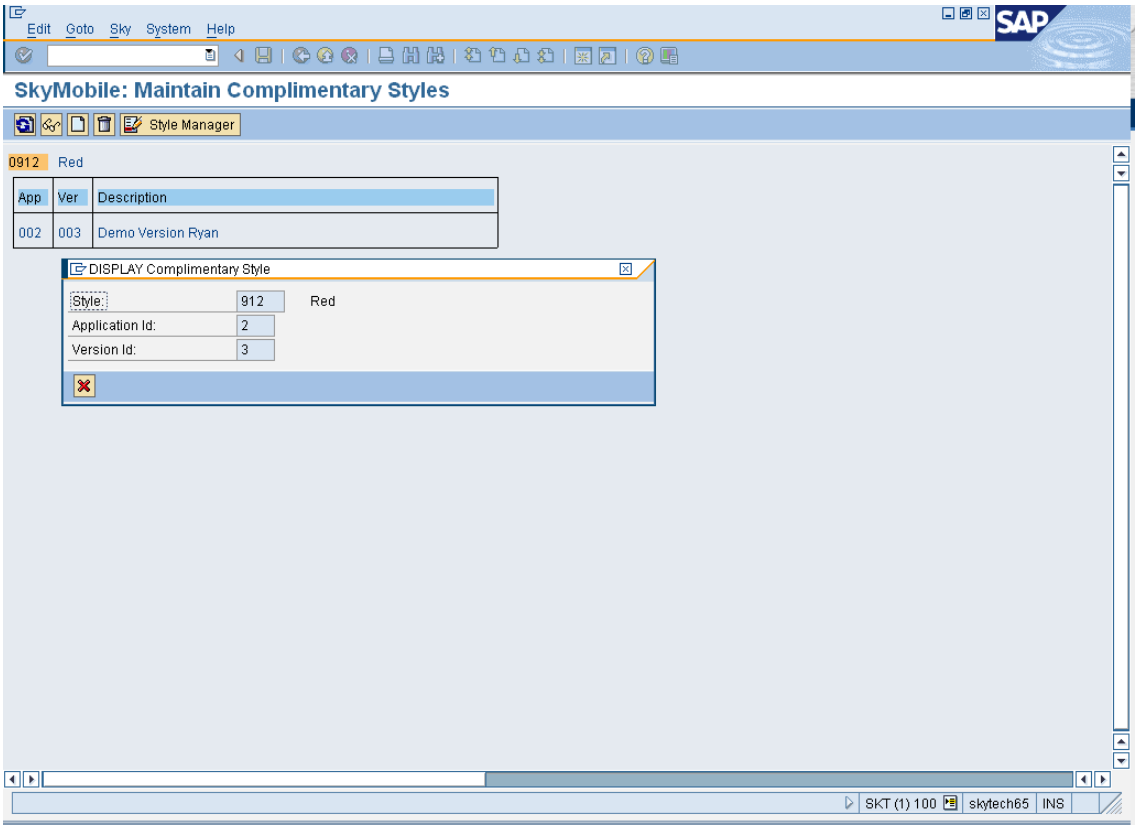
You use this option to specify additional colour definitions that you may download to a specific Application Server. By default, Application Servers automatically download all colours referenced in screen definitions. Sometimes you need to download additional colours as well, but they are not directly referenced, for example, used in exit programs. You do this by specifying a cross reference of application versions and colour definitions. You may use a selection screen to restrict the cross reference display.



You may create or remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option. The workbench highlights entries that are directly referenced in screens and thus do not need to directly cross reference.

4.11.7.8 Complimentary Styles

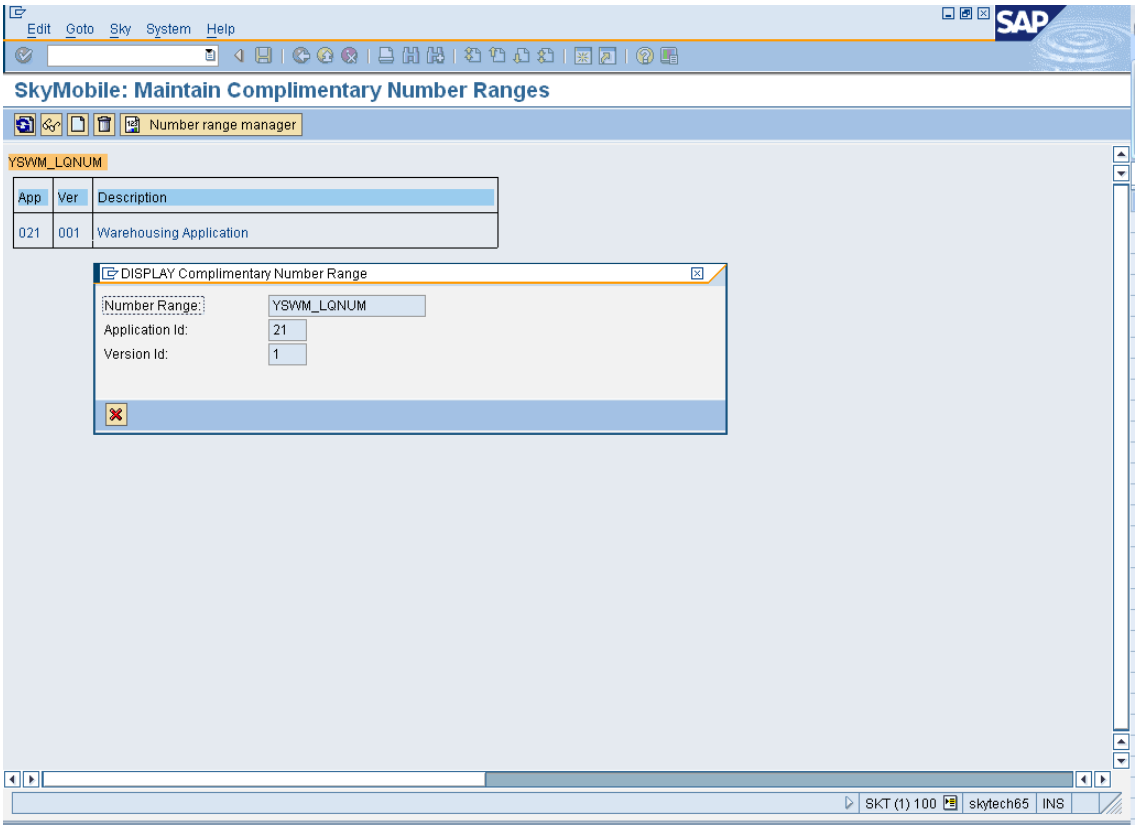
You use this option to specify additional screen style definitions that you may download to a specific Application Server. By default, Application Servers automatically download all styles used in screen functions. Sometimes, you need to download additional styles as well, but they are not directly referenced in screen functions. You do this by specifying a cross reference of application versions and styles names. You may use a selection screen to restrict the cross reference display.



You may create and remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option. The workbench highlights entries that are directly referenced in screens and thus do not need to directly cross reference.

4.11.7.9 Complimentary Number Ranges

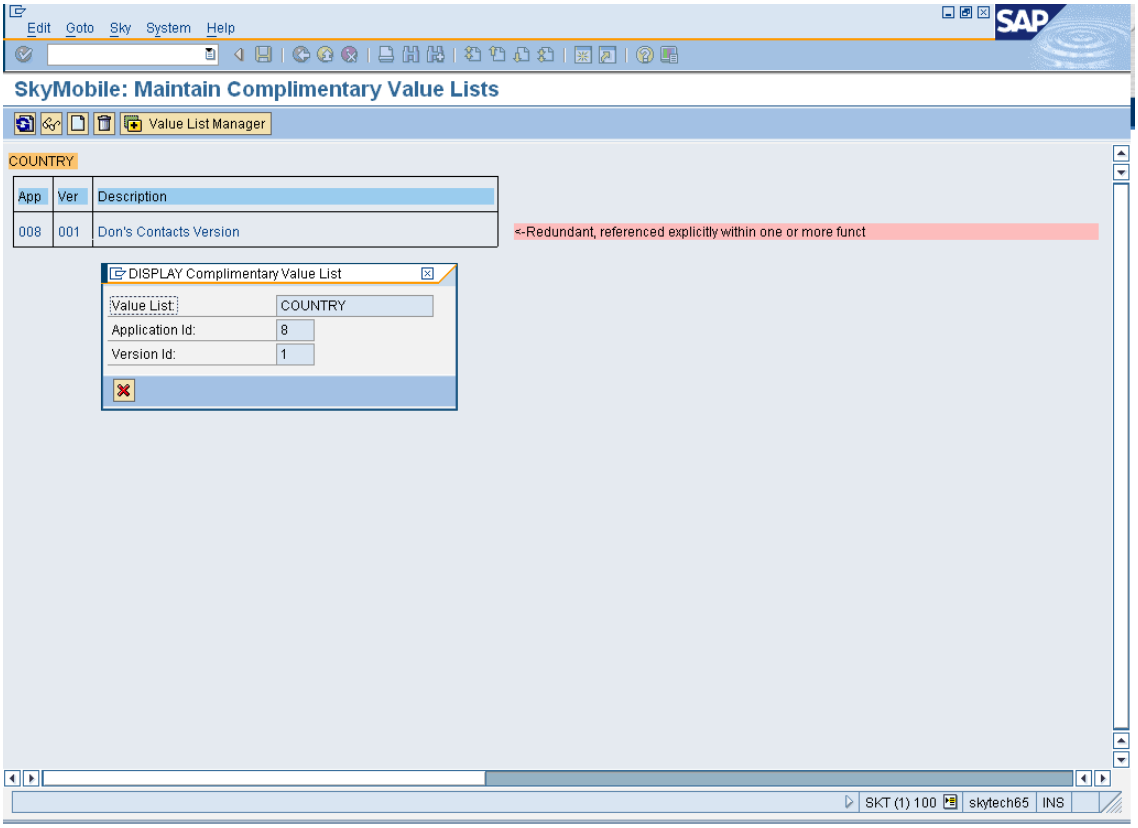
This option is used to specify additional number range definitions that may be downloaded to a specific Application Server. By default, Application Servers automatically download all number ranges referenced in screen functions. If number ranges are only referenced in programming exits, you need to define a complimentary number range cross reference. You may use a selection screen to restrict the display.



You may create and remove entries by positioning your cursor on the appropriate level and clicking on the application toolbar option. The workbench highlights entries that have are directly referenced in screens and thus do not need to directly cross reference.

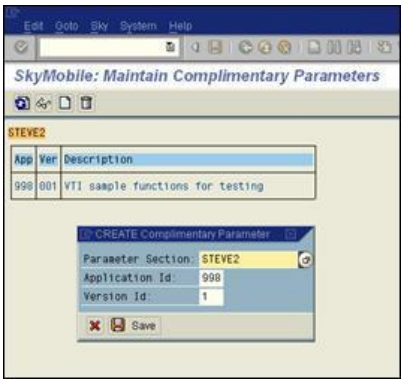
4.11.7.10 Complimentary Value Lists

You use this option to specify additional value list definitions that you may download to a specific Application Server. By default, Application Servers automatically download all value lists that the screen functions reference. Sometimes you need to download additional value lists as well, but they are not directly referenced in screen functions. You can do this by specifying a cross reference of application versions and value list names. You may use a selection screen to restrict the cross reference display.



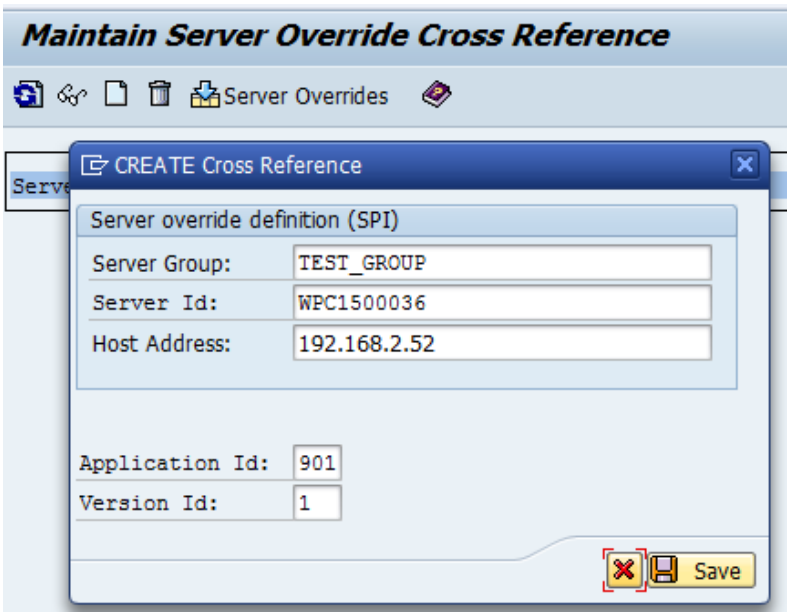
4.11.7.11 Complimentary Parameter Values

This option is used to specify additional parameter value definitions that may be downloaded to a specific Application Server. By default, Application Servers automatically download all parameter values referenced by screen functions. Sometimes you need to download additional parameter values download as well, but are not directly referenced in screen functions. You can do this by specifying a cross reference of application versions and parameter value names. You may use a selection screen to restrict the cross reference display.



4.11.7.12 Complimentary Server Overrides

This facility enables you to associate server override definitions with a application version. This is useful when transporting application definitions across systems, ensuring that dependent override definitions are automatically implemented as well. To maintain, double-click the Complimentary Server group/id overrides node in the IDE hierarchy and the following screen appear. You may also use the Complimentary Object option on the **Utilities** menu.



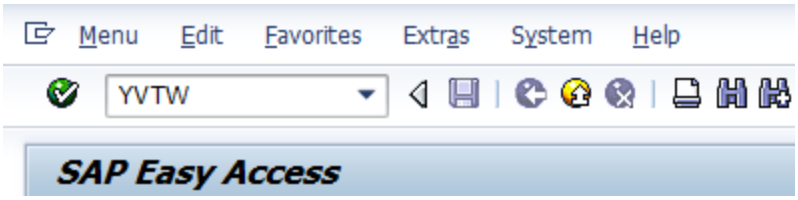
On this screen, you maintain a list of cross reference entries that tie the application version with a server override definition.

4.11.8 Working with a Group

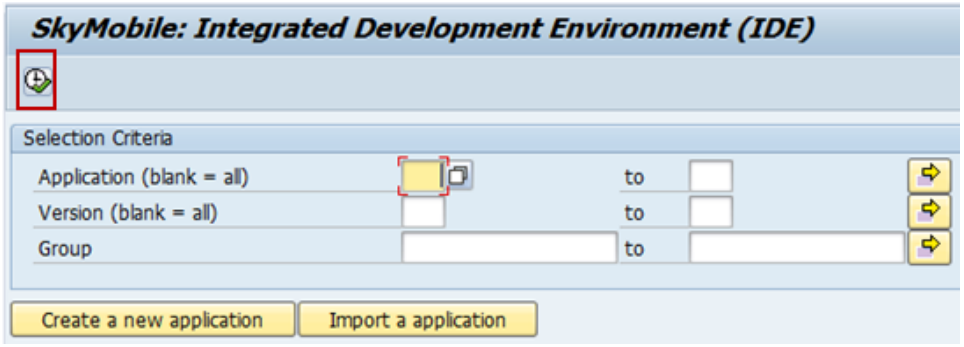
You may allocate application versions, database table definitions and binary objects to a group to manage and list them efficiently. You may maintain groups and group descriptions.

To work with the groups, follow these steps:

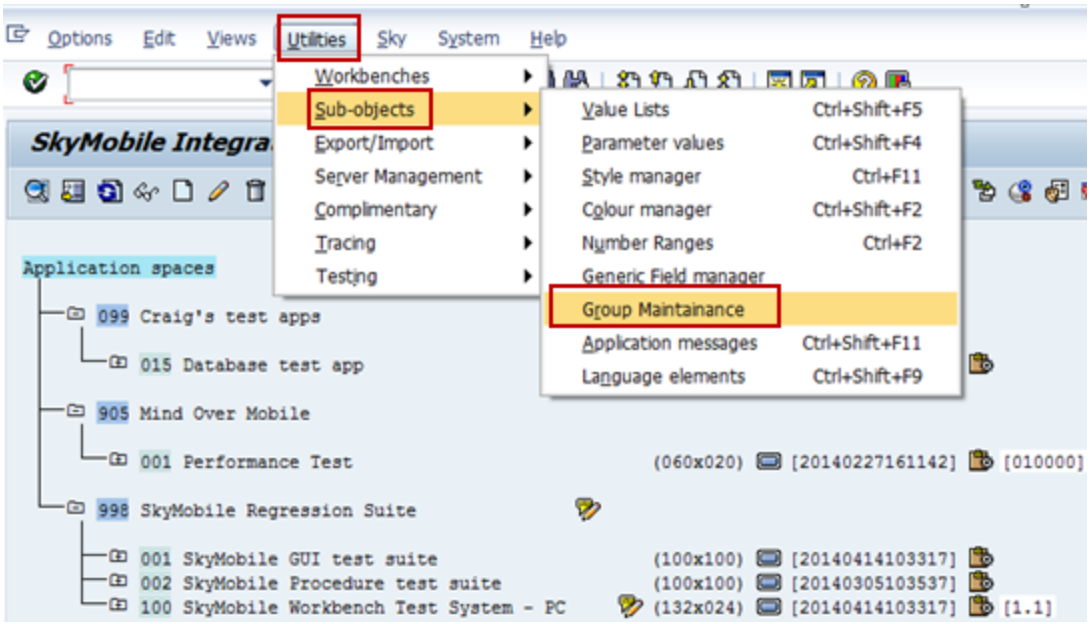
1. Execute the transaction,**YVTW** in SAP system.



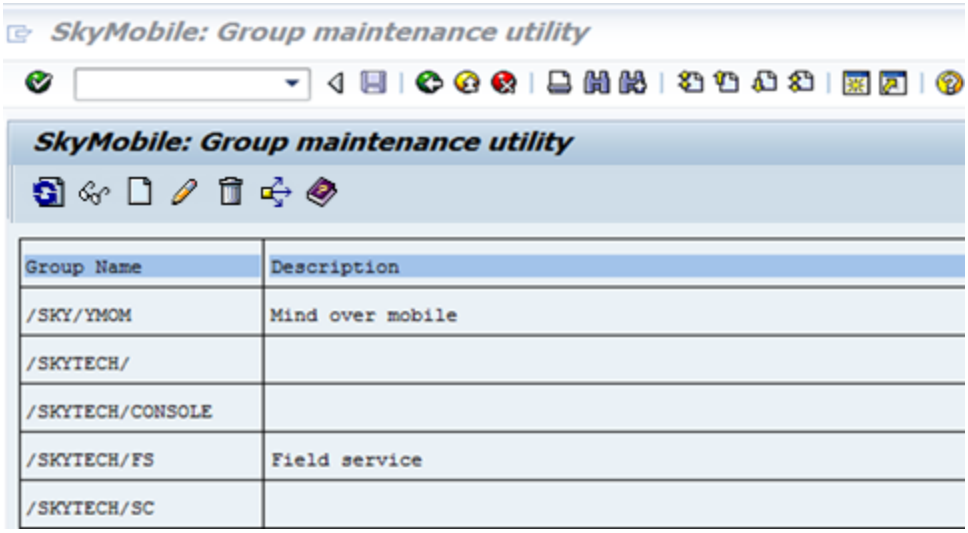
   The **SkyMobile: Integrated Development Environment (IDE)** window appears.



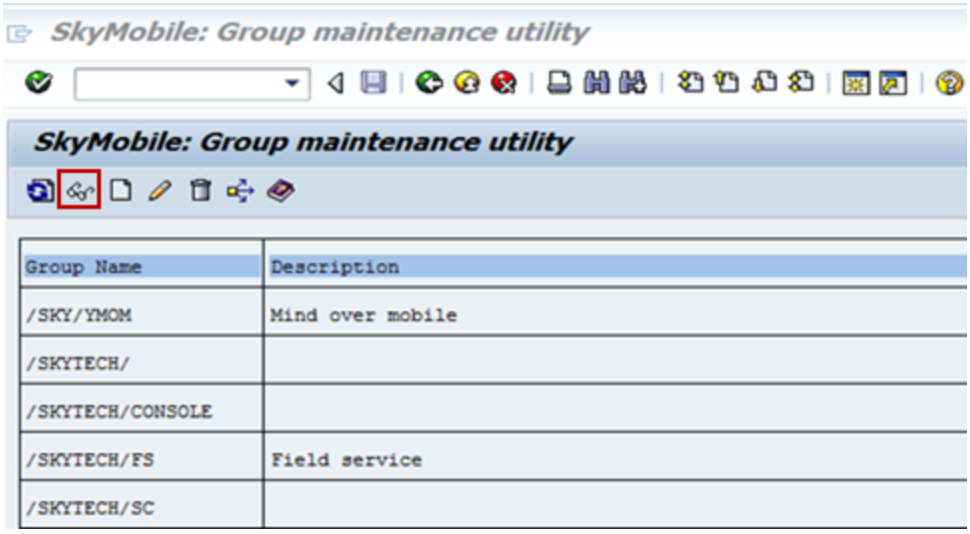
2. Go to **Utilities** on the Application Workbench > **Sub-objects** > **Group Maintenance**.



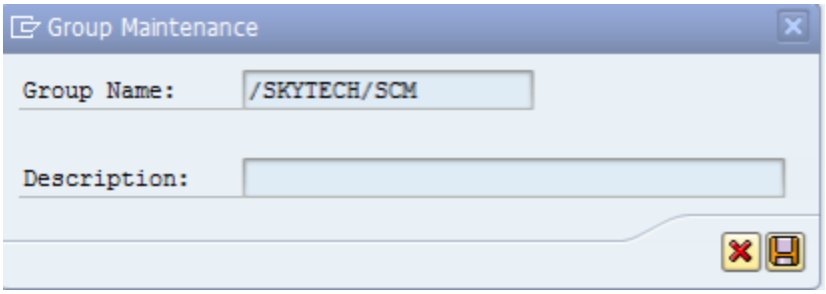
The **SkyMobile: Group Maintenance Utility** dialog appears with the existing list of groups and their descriptions.



3. To display the group maintenance dialog, click **Display**.

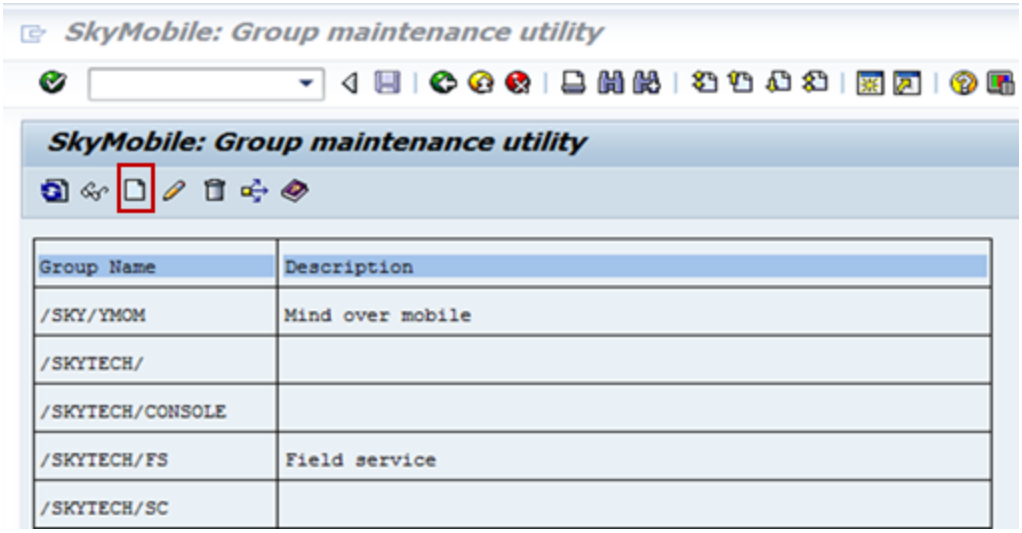


The **Group Maintenance** dialog appears in a separate window

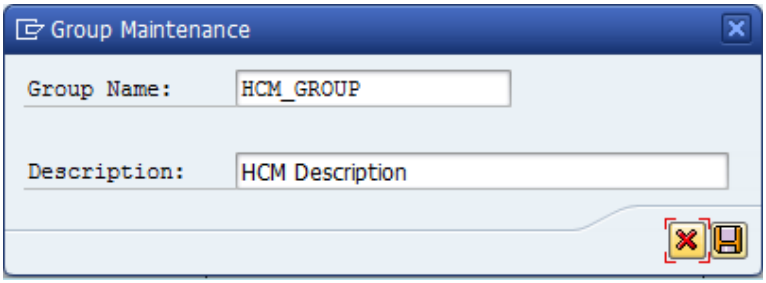


You may view the name and description of the group.

4. To create a group in the system, click anywhere in the table and click **Create**.



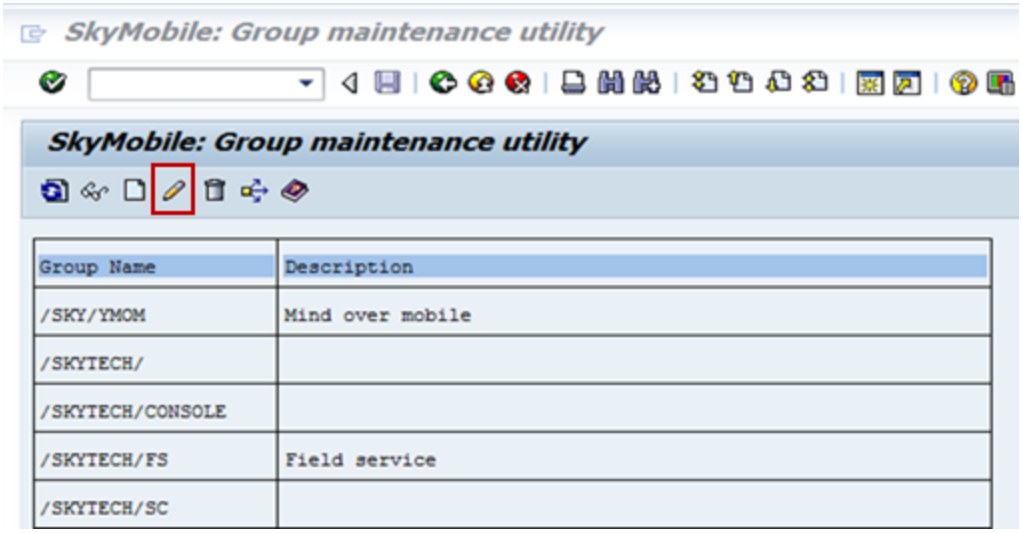
The **Group Maintenance** dialog appears.



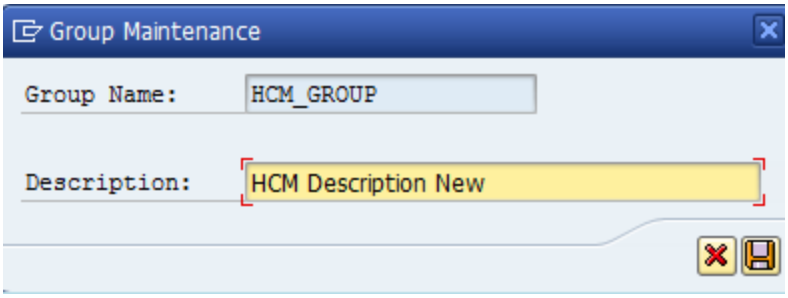
   a. Enter name and description of the group in **Group Name** and **Description**.

   b. Click **Save** to save the name and description of the group.

5. To change the description of a group, click anywhere in the table and click **Change**.



The **Group Maintenance** dialog appears.

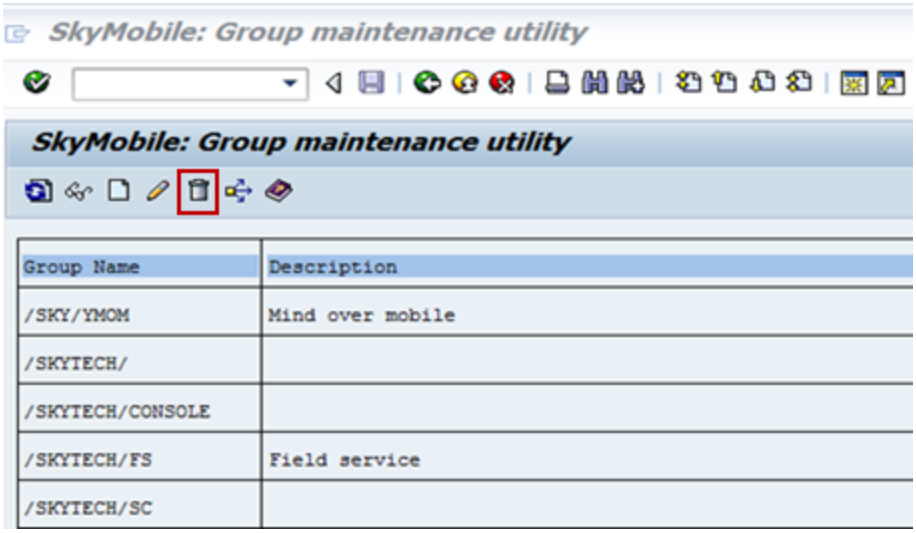


a. Modify the description of the group in **Description**.

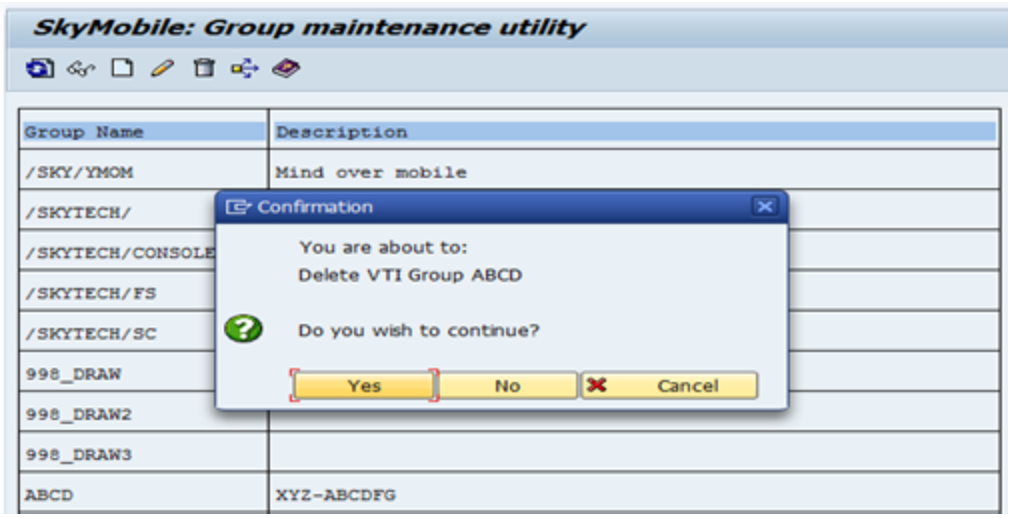
> ***Note:*** You cannot change the name of the existing group.

b. Click **Save** to save the name and description of the group.

6. To delete an existing group, click anywhere in the table, and click **Delete**.

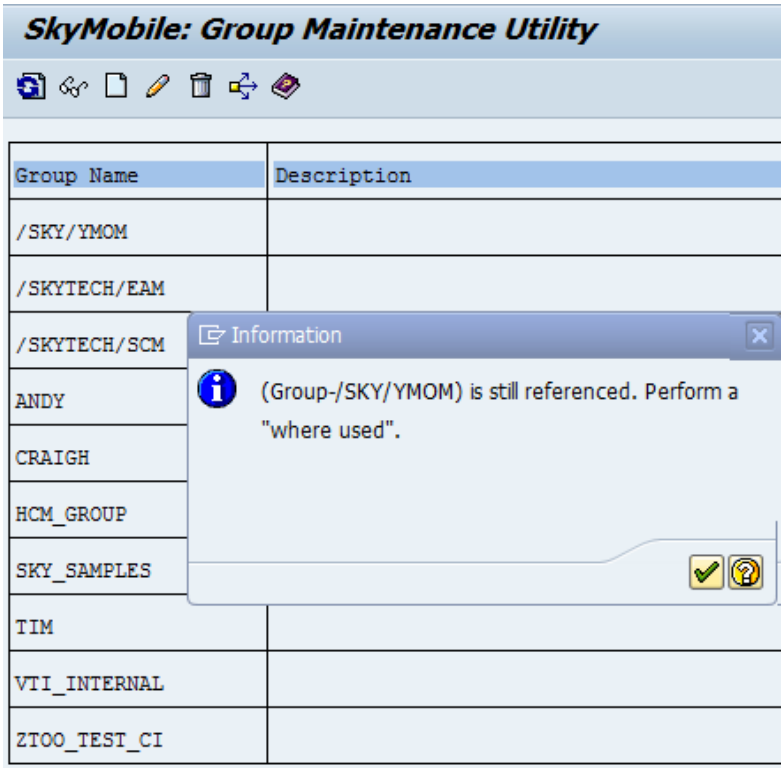


- A **Confirmation** dialog appears.



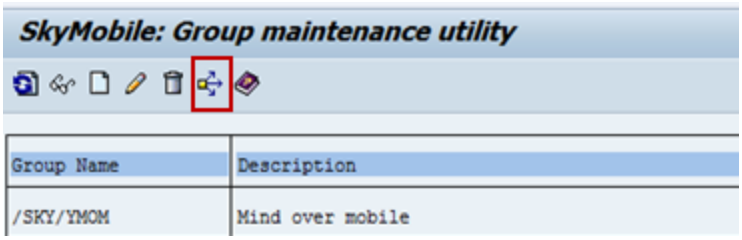
  Click **Yes** to continue.

- The **Information** dialog appears when you try to delete a group that is currently referenced by some other object.

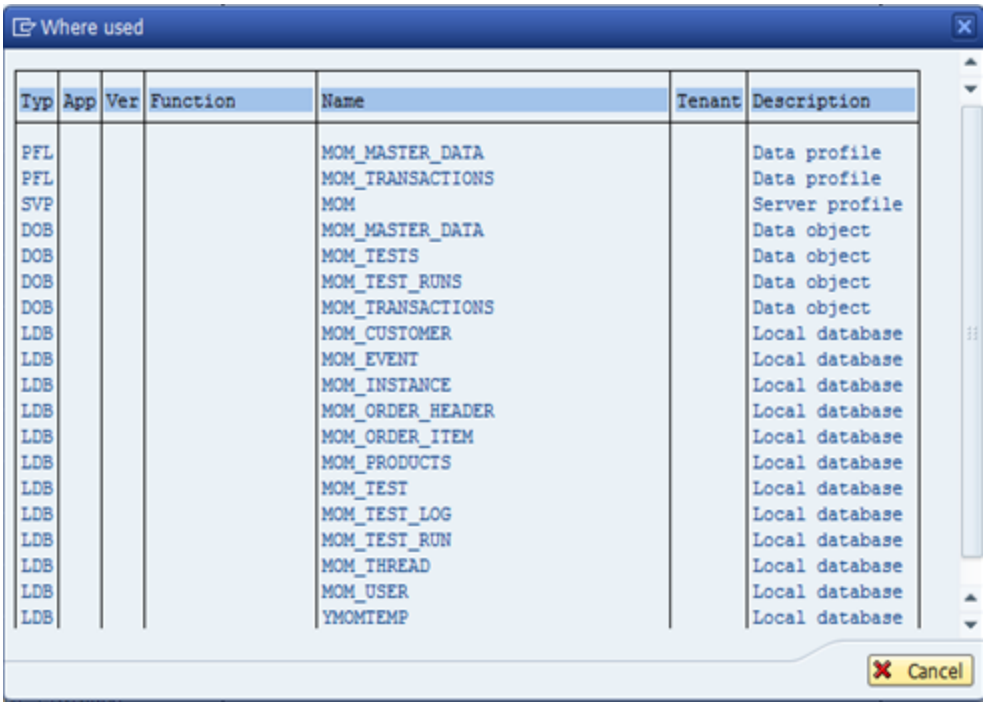


Click the green tick mark icon to continue.

7. To find out where the group is used, place the cursor on any row of the group and click **Where Used**.

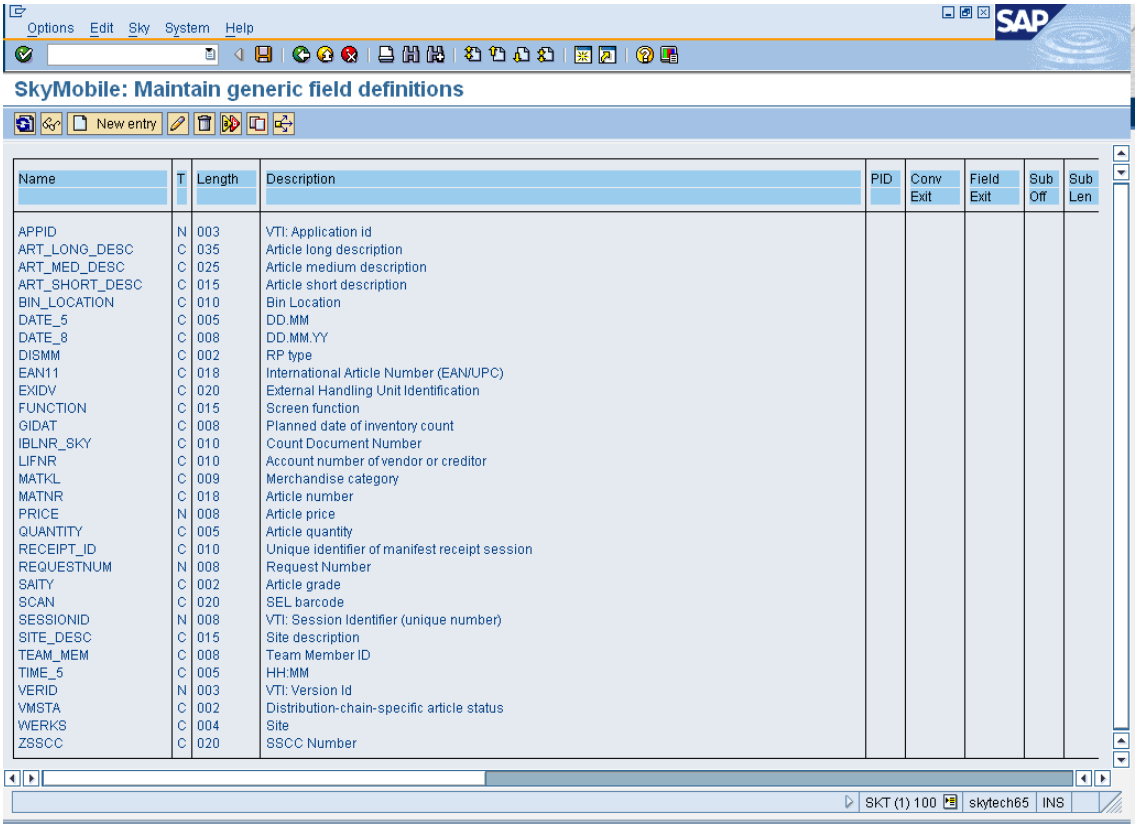


The **Where used** dialog appears with the list of applications, LDBs, DOBs, and so on from where the group is referenced.



4.11.9 Generic Fields

You use the generic fields to define standard field definitions to use across multiple screen functions. This is to encourage consistency across applications and have the ability to cross reference where fields are used. For example, you may define a material number once and implemented in screens using the 'like' option. You may invoke the generic field functionality from multiple points; directly from the workbench, from the screen painter or from the field definition screen or through transaction YVTG. You may use a standard icon: on the application toolbar to invoke the generic field manager. You may use a selection screen to restrict the fields that appear. To select a specific field to process, you position your cursor on the field name and click the relevant icon on the application toolbar.



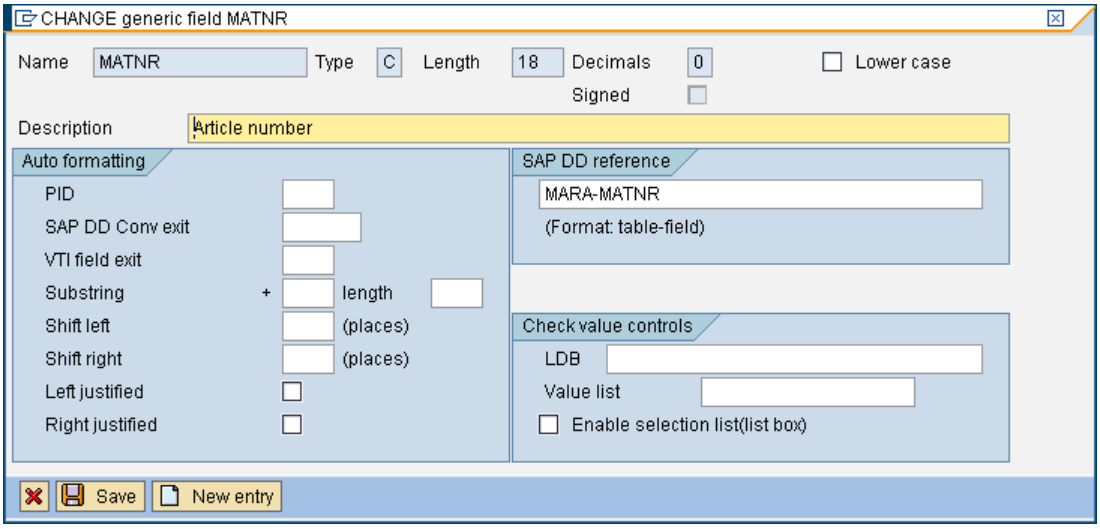
Name	T	Length	Description	PID	Conv Exit	Field Exit	Sub Off	Sub Len
APPID	N	003	VTI: Application id					
ART_LONG_DESC	C	035	Article long description					
ART_MED_DESC	C	025	Article medium description					
ART_SHORT_DESC	C	015	Article short description					
BIN_LOCATION	C	010	Bin Location					
DATE_5	C	005	DD.MM					
DATE_8	C	008	DD.MM.YY					
DISMM	C	002	RP type					
EAN11	C	018	International Article Number (EAN/UPC)					
EXIDV	C	020	External Handling Unit Identification					
FUNCTION	C	015	Screen function					
GIDAT	C	008	Planned date of inventory count					
IBLNR_SKY	C	010	Count Document Number					
LIFNR	C	010	Account number of vendor or creditor					
MATKL	C	009	Merchandise category					
MATNR	C	018	Article number					
PRICE	N	008	Article price					
QUANTITY	C	005	Article quantity					
RECEIPT_ID	C	010	Unique identifier of manifest receipt session					
REQUESTNUM	N	008	Request Number					
SAITY	C	002	Article grade					
SCAN	C	020	SEL barcode					
SESSIONID	N	008	VTI: Session Identifier (unique number)					
SITE_DESC	C	015	Site description					
TEAM_MEM	C	008	Team Member ID					
TIME_5	C	005	HH:MM					
VERID	N	003	VTI: Version Id					
VMSTA	C	002	Distribution-chain-specific article status					
WERKS	C	004	Site					
ZSSCC	C	020	SSCC Number					

From the main screen, you may perform the following:

- Create, change delete generic fields
- Rename generic fields

- Copy generic fields
- List generic field dependencies

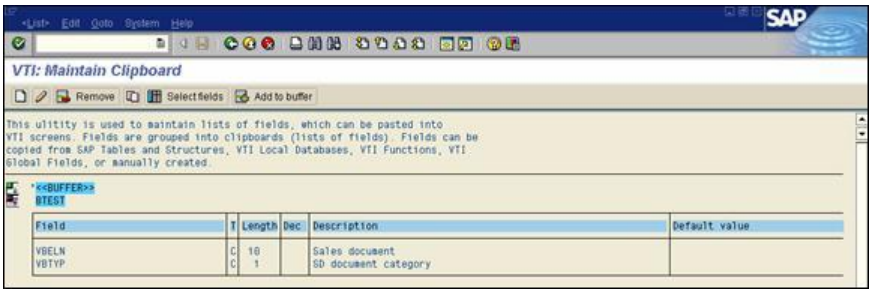
> ***Important:*** If you change generic fields, all dependent definitions; screens change as well.



You may use the field definition on a SAP DD field. The other attributes are defaulted onto the screen field definition.

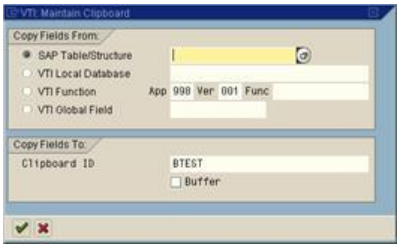
4.11.10 Clipboard Manager

You use the clipboard manager (transaction YVTZ) to define groups of fields that you regularly paste into screen definitions. This saves considerable time; developers don't need to constantly type in names, types and lengths; but can just copy and paste from a clipboard buffer. The clipboard has the capability to copy fields from multiple sources, for example, SAP tables, other screen functions, generic field definitions, SAP function modules. It also has the ability to create different groups of fields that is useful for different applications, and personal lists. A standard <<BUFFER>> is used to store common fields between the clipboard manager and the screen function builder. You copy fields into the buffer and copy and paste from it onto the screen layout. You may use a standard icon: on the application toolbar to invoke the clipboard manager. You may use a selection screen to restrict the clipboards displayed. To select a specific clipboard or field to process, you position your cursor on the name and click the relevant icon on the application toolbar.



From the main clipboard screen, you can:

- Create, change and remove clipboards and fields
- Select fields into the clipboard from various sources
- Add clipboards and/or fields into the common <<BUFFER>> area



You may copy the clipboard entries from various sources. A selection screen then appears, where you may select/deselect the fields required.

4.11.11 Standard Object Framework

The Standard Object Framework (SOF) facility allows you to add common modules and their call interfaces that you may invoke from one or more screen functions. It is essentially an extension of the exits functionality in SkyMobile. The idea behind SOF is to provide a collection of pre-coded exits for commonly used functionality such as:

- Application authentication (to SAP, Active directory)
- Triggering the launch of documents
- Triggering URLs
- Retrieving GPS coordinates

Because these requirements are often quite specific for each different platform (for example, programmatically triggering a URL or launching a document in Windows is different to Blackberry) SOF objects provide a mechanism to abstract common utilities from the "core" SkyMobile Java Server that is portable. It also enables a 'shareware' library of widgets for the developer so that they don't have to write code or can base a solution on a sample code provided in the SOF repository.

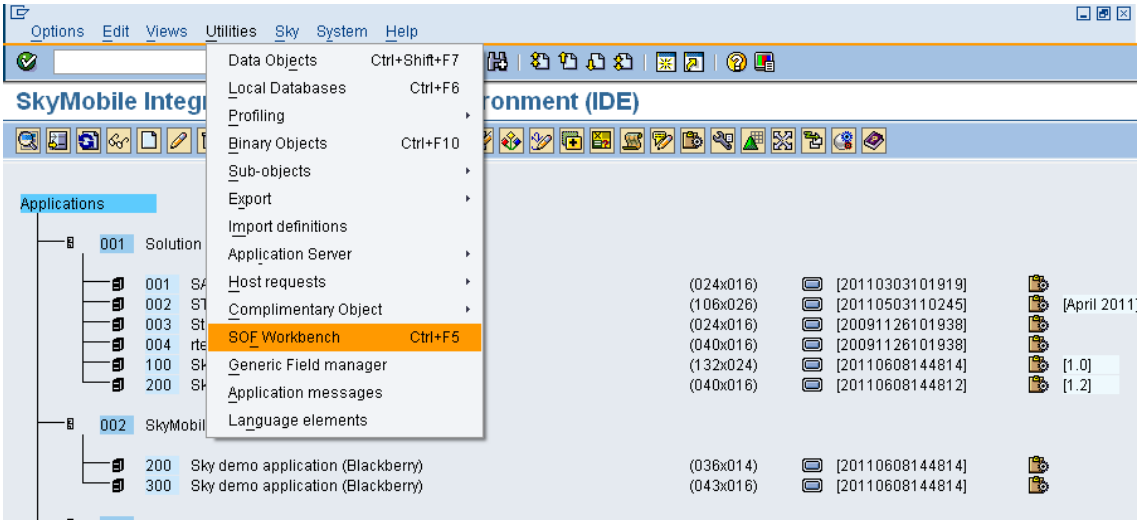
Sky plans to provide more and more standard SOF functionality and encourages its partners and customers to contribute common programs (ABAP, Java, JNI calls to C++) to SOF so that you can freely distribute a powerful 'shareware' repository.

> ***Note:*** A default SOF library is available from Sky and you must import separately; is not implemented as part of the SkyMobile software installation.

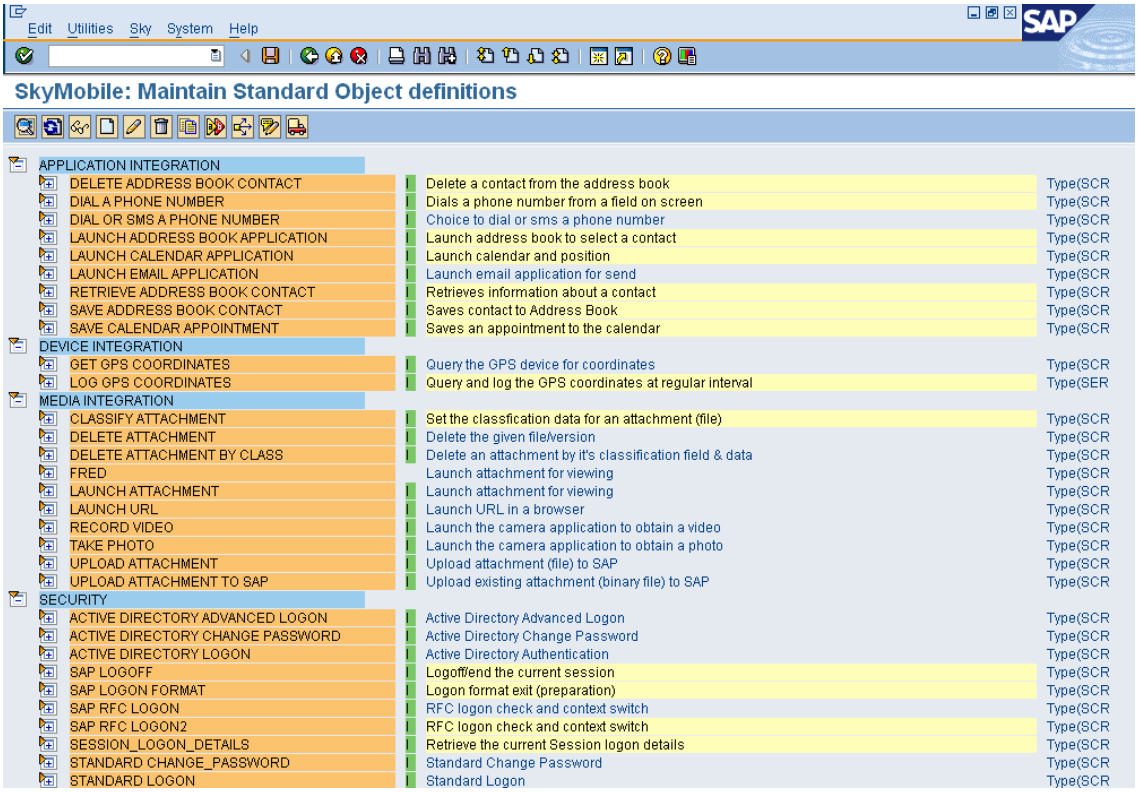
Each SOF object is categorised depending on its function, for example, media integration, security, and technical information. The whole concept is that SOF provides a "Swiss army knife" of standard utilities and connectors that you add to applications as required. This saves having to write code and provides an effective abstraction layer between the application and OS specific functions. It is important to note that though SOF objects are designed to be generic, they may only support certain operating systems. A list of supported operating systems is declared against each object and the

equivalent SOF program library is organised by operating system. This is because most SOF objects often have different and proprietary implementations according to the operating system they are executing under, for example, the mechanism to invoke the calendar application on a Blackberry is different to that used in a Windows environment.

You launch the SOF maintenance utility from the main workbench hierarchy utility menu.



A selection screen is displayed to help filter the list (if required). The following list is displayed.

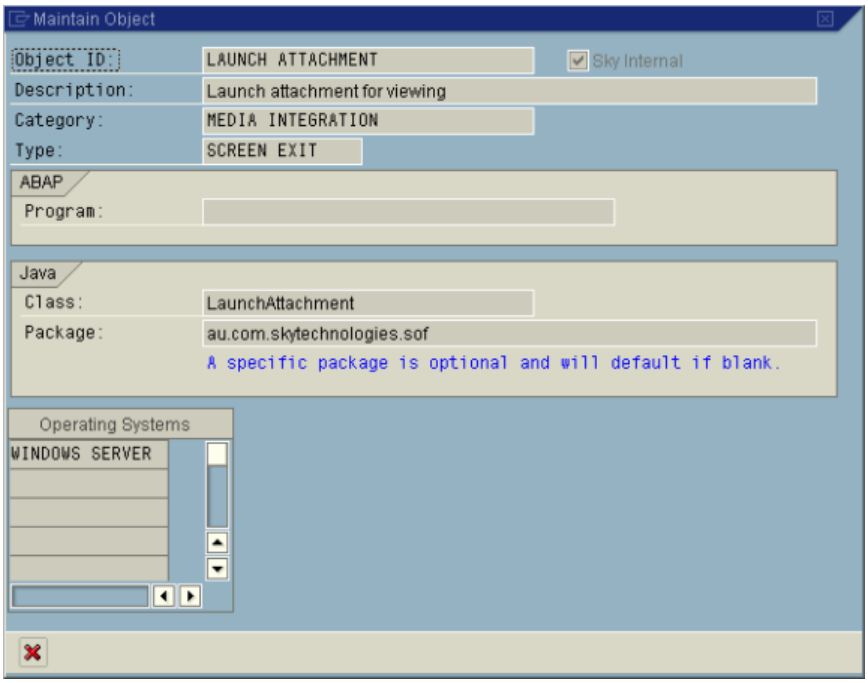


The list is grouped by category. Each object has a defined call interface; input output parameters and these are listed as fields under the object node. The objects and the field definitions are maintained by double clicking the level or positioning your cursor and clicking the application toolbar. The Object node also displays the object type, Java/ABAP attributes and operating systems supported.

> ***Important:*** You should not directly maintain Internal Sky definitions. You indicate these by a highlighted 'I' after the name. You may create custom copies of them.

4.11.11.1 Defining SOF Objects

Double-click an existing object or position your cursor and click the icon on the application toolbar. The following pop-up appears:



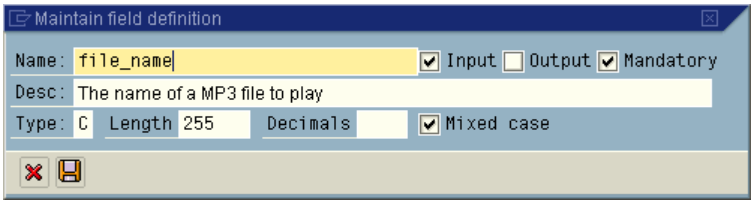
If the object is flagged as "Sky Internal", then you may not change its attributes. The SkyMobile check install process automatically generates or imports these objects.

Field	Description
ID	This is the unique name of the SOF object
Category	You must assign each object into a category that helps describe its purpose and makes listing and organization easier.
Type	Currently, only screen type exits are supported.
ABAP	The name of the ABAP program to execute (optional)

Field	Description
Java	The name of the Java class and package to execute (optional). If the package name is left blank, SkyMobile attempts to default it from either the application version or generic loaded classes.
Operating systems	A list of supported operating systems.

4.11.11.2 Defining SOF Fields

If the object requires input or output parameters, then you need to add these to the object. Either double-click an existing object or position your cursor and click a icon on the application toolbar. The following pop-up appears. Currently tables are not supported. You may add as many field definitions as required.



You specify the attributes of the field; input/output/type/length. You use these to define the call interface for the object; the sending and receiving data.

4.11.11.3 Calling and Implementing SOF Objects

Screen Exits

When you define a screen exit, you may configure it to invoke a SOF object using the supplied drop-down. The workbench then prompts you to define fields for the call interface. You must define fields with the same name; there is no mapping. Refer defining screen exits for more information. For internal Sky objects, both the source and executable libraries for each operating system are provided with the Kony for SAP installation. You must ensure that you copy the appropriate library to the Application Server. Refer to the SkyMobile XAI/SOF guide and SkyMobile installation guide for the relevant platform more information.

4.11.11.4 SOF Object Programs and Sample Code

The final SOF component is the exit programs themselves. In case of ABAP exits, these are included with the SkyMobile SAP add-in transport. In the case of Java SOF objects, these are provided in the sample programs.

> ***Note:*** All Sky supplied Java SOF objects are defined in the package au.com.skytechnologies.sof and come precompiled for a limited number of platforms. The source is provided so that you can compile the exits for alternate JVMs.

4.11.12 ABAP Programming Guide

4.11.12.1 About

This guide provides information on the SkyMobile ABAP SDK with example code. It also has a section documenting some of the special APIs, utilities and techniques that are available to integrate with SAP and SkyMobile sub-systems. It is assumed that the reader has a good understanding of the ABAP programming language and the SkyMobile workbench. For more details on some of the concepts discussed, refer to the SkyMobile Workbench and System Management sections.

4.11.12.2 ABAP Programming Guide

Each SDK command is prefixed with 'VTI_....' and the internal variables with 'V_VTI_...' and 'IT_VTI_....' for internal tables respectively. For a list of commands and their usage, either see the exit command reference in this guide or double-click the /SKY/YVTISDK include and scroll to the relevant section.

4.11.12.3 The SDK Include

Before you use any SkyMobile ABAP SDK commands, you must first include /SKY/YVTISDK at the start of your program. This one include contains all that you need to interact with SkyMobile. The SDK commands are in the form of ABAP macros that are embedded into your ABAP program code. This makes it extremely simple for programmers to use and abstracts any internal dependencies away from the custom program. You may use any ABAP syntax in conjunction with the SDK, thus the programmer has the full power of SQL, function calls, BAPI calls, IDOC processing and BDC processing.

4.11.12.4 Screen Exits

You may use screen exits to branch out from 'normal' SkyMobile application processing to perform more complex processing and screen manipulation. Strong ABAP and Java SDKs are provided to assist developers to interact with the SkyMobile functionality and dramatically simplify code. As well as custom modules, you may also incorporate standard "objects" into the framework using the SOF (Standard Object Framework), see the workbench documentation for more details. For example, launch an attachment, trigger a URL, perform an active directory (ADD) logon check.

A screen exit is identified by a unique three letter identifier and a process code used to help categorize its use. You may invoke exits from the following events:

- Screen formatting
- Screen action to either verify content or perform a transaction (or both)

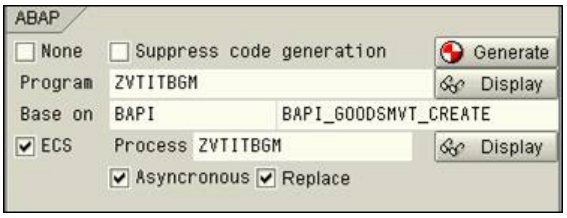
You should be aware of the context that an exit is being processed in, for example, you should not define an ABAP (host) formatting exit on a 'local' screen. The workbench attempts to identify and warn you of these conflicts. If a screen is local and an ABAP exit is configured, all local screen processing is performed first and then a 'post local' call is made to the SAP host. The same is true of any host based database operations.

The execution of ABAP and Java screen exits may be mixed, that is, may be equivalent, meaning the Java exit is used when executing from a remote SkyMobile server and the ABAP exit is used from within SAP using the SAP emulator; or complimentary when both are executed.

> ***Important:*** Read "Screen processing exits" in the Workbench guide for details on creating SkyMobile screen exit definitions.

Generating ABAP Exit Code

You may configure the ABAP exit name and usage using the workbench. You may then automatically generate and maintain ABAP exit code. See the workbench section on exits for more information. When the ABAP exit code is generated, SkyMobile uses code markers in the source to insert screen variable and table definitions. It also inserts sample API calls to retrieve and set the screen field values. The following pop-up appears when creating or maintaining programming exits from the workbench.



A default name is generated using the prefix specified in the version configuration, type of exit, for example, F, V, T or S and the exit suffix code.

Field	Description
None	If you select this check box, no ABAP exit is called. This is usually set if a Java exit is configured and there is no SAP equivalent.
Suppress code generation	Suppresses the generation of the ABAP exit code. This is useful, if the exit belongs to another function and is shared, or you do not wish to re-generate the ABAP code after you have customised it.
Program	You may specify a custom ABAP name, or the default name is used. The name must start with 'Z' as per the SAP custom program convention.
Generate button	Create a new or analyse an existing ABAP program. If the program already exists, only the code between the 'code markers' is replaced.
Display button	This button invokes the SAP ABAP editor.
Base on	Generate an extra code to call a nominated SAP BAPI, function module, and IDOC.

Field	Description
ECS options	Valid for transaction type exits only. This instructs SkyMobile to invoke the exit program using ECS interface management. The specified ECS Process name are automatically created. Usually this option is used to support asynchronous exit processing.

Important: The ABAP exit code is generated every time you press the generate button. This means that the standard sections in the code, identified by <<....>> code markers are maintained. If you wish to suppress automatic generation, select the "Don't generate ABAP code" check box. If you wish to retain your own custom definitions and syntax, make sure you code outside of these markers.

Example of Generated Exit Code

The following sample generated code retrieves a list of materials from the SAP material master (MARA) table. The highlighted 'code markers' show the standard sections that the ABAP code generator maintains.

4.11.12.5 Example

```
*$*$
=====================================================================
*$*$ This ABAP VTI exit has been generated automatically by the VTI
*$*$ Workbench function builder. Please note that the generated code
*$*$ between the <VTI_xxxxxxx_START> and <VTI_xxxxxxxx_END> place
*$*$ markers will be automatically maintained whenever the exit
*$*$ definition is generated. Remove the pair of place markers to
*$*$ prevent this from happening.
*$*$
=====================================================================
REPORT ZVTIFSAM                                 .

* <VTI_COMMENT_START>
* ---------------------------------------------------------------------
--
* VTI exit definition: FORMAT SAM
* Description: ABAP list of SAP materials
* Generated: 19.03.2007 14:24:01 SKYTECH
* ---------------------------------------------------------------------
--
* <VTI_COMMENT_END>

*
=====================================================================
==
* Data declarations.
*
=====================================================================
==

* Declare VTI/ECS SDK.
INCLUDE: /SKY/YECSSDK,
```

```
         /SKY/YVTISDK.

* VTI screen field definitions.
* <VTI_FIELD_START>
data:
      dummy(1)                type c.
data: begin of it_MATERIALTABLE occurs 0,
      MATNR(018)              type C,
      MEINS(003)              type C,
      MATKL(009)              type C,
      MAKTX(060)              type C,
      end of it_MATERIALTABLE .
* <VTI_FIELD_END>

*
=====================================================================
==
* Main processing logic.
*
=====================================================================
==
Initialization.
  perform f_initialisation.

start-of-selection.
  perform f_custom_processing.

end-of-selection.
  perform f_termination.
```

The <start> <end> code markers enables SkyMobile to safely re-generate the ABAP code i.e. include any changes without losing your custom code. Therefore you may re-generate exits at any stage. The basic rule is to avoid placing your own custom code within the internal code markers. The following is a table of code markers and their use:

Code Markers	Description
<VTI_COMMENT_START> <VTI_COMMENT_END>	Generated information comment block
<VTI_FIELD_START> <VTI_FIELD_END>	Contains all screen field and table definitions
<VTI_GET_FIELD_START> <VTI_GET_FIELD_END>	Select screen fields and tables
<VTI_SET_FIELD_START> <VTI_SET_FIELD_END>	Update screen fields and tables
<VTI_ECS_EXPORT_ START> <VTI_ECS_EXPORT_END>	Export screen data to ECS interface parameter data (if configured)
<VTI_OBJECT_TYPE_ START> <VTI_OBJECT_TYPE_END>	"Based on" code generation for example, BAPI call

Importing and Exporting Screen Parameters

SkyMobile passes the internal screen definition to the ABAP program through ABAP memory. Therefore, the first thing that you must do is a VTI_IMPORT_PARAMETERS, and the last thing you must do is a VTI_EXPORT_PARAMETERS to return any changes back to SkyMobile.

Structuring the Exit Programs

Normal ABAP standards and techniques apply, but there are some SDK features that can help make the code more modular and easier to maintain. In order to process a screen, the program must import and export the screen definition. In-between, it is best to use forms to modularise and perform specific tasks. The first statement in each form should be a VTI_IGNORE_IF_ERROR command that ignores processing the form and continue with the next if an error is raised.

```
VTI_IMPORT_PARAMETERS

Perform f_verify_material.
Perform f_verify_uom.
Perform f_verify_quantity.
.....
VTI_EXPORT_PARAMETERS

Form f_verify_material.
  VTI_IGNORE_IF_ERROR.
...
Endform.

Form f_verify_uom.
  VTI_IGNORE_IF_ERROR.
...
Endform.

Form f_verify_quantity.
  VTI_IGNORE_IF_ERROR.
...
Endform.
```

Using this technique, any error set in the forms causes processing to drop through the other forms, ignoring further processing, but still performing the VTI_EXPORT_PARAMETERS. This makes for more modular code, avoiding the use of flags and nested conditions.

Getting and Setting Screen Data

Commands are used to import screen field and table data into equivalent program variables. You may manipulate these and then export back to the screen.

Screen Fields

You may retrieve screen values into your program values using VTI_GET_SCREEN_FIELD and update screen fields using VTI_SET_SCREEN_FIELD. Internally, SkyMobile expects the target and source variable to be named 'v_{screen field name}'.

Example:

```
GET_SCREEN_FIELD 'TYPE'.
SET_SCREEN_FIELD 'TYPE'.
```

Screen Fields

You may retrieve screen values into your program values using VTI_GET_SCREEN_FIELD and update screen fields using VTI_SET_SCREEN_FIELD. Internally, SkyMobile expects the target and source variable to be named 'v_{screen field name}'. For example:

```
GET_SCREEN_FIELD 'TYPE'.
SET_SCREEN_FIELD 'TYPE'.
```

Screen Tables

A screen table is represented in ABAP as an internal table and is handled in the same fashion as any other ABAP internal table. SkyMobile expects the internal table name to be 'IT_{screen table name}'. Special processing is required to import and export table data from the screen. The internal table structure must match the screen table exactly (as generated by SkyMobile).

Importing table data from the screen into the internal table

```
vti_import_table_control 'NAMETABLE'.
```

Exporting table data from the internal table back to the screen

```
vti_export_table_control 'NAMETABLE'.
```

You must remember that the entire table is processed; not just what is displayed on the screen. Therefore, special consideration is required if you are scrolling and/or manipulating what currently appears on the screen, for example, changing styles, graphics, and hiding fields. Each 'displayable' element of the table is represented as a unique internally generated field name of the format '$TBL_TT_RR_LL_CC' where 'TT' is the unique table number, 'RR' is the table row and 'CC' is the column. Instead of you having to specify this internal name, you may use the VTI_GET_TABLE_FIELD_NAME command using the actual screen field name and the row you are interested in.

In the following example code, loop through all the entries in the table and set a style on the screen field. The dimensions of the table (total rows, screen occurs) are retrieved and the internal table field name for the current row is retrieved using the VTI_GET_TABLE_FIELD_NAME command. Note how the table occurs (displayable) size, 'top row' position and current row counters are used to indicate where the table is displayed from, how many lines are displayed and what table elements (fields) are being addressed. If you simply scroll from a point in the table, then you simply set the 'top row' to the index count of the internal table, for example, sy-tabix. Unfortunately, because of the flexibility required with table manipulation and the diverse functionality of the screen elements, there is no automatic way to avoid coding loops and using counters.

Getting Screen Table Fields

```
* Working variables.
data: l_count            type i,
      l_field_name       type vti_field_name,
      l_tabix            like sy-tabix,
```

```
      l_found(1)          type c,
      l_top_row           type i,
      l_row_count         type i,
      l_total_rows        type i.

* Highlight a field in the list.
  VTI_GET_TABLE_SCREEN_OCCURS 'MY_LIST' l_count.
  VTI_GET_TABLE_TOP_ROW 'MY_LIST' l_top_row.
  if l_top_row = 0. l_top_row = 1. endif.
  do l_count times.
     l_tabix = ( l_top_row + sy-index ) - 1.
     l_row_count = sy-index.
     clear: it_my_list, l_found.
     read table it_my_list index l_tabix.
     if sy-subrc = 0.
        l_found = c_vti_true.
     endif.

*    Analyse the table field & set a style.
     VTI_GET_TABLE_FIELD_NAME 'my_field'
                              l_row_count
                              l_field_name.
     if l_found = c_vti_true.
        If it_my_list-my_field = 'HIGHT'.
        vti_set_style l_field_name '1'.  "<- red colour
     else.
        VTI_SET_STYLE l_field_name '0'.  "<- reset colour
     endif.
  enddo.
```

Important: A common problem with tables is forgetting to re-generate the exit when the table control layout on the screen changes, that is, the screen table does not match the ABAP structure. This can cause unpredictable problems with data formatting.

Message Control and Error Processing

Each session has a return code and a message text. SkyMobile displays a standard message on the screen. If the return code is blank, the message is treated as information. If the return code is set, the message is treated as an error message. The SDK has commands to set the return code and/or message text. See the exit command reference for all the available options, but you are recommended the following options.

```
VTI_SET_MESSAGE 'Material not found'.
VTI_SET_RC '100'.
```

OR

```
VTI_SET_ERROR_VIA_SAP_MESSAGE 'YV' '001' v_vbeln v_posnr.
VTI_SET_MESSAGE_VIA_SAP 'YV' '001' v_vbeln v_posnr.
```

Manipulating Screen Definitions

You may dynamically temporarily alter the screen definition, for example, change field from input to output, character to numeric. There are many options available to do this (see the command reference table). The following is possible:

- Change the characteristics of a screen field (input or output, type, style, colour)
- Hide, remove or re-instate screen fields and/or function keys
- Maintain the standard screen title (with or without variables)
- Clear all screen values (unless explicitly kept)
- Position the cursor on a field.

Example

```
VTI_SET_FIELD_ATTR 'MATERIAL' OUTPUT.
VTI_POSITION_CURSOR 'QUANTITY'.

VTI_REMOVE_FIELD 'CREDIT_LIMIT'.
VTI_REINSTATE_FIELD 'AMOUNT_OWING'.

VTI_CLEAR_SCREEN.
```

Storing Data Definitions for Later Use

SkyMobile provides a standard facility to export and re-import ABAP data definitions, for example, fields and internal tables to permanent database storage, so that they may retrieve and reuse later, or in another ABAP exit program. For example, you may want to pass a list of calculated product prices around a number of ABAP exits. You use the VTI_EXPORT_DATA and VTI_IMPORT_DATA commands for this purpose. These data definitions are stored for the life of the session.

> ***Note:*** Both the exported and import data structures must be identical. You may use any ABAP data type, for example, internal tables, structures, and individual variables.

ABAP screen exit 1

```
VTI_EXPORT_DATA IT_ORDER_ITEMS
```

ABAP screen exit 2

```
VTI_IMPORT_DATA IT_ORDER_ITEMS.
```

Logical Locking

Another common requirement is to lock a resource to prevent concurrent access, for example, prevent an update, if already in progress elsewhere. SkyMobile provides a basic locking mechanism that you may use to create, check and release a lock value. You use the VTI_CREATE_LOCK, VTI_CHECK_LOCK and VTI_RELEASE_LOCK commands for this purpose.

> ***Note:*** The context of the lock is really only relevant within the SkyMobile application; stop two mobile users from updating the same order.

Using the Flow Trace Facility

The SAP add-in has many trace options to help diagnose processing flow and logic. On of these, the flow trace may be utilised by ABAP programs to trace logic flow and data values. The trace must be activated for the session, or else trace commands will be ignored. In this way, trace commands may be left in the program code and then activated when required. The VTI_FLOW_TRACE command is used to write a single string to the trace. The trace is viewed using the session manager.

Another useful aspect of the flow trace is to highlight potential performance problems, embed strategic flow trace commands and view the run time taken between them using the trace list. You should also take care to avoid excessive tracing; avoid high volume processing and turn it off when finished.

Invoking SAP Interfaces

Because these are ABAP exits, in SAP, you may use any SAP interfacing technique, for example, BAPI call, function call, IDOC processing, BDC, custom ABAP, and SQL. Whatever the interface technique you use, you should perform the posting of transactions asynchronously; in the background. This means that the user does not need to wait for SAP to complete processing, thus they are released to continue working. In order to perform asynchronous transaction updates, you should verify the data to eliminate failure due to the input data. Another useful mechanism is to utilise the SkyConnect ECS interface management that comes bundled with SkyMobile. This provides true end-to-end transaction management, load balancing, dependency controls and restart or recovery.

Using SkyConnect ECS Interface Management

The Sky Technologies ECS interface management framework can effectively manage and monitor both synchronous (real-time) and asynchronous (background) processing. The ECS run-time software is installed with SkyMobile and is available for interface processing. It provides an excellent mechanism for scheduling, processing and monitoring interface runs. The ECS user guide explains all the functionality in detail, but the basics are covered here.

> ***Note:*** ECS processing may be automatically generated for transaction exits by specifying the ECS options on the screen exit definition screen, however you may also achieve this through custom code using the ECS ABAP SDK.

Custom Coding to Start an ECS Process

It is assumed here that the programmer knows how to define SkyConnect ECS processes. In order to start an ECS process, you must include the ECS SDK (YECSSDK) at the start of the program. You may then use the following commands to trigger a new interface run, passing data.

Example

```
ecs_initialization.
ecs_reserve_run_number.
v_ecs_*rnumber = v_ecs_rnumber_reserve.
ecs_export_data: v_order_header, it_order_items.
ecs_start_process 'CREATE_SALES_ORDER'.

if v_ecs_*rc ne 0.
   vti_set_message 'ECS start failed'.
   vti_set_return_code '100'.
endif.
```

The ABAP program receiving the data, in the ECS process CREATE_SALES_ORDER would issue an equivalent ecs_import_data: v_order_header, it_order_items. to receive the passed data definition. You may pass any number of variables, structures or internal tables using this technique. A unique run number is returned identifying the unique instance of the interface. See the SkyConnect online guide for more details.

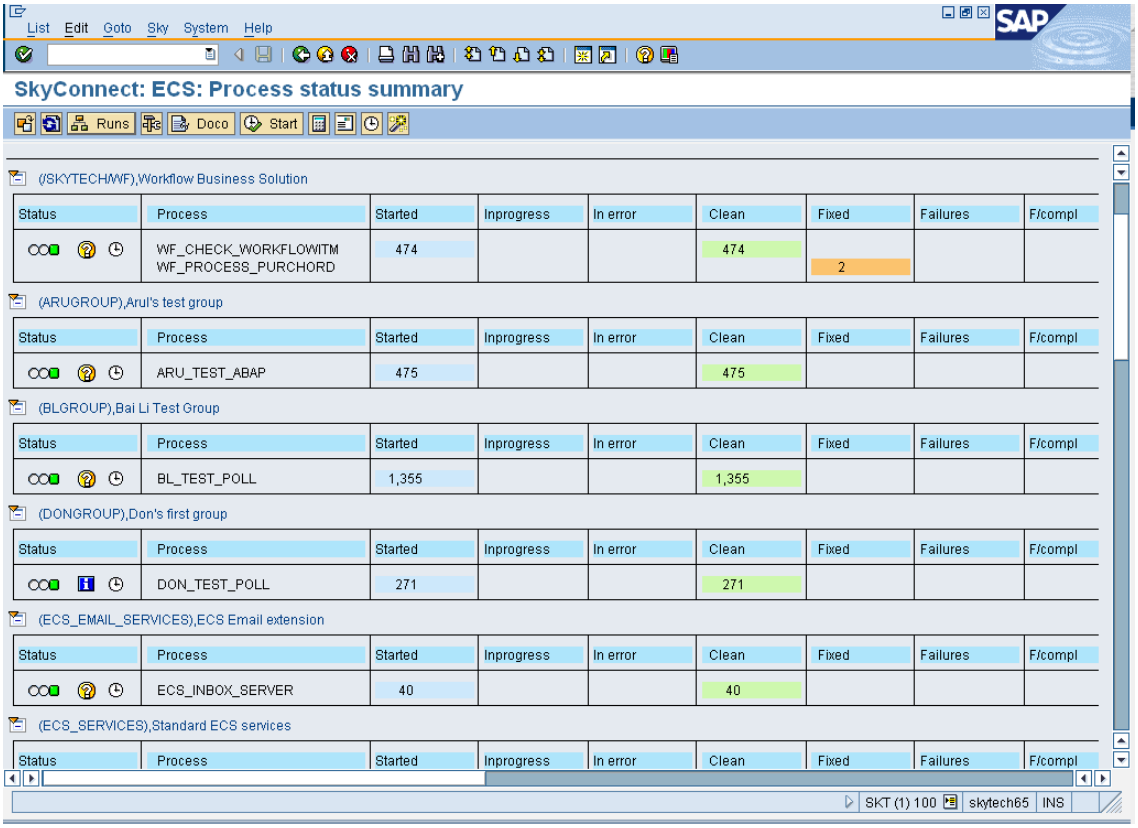
Using the ECS BDC Interface

A specialized BDC interface is available for using ECS to process generated BDC sessions. The following example posts a material movement. Note that the generated ECS process name is {VTI BDC prefix}_{screen function name} for example, VTI_GOODS_RECEIPT. You may use your own BDC processing, but you may find this method easy to use. To use this option, you must include the ECS SDK (YECSSDK) at the start of your program.

Example

```
* Create BDC session.
VTI_SET_BDC_PREFIX 'VTI_'.
ECS_VTI_BDC_OPEN_GROUP.

* Create MB0A transaction.
ECS_BDC_BEGIN 'SAPMM07M' '0201'.
ECS_BDC_FIELD: 'RM07M-BWARTWE'   '101',
                                 'RM07M-WERKS'   V_TO_PLANT,
                                 'RM07M-LGORT'    V_TO_STORE,
                                 'RM07M-LFSNR'     V_DELIVERY.
* Save BDC transaction
ECS_BDC_INSERT 'MB0A'.

* Close BDC session
ECS_VTI_BDC_CLOSE_GROUP.
```

These commands automatically create, format and submit a BDC session to ECS. You must set up a ECS Process name of {VTI BDC prefix}_{VTI screen function} with two phases using the ECS workbench (YECS). The first phase is object type NONE and the second phase is object type BDC. You can then subsequently monitor the BDC run and any failures reprocessed using ECS.



Calling BAPIs

As SkyMobile can invoke ABAP exits, you may use any interfacing technique available to SAP for example, IDOC, BAPI, function calls, and BDC. A useful mechanism is to call standard SAP BAPI functions to process transactions. Many BAPIs are available to easily interface to SAP business functionality. All that is required is to retrieve the data values that are required from the screen or SAP database, performing the BAPI function call and monitoring the result.

```
* Perform goods movement using the standard SAP BAPIs.
CALL FUNCTION 'BAPI_GOODSMVT_CREATE'
     exporting
       GOODSMVT_HEADER = V_GOODSMVT_HEADER
        GOODSMVT_CODE   = V_GOODSMVT_CODE
     importing
        GOODSMVT_HEADRET = V_GOODSMVT_HEADRET
     tables
        GOODSMVT_ITEM   = IT_GOODSMVT_ITEM
        RETURN          = IT_RETURN.

READ TABLE IT_RETURN WITH KEY TYPE = 'E'.
IF SY-SUBRC = 0.
   PERFORM F_BAPI_ERROR.
   V_ERROR_FOUND = C_TRUE.
   EXIT.
ENDIF.
```

Submitting IDOCs

You may format and submit IDOC definitions for processing in either synchronous or asynchronous modes. You use Standard SAP IDOC functions for this. You code the declarations, assignments and function module calls in exactly the same way as you would in a normal ABAP program. If more sophisticated IDOC interfacing is required, consider using the SkyConnect IDOC extension for total interface management.

Common Types

You may need to declare variables based on SkyMobile definitions. A common suite of types is provided with the ABAP SDK that you should use to ensure programs are upgrade compatible.

Object type
VTI_FIELD_NAME
VTI_FIELD_VALUE
VTI_TABLE_ROW
VTI_TIMESTAMP
VTI_RETURN_CODE
VTI_SEQUENCE
VTI_TRANSFER_HEADER
VTI_MESSAGE
VTI_STYLE
VTI_COLOUR
VTI_FUNCTION_KEY

Example

```
data v_field_name like vti_field_name
```

4.11.12.6 Working with Local Database Tables

You may specify custom LDB download, refresh and upload ABAP functions against local database definitions. These functions must follow a specific convention to accept parameters and return data back to SkyMobile. You may use these custom functions to perform any ABAP syntax and SkyMobile provides APIs in its ABAP SDK to assist the developer. For more information on the local database concept, see the section in the workbench documentation. SkyMobile provides automated LDB operations and you should explore prior to coding a custom function. You may monitor LDB data transfer requests using the LDB trace facility. See the section in the system management guide for more details.

You may fully integrate LDB definitions into one or more data objects. Refer to the Data Object programming guide (DOB) for more information. In the context of a data object, LDBs provide the underpinning table level data access and storage routines. You may use the DOB SKD command set to directly invoke LDB definitions.

Example Functions:

Standard example functions are provided in the install that you may use as a basis (that is copy the example function). They contain all the logic and techniques necessary to perform local database data transfer processing. The examples use the sample yvti_cn* tables and VTI_SAMPLE_* local database definitions.

Field	Description
/SKY/VTI_EXAMPLE_ DOWNLOAD_FUNC	Custom download routines to extract SAP data.
/SKY/VTI_EXAMPLE_ REFRESH_FUNC	Custom refresh routines to extract SAP data using time stamping.
/SKY/VTI_EXAMPLE_ UPLOAD_FUNC	Custom upload routine to maintain SAP table data.
/SKY/VTI_EXAMPLE_ DELETE_FUNC	Custom routine to demonstrate how to flag data for deletion on the calling Java Server.

Field	Description
/SKY/VTI_EXAMPLE_ REFRESH_SQL	Custom routine performing a refresh using dynamic SQL where conditions.

How SkyMobile Transfers Data between SAP and VTI Java Servers

As both the SAP add-in and the Java Server know the local database layout, only the data components are passed. A 'spreadsheet' technique is used; table containing row/column/data and a control header is passed with the attributes of the request. These layouts are contained in the /SKY/VTI_TSFHD (header), /SKY/VTI_TSFFL (field attribute) and /SKY/VTI_TSFDT (field data) structures. The custom ABAP function formats data for download or refresh and receives data from upload operations using commands. These abstract the internal intricacies from the developer, thus making programming easier. If the data exceeds 40 characters, an object reference issued and this is indicated by the /SKY/VTI_TSFDT-OBJECTDATA flag. The associated object definition is stored in the object header (/SKY/VTI_OBJH) and object data (/SKY/VTI_OBJD).

Identifying Requests

It is important when processing LDB transfer information to identify from which Java Server the request came. The header (e_header) contains a number of fields that the programmer may use.

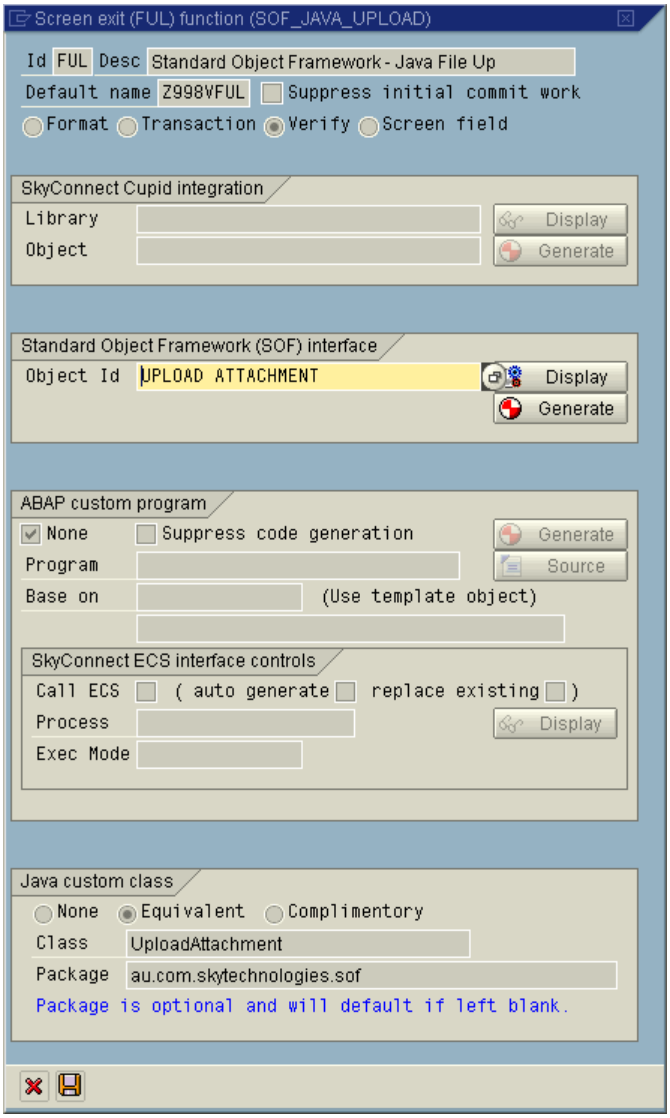
Field	Description
TRNSFR_ MOD	The data transfer mode; DOWNLOAD, REFRESH or UPLOAD.
LOCAL_ DB	The name of the local database that is processed.
VTI_ SERVER	The name of the Java server issuing the request. Configure using the 'VtiServerId =' parameter in the Java server configuration.
SERV_ GROUP	The name of the Java server group issuing the request. SetConfigure using the 'VtiServerGROUP =' parameter in the Java server configuration.

4.11.12.7 Implementing and Using Common SOF Objects

The Standard Object Framework (SOF) enables you to manage a library of common Java and ABAP modules that you may re-use in applications. Sky provides a standard suite of SOF objects as part of the standard installation to do things such as launch attachments, link to a URL, and perform an active directory (ADD) check. In this way, you may re-use the common code and functionality without having to write it. Sky took this approach because of the differences across operating systems to perform system tasks, for example, invoking a URL on Windows is different to a Blackberry.

You may declare your own custom SOF objects using the SOF workbench that is fully described in the Workbench online help. Once you do this, you may refer to the object in the screen exit attributes and the call interface is automatically generated. For example, you may have a common screen exit to

check stock availability; you just need to add this to the SOF library and now wherever you need to perform a stock check you just refer to the SOF object in the screen exit definition.



In the above pop-up, the standard SOF object 'UPLOAD_ATTACHMENT' is defined. You may observe that everything else is disabled. In addition, you may automatically generate the parameter requirements (screen fields). The ABAP end /or Java references are automatically populated from the SOF library definition.

Working with Local Database Tables

You may specify custom LDB download, refresh and upload ABAP functions against local database definitions. These functions must follow a specific convention to accept parameters and return data back to SkyMobile. You may use these custom functions to perform any ABAP syntax and SkyMobile provides APIs in its ABAP SDK to assist the developer. For more information on the local database concept, see the section in the workbench documentation. SkyMobile provides automated LDB operations and you should explore these prior to coding a custom function. You may monitor LDB data transfer requests using the LDB trace facility. See the section in the system management guide for more details.

You may fully integrate LDB definitions into one or more data objects. Refer to the Data Object programming guide (DOB) for more information. In the context of a data object, LDBs provide the underpinning table level data access and storage routines. You may use the DOB SKD command set to directly invoke LDB definitions.

Example Functions:

Standard example functions are provided in the install that you may use as a basis (that is, copy the example function). They contain all the logic and techniques necessary to perform local database data transfer processing. The examples use the sample yvti_cn* tables and VTI_SAMPLE_* local database definitions.

Field	Description
/SKY/VTI_EXAMPLE_ DOWNLOAD_FUNC	Custom download routines to extract SAP data.
/SKY/VTI_EXAMPLE_ REFRESH_FUNC	Custom refresh routines to extract SAP data using time stamping.
/SKY/VTI_EXAMPLE_ UPLOAD_FUNC	Custom upload routine to maintain SAP table data.
/SKY/VTI_EXAMPLE_ DELETE_FUNC	Custom routine to demonstrate how to flag data for deletion on the calling Java Server.

Field	Description
/SKY/VTI_EXAMPLE_REFRESH_SQL	Custom routine performing a refresh using dynamic SQL where conditions.

How SkyMobile transfers data between SAP and VTI Java Servers

As both the SAP add-in and the Java Server know the local database layout, only the data components are passed. A 'spreadsheet' technique is used that is, table containing row/column/data and a control header is passed with the attributes of the request. These layouts are contained in the /SKY/VTI_TSFHD (header), /SKY/VTI_TSFFL (field attribute) and /SKY/VTI_TSFDT (field data) structures. The custom ABAP function formats data for download or refresh and receives data from upload operations using commands. These abstract the internal intricacies from the developer, thus making programming easier. If the data exceeds 40 characters, an object reference is issued and this is indicated by the /SKY/VTI_TSFDT-OBJECTDATA flag. The associated object definition is stored in the object header (/SKY/VTI_OBJH) and object data (/SKY/VTI_OBJD).

Identifying requests

It is important when processing LDB transfer information to identify from which Java Server the request came. The header (e_header) contains a number of fields that the programmer may use.

Field	Description
TRNSFR_MOD	The data transfer mode; DOWNLOAD, REFRESH or UPLOAD.
LOCAL_DB	The name of the local database that is processed.
VTI_SERVER	The name of the Java server issuing the request. Set using the 'VtiServerId =' parameter in the Java server configuration.
SERV_GROUP	The name of the Java server group issuing the request. Set using the 'VtiServerGROUP =' parameter in the Java server configuration.

Primary key traversals

Support for using a primary key traversal technique for processing LDB downloads and refreshes is available. This functionality is aimed at incremental downloads of large volumes of data from SAP and avoids having to extract and save the entire data set first. As well as system level configuration, support is also added to the ABAP API to perform this functionality from custom functions.

See the download and refresh sections for sample code taken from the /sky/vti_example_download_ func and /sky/vti_example_refresh_func usec to demonstrate the primary traversal technique using SDK commands.

The primary advantage of a primary key traversal is that only the extracted data that fits in the specified transfer buffer is retrieved and thus only the required memory footprint is used. In a normal extract scenario, all the data is retrieved and stored into an internal table in SAP and then chunks of the table are returned as per the transfer buffer size. This can cause a problem with memory and performance, since you need to store the entire extracted set up front.

LDB Error Processing

You indicate failures in the LDB refresh by setting a return code (RC) and explanatory message (MSG) in the transfer header (E_HEADER). Take care to set these values in the header structure that is exported back to the Java server. It is recommended that you use return codes starting from 100, since internal codes start from 001-099. The error is logged in the Java server system log.

```
e_header-rc = 100.
e_header-msg = 'Duplicate order'.
```

4.11.12.8 Primary Key Traversals

Support for using a primary key traversal technique for processing LDB downloads and refreshes is available. This functionality is aimed at incremental downloads of large volumes of data from SAP and avoids having to extract and save the entire data set first. As well as system level configuration, support is also added to the ABAP API to perform this functionality from custom functions.

See the download and refresh sections for sample code taken from the /sky/vti_example_download_func and /sky/vti_example_refresh_func usec to demonstrate the primary traversal technique using SDK commands.

The primary advantage of a primary key traversal is that only the extracted data that fits in the specified transfer buffer is retrieved and thus only the required memory footprint is used. In a normal extract scenario, all the data is retrieved and stored into an internal table in SAP and then chunks of the table are returned as per the transfer buffer size. This can cause a problem with memory and performance, since you need to store the entire extracted set up front.

4.11.12.9 LDB Error Processing

You indicate failures in the LDB refresh by setting a return code (RC) and explanatory message (MSG) in the transfer header (E_HEADER). Take care to set these values in the header structure that is exported back to the Java server. It is recommended that you use return codes starting from 100, since internal codes start from 001-099. The error is logged in the Java server system log.

Example

```
e_header-rc = 100.
e_header-msg = 'Duplicate order'
```

> ***Important:*** If an error is triggered, SkyMobile saves no data. You should be wary of when database data is committed and the importance of transaction boundaries to avoid data integrity issues.

4.11.12.10 LDB Download Operation

A download operation replaces all data in the LDB table. It is called when the table is first created, manually through the Web status page; when the Java server starts (if configured) or through a LDBDOWNLOAD heartbeat command is received. You may use all data or a custom download function. The custom download function may perform all the functionality of a refresh function (refer to the following refresh section for details); the only difference is the timing of the request. Refer to the /SKY/VTI_EXAMPLE_DOWNLOAD_FUNC for details.

Example using Primary Key Traversal

The primary key traversal technique is recommended for high volume downloads, for example, thousands of records. The following code example demonstrates how to use the commands to perform this effectively.

```
* Initialisation
  E_HEADER = I_HEADER.
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.

  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Indicate to the SAP LDB manager that we are doing a primary key
* traversal.
  vti_ldb_primary_key_traversal f_header.

* Get the last stored primary key (if any) and build a dynamic SQL
* where condition. Any stored primary key is then removed.
  clear /SKY/YVTI_CNTCT.
  vti_ldb_import_primary_key f_header.        "<- get stored key
  refresh it_sql_where.                       "<- build dynamic where
  append 'deleted = space' to it_sql_where.       " logical delete
```

```
ind
  vti_ldb_primary_key_where f_header it_sql_where. "primary key
  vti_ldb_delete_primary_key f_header.        "<- remove stored key.

  SELECT *
    FROM /SKY/YVTI_CNTCT
   where (it_sql_where)                  "<- dynamic where condition
   order by primary key.                 "<- force primary key sequence

*         assign the next row number.
          VTI_LDB_CALC_NEXT_ROW_NUMBER.

*         load the transfer data from the structure. Note that the
field
*         names in the structure must match the LDB definition.
          vti_ldb_set_structure /SKY/YVTI_CNTCT.

*         check to see if we have exceeded the maximum transfer buffer
*         size specified by the Java Server (0 = unlimited). If the max
*         buffer size has been reached, then we store the current
*         primary key field values for next time and exit the select.
We
*         also indicate to the Java Server that more data is available
*         i.e. call back.
          VTI_LDB_check_max_buffer_size f_header.
          if v_vti_*rc ne 0.
             vti_ldb_append_primary_key /SKY/YVTI_CNTCT.
             vti_ldb_export_primary_key f_header.
             vti_ldb_set_more_data f_header.  "<- More data is
available
             exit.
          endif.
  ENDSELECT.

  if e_header-rc is initial.
```

```
*    copy our working data set to pass back to VTI.
     VTI_LDB_COPY_DATA.
  endif.
```

4.11.12.11 LDB Refresh Operation

The refresh operation is intended to incrementally update the LDB table with any changes. Refresh calls may be based on an interval, for example, every five minutes, scheduled to occur at a specific time. For example, 18:00:00 and/or initiated through heartbeat command; 'push'. The timestamp and sequence number of the last refresh operation is passed in the LASTTMSTMP field in the header. You may use these to perform incremental refreshes (see the following section on incremental data transfer operations).

Example using Table Level Assignments

This technique matches and copies the fields from the input structure into the LDB data container, that is, at structure level. Fields are matched by name.

```
INCLUDE YVTISDK2.     "<- include the SDK to use standard commands


* Initialisation
  E_HEADER = I_HEADER.


* Get the local database field definitions
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

  SELECT *
    FROM YVTI_CNTCT.

*         Assign fields to data table using by offset method.
          VTI_LDB_CALC_NEXT_ROW_NUMBER.
          VTI_LDB_SET_STRUCTURE YVTI_CNTCT.
  ENDSELECT.
```

```
  IF SY-SUBRC = 0.
     VTI_LDB_COPY_DATA IT_DATA.
     VTI_LDB_SET_TOTAL_ROWS E_HEADER-TOTAL_ROWS.
  ELSE.
*    Only do this if you want messages to appear in the VTI Java
server log
     E_HEADER-RC = '101'.
     E_HEADER-MSG = 'No data was selected for the refresh operation'.
  ENDIF.
```

Example using Primary Key Traversal

The primary key traversal technique is recommended for high volume refreshes, for example, thousands of records. The following code example demonstrates how to use the commands to perform this effectively.

```
* Indicate to the SAP LDB manager that we are doing a primary key
* traversal.
  vti_ldb_primary_key_traversal f_header.


* Get the last stored primary key (if any) and populate key fields for
* greater/than selection. Any stored primary key is then removed.
  clear /SKY/YVTI_CNTCT.
  vti_ldb_import_primary_key f_header.


* Get the last stored primary key (if any) and build a dynamic SQL
* where condition. Any stored primary key is then removed.

  clear /SKY/YVTI_CNTCT.
  vti_ldb_import_primary_key f_header.        "<- get stored key
  refresh it_sql_where.                       "<- build dynamic where
  append 'deleted = space' to it_sql_where.         " logical delete
  concatenate c_quote F_HEADER-LASTTMSTMP c_quote   " timestamp
```

```
         into it_sql_where.

  concatenate 'and LDBTSTAMP GE' it_sql_where
         into it_sql_where separated by space.

  append it_sql_where.
  vti_ldb_primary_key_where f_header it_sql_where.  " primary key
  vti_ldb_delete_primary_key f_header.        "<- remove stored key.


* Select all contacts changed since the last refresh.
  SELECT *
    FROM /SKY/YVTI_CNTCT
   where (it_sql_where)                 "<- dynamic where condition
   order by primary key.                "<- force primary key sequence

*        assign the next row number.
         VTI_LDB_CALC_NEXT_ROW_NUMBER.

*        load the transfer data from the structure. Note that the
field
*        names in the structure must match the LDB definition.
         VTI_LDB_SET_STRUCTURE /SKY/YVTI_CNTCT.

*        check to see if we have exceeded the maximum transfer buffer
*        size specified by the Java Server (0 = unlimited). If the max
*        buffer size has been reached, then we store the current
*        primary key field values for next time and exit the select.
We
*        also indicate to the Java Server that more data is available,
call
*  call back.
         VTI_LDB_check_max_buffer_size f_header.
         if v_vti_*rc ne 0.
            vti_ldb_append_primary_key /SKY/YVTI_CNTCT.
```

```
            vti_ldb_export_primary_key f_header.
            vti_ldb_set_more_data f_header.  "<- More data is
available
            exit.
         endif.
  ENDSELECT.
```

Example using Field Level Assignments

You may configure individual fields using this method. As well as the field name and value, you may specify a special flag known as an 'update indicator' that determines the data maintenance mode. You may configure this flag to either space/I (Insert), U (Update) or D (Delete). Refer to the following sections on inserting, updating and deleting entries for more information.

```
INCLUDE YVTISDK2.     "<- include the VTI SDK to use standard commands

* Initialisation
  E_HEADER = I_HEADER.


* Get the local database field definitions
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.


* Select data and load into transfer data set.
  SELECT *
    FROM YVTI_CNTCT.

         VTI_LDB_CALC_NEXT_ROW_NUMBER.
         VTI_LDB_SET_FIELD_DATA: 'NAME'      YVTI_CNTCT-NAME
```

```
SPACE,
                                  'PHONE'     YVTI_CNTCT-PHONE
SPACE,
                                  'EMAIL'     YVTI_CNTCT-EMAIL
SPACE,
                                  'CONTACTED' YVTI_CNTCT-CONTACTED
SPACE,
                                  'TYPE'      YVTI_CNTCT-TYPE
SPACE,
                                  'COUNTRY'   YVTI_CNTCT-COUNTRY
SPACE,
                                  'TIMESTAMP' YVTI_CNTCT-LDBTSTAMP
SPACE,
                                  'SEQUENCE'  YVTI_CNTCT-LDBSEQNUM
SPACE,
                                  'TESTPACK'  YVTI_CNTCT-TESTPACK
SPACE.
  ENDSELECT.

  IF SY-SUBRC = 0.
     VTI_LDB_COPY_DATA IT_DATA.
     VTI_LDB_SET_TOTAL_ROWS E_HEADER-TOTAL_ROWS.
  ELSE.
*    Only do this if you want messages to appear in the VTI Java
server log
     E_HEADER-RC = '101'.
     E_HEADER-MSG = 'No data was selected for the refresh operation'.
  ENDIF.
```

Inserting Data

By default, the Java server inserts rows that don't exist and totally replaces rows that do exist, based on the primary key. If a field is not specified, its value is defaulted to its initialized value. You may explicitly set the 'update indicator' to 'I" or leave it as blank to default to 'I'. You can prevent duplicates by specifying the vti_ldb_no_duplicates command prior to any set structure/field command; then if a

duplicate is detected, an error is written to the Java Server log and all updates are rolled back. You can suppress the error and continue processing using the vti_ldb_set_error_severity 'I' command, that is, make the duplicate messages information only.

Updating/Inserting Data

By default, the Java server replaces any existing entry, based on the primary key. You may update specific fields by setting the update indicator (3rd parm) on the vti_ldb_set_field_data to 'U'. Any other fields with no update indicator (space) are then used only as key fields. In this way, restricted or generic updates are possible. To ensure that a specified row exists, use the vti_ldb_row_must_exist command, prior to each set structure/field command or else a new row is inserted. In the following example, only the email address is updated (EMAIL) where the contact name (NAME) matches.

Updating/Inserting Data Example

```
SELECT *
    FROM YVTI_CNTCT
   WHERE UPD_FLAG = C_TRUE.
         VTI_LDB_CALC_NEXT_ROW_NUMBER.
         VTI_LDB_SET_FIELD_DATA: 'NAME' YVTI_CNTCT-NAME   ' '.
         VTI_LDB_SET_FIELD_DATA: 'EMAIL' YVTI_CNTCT-EMAIL 'U'.
  ENDSELECT.
```

Deleting Entries

The refresh function may delete any existing entries by setting the update indicator (3rd parm) on the vti_ldb_set_field_data to 'D'. You should do this for all fields that make up the key or the range of the delete. For example, vti_ldb_set_field_data 'NAME' 'JOHN SMITH' 'D' causes the JOHN SMITH contact to be deleted from the local database. Generic deletions are possible using a partial or non key field. See the /SKY/VTI_EXAMPLE_DELETE_FUNC function for an example on how to delete local database data.

Deleting Entries Example

```
SELECT *
    FROM YVTI_CNTCT
   WHERE DEL_FLAG = C_TRUE.
         VTI_LDB_CALC_NEXT_ROW_NUMBER.
         VTI_LDB_SET_FIELD_DATA: 'NAME' YVTI_CNTCT-NAME  'D'.
  ENDSELECT.
```

Clearing all Existing Table Contents

The transfer header (E_HEADER) contains a field, CLRCNTENTS that you may set to true (X) to indicate to the Java server that all existing contents of the LDB are to be deleted, before the refresh data is loaded. You may also use this mechanism just to clear an LDB, by configuring the indicator and passing a null it_data table back. You may use the VTI_LDB_CLEAR_CONTENTS command to set this indicator.

4.11.12.12 LDB Upload Operation

An upload of data may be triggered by, scheduled to occur on an interval or specific time and/or triggered by a LDBUPLOAD heartbeat command. You may also trigger a SkyConnect ECS process to perform post upload processing, for example, execute SAP interfaces.

Example using Table Level Automatic Update

The underlying SAP table is automatically modified using all the uploaded data rows using a single command, vti_ldb_modify_sap_table.

Table Level Automatic Update Example

```
INCLUDE /SKY/YVTISDK.     "<- include the VTI SDK to use standard
commands

* Initialisation
  E_HEADER = I_HEADER.

* Get the local database field definitions
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Load the ldb field data into a internal VTI table for processing.
  VTI_LDB_LOAD_DATA IT_DATA.

* Insert/Modify the SAP table associated with the ldb definition.
  VTI_LDB_MODIFY_SAP_TABLE.
```

Example using Table Level Assignment

This technique involves looping through the uploaded rows, populating the physical table structure and then performing the database insert, update or modify. The advantage of this is that extra row field formatting is possible.

```
INCLUDE /SKY/YVTISDK.     "<- include the VTI SDK to use standard
commands


* Initialisation
  E_HEADER = I_HEADER.

* Get the local database field definitions
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Load the ldb field data into a internal VTI table for processing.
  VTI_LDB_LOAD_DATA IT_DATA.


* Loop through the uploaded data rows and modify the SAP table.
  DO F_HEADER-TOTAL_ROWS TIMES.
     VTI_LDB_SET_ROW_NUMBER SY-INDEX.
     VTI_LDB_GET_STRUCTURE YVTI_CNTCT.
     MODIFY YVTI_CNTCT.
  ENDDO.
```

Field Level Assignment

You can loop through the uploaded rows and retrieve the values field by field if you wish. This technique is useful if there are multiple target SAP tables and /or the target field names are different.

```
INCLUDE /SKY/YVTISDK.     "<- include the VTI SDK to use standard
commands

* Initialisation
  E_HEADER = I_HEADER.

* Get the local database field definitions
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Load the ldb field data into a internal VTI table for processing.
  VTI_LDB_LOAD_DATA IT_DATA.

* Loop through the uploaded data rows and modify the SAP table.
  DO F_HEADER-TOTAL_ROWS TIMES.
     VTI_LDB_SET_ROW_NUMBER SY-INDEX.
     VTI_LDB_GET_FIELD_DATA: 'NAME'      YVTI_CNTCT-NAME,
                             'PHONE'     YVTI_CNTCT-PHONE,
                             'EMAIL'     YVTI_CNTCT-EMAIL,
                             'CONTACTED' YVTI_CNTCT-CONTACTED,
                             'TYPE'      YVTI_CNTCT-TYPE,
                             'COUNTRY'   YVTI_CNTCT-COUNTRY.
     MODIFY YVTI_CNTCT.
  ENDDO.
```

Clearing Existing SAP Data First

To clear any existing data in the SAP table prior to the upload operation, configure 'Clear existing data' in the LDB definition. The SQL delete is performed prior to the user upload function call. You may use this facility to clear SAP tables from the Java server by configuring a 'Clear existing data' and passing a null list of data. You may use the vli_ldb_clear_contents command to set this indicator.

Responding with LDB Maintenance Requests (reply data)

It is sometimes useful to return updates to the Java server once an upload is successfully processed. To do this, the custom function module must create entries in IT_DATA in exactly the same manner as a (see /SKY/VTI_EXAMPLE_REFRESH_FUNC for details). You must issue the VTI_RESPOND_WITH_DATA command to prevent the upload manager from initializing IT_DATA (transfer data) on return. See the example refresh function for details on the available commands. This option enables the 'two phase commitment' of upload operations, for example, update / delete LDB data on the Java Server when an upload is successfully processed in SAP.

> ***Note:*** You need have the reply updates against the LDB.

LDB Upload Interface

You can automatically trigger a SkyConnect ECS process after a successful LDB upload event. All the data processed is passed to the ABAP exit through ECS exported data definitions. The process to execute is configured in the LDB upload attributes. See the Workbench online help for more details. The following examples show how you receive the data; you may then process against it accordingly. This is a useful mechanism to automatically trigger integration processing once the data is uploaded from a remote Java Server.

Example LDB Upload Exit

The LDB upload header and the primary key data values are exported to ECS memory and thus an ABAP program may use to perform additional processing. See the sample ABAP code /SKY/YVTI8022 for details of how to retrieve the data. Note that the primary key values are as received by the SAP LDB processor and so if these are further filtered or manipulated by a custom function, you may need to perform your own ECS start processing. See the "VTI SAMPLE CONTACTS" LDB example.

```
include: /sky/yecssdk.

data: v_tsfhd  like /sky/yvti_tsfhd,   "<- LDB upload header
      it_tsfpk like /sky/yvti_tsfpk    "<- LDB upload primary key
values
               occurs 0 with header line.

initialization.

  ecs_import_parameters.
  if v_ecs_*rc ne 0.
     write 'ECS import parameters failed'.
     exit.
  endif.


start-of-selection.
```

```
* Import LDB transfer header and primary key values.
  ecs_import_data: v_tsfhd,
                   it_tsfpk.

  if v_ecs_*rc ne 0.
     write 'ECS import data failed'.
     exit.
  endif.


end-of-selection.

  write: /01 'LDB',      25 v_tsfhd-LOCAL_DB,
         /01 'Group',    25 v_tsfhd-server_group,
         /01 'Server',   25 v_tsfhd-vti_server.

  skip 1.

* Loop through the primary key values.
  loop at it_tsfpk.
       at new row_number.
          write: /01 'Row:', 06 it_tsfpk-row_number.
       endat.
       write: /06 it_tsfpk-column_num,      "6
                 12 it_tsfpk-field_id,      "32
                 44 it_tsfpk-field_val(200).
  endloop.
```

4.11.12.13 Incremental Data Transfer Operations

The local database download, refresh and upload operations provide an effective timestamp mechanism to help detect and manage incremental downloads and uploads of data. It works like this:

- The ABAP function declares the timestamp to use using the vti_ldb_set_timestamp command.
- The timestamp is passed back to the Java server who stores it against the local database definition, and passes it back with the next download/refresh or upload operation.
- The default LDB operation or custom function module then uses the last timestamp and current timestamp range to determine any new updates that occurred in SAP.

The Java server may also perform automatic timestamp updates on the LDB, by specifying timestamp fields in the local database table as char(14). You have to configure these against the LDB definition; the fields in the table to use. The Java servers then automatically updates the database rows in the refresh or upload data set, once the corresponding transfer operation is complete.

You may use the following commands to control timestamps / sequence numbers: {h} = header (for example, e_header), {f} = input field.

Command	Description
vti_ldb_set_timestamp {h} {f}.	Configure the timestamp value from a field (type vti_timestamp).
vti_ldb_get_timestamp {h}	Retrieve the current timestamp into v_vti_*timestamp.
vti_ldb_calculate_timestamp	Configure the timestamp using the current SAP date and time.

Note: Download and upload operations always configure the timestamp. You have to configure the Refresh operations timestamp using the SDK.

Using Dynamic SQL where Conditions

You may dynamically generate your own SQL where conditions and invoke a LDB operation (or recursively) to select a subset of data depending on runtime criteria. Be aware that you may configure additional SQL where conditions against the LDB definition using the workbench (see the workbench online guide for details). An example function module /SKY/VTI_EXAMPLE_REFRESH_SQL demonstrates how the SDK WHERE commands are used to generate extra selection conditions and then re-call the LDB definition to select the data.

> ***Important:*** You may use the Dynamic SQL conditions in download and refresh custom function modules.

```
* include /SKY/YVTIsdk.     "<- include the VTI SDK to use API
commands.

* Initialisation
  E_HEADER = I_HEADER.
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Check to see if a where condintion has been passed. If it has,
* execute the dynamic SQL ... otherwise construct it and call
* ourselves again one or more times, appending the result set.
  if it_where[] is initial.
*    Get current system timestamp. Returned into v_vti_*timestamp.
*    For use in where condition.
     vti_ldb_current_timestamp.
*    1st call.
     vti_ldb_where_initialise.   "<- clear the where conditions.
     vti_ldb_where_append 'LDBTSTAMP' 'GE' e_header-LASTTMSTMP space.
     vti_ldb_where_append 'LDBTSTAMP' 'LT' v_vti_*timestamp    space.
```

```
     vti_ldb_where_append 'DELETED'   'EQ' space                space.
     vti_ldb_where_append 'TYPE'      'EQ' 'CUSTOMER'           space.
     vti_ldb_refresh e_header.   "<- call ourselves again with SQL
where
     check v_vti_*rc is initial. "<- exit if any errors.
     vti_ldb_append_data.        "<- add result data to the returned
set
*    2nd call with different criteria.
     vti_ldb_where_initialise.   "<- clear the where conditions.
     vti_ldb_where_append 'LDBTSTAMP' 'GE' e_header-LASTTMSTMP space.
     vti_ldb_where_append 'LDBTSTAMP' 'LT' v_vti_*timestamp    space.
     vti_ldb_where_append 'DELETED'   'EQ' space                space.
     vti_ldb_where_append 'TYPE'      'EQ' 'VENDOR'             space.
     vti_ldb_refresh e_header.   "<- call ourselves again with SQL
where
     check v_vti_*rc is initial. "<- exit if any errors.
     vti_ldb_append_data.        "<- add result data to the returned
set
*    Set the latest timestamp in the transfer header for next time.
     vti_ldb_calculate_timestamp e_header.
  else.
*    Dynamically generate the SQL where condition from the input
*    internal where criteria passed in.
     vti_ldb_load_where_data it_where.
     vti_ldb_where_generate.
*    Select the data and populate the returned data set.
     select *
       from /SKY/YVTI_CNTCT
      where (it_VTI_SQL_where).
            VTI_LDB_CALC_NEXT_ROW_NUMBER.
            VTI_LDB_SET_STRUCTURE /SKY/YVTI_CNTCT.
     endselect.
  endif.
```

4.11.12.14 Using Dynamic SQL where Conditions

You may dynamically generate your own SQL where conditions and invoke a LDB operation (or recursively) to select a subset of data depending on runtime criteria. Also be aware that you may configure additional SQL where conditions against the LDB definition using the workbench (see the workbench online guide for details). An example function module /SKY/VTI_EXAMPLE_REFRESH_SQL demonstrates how the SDK WHERE commands are used to generate extra selection conditions and then re-call the LDB definition to select the data.

> ***Important:*** You may use dynamic SQL conditions only in download and refresh custom function modules.

```
* include /SKY/YVTIsdk.     "<- include the VTI SDK to use API
commands.

* Initialisation
  E_HEADER = I_HEADER.
  VTI_LDB_SELECT_DEFINITION E_HEADER-LOCAL_DB.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Local database not found'.
     EXIT.
  ENDIF.

* Check to see if a where condintion has been passed. If it has,
* execute the dynamic SQL ... otherwise construct it and call
* ourselves again one or more times, appending the result set.
  if it_where[] is initial.
*    Get current system timestamp. Returned into v_vti_*timestamp.
*    For use in where condition.
     vti_ldb_current_timestamp.
*    1st call.
     vti_ldb_where_initialise.   "<- clear the where conditions.
     vti_ldb_where_append 'LDBTSTAMP' 'GE' e_header-LASTTMSTMP space.
     vti_ldb_where_append 'LDBTSTAMP' 'LT' v_vti_*timestamp    space.
```

```
     vti_ldb_where_append 'DELETED'   'EQ' space                space.
     vti_ldb_where_append 'TYPE'      'EQ' 'CUSTOMER'           space.
     vti_ldb_refresh e_header.   "<- call ourselves again with SQL
where
     check v_vti_*rc is initial. "<- exit if any errors.
     vti_ldb_append_data.        "<- add result data to the returned
set
*    2nd call with different criteria.
     vti_ldb_where_initialise.   "<- clear the where conditions.
     vti_ldb_where_append 'LDBTSTAMP' 'GE' e_header-LASTTMSTMP space.
     vti_ldb_where_append 'LDBTSTAMP' 'LT' v_vti_*timestamp    space.
     vti_ldb_where_append 'DELETED'   'EQ' space                space.
     vti_ldb_where_append 'TYPE'      'EQ' 'VENDOR'             space.
     vti_ldb_refresh e_header.   "<- call ourselves again with SQL
where
     check v_vti_*rc is initial. "<- exit if any errors.
     vti_ldb_append_data.        "<- add result data to the returned
set
*    Set the latest timestamp in the transfer header for next time.
     vti_ldb_calculate_timestamp e_header.
  else.
*    Dynamically generate the SQL where condition from the input
*    internal where criteria passed in.
     vti_ldb_load_where_data it_where.
     vti_ldb_where_generate.
*    Select the data and populate the returned data set.
     select *
       from /SKY/YVTI_CNTCT
      where (it_VTI_SQL_where).
            VTI_LDB_CALC_NEXT_ROW_NUMBER.
            VTI_LDB_SET_STRUCTURE /SKY/YVTI_CNTCT.
     endselect.
  endif.
```

4.11.12.15 Working with Data Objects

You may specify custom DOB download, refresh and upload ABAP functions against data object definitions. These functions must follow a specific convention to accept parameters and return data back to SkyMobile. You may use these custom functions to perform any ABAP syntax and SkyMobile provides APIs in its ABAP SDK to assist the developer. For more information on the data object concept, see the section in the workbench documentation. SkyMobile provides automated DOB operations and you should explore these prior to coding a custom function. You may monitor DOB data transfer requests using the data object trace facility. See the section in the system management guide for more details.

Data objects normally utilize LDB definitions for all data management. They are effectively a 'front end' that enables multiple LDB tables and their relationships to be defined and processed. An obvious advantage of a data object is that all related data is processed as a single transaction and that a single data object may contain multiple LDB requests, thus significantly cutting down on the calls between SAP and SkyMobile Java servers. Data object interval and time scheduling options supersedes any LDB scheduling that is nullified down on the Java server. It is important to note that data objects work in a very similar fashion to LDBs except that multiple tables and their data definitions are processed and passed between SAP and the Java server.

> ***Important:*** The data object root tables (those with no parent) are the transaction boundary, that is, all child data is retrieved for a parent row irrespective of the transfer buffer size. Thus only use transaction tables, for example, ORDER as the root where there are potentially large volumes of data and complex child relationships involved.

The following areas support custom development:

- Download and refresh operations
- LDB upload operation
- Dynamically invoking data objects
- Cascading Updates
- Data profile operations

For a list of all data object SDK commands, refer to the DOB Command Reference.

Download and Refresh Operations

These are very similar in concept, except that the refresh typically utilises a timestamp to identify changes. The following example code shows how a simple nested SQL selection of data is used to populate data object table definitions and dependencies. See example function /SKY/VTI_EXAMPLE_DOB_DOWNLOAD. As well as coding your own selects, you may also dynamically invoke LDB definitions with additional SQL conditions and then perform a append data. In the case of refresh operations, you would utilise timestamp and delete indicator fields to select data in the same way you do custom programming with LDB's.

Example

In this example, a data object is populated directly from SAP tables. The basic concept is that you select the data and then position on the target data object table and append the data to the data object.

```
include /SKY/YVTIsdk.     "<- include the VTI SDK to use API commands.

tables: /sky/yvti_cnord, /sky/yvti_cnitm, /sky/yvti_cncpy,
        /sky/yvti_cnprd, /sky/yvti_cnicm, /sky/yvti_cncmp.

* Initialisation
  E_HEADER = I_HEADER.
  VTI_DOB_SELECT_DEFINITION E_HEADER-object_name.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Data Object not found'.
     EXIT.
  ENDIF.

* Select orders.
  select *
    from /sky/yvti_cnord
   where deleted = space.
*        load order data
         VTI_DOB_POSITION_TABLE 'VTI_SAMPLE_ORDER'.
```

```
          vti_dob_set_structure /sky/yvti_cnord.
          vti_dob_append_data.
*         select company data.
          select *
            from /sky/yvti_cncpy
           where company = /sky/yvti_cnord-company.
*               load company data
                VTI_DOB_POSITION_TABLE 'VTI_SAMPLE_COMPANY'.
                vti_dob_set_structure /sky/yvti_cncpy.
                vti_dob_append_data.
          endselect.
*         select items.
          select *
            from /sky/yvti_cnitm
           where order_number = /sky/yvti_cnord-order_number.
*               load item data
                VTI_DOB_POSITION_TABLE 'VTI_SAMPLE_ORDER_ITEM'.
                vti_dob_set_structure /sky/yvti_cnitm.
                vti_dob_append_data.
*               select product data
                select *
                  from /sky/yvti_cnprd
                 where product = /sky/yvti_cnitm-product.
*                     load product data
                      VTI_DOB_POSITION_TABLE 'VTI_SAMPLE_PRODUCT'.
                      vti_dob_set_structure /sky/yvti_cnprd.
                      vti_dob_append_data.
                endselect.
*               select item components.
                select *
                  from /sky/yvti_cnicm
                 where order_number = /sky/yvti_cnitm-order_number
                   and item_number  = /sky/yvti_cnitm-item_number.
*                     load item component data
                      VTI_DOB_POSITION_TABLE 'VTI_SAMPLE_ITEM_COMP'.
```

```
                        vti_dob_set_structure /sky/yvti_cnicm.
                        vti_dob_append_data.
*                       select component data
                        select *
                          from /sky/yvti_cncmp
                         where component = /sky/yvti_cnicm-component.
*                            load component data
                             VTI_DOB_POSITION_TABLE
                                              'VTI_SAMPLE_COMPONENT'.
                             vti_dob_set_structure /sky/yvti_cncmp.
                             vti_dob_append_data.
                        endselect.
                endselect.
         endselect.
  endselect.
```

DOB Upload Operation

A data object upload request from a Java server may consist of one or more of the data object tables, depending on what is changed. Thus, the code needs to be flexible to detect what is changed and process it accordingly. The internal table it_tsfhd contains a list of the LDB headers. Each entry contains the datarow_st and datarow_en attributes that contain the physical row references in it_data. You may customise how you traverse these tables a number of ways that best suits your purpose. The SDK contains a number of DOB commands that you may use to simplify the process. See example function module: /SKY/VTI_EXAMPLE_DOB_UPLOAD.

Example

The following example loops through and positions on each data object table (VTI_DOB_POSITION_TABLE). The associated data for the table is loaded from the function parameters into the SDK internal work areas. If there is no data, the table is ignored. You may then use the LDB commands to maintain the data as per normal LDB processing.

```
include /SKY/YVTIsdk.     "<- include the VTI SDK to use API commands.


 tables: /sky/yvti_cnord, /sky/yvti_cnitm, /sky/yvti_cncpy,
        /sky/yvti_cnprd, /sky/yvti_cnicm, /sky/yvti_cncmp.


* Initialisation
  E_HEADER = I_HEADER.
  VTI_DOB_SELECT_DEFINITION E_HEADER-object_name.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Data Object not found'.
     EXIT.
  ENDIF.


* This technique effectively loops through the data object LDB tables
* and selectively copies the associated data into the SDK work areas.
```

```
* then you may use LDB commands (as per LDB custom upload) or directly
* invoke LDB upload processing.
  loop at it_vti_*dobt. "<- internal table list from select
definition.

*       Position on the data object table definition.
        vti_dob_position_table it_vti_*dobt-table_name.
        check v_vti_*rc = 0.

*       Selectively load the table data (if any) into the SDK work
area.
        vti_dob_load_table_data.
        check not it_vti_*ldb_data[] is initial.

*       Process all the data rows for the table as per normal LDB
command
*       processing, or invoke the LDB upload routine. For LDB command
*       examples, see /SKY/VTI_EXAMPLE_UPLOAD_FUNC.
        VTI_LDB_MODIFY_SAP_TABLE.

  endloop.
```

DOB Upload Interface

This is a useful mechanism to automatically trigger integration processing once data has been uploaded from a remote Java Server. The ABAP exit program is passed the data object header, a list of LDB headers and a list of the primary key values for each table. An example program /SKY/YVTI8023 and ECS process (VTI EXAMPLE DOBUPL) is provided to demonstrate how to configure and receive the parameters.

```
REPORT  /SKY/YVTI8023 line-size 255.

*---------------------------------------------------------------------
* This sample DOB Upload ECS program demonstrates how to receive the
DOB
* parameters. It simply imports and lists attributes of the DOB header
* (v_tdobh), LDB headers (it_tsfhd) and primary key values (it_tsfpk).
* The primary key values are as passed to the DOB upload routine as an
* indication of what data has been updated. See the associated
* structures for more details. Sample ECS process: VTI_EXAMPLE_
DOBUPL'.
*
* Note:
* -----
* The order of the LDB tables in internal table it_tsfhd relates to
the
* logical table_num in the primary key internal table it_tsfpk e.g.
the
* primary key data for the 1st LDB entry will have a table_num of 01.
*---------------------------------------------------------------------
--

include: /sky/yecssdk.

data: v_tdobh           like /sky/yvti_tdobh,  "<- DOB header
      v_table_num       like /sky/yvti_tsfpk-table_num,
      it_tsfhd          like /sky/yvti_tsfhd   "<- LDB headers
                        occurs 0 with header line,
```

```
      it_tsfpk          like /sky/yvti_tsfpk   "<- LDB primary key
data
                        occurs 0 with header line,
      v_error_found(1)  type c.

constants: c_true(1)    type c value 'X'.

* ---------------------------------------------------------------------
--
* Main processing.
* ---------------------------------------------------------------------
--

initialization.
  perform f_initialise.


start-of-selection.
  perform f_process_key_data.

* ---------------------------------------------------------------------
--
* Initialisation.
* ---------------------------------------------------------------------
--
form f_initialise.

  clear v_error_found.


* Get the ECS run details.
  ecs_import_parameters.
  if v_ecs_*rc ne 0.
     write 'ECS import parameters failed'.
     v_error_found = c_true.
```

```
    exit.
  endif.

* Retrieve the exported data definitions.
  ecs_import_data: v_tdobh,
                   it_tsfhd,
                   it_tsfpk.

  if v_ecs_*rc ne 0.
     write 'ECS import data failed'.
     v_error_found = c_true.
     exit.
  endif.

endform.

* ---------------------------------------------------------------------
--
* Process the primary key data.
* ---------------------------------------------------------------------
--
form f_process_key_data.

  check v_error_found is initial.

* List DOB header attributes:
  write: /01 'DOB',      25 v_tdobh-object_name color col_key,
         /01 'Group',    25 v_tdobh-server_group,
         /01 'Server',   25 v_tdobh-vti_server.

* List LDB header details.
  clear v_table_num.
  loop at it_tsfhd.
       add 1 to v_table_num.   "<- logical table number sequence
       skip 1.
```

```
      write: /01 it_tsfhd-local_db color col_group,        "32
              34 v_table_num.                              "2

*     List the primary key values for the current table (table_num).
*     Note that the table_num is the logical sequence of the LDB
*     header i.e. it_tsfhd.
      skip 1.
      loop at it_tsfpk where table_num = v_table_num.
           write: /01 it_tsfpk-row_number,      "6
                     08 it_tsfpk-column_num,    "6
                     15 it_tsfpk-field_id,      "32
                     48 it_tsfpk-field_val(200).
      endloop.
  endloop.

endform.
```

Dynamically Invoking Data Objects

It is possible to programmatically invoke one or more data object definitions to retrieve and process data. You may also have a dummy 'shell' data object that has no tables defined i.e. it purely acts as a broker to invoke other data objects. This provides the flexibility to dynamically determine selection criteria and processing options at run time. See the example function /SKY/VTI_EXAMPLE_DOB_REF_SQL.

Example:

This simple example selects data object definition 'VTI_SAMPLE_ORDERS', sets some additional where criteria, invokes the data object and returns the result data set back to the caller.

```
include /SKY/YVTIsdk.     "<- include the VTI SDK to use API commands.

* Initialisation
  E_HEADER = I_HEADER.

* Select the new DOB definition to execute.
  VTI_DOB_SELECT_DEFINITION 'VTI_SAMPLE_ORDERS'.
  IF V_VTI_*RC NE 0.
     E_HEADER-RC = '100'.
     E_HEADER-MSG = 'Data object not found'.
     EXIT.
  ENDIF.

* Create a dynamic SQL condition to select a specific order.
  vti_dob_position_table 'VTI_SAMPLE_ORDER'.
  vti_dob_where_initialise.
  vti_dob_where_append 'ORDER_NUMBER' 'EQ' 'WOOLWORTHS' space.

* Execute the alternative DOB and return the result.
  vti_dob_execute.
  vti_dob_copy_data.
```

Cascading Updates

There are useful commands to automatically cascade delete indicator, timestamp and physical deletions on data object data. This saves programming effort and helps abstract the data object definition from code dependencies. The cascading takes all dependent data and binary objects into account and only processes relationships based on the primary keys of both tables; only process the transactional data and exclude any master data references.

You may use the following commands:

```
VTI_CASCADE_DELETE
```

OR

```
VTI_CASCADE_DELETE_DB
```

OR

```
VTI_CASCADE_TIMESTAMP
```

See the DOB command reference for more information on the usage and parameters required.

Example code

Time stamp cascade upwards

```
VTI_DOB_WHERE_INITIALISE.

VTI_DOB_WHERE_append 'order_number' 'eq' v_order_number ''.
vti_dob_cascade_timestamp 'VTI_SAMPLE_ORDERS' 'VTI_SAMPLE_ORDER' v_
timestamp '' 'UP'.
```

Logical delete downwards

```
VTI_DOB_WHERE_INITIALISE.

VTI_DOB_WHERE_append 'order_number' 'eq' v_order_number ''.
vti_dob_cascade_delete 'VTI_SAMPLE_ORDERS' 'VTI_SAMPLE_ORDER' v_
timestamp ''
```

Physical delete downwards

```
VTI_DOB_WHERE_INITIALISE.

VTI_DOB_WHERE_append 'order_number' 'eq' v_order_number ''.
vti_dob_cascade_delete_db 'VTI_SAMPLE_ORDERS' 'VTI_SAMPLE_ORDER
```

Data Profile Operations

Invoking the Profiler to Issue Data Object Refreshes

You are provided with SDK commands to invoke the SkyMobile data profile functionality. This is designed primarily for ABAP extract programs to automatically notify Java Servers of updates to data on the SAP host, that is, notify the impacted servers after updating data associated with a data object. In this way, you can push out the notifications when update events happen so that data is synchronised in a more timelyfashion without regular polling by the servers. For more information on the data profile functionality, refer to the SkyMobile Workbench guide. The following commands are provided:

Command	Description
VTI_ PROFILE_ REFRESH_ ATTRIBUTES	Clear the profile attribute table.
VTI_ PROFILE_ APPEND_ ATTRIBUTE {attribute} {value}	Add an attribute name and corresponding value to the attribute table. You use these to select a profile instance using the configured selection criteria.
VTI_ PROFILE_ REFRESH VTI_ PROFILE_ DOWNLOAD VTI_ PROFILE_ UPLOAD	Invoke the profiler to generate data object download, refresh or upload heartbeat commands. You must have populated the attribute value list prior to this, using the vti_profile_append_attribute command. If a server is not yet registered with SAP, it is ignored. Table "it_vti_*pflcmd" contains a result list of all the commands issued. An internal flow trace is returned in the it_vti_*text table. You may use this to audit/diagnose the processing and decisions made.

Example call

```
vti_profile_refresh_attributes.
   vti_profile_append_attribute 'COMPANY' 'WOOLWORTHS'.
   vti_profile_refresh 'VTI_SAMPLE_ORDERS'.
  If v_vti_*rc ne 0.
     "error. Message in v_vti_*msg.
  Endif.
```

This example invokes the profile "VTI_SAMPLE_ORDERS" to select the instance where company is "WOOLWORTHS". A DOBREFRESH heartbeat command is generated for all servers associated with the instance. If no instances were selected or a problem is encountered, the v_vti_*rc return code is set to non-zero and v_vti_*msg is populated with the reason. A sample program /SKY/YVTI8026 is provided.

API to Dynamically Maintain Profile Instances

You may use a function module /SKY/VTI_PROFILE_INSTANCE_UTIL to programmatically maintain a profile instance. If the instance does not exist, it is created... if it does exist, replace it... or if it exists and the delete flag is set, remove the instance. This utility is provided as an easy way to dynamically create, change or remove profile instances, for example, as a work order gets extracted, generate profile criteria based on the order work centre (or as part of a mass data load utility). The timestamp of the profile is updated so that any in-memory cache is flushed and reloaded from the database, that is, any download / refresh / upload operations pick up the latest definition. No where used is performed since there is no impact on the underlying applications.

Parameters

Parameter	Description
I_PROFILE	The name of an existing profile definition.
I_INSTANCE	The unique name of the instance.
I_DESCRIPTION	A description of the instance (not required for delete).

Parameter	Description
I_DELETE	Delete all instance data (only i_profile / instance reqd).
IT_CRITERIA	List of one or more instance selection criteria.
IT_SERVERS	List of server group / ids that qualify.
IT_USERS	List of work group / user ids that qualify.

4.11.12.16 Working with Binary Objects

APIs (function modules) are available to manipulate and integrate SkyMobile binary objects from ABAP programs and functions. Comprehensive functions are provided to:

- Import objects into the repository
- Export objects to a external source
- Maintain repository information
- Delete / Housekeep objects
- Integrate with the SAP GOS (Generic Object) sub-system
- Integrate with SAP Script

Maintaining Repository Objects

The following is a summary of the function modules available:

Function Module	Description
/sky/vti_import_binary_data	Import a binary object into the SkyMobile repository from the following sources: • A static or variable input string of type 'C' or 'X'. • A host file. • Raw binary data table with a simple structure of a single field of type 'C' or 'X' (raw). • Internal SkyMobile data table format based on structure /sky/yvti_filed.
/sky/vti_import_binary_file	Front ends the /sky/vti_import_binary_data function to import a host file into the SkyMobile repository, allowing the added ability to delete the host file after a successful completion.
/sky/vti_check_binary_object	Check for the existence of a binary object and return its attributes.
/sky/vti_update_binary_object	Update certain attributes of a binary object definition.
/sky/vti_rename_binary_object	Rename an existing binary object.

Function Module	Description
/sky/vti_copy_binary_object	Copy an existing binary object.
/sky/vti_delete_binary_object	Logically or physically delete an existing binary object.
/sky/vti_export_binary_data	Export a binary object from the SkyMobile repository into the following formats: • A static or variable input string of type 'C' or 'X'. • A host file. • Raw binary data table with a simple structure of a single field of type 'C' or 'X' (raw). • Internal SkyMobile data table format based on structure /sky/yvti_filed.

These functions enable custom ABAP programs to effectively import and export objects for use in applications, for example, graphics, data files, and reports. These are provided to enable a simple robust interface and abstract custom programs away from the internal data structures; that could change in future releases. View the comments in each function module for more information on usage. There is a provision for simple string, file and the internal data table format depending on where the source data is and the volume. The internal data table is simply an ABAP internal table based on structure /sky/yvti_filed that breaks the data up into fixed chunks, that is length and data segments.

Importing Data into the Repository

SKY/VTI_IMPORT_BINARY_DATA

This is a general utility to import a binary object to the SkyMobile repository. You may input data in internal table format (it_filed), raw binary data table (it_table), a string (i_string) or host file (i_host_file). The format of the data may be either in raw binary (byte level) or character hex (2 character hex code). You may also pass in an exact byte length of the input data using i_total_bytes parameter. The routine basically loads the source data into an internal format and then invokes the central upload utility.

By default, an object is created even if there is no input data. Configure i_check_no_input to true (X) to ignore creation and raise exception NO_INPUT_DATA.

Source data

it_filed Internal SkyMobile data table format based on structure /sky/yvti_filed.
it_table Raw binary data table with a simple structure of a single field of type 'C' or 'X' (raw).
i_string A static or variable input string of type 'C' or 'X'.
i_host_file The full path name of a host file.
You may specify only one source, otherwise an exception is raised.

Binary/hex modes

Two data modes are supported: hex character representation and raw binary (default). You must configure i_hex_mode to true (X) if the caller sent the data in hex character representation. You may require to avoid SAP RFC conversion issues, for example, Unicode, and ebcdic/asci formats. In both cases, the binary object data is stored in raw byte format and is retrieved as such.

```
CALL FUNCTION '/SKY/VTI_IMPORT_BINARY_DATA'
     EXPORTING
       I_BINARY_NAME                = 'my_picture'
       I_EXTENSION                  = 'JPG'
       I_CATEGORY                   = 'GRAPHIC'
       I_DESCRIPTION                = 'example binary import'
       I_LOCATION                   = 'GRAPHICS'
       I_REPLACE                    = 'X'
       I_GROUP                      = 'my_album'
       i_total_length               = l_total_length
    TABLES
```

```
    IT_TABLE                          = it_raw_table
  EXCEPTIONS
    INVALID_BINARY_NAME               = 1
    INVALID_BINARY_VERSION            = 2
    INVALID_EXTENSION                 = 3
    INVALID_CATEGORY                  = 4
    INVALID_DESCRIPTION               = 5
    INVALID_LOCATION                  = 6
    BINARY_OBJECT_ALREADY_EXISTS      = 7
    NO_INPUT_DATA                     = 8
    STRING_AND_TABLE_DATA             = 9
    OTHERS                            = 10.
```

Note:

- If you are using a host file as input, the binary name (i_binary_name) and extension (i_extension) parameters are not necessary and are defaulted from the i_host_file name.
- The binary name and version uniquely identifies the binary object. Version is used to store different iterations of the same binary object and may be 0-99. If you are importing graphics for use in SkyMobile screens, specify a version of 0. The default is 0. Typically, you leave the version to default.
- The location must exist in table /sky/yvti_pkgl. This determines where binary objects are stored by the Java server when they are downloaded from the host. You must specify this parameter irrespective of whether the object is used by Java Servers or not.
- The i_keep_in_days parameter determines how long SkyMobile keeps the binary object for. Leave out or specify a value of 0 to keep indefinitely.
- You may use the i_group parameter to include the binary object in a SkyMobile group for easier listing in the workbench. If blank, the object is added to the general group. If the group does not exist, it is automatically created.

- You may use the i_server_group / id parameters to assign the binary object to "belong" to a specific Java server or group of servers.
- If you do not specify a physical file name, it is defaulted by concatenating binary name|.|extension.
- If you do not specify the exact i_total_bytes, the routine defaults to either the string length or table rows x length. Certain types of input object such as graphics require an exact length.

Exporting Data from the Repository

/SKY/VTI_EXPORT_BINARY_DATA

General utility to export a binary object from the SkyMobile repository to a number of target data types as defined by the I_TARGET_TYPE parameter, that is, internal table format (it_filed), raw binary data table (it_table), a string (e_string) or host file (e_host_file). The format of the data may be either in raw binary (byte level) or character hex (2 character hex code). See the binary/hex explanation below.

Target data types

INTERNAL (it_filed) Internal SkyMobile data table format based on structure /sky/yvti_filed.
TABLE (it_table) Raw binary data table with a simple structure of a single field of type 'C' or 'X' (raw).
STRING (e_string) A static or variable output string of type 'C' or 'X'. Variable strings are not currently supported until SAP supports offset / length operations.
FILE (i_host_file) The full path name of a host file to create with the data.

Binary/hex modes

Two data modes are supported: hex character representation and raw binary (default). You must set i_hex_mode to true (X) if the caller requires the data in hex character representation. You may require this to avoid SAP RFC conversion issues, for example, Unicode, and ebcdic/asci formats. In both cases, the binary object data is stored in raw byte format and is retrieved as such.

Example call

```
CALL FUNCTION '/SKY/VTI_EXPORT_BINARY_DATA'
    EXPORTING
      I_BINARY_NAME                  = 'my_picture'
      I_TARGET_TYPE                  = 'TABLE'
   TABLES
      IT_TABLE                       = it_raw_table
   EXCEPTIONS
      INVALID_BINARY_NAME            = 1
      INVALID_BINARY_VERSION         = 2
      BINARY_OBJECT_NOT_FOUND        = 3
```

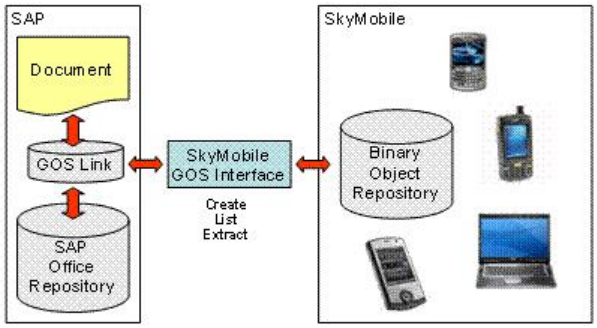
```
    INVALID_TARGET_TYPE                = 4
    SELECTION_FAILED                   = 5
    CONVERSION_FAILED                  = 6
    OTHERS                             = 99.
```

Note:

- The binary object is uniquely identified by the binary name and version. Version is used to store different iterations of the same binary object and may be 0-99. The default is 0. Typically, you leave the version to default.
- The attributes of the binary object header are returned in `e_binh` that is based on structure `/sky/yvti_binh`.

SAP GOS Interface API

The SAP Generic Object Services sub-system allows you to attach files and binaries to SAP documents, such as orders, deliveries, and notifications. Sky provided a standard interface with SkyMobile that allows developers to easily import or export binaries between the SkyMobile binary object repository and the SAP GOS system. ABAP function module APIs are provided to create, list and extract GOS objects. This enables you to extract data, such as signatures, and photographs automatically from the mobile system and to attach to a SAP document and vice versa.



The following new functions are provided to easily interface SkyMobile binary objects and SAP documents and attachments, for example, attached documents with a SAP notification / delivery / sales order. This enables to create SAP document links from SkyMobile binary objects and vice versa, for example, attach a customer signature to the SAP delivery as part of mobile "proof of delivery".

SKY/VTI_CREATE_SAP_BINARY

This utility creates a binary reference (attachment) for an existing SAP object (for example, Delivery/Notification) through SAP object services; from an existing VTI binary object. Refer to table TOBJ or transaction SW01 for the SAP object type to use. Also ensure that the SAP object number (notification, delivery, and order) is right justified and fully zero padded out to its maximum length. If you wish to delete the source VTI binary object when the SAP object is successfully created, then specify the input parameter I_DELETE = 'X'. You may specify the SAP object attachment name using the I_SAP_ATTACHMENT_NAME parameter. If you do not specify this, the VTI binary object name (I_FILE_NAME) is used.

Example call

```
CALL FUNCTION '/SKY/VTI_CREATE_SAP_BINARY'
  EXPORTING
    I_FILE_NAME              = 'my_vti_binary_object'
    I_FILE_VERS              = '00'
    I_SAP_OBJECT_TYPE        = 'BUS2080'
    I_SAP_OBJECT_NUMBER      = '000300000053'
    I_DELETE                 = 'X'   "<- optional
 EXCEPTIONS
    BINARY_OBJECT_NOT_FOUND            = 1
    BINARY_OBJECT_EXTRACT_FAILED       = 2
    INVALID_SAP_OBJECT                 = 3
    SAP_OBJECT_LINK_FAILED             = 4
    OTHERS                             = 5.
case SY-SUBRC.
       when 0.
     when 1......
endcase.
```

/SKY/VTI_SELECT_SAP_BINARY

This utility selects a list of SAP binary objects (attachments) for an existing SAP object (for example, Delivery / Notification) through SAP object services. Refer to table TOBJ or transaction SW01 for the SAP object type to use. Also ensure that the SAP object number (notification, delivery, and order) is right justified and fully zero padded out to its maximum length. A list of objects is returned in table it_saplink that is based on structure `/sky/yvti_saplink`. This routine is usually called in conjunction with /sky/import_sap_binary to create VTI binary objects from SAP document attachments.

Example call

```
DATA: IT_SAPLINK   like /sky/yvti_saplink
                   occurs 0 with header line.
CALL FUNCTION '/SKY/VTI_SELECT_SAP_BINARY'
  EXPORTING
    I_SAP_OBJECT_TYPE                = 'BUS2080'
```

```
    I_SAP_OBJECT_NUMBER                 = '000300000053'
 TABLES
    IT_SAPLINK                          = IT_SAPLINK
 EXCEPTIONS
    INVALID_SAP_OBJECT    = 1
    INTERNAL_SAP_ERROR    = 2.
case SY-SUBRC.
     when 0.
     when 1......
endcase.
```

SKY/VTI_IMPORT_SAP_BINARY

This utility imports a SkyMobile binary object from an SAP object that the SAP Generic Object Services (GOS) created. This is usually used in conjunction with the /SKY/VTI_SELECT_SAP_BINARY, that returns a list of GOS objects associated with a document. The two mandatory parameters are i_sap_document_id and i_location. The others are optional and you can default them. The SAP document id is the internal key assigned to the object when the /sky/vti_create_sap_binary (or SAP) created. It usually consists of a SAP office folder and extension id, for example, FOL31000000000004EXT33000000000004. The /sky/vti_select_sap_binary routine passes back a list of associated SAP documents. The internal id is passed back in INSTID. The location must exist in `/sky/yvti_pkgl`. This determines where the Java server stored the binary objects when they are downloaded from the host. You must specify this parameter irrespective of whether the object is used by servers or not. The binary object is uniquely identified by the binary name and version. Version is used to store different iterations of the same binary object and may be 0-99. If you are importing graphics for use in screens, specify a version of 0. The default is 0. Typically, you leave the version to default. If not specified, the binary name is defaulted from the SAP document description. The i_keep_in_days parameter determines how long VTI keeps the binary object for. Leave out or specify a value of 0 to keep indefinitely. You may use the i_group parameter to include the binary object in a group for easier listing in the workbench and /or management using a binary group cross reference. If blank, the object is added to the general group. If the group does not exist, it is automatically created. You may use the i_server_group / id parameters to assign the binary object to "belong" to a specific Java server or group of servers. If you do not specify a physical file name, concatenating binary name|.|extension defaults it.

Example call

```
CALL FUNCTION '/SKY/VTI_IMPORT_SAP_BINARY'
  EXPORTING
    I_SAP_DOCUMENT_ID  = 'FOL31000000000004EXT33000000000004'
    I_LOCATION         = 'DOCUMENTS'
  EXCEPTIONS
    INVALID_BINARY_VERSION       = 1
    INVALID_LOCATION             = 2
    BINARY_OBJECT_ALREADY_EXISTS = 3
    SAP_DOCUMENT_NOT_FOUND       = 4
    UNSUPPORTED_EXTENSION        = 5
    INVALID_DOCUMENT_LENGTH      = 6
    OTHERS                       = 7.
IF SY-SUBRC NE 0.
   ........
ENDIF.
```

The following additional parameters are optional, but you may use to specify additional attributes for the SkyMobile binary objects:

- I_BINARY_NAME
- I_BINARY_VERSION
- I_DESCRIPTION
- I_LOCATION
- I_REPLACE
- I_KEEP_IN_DAYS
- I_GROUP
- I_PHYSICAL_NAME

- I_SERVER_GROUP
- I_SERVER_ID

SAP Script Interface API

/SKY/VTI_CREATE_SAPSCRIPT_BMP

This utility creates a SAP Script BDS bit map object from a SkyMobile binary object. The default is to use the same name as the binary object (i_binary_name), but you may specify an alternate name and description through the the optional i_sapscript_name / description parameters (otherwise defaulted from object). You may also delete the source SkyMobile binary object once the SAP Script bit map is created by setting the i_delete parameter to true (X). The binary version (i_binary_version) is optional and defaults to 0 if not specified. If you are happy to replace any existing SAP Script object, then set the i_replace parameter to *true* (X).

```
CALL FUNCTION '/SKY/VTI_CREATE_SAPSCRIPT_BMP'
  EXPORTING
    I_BINARY_NAME                    = 'my_skymobile_binary_object'
    I_REPLACE                        = 'X'
 EXCEPTIONS
    SAPSCRIPT_ALREADY_EXISTS         = 1
    BINARY_OBJECT_NOT_FOUND          = 2
    SAP_ENQUEUE_FAILED               = 3
    SAP_BITMAP_CONVERSION_FAILED     = 4
    SAP_BDS_UPLOAD_FAILED            = 5
    BINARY_OBJECT_DELETE_FAILED      = 6
    OTHERS                           = 7.
```

> ***Note:*** This utility is designed with bit map images in mind, for example, drawing object signatures. Other formats may not work or you may have to convert separately.

4.11.12.17 Heartbeat Commands

ABAP SDK commands are provided that enable you to issue heartbeat commands to either Java Servers or Work groups / User ids. You specify a command and any parameter data. The command is placed on the command queue.

> ***Important:*** You should take care to nominate the target, for example, Server group or User Id; otherwise the command is issued to all registered servers.

DOBREFRESH command for Data Object 'VTI_SAMPLE_ORDERS' to Java Server 'MYSERVER', group 'MYGROUP'

```
* Issue a heartbeat command to initiate a data object refresh.
  VTI_SERVER_COMMAND 'MYGROUP' 'MYSERVER' 'DOBREFRESH'
                     'VTI_SAMPLE_ORDERS' '' ''.
  IF V_VTI_*RC ne 0.
     " command failed
  ENDIF.
```

> ***Note:*** You can exclude servers from command generation using the VTI_REFRESH_EXCLUDE_LIST and VTI_EXCLUDE_SERVER commands prior to the call.

DOBREFRESH command for Data Object 'WORK_ORDERS' to all registered users in work group CREW15

```
* Issue a heartbeat command to initiate a data object refresh.
  VTI_USER_COMMAND 'CREW15' '*' 'DOBREFRESH'
                     'WORK_ORDERS' '' ''.
  IF V_VTI_*RC ne 0.
     " command failed
  ENDIF.
```

> ***Note:*** You can exclude servers from command generation using the VTI_REFRESH_EXCLUDE_USER_LIST and VTI_EXCLUDE_COMMAND's command prior to the call.

Heartbeat commands

Command	Description
RESTART	Restart Java Server
STOP	Stop Java Server
TRACEMINUS	Decrease the trace level
TRACEPLUS	Increase the trace level
RESTART	Restart Java Server
APPCHECK	Check application definitions
APPSYNCH	Force application synchronization
BINARYCHECK	Check binary files and packages
BINARYDOWNLOAD	Download a specific binary object from the host
BINARYUPLOAD	Upload a specific binary object to the host
CONFIGCHANGE	Change CFG entry section/option/value
CONFIGSAVE	Commit configuration changes to file
DISABLE	Disables the Java Server from use
LDBDOWNLOAD	Invoke download of a database table (LDB)
LDBREFRESH	Invoke refresh of a database table (LDB)
LDBUPLOAD	Invoke upload of a database table (LDB)

Command	Description
DOBDOWNLOAD	Invoke download of a data object (DOB)
DOBREFRESH	Invoke refresh of a data object (DOB)
DOBUPLOAD	Invoke upload of a data object (DOB)
RELAYOPEN	Open a gateway http relay connection
RELAYCLOSE	Close a gateway http relay connection

The following screen exit commands are available to more easily issue heartbeat commands:

Command	Description
VTI_SERVER_COMMAND {group} {id} {cmd} {parm1} {parm2} {parm3}	Issue the specified heartbeat command and parameters (as above) to the server group / ID.
VTI_CURRENT_SERVER_ COMMAND {cmd} {parm1} {parm2} {parm3}	Issue a heartbeat command to the current server, that is, server that called SAP.
VTI_CURRENT_SERVER_LDB_ REFRESH {LDB}.	Issue a LDB refresh heartbeat command to the current server that called SAP.
VTI_CURRENT_SERVER_DOB_ REFRESH {DOB}.	Issue a Data Object refresh heartbeat command to the current server that called SAP.
VTI_REFRESH_EXCLUDE_LIST	Clear the excluded server table.
VTI_EXCLUDE_SERVER {group} {id}	Add a server to the excluded server table.
VTI_USER_COMMAND {work group} {user id} {cmd} {parm1} {parm2} {parm3}	Issue the specified heartbeat command and parameters (as above) to the registered work group and user(s).

Command	Description
VTI_CURRENT_USER_COMMAND {cmd} {parm1} {parm2} {parm3}	Issue a heartbeat command to the registered work group and user(s).
VTI_CURRENT_USER_LDB_ REFRESH {LDB}.	Issue a LDB refresh heartbeat command to the registered work group and user(s).
VTI_REFRESH_EXCLUDE_USER_ LIST	Clear the excluded work group / user id table.
VTI_EXCLUDE_USER {group} {id}	Add a work group / user id to the excluded server table.

4.11.12.18 Handling Drawing Objects

Drawing object data is passed to the ABAP exit in character hex encoding. This is to provide total portability across different Unicode and non-Unicode platforms. Character hex encoding is when the hex values are listed as actual character values and not as a single byte, for example, 'FF0A0D'.

It is important to note that drawing objects always return a string of values. This is because even a 'null' bit map object has a header and footer. Because of this, a special command is provided to test whether any input is actually drawn since the screen is first formatted. Special commands are also provided to manipulate this flag if the exit is processing the drawing object data, that is, set it on or off programmatically.

Command	Description
vti_check_for_ drawn_object {name}	Test to see if the specified drawing object is drawn. A return code of 0 indicates that it has.
vti_indicate_drawn_ object {name}	Indicates that the drawing object is drawn. This notifies VTI that the exit is manipulated the content of the drawing object.
vti_initialise_drawn_ object {name}	Initialize (set off) the drawn object indicator.

Example call

```
* Check for signature.
  vti_check_for_drawn_object 'd_signature'.
  If v_vti_*rc ne 0.
     vti_set_rc 100.
     vti_set_msg 'Please sign the screen'.
     exit.
  Endif.
```

The following function modules are useful to convert character to hex and vice versa.

Function	Description
/sky/convert_char_hex_to_raw	Convert a character hex string to raw hex string and vice versa.
/sky/convert_hex_to_character	Converts a single hex byte to an equivalent Unicode single character and vice versa.

Example call

```
data: v_xstring        type xstring,
      v_string         type string,
      v_file_name(60)  type c value '/tmp/myimage.bmp'.

* Read in binary image from file.
open dataset v_file_name for input in binary mode.
read dataset v_file_name into v_xstring.

* Convert to character hex encoding.
CALL FUNCTION '/SKY/CONVERT_CHAR_HEX_TO_RAW'
  EXPORTING
    I_MODE             = 'RAW_HEX_TO_CHAR'
    I_RAW_HEX          = v_xstring
 IMPORTING
    E_CHAR_HEX         = v_string
  EXCEPTIONS
    INVALID_MODE       = 1
    OTHERS             = 2.

* Configure screen drawing object field.
v_d_signature = v_string.
```

4.11.12.19 Sending Messages

You may send messages to one or more SkyMobile servers from an ABAP program. You can notify mobile users of events, such as successful processing or an error. The following screen exit commands are available to more easily issue heartbeat commands:

Command	Description
VTI_USER_ MESSAGE {work group} {user id} {priority} {message}.	Create a SkyMobile Server message queue entry by work group / user id. You may use Wild card '*' entries for generic selections. The priority can be L-low, M-medium, and H-High. The message text is free format up to 255 characters.
VTI_SERVER_ MESSAGE {server group} {server id} {priority} {message}.	Create a SkyMobile Server message queue entry by server group / id. You may use Wild card '*' entries for generic selections. The priority can be L-low, M-medium, and H-High. The message text is free format up to 255 characters.

4.11.12.20 Command Reference

Screen Exit Commands

Initialization and Termination

You use the following commands to import / export screen variables.

Command	Description
VTI_IMPORT_ PARAMETERS	Import a working copy of the current SkyMobile screen definition for processing.
VTI_EXPORT_ PARAMETERS	Export the working copy of the screen definition back to SkyMobile.

Setting Session Attributes

You may set the following session variables as required. These standard variables are stored permanently in the session header and you may retrieve.

Command	Description
VTI_SET_ USER {value}	Configure a user name or ID. Maximum 12 character.
VTI_SET_ PASSWORD {value}	Configure a password value. Maximum eight character.
VTI_SET_ PLANT {value}	Configure the SAP plant. Maximum four character.
VTI_SET_ WORK_AREA {value}	Configure a work area ID. Maximum four character.

Command	Description
VTI_SET_ WORK_ GROUP {value}	Configure a work group ID. Maximum fifteen character.
VTI_SET_ SESSION_ID {value}	Configure the internal session ID that is an eight digit number. ***Important:*** The session ID is handled internally and you are not recommended to configure this value unless absolutely necessary.
VTI_SET_ TRAN_ID {value}	Configure the unique transaction ID that is a 10 digit number. This is usually the ECS run number. Used to uniquely identify export / import data operations.

Screen Manipulation

Command	Description
VTI_GET_ SCREEN_FIELD {field name}	Copy the value of a screen field to a working variable of the name 'v_{field name}'.
VTI_SET_ SCREEN_FIELD {field name}	Configure the value of a screen field from a working variable of the name 'v_ {field name}'.
VTI_IMPORT_ TABLE_ CONTROL {table name}	Import a screen tables content into an internal table of the name 'it_{table name}'.
VTI_EXPORT_ TABLE_ CONTROL {table name}	Populate a screen table from a internal table of the name 'it_{table name}'.

Command	Description
VTI_GET_TABLE_FIELD_NAME {field name} {row number} {internal name}	If you wish to manipulate a specific field in a table, you must first determine the internal field name. You may use the resulting field name as input to other commands, for example, VTI_POSITION_CURSOR. You may use the standard type VTI_FIELD_NAME to declare the result variable.
VTI_GET_TABLE_TOP_ROW {table name} {row number}	Retrieve the table control 'top row' for the current screen. This row number is used to populate the screen table from this position in the table.
VTI_SET_TABLE_TOP_ROW {table name} {row number}	Configure the table control 'top row' for the current screen. This effectively instructs the session manager to populate the screen table from this position in the table.
VTI_GET_TABLE_ACTIVE_ROW {table name} {variable}	Populates the supplied variable with the current active row number. This equates to the row in the internal table. An active row may be triggered using a link or button etc.
VTI_SET_TABLE_ACTIVE_ROW {table name} {variable}	Programmatically configure the current active row in the table. The row is configured as the 'top row', that is, from which the screen table is displayed.

Command	Description
VTI_GET_TABLE_SCREEN_OCCURS {table name} {variable}	Retrieve the number of displayable table rows on the current screen for the specified table control.
VTI_SET_SCREEN_ATTR {field name} {action}	Manipulate the screen field attributes. The following actions are supported: OUTPUT, INPUT, CURSOR, NOCURSOR, LOWERCASE, UPPERCASE, HIDE, NOHIDE, NOCLEAR, CLEAR, LEFTJUSTIFY, RIGHTJUSTIFY, NOJUSIFY, MANDATORY, NOMANDATORY, BORDER, NOBORDER, NUMERIC, INVISIBLE, CHARACTER, DATE, TIME, BUTTON, CHECKBOX, HOTLINK, RADIOBUTTON. ***Note:*** If you wish to configure the attributes of a field in a table, you must first issue the VTI_GET_TABLE_FIELD_NAME.
VTI_POSITION_CURSOR {field name}	Position the cursor on a specific field on the screen. You can also use VTI_SET_CURSOR. ***Note:*** If you wish to configure the attributes of a field in a table, you must first issue the VTI_GET_TABLE_FIELD_NAME.
VTI_SET_STYLE {field name} {style number}	Configures the VTI style number for the field. ***Note:*** If you wish to configure the attributes of a field in a table, you must first issue the VTI_GET_TABLE_FIELD_NAME.

Command	Description
VTI_SET_GRAPHIC {field name} {binary object name}	Configures the binary file name for the graphic i.e. the image to display. ***Note:*** If you wish to configure the attributes of a field in a table, you must first issue the VTI_GET_TABLE_FIELD_NAME.
VTI_CLEAR_SCREEN	Initialize all input / output fields on the screen that are not explicitly flagged as kept.
VTI_SET_TITLE	Configure the screen title without any variable substitution. If you wish to configure your own title, or substitute variables, you may use the VTI_SET_TITLE_DATA command.
VTI_SET_TITLE_DATA {variable1, variable2, variable3, variable4, variable5}	Configure the screen title passing five variable substitution parameters. Specify space for variables that are not required.
VTI_REMOVE_FIELD {field name}	Suppress field definition by removing it from the screen definition. If you wish to temporarily hide the field, use the VTI_SET_SCREEN_ATTR HIDE option. ***Note:*** You cannot use this option on a field in a table control. You may also use VTI_SUPPRESS_FIELD.
VTI_REINSTATE_FIELD {field name}	Adds a removed field definition back into the screen definition. ***Note:*** This option cannot be used on a field in a table control.

Command	Description
VTI_ REMOVE _ FKEY_ID {fkey id}	Suppress function key entry from the screen using a specific function key, for example, F1, F2 (You are recommended to use the action code). You may also use VTI_SUPPRESS_FKEY_ID.
VTI_ REINSTATE_ FKEY_ID {fkey id}	Restores a removed function key definition back into the screen definition.
VTI_ REMOVE _ FKEY_ACTION {action}	Suppress function key entry from screen using an fkey action code. e.g. UNDO. You are recommended to use the action code over specific keys IDs. You may also use VTI_SUPPRESS_FKEY_ACTION.
VTI_ REINSTATE_ FKEY_ACTION {action}	Restores a removed function key action definition back into the screen definition.
VTI_GET_ ACTIVATED_ FIELD {field name}	Return the name of the activated field on the screen, that is, what field triggered the event, for example, link, drop down, and input aid.

Command	Description
VTI_CHECK_FOR_DRAWN_OBJECT {name}. VTI_INDICATE_DRAWN_OBJECT {name} VTI_INITIALISE_DRAWN_OBJECT {name}	Returns 0 in v_vti_*rc if the drawing object has been drawn into, otherwise 1 if not. Configure the 'drawn in' indicator. Reconfigure the 'drawn in' indicator.
VTI_SET_NEXT_FUNCTION {function name}	Instruct SkyMobile to switch to another screen function within the same application version.

Message Control

You use these commands to configure the VTI return code and message text.

Command	Description
VTI_SET_MESSAGE {value}	Configure the message text. You may also use VTI_SET_MSG.
VTI_SET_MESSAGE_IF_OK {value}	Only configure the message text, if there is no return code (error).
VTI_SET_RETURN_CODE {number/space}	Configure the return code, any numeric value up to three digits. A value of space is ok. You may also use VTI_SET_RC.

Command	Description
VTI_SET_ERROR_ VIA_TEXT_ ELEMENT {number}	You can configure the number as the return code and is used to identify a text element, for example, text-001.
VTI_SET_ERROR_ VIA_VARIABLE {variable}	Configures the error message as the value of the specified variable and configures a return code of 99.
VTI_SET_ERROR_ VIA_SAP_ MESSAGE {msgid} {msgno} {v1} {v2}	Configures the error message using the specified SAP message and the return code as the specified message number.
VTI_SET_ MESSAGE_VIA_ SAP {msgid} {msgno} {v1} {v2}	Configures the message text using the specified SAP message.
VTI_SET_ MESSAGE_POP_ UP {severity} Or VTI_SET_ INFORMATION_ POPUP VTI_SET_ WANING_POPUP VTI_SET_ERROR_ POPUP	Instructs SkyMobile to issue a popup message box for the message. You must have performed a VTI_SET_MSG prior to this. You can have the severity as I,W,E. Standard commands are available to help make this easier.
VTI_IGNORE_IF_ ERROR	Do not proceed if you have configured a return code. This is the equivalent of an ABAP check and is usually specified at the top of a form to prevent execution if an error occurred.

Command	Description
VTI_SET_SOUND {name}	Configures a sound to play in the header. A value of $DEFAULT uses whatever is configured for the operating system. This is useful to play audible sounds to emphasize an error.

Tracing Facility

Use the following commands with the VTI flow trace facility to help diagnose processing logic.

Command	Description
VTI_FLOW_TRACE {value}	Write a value to the flow trace for the session. This is ignored if the trace is not on.
VTI_SET_FLOW_TRACE_ ON (i)	Activates the flow trace for the session.
VTI_SET_FLOW_TRACE_ OFF (i)	Deactivates the flow trace for the session.

Note: You normally do not require these commands, as you may dynamically turn on and off at run time. Use these commands to explicitly override any run time configuration.

Data Management

Use the following commands to retrieve and update data definitions for later use or formatting.

Command	Description
VTI_SHIFT_ RIGHT_AND_ ZERO_FILL {variable}	Useful command to format a numeric value in a character field.
VTI_SET_PID {VTI pid} {value}	Configure the value of the PID from the specified value.

Command	Description
VTI_GET_PID {VTI pid}	Retrieve the value of the PID into the v_vti_pid_value variable.
VTI_EXPORT_DATA {variable/table}	Export the specified data definition to the export database.
VTI_IMPORT_DATA {variable/table}	Import the specified data definition from the export database.
VTI_FREE_DATA	Free all exported data definitions for the session.
VTI_FREE_DATA_ITEM {variable/table}	Free a specific exported data item.
VTI_GET_NEXT_NUMBER {VTI number range} {next value}	Returns the next available number for a number range definition.
VTI_EXPORT_ECS_PARAMETERS	Special internal command to export SkyConnect ECS parameter definitions. You use this to support ECS interface processing, that is, asynchronous submissions of ABAP exits. You use normal ECS_IMPORT_DATA commands to receive the data in an ECS process using the ECS SDK.
VTI_IMPORT_ECS_PARAMETERS	Special internal command to import SkyConnect ECS parameter definitions. You use this to support ECS interface processing, that is, asynchronous submissions of ABAP exits. You use normal ECS_EXPORT_DATA commands to update and return the data in an ECS process using the ECS SDK (only relevant for synchronous calls).

Command	Description
VTI_GET_ PARAMETER	Retrieve a SkyMobile static parameter value as defined using the workbench, for example, vti_get_parameter_value 'salescfg' 'salesorg' v_salesorg. Parameters are useful to configure constants that you can derive at run time to help avoid hard coding static values.
VTI_SET_ PARAMETER	Update a SkyMobile static parameter value as defined using the workbench, for example, vti_set_parameter_value 'salescfg' 'salesorg' v_salesorg.

Logical Locking Facility

The lock functionality enables ABAP exits to perform logical locking of resources. This is useful to prevent concurrent processing, for example, update an order.

Command	Description
VTI_CREATE_ LOCK {value}	Create a logical lock entry. A return code of 4 is returned in v_vti_*rc if the lock entry already exists.
VTI_CHECK_ LOCK {value}	Check to see if a lock exists. A return code of 2 indicates the lock already exists and a return code of 3 indicates that the lock does not exist.
VTI_ RELEASE_ LOCK {value}	Release a lock entry.

Miscellaneous

You are provided the following commands for special processing options.

Command	Description
VTI_ EMULATION_ SETTINGS	Invokes the pop up prompt for the VTI SAP Emulator, that is, application / version.

Command	Description
VTI_ EMULATOR	Invoke the VTI SAPGUI emulator.
VTI_CHECK_ BAPI_ RETURN	Check the BAPI return code and Configure the header accordingly. This command is designed to use with BAPI return structures based on BAPIRETURN and BAPIRETURN1.
VTI_EXPORT_ ECS_ PARAMETERS	Special internal command to export SkyConnect ECS parameter definitions. This is used to support ECS interface processing, that is, asynchronous submissions of ABAP exits. You use normal ECS_IMPORT_DATA commands to receive the data in an ECS process using the ECS SDK.
VTI_IMPORT_ ECS_ PARAMETERS	Special internal command to import SkyConnect ECS parameter definitions. This is used to support ECS interface processing, that is, asynchronous submissions of ABAP exits. You use normal ECS_EXPORT_DATA commands to update and return the data in an ECS process using the ECS SDK (only relevant for synchronous calls).
VTI_SET_ BDC_PREFIX	Configure the prefix to use to generate the ECS process name, for example, 'VTI_'. The process name takes the form {BDC prefix}_{screen function name}.
ECS_VTI_ BDC_OPEN_ GROUP	Start the first ECS dummy phase and create a BDC session. This is a ECS command, thus in order to use it, you must have included the ECS SDK (/SKY/YECSSDK).
ECS_VTI_ BDC_CLOSE_ GROUP	Close the BDC session, complete the dummy first phase and start the second BDC phase. This is an ECS command, thus in order to use it, you must have included the ECS SDK (/SKY/YECSSDK).

Note: For server heartbeat commands refer to: Heartbeat commands API.

LDB Commands

Use the following commands in custom local database download, refresh and upload functions. For more details in the concept, see the Workbench local database section. For documentation on how to use these commands, see the Local database programming guide section.

General

Command	Description
VTI_LDB_ SELECT_ DEFINITION {LDB name}	Select the local database definition into SDK working variables.
VTI_LDB_ CALC_NEXT_ ROW_ NUMBER {variable}	Calculate the next LDB data transfer row number.
VTI_LDB_ SET_TOTAL_ ROWS {variable}	Configure total rows from the internal total row variable.
VTI_LDB_ SET_ROW_ NUMBER	Configure the current row number to a specific value.

Command	Description
VTI_LDB_ SET_FIELD_ DATA {ldb field} {value} {indicator}.	Configure an LDB transfer field element. ***Note:*** The indicator is used to denote delete / update operations. Ordinarily, this is space, but may be D-delete or U-update or I-Insert. You may use the vti_ldb_row_must_exist and vti_ldb_no_duplicates to explicitly prevent automatic modification if existing row(s) do or dont exist in the table. ***Important:*** You must issue 'vti_ldb_select_definition' prior to this option.
VTI_LDB_ SET_ROW_ MUST_EXIST	Configure the row must exist indicator. You use this when performing refresh operations to ensure that an update request locates an existing row in the table. You must issue this command every time you use a configure field data, configured by position or configured by offset.
VTI_LDB_ SET_NO_ DUPLICATES	Configure the no duplicates indicator. You use this when performing refresh operations to ensure that an insert request does not encounter an existing row in the table. You must issue this command every time you use a data, configured by position or configured by offset. Usage: vti_ldb_set_no_duplicates.
VTI_LDB_ SET_ STRUCTURE {structure}	Configure ldb transfer field data from a nominated structure by field name. ***Important:*** You must issue 'vti_ldb_select_definition' prior to this option.
VTI_LDB_ COPY_DATA	Copy the internal LDB field data elements to a local copy to reference and / or manipulate. ***Note:*** If you wish to control the target tables yourself, use the vti_copy_ field_data {internal table based on /sky/yvti_tsfdt} and vti_copy_object_data {internal tables based on /sky/yvti_objh and /sky/yvti_objd}.

Command	Description
VTI_LDB_ LOAD_DATA	Load the internal LDB field data elements from the local copy to return back to SkyMobile after processing. If you wish to control the source tables yourself, use the vti_load_field_ data {internal table based on /sky/yvti_tsfdt} and vti_load_object_data {internal tables based on /sky/yvti_objh and /sky/yvti_objd}.
VTI_LDB_ APPEND_ DATA	Append the internal LDB field data elements to the function parameter data table to return back to the caller after processing.
VTI_LDB_ MODIFY_ SAP_TABLE_ FROM {structure}	Take the input LDB field data and modify the associated SAP table, using the defined structure. The structure must match the SAP table exactly, since the LDB offsets are used. This command also issues a commit work, once the SAP database table is modified. ***Important:*** The structure must exactly match the SAP table. ***Important:*** You must use 'vti_ldb_select_definition' prior to this option.
VTI_LDB_ MODIFY_ SAP_TABLE	Extension of 'vti_ldb_modify_sap_table_from', defaulting the ldb SAP table name as the structure. This command assumes that the structure is previously declared, using a tables statement.
VTI_LDB_ GET_ STRUCTURE	Extract the ldb field data for a specified row into the specified structure by field name. By default, the receiving structure is cleared. Use the vti_ldb_ignore_ initialize command prior to this to keep existing structure values. `'vti_ldb_select_definition' must be issued prior to this option. 'vti_ldb_calc_next_row_number' is used to increment the row #. 'vti_ldb_set_row_number' is used to set a specific row #.`

Command	Description
VTI_LDB_ IGNORE_ INITIALISE	For use with VTI_LDB_GET_STRUCTURE. This option suppresses the automatic initialization of the target structures fields, if no equivalent value exists in the internal data table, that is, if configured, only the values passed are updated into the receiving structure.
VTI_LDB_ GET_FIELD_ DATA {ldb field} {value}.	Retrieve the value of a specific field component of the current row in an upload. `'vti_ldb_select_definition' must be issued prior to this option. 'vti_ldb_calc_next_row_number' is used to increment the row #. 'vti_ldb_set_row_number' is used to set a specific row #.`
VTI_LDB_ RESPOND_ WITH_DATA	Indicate to the upload manager to return back the LDB maintenance requests to the Java Server (ala refresh). If you do not specify this, the transfer data is automatically cleared after a successful upload.
VTI_LDB_ CLEAR_ TRANSFER_ DATA	Clears the contents of the transfer data table. This is typically used to remove the upload contents and insert response data.
VTI_LDB_ CLEAR_ CONTENTS	Indicate to the Java server to clear all data after a successful transfer operation. You may configure this as part of a refresh operation, that is, clear all existing contents.

Command	Description
VTI_LDB_SET_TIMESTAMP VTI_LDB_GET_TIMESTAMP VTI_LDB_CALCULATE_TIMESTAMP VTI_LDB_CURRENT_TIMESTAMP	Configure / get the last timestamp for a download / refresh or upload operation. You may use the vti_timestamp type to declare a receiving field. The calculate timestamp option automatically configures the timestamp based on the current SAP date and time. The current timestamp option calculates the system timestamp, populating V_VTI_*TIMESTAMP.
VTI_LDB_SET_ERROR_SEVERITY {severity}	Configure the error processing severity flag on the LDB data element. Valid values are: I-Info, W-Warning, E-Error, A-Abort or space (default processing). You need to configure this prior to every vti_ldb_set..... operation. This flag instructs SkyMobile on how to handle any errors that occur when processing the data element.
VTI_LDB_SET_MAX_BUFFER	Configure the size of the transfer buffer in bytes.
VTI_LDB_SET_MORE_DATA	Indicate to the calling Java Server that there is another buffer available after this one, that is, call SAP back again.
VTI_LDB_CHECK_MAX_BUFFER_SIZE	Check to see if the max buffer size is exceeded.

Command	Description
VTI_LDB_USE_DATABASE_STORAGE	Force all internal data elements to store into an internal database table instead of in memory. You should use this only in very special High volume circumstances.

Primary Key Traversal

Command	Description
VTI_LDB_PRIMARY_KEY_TRAVERSAL	Indicate to SkyMobile that a primary key traversal is performed.
VTI_LDB_CLEAR_PRIMARY_KEY	Initialize the internal primary key table.
VTI_LDB_IMPORT_PRIMARY_KEY	Import the last saved primary key definition.
VTI_LDB_EXPORT_PRIMARY_KEY	Store the current primary key definition.
VTI_LDB_DELETE_PRIMARY_KEY	Delete (remove) any saved primary key definition.
VTI_LDB_APPEND_PRIMARY_KEY	Append the primary key field values to the internal primary key table definition.
VTI_LDB_GET_PRIMARY_KEY	Get the primary key field values from the current LDB row and populate the target structure.
VTI_LDB_PRIMARY_KEY_WHERE	Generate a dynamic SQL where clause from the current primary key saved definition. If there is no definition, it is ignored.

Dynamic SQL where Conditions

Command	Description
VTI_LDB_LOAD_WHERE_DATA	Populate the internal SQL condition table from the input function parameter table. This is necessary to 'pass in' any additional dynamic SQL that is passed into the function for processing.
VTI_LDB_WHERE_INITIALISE	Clear the working where condition table.
VTI_LDB_WHERE_APPEND {field id} {operation} {value1} {value2}	Append a condition to the internal SQL where table, for example, VTI_LDB_WHERE_APPEND 'ORDER_NUMBER' 'EQ' '1000213' space. The second value is to facilitate between (range) operations. Valid operations are: EQ, NE, GT, GE, LT, LE, IN / NOT IN, BETWEEN / NOT BETWEEN, LIKE / NOT LIKE
VTI_LDB_WHERE_GENERATE	Generate an SQL where clause from the internal SQL condition table.
VTI_LDB_COPY_WHERE_DATA	Populate the function parameter where table from the internal SQL condition table. This is necessary to 'pass back' any additional dynamic SQL that is generated to the caller for processing.

Directly executing LDB Operations

Command	Description
VTI_LDB_ DOWNLOAD {header} VTI_LDB_ REFRESH {header} VTI_ LDB_ UPLOAD {header}	Invoke an LDB operation for the current LDB. This is designed to use from within a custom function module. You can now share files within our team and take backups of your files. 1. Connect to \\10.10.4.5 (file://10.10.4.5/) 2. Select use another account 3. If it asks for login credentials, use the following: Userdid: platform\klhdt121 Password : Docteam123 where dynamic where conditions are generated programmatically. Take care to avoid recursive loops by checking that a where clause is specified, for example, IF NOT IT_WHERE[] IS INITIAL ... select the data . Data is returned into the standard areas and you must use a vti_ldb_append_data or vti_ldb_copy_data to populate the returned configured. 'vti_ldb_select_definition' is a prerequisite.

DOB Commands

The following commands are used in custom data object download, refresh and upload functions. For more details on the concept, refer the Workbench Data Object section in the online help.

General

Command	Description
VTI_DOB_SELECT_ DEFINITION {name}	Select the data object definition, ready for processing.
VTI_DOB_LOAD_ DATA	Select the function parameter data that the caller provided into the SDK internal data areas.

Command	Description
VTI_DOB_LOAD_ TABLE_DATA	Load the data for the current table (VTI_DOB_POSITION_TABLE) that the caller provided into the SDK internal data areas.
VTI_DOB_COPY_ DATA	Copy data from the internal SDK work areas into the function parameters to return to the caller.
VTI_DOB_POSITION_ TABLE {name}	Initialize and load the definition for a data object table, ready to be processed against.
VTI_DOB_SET_ STRUCTURE {name}	Load table data from a structure into the internal SDK data areas.
VTI_DOB_APPEND_ DATA	Append all table data from the SDK internal work areas to the function parameters to return to the caller.
VTI_DOB_CLEAR_ LDB_DATA	Clears all data entries for the current table. This is used primarily to support upload reply data processing.

Dynamic SQL Functions

Command	Description
VTI_DOB_ WHERE_ INITIALISE	Initialize the internal condition table prior to appending conditions.
VTI_DOB_ WHERE_ APPEND	Append a condition to the internal SQL where table, for example, VTI_DOB_ WHERE_APPEND 'ORDER_NUMBER' 'EQ' '1000213' space. The second value is to facilitate between (range) operations. Valid operations are: EQ, NE, GT, GE, LT, LE, IN/NOT IN, BETWEEN/NOT BETWEEN, LIKE/NOT LIKE.
VTI_DOB_ EXECUTE	Execute the current DOB definition with the same transfer mode.

Command	Description
VTI_DOB_ DOWNLOAD {header)	Invoke a download operation for the current DOB. This is designed to use from within a custom function module where dynamic where conditions are generated programatically. You must take care to avoid recursive loops by checking that a where clause is specified, for example, IF NOT IT_ WHERE[] IS INITIAL ... select the data. Data is returned into the standard areas and you must use a vti_dob_append_data to populate the returned set. ***Note:*** You must issue a 'vti_dob_select_definition' prior to this option.
VTI_DOB_ REFRESH {header)	Invoke a refresh operation for the current DOB. This is designed to use from within a custom function module where dynamic where • conditions are generated programatically. You must take care to avoid recursive loops by checking that a where clause is specified. for example, IF NOT IT_WHERE[] IS INITIAL ... select the data. Data is returned into the standard areas and you must use a vti_dob_append_data to populate the returned set. ***Note:*** You must use a 'vti_dob_select_definition' prior to this option.
VTI_DOB_ UPLOAD {header)	Invoke an upload operation for the current DOB. Data is processed from the standard areas and you must use a vti_DOB_append_data or vti_DOB_copy_data to populate the returned set. ***Note:*** A vti_dob_select_definition' must be issued prior to this option.

Cascading Functions

Command	Description
VTI_DOB_ CASACDE_ DELETE {object} {start table} {timestamp} {ignore start rows}	Perform a cascade delete (logical) on the specified data and all subordinate child data. You must pass the Dynamic SQL where conditions in as constructed using the VTI_DOB_WHERE_INITIALISE and VTI_DOB_WHERE_APPEND commands, that is, it_vti_where. If you do not pass an explicit time stamp yyyymmddhhmmss in (parm 3), then the current system date / time is used. Specify ignore start rows (parm 4 = 'X') if you want to skip updating the selected starting position rows. ***Note:*** This function issues a database commit.
VTI_DOB_ CASCADE_ DELETE_DB {object} {start table}	Perform a cascade delete (physical) on the specified data and all subordinate child data. You must pass Dynamic SQL where conditions in as constructed using the VTI_DOB_WHERE_INITIALISE and VTI_DOB_WHERE_APPEND commands, that is, it_vti_where. ***Note:*** • The data is deleted irrespective of timestamp / delete indicator. • This function issues a database commit.
VTI_DOB_ CASCADE_ TIMESTAMP {object} {start table} {timestamp} {ignore start rows} {UP/DOWN}	Perform a cascade timestamp update on the specified data and related parent / child data (up / down). You must pass dynamic SQL where conditions in as constructed using the VTI_DOB_WHERE_INITIALISE and VTI_DOB_WHERE_ APPEND commands, it that is,it_vti_where. If you do not pass an explicit time stamp yyyymmddhhmmss in (parm 3), then the current system date / time is used. Specify ignore start rows (parm 4 = 'X') if you want to skip updating the selected starting position rows. You may specify a direction (up / down) in parm 5, if blank the default is 'up'. ***Note:*** This function issues a database commit.

Profile functions

Command	Description
VTI_ PROFILE_ REFRESH_ ATTRIBUTES	Refresh the profile attribute value table.
VTI_ PROFILE_ APPEND_ ATTRIBUTE {name} {value}	Append a profile attribute value.
VTI_ PROFILE_ DOWNLOAD {profile} VTI_ PROFILE_ REFRESH {profile} VTI_ PROFILE_ UPLOAD {profile}	Invoke the profiler to generate data object download, refresh or upload heartbeat commands. You must have populated the attribute value list prior to this, using the vti_profile_append_attribute command. If you did not register a server with SAP yet, it is ignored. Table it_vti_*pflcmd contains a result list of all the commands issued.

4.11.13 Java Programming Guide

In addition to the ability to provide ABAP exits, Sky Mobile allows Java code to execute to perform tasks that may have specific, complex and / or unusual requirements. These Java exits or 'hooks' into the application, are provided by extending classes for each exit that you wish to specify, and providing the class names within the exit definition in SkyMobile. You must place the compiled class files in the SkyMobile "classes" directory and whilst not required, you may package into a jar file for convenience.

> ***Note:*** While Java exits provide the ability to customize your application extensively, you must take care as errors in the code have the power to cause SkyMobile to fail.

If you plan to provide your own exits, ensure that you have a thorough understanding of at least the basic concepts discussed in this document.

4.11.13.1 Java Programming Guide

Getting started with Java Exits
Common Exit Functionality
Java Exit Types

4.11.13.2 Getting Started with Java Exits

Java exits are implemented as standard Java classes. In order to write one, you should be familiar with the syntax and general format of the Java language. Most Java exits are quite simple, and do not require a complete understanding of the intricacies of the Java language. However, there are still some basic concepts that are essential for the developer to understand before you write any exits. To assist those with limited experience in Java programming, a summary of these Essential Java Concepts is provided. You are highly recommended that you research these topics further, if you do not fully understand any of these concepts. Oracle provides excellent tutorials on learning java at http://docs.oracle.com/javase/tutorial.

Ensure that you are aware of our Java tips and traps.

If on the other hand, you write Java exits that need to interact with each other, you require a much deeper understanding of Java. Perhaps, the most significant is an understanding of multi-threaded environments. SkyMobile is a multi-threaded environment that means that different exits run in different threads and are quite likely to execute at exactly the same time. If your exits require to pass data between each other, you ocan introduce timing errors, and unfortunately cause unpredictable results that are hard to find and fix. When you approach these kinds of tasks, you are highly recommended that you understand fully these more Advanced Java Concepts.

In order to compile the Java class you wrote, you need either a Java compiler or an Integrated Development Environment such as Eclipse, NetBeans, JBuilder, or Visual J++.

The general process for writing user exits is as follows:

- Install and Configure your Development Environment
- Write and Compile your User Exits
- Install and Configure the Device

> ***Note:*** You can find documentation for the Java exits API in the `doc` directory that is located in the `SkyMobile installation` directory. It is in javadoc format and you can interrogate using any standard browser.

Configuring your Development Environment

The only way in which compiling an exit class differs from compiling any other Java class is that you need to incorporate the standard classes into your CLASSPATH. The VTI classes are all contained within the `vti.jar` file that is located in the VTI installation directory. Check the documentation for your Java compiler or IDE for details on how to incorporate additional classes into your CLASSPATH.

Once you do this, you are able to compile your exit class and correct any errors that may be present in the code. Once you have a clean compilation, the resulting Java class file is ready for installation.

Using Eclipse

Eclipse is an integrated development environment that assists greatly with the development of Java code, is free, and is used widely around the world. The Java compiler is largely platform independent and so most versions of Oracle / Sun SDKs suffice. You also need to have SkyMobile installed on your computer. To get Eclipse up and running on your PC, follow these steps:

1. Install the latest Oracle Java SE SDK from: http://java.com.
2. Install the latest version of Eclipse from: http://www.eclipse.org
3. Install SkyMobile.
4. Start Eclipse
5. Select **File** > **New** > **Java Project** to open the **New Java Project** window.
6. Enter a name for your project and click **Next**.
7. Select the **Libraries** tab and press **Add External JARs...**
8. Navigate to the `vti.jar` file, select it, and press **Open**. You can find it under `C:\Program Files\Sky Technologies\SkyMobile_Release_XX\SkyMobile Server\Windows\Windows-32\vti\vti.jar`.
9. Expand the `vti.jar` by clicking the **+** to the left of it, select **Javadoc location** and press **Edit...**

10. Click browse, navigate to the doc folder in the same directory as your `vti.jar`, and select **OK**, and then **OK** again.
11. If you have any other libraries that you want to add for your project add them now.
12. Press **Finish**.
    Your project is created.
13. Select **Project** > **Properties** from the menu.
14. Select **Java Compiler** and select **Enable project specific settings**.
15. Select the compiler compliance level to match the end JVM that you intend to run on. If you deploy to Windows Mobile using the CrEme JVM for example, you select 1.3 and ensure that the generated `.class` file compatibility is set to *1.1*.
16. Select **OK** and now you are ready to go.

How to Write your Exits

Once you decide what type of exit you wish to develop, writing the exit is simply a matter of developing a class that extends the VTI class provided. For example, in order to write a custom Java user exit, you develop a new class (**MyUserExit**, for example) that extends VtiUserExit. Each of the major types of exit has one or more abstract methods that provide the "hooks" that the server uses to invoke your code. You must provide implementations for these methods so that the server is able to do this.

Using Eclipse

To create a new exit in your project, follow these steps:

1. Right-click your project name in the package explorer window and select **New** > **Class**.
2. Enter the package name. (In the **your.company.domain.project** format or **au.com.skytechnologies.demo**.
3. Enter a name for this exit. (that is, **VtiUserExit**).
4. Click **Browse** for the **Superclass** and type in the name of the exit type and press **OK**.
5. Ensure that **Inherited abstract methods** is selected and click **Finish**.
6. The class is added to your project and you may start to add your custom code.

Handy Shortcuts

- Right-click a variable, function or class name in the editor and select **Open declaration** to go to where it is defined.
- Right-click a function name and select **Open call hierarchy** to see whereabouts this function is called from.
- Press <Ctrl><Shift>F to automatically format your code. You can set how it formats in your window preferences.
- Press <Ctrl><Shift>M to automatically add an import for a class that is currently not imported.

- Press <Ctrl><Shift>O to organize your imports. This deletes import statement lines not used and removes duplicates.
- Right-click in the editor and select **Source** > **Override/Implement Methods** to add method skeletons.

How to Install a Java Exit

The server automatically detects any additional classes that are placed inside the "classes" directory located in the installation directory. Making your exit available to the SkyMobile application server is usually a matter of copying your Java `.class` file into the `classes` directory and adding the appropriate configuration either to your application, or to the SkyMobile configuration file. If you change the configuration file, it is necessary to restart the server before it picks up the changes. You can find the details of the configuration required for each java exit type in the documentation for your exit type here.

In addition to raw `.class` files, the server is also able to load classes from `.jar` (Java archive) and `.zip` files placed in the `classes` directory. You are recommended to keeping your classes in a jar file, as it simplifies the management and versioning of your project files.

Using Eclipse

Is it easiest to package all of your classes into a single jar file using eclipse. This makes keeping all of your class files together, simple. It is also possible to keep a copy of the source code within the jar file so that everything is always together.

To export your project, follow these steps:

1. Right-click your project name in the package explorer window and select **Export...**.
2. Expand **Java**, select `JAR` file and press **Next**.
3. If you want to include the source files in this jar, select **Export java source files and resources**.
4. Specify the name and path of the JAR file that you like to create.
5. Press **Finish** and your jar file is created.
6. Deploy the **.jar** file to your `vti/classes` directory and you are done.

Java Tips and Traps

Coding Standards

Coding to a standard assists other programmers to make sense of your code. The standard conventions in Java are:

- Package names are always lowercase, and follow a "reverse domain name" convention, for example "com.company.project".
- Class names and interfaces are always "camel case" without spaces, for example "MyGreatClass".
- Function and variable names are the same as class names, but the first letter is always lowercase, for example "myGreatFunction".
- Constants are always upper case with underscores instead of spaces, for example "MY_CONSTANT".

Assignment

The following three assignment / comparison operators are often confused with each other and misused. Be careful to use the right one in each situation.

- '=' is used to assign a value to another value.
- '==' is used for reference comparison. Ie object1 == object2 only returns true when they are exactly the same object instance - regardless of whether the contents are the same.
- .equals() is used for value comparison. Ie object1.equals(object2) returns true if the contents are the same.

Mathematics

- Java maths functions are all stored in the `Math` class, and you can obtain random numbers using the `Random` class.

- When you divide integers, it is ESSENTIAL to cast at least one of these to a double FIRST if you want the result of the calculation to include decimals. If you don't, Java assumes you want the result as an integer EVEN if you put the result into a double.

  Example:

```
double result = 1 / 5; //Wrong! result = 0.0
double result = (double)1 / 5 // Correct! result = 0.2
```

Arrays and Collection Classes

- Arrays are created using the [ ] operators and you can define the size at runtime. Ie String[] myArray = new String[100];
- Arrays have a fixed length that you can obtain by the .length operator. Ie myArray.length == 100
- Utility functions for sorting arrays are found in the `Arrays` class.
- Collection classes are useful where the size of your array is unknown or can vary, the retrieval time matters, or the order or duplication of members matters. Vector, HashSet, TreeSet, SortedSet, HashMap are all commonly used.
- Collection classes are traversed using the iterator class returned from their .terator() function.

Operator Precedence

- When you parse your statement or equation, java processes each operator according to the priorities listed below. Where operators have equal priority, they are processed from left to right. Ie (int)i++, 'I' is cast to an integer before incrementing.
- Array operators "[ ]"
- Pre / Post fix operations and casting. "v++, --v, !v"
- Multiplication, Division, Modulus
- Addition, Subtraction

Essential Java Concepts

Primitive Types

Variables in Java are either objects or primitive types. Primitive types have lowercase names, are stored in the stack, and passed 'by value' when submitted to functions. Example types are boolean, double, char, float, int, and long.

References

If a variable is not a primitive type, it is a reference to an object that is stored on the heap. All objects extend the base class `Object` and their reference is passed around and submitted to functions. An important consideration is that references are allowed to be "null" that means that they do not point to any object yet. You should ALWAYS take this into account in your code.

Garbage Collection

You do not have to worry about deleting objects when you finish with them. Java periodically checks which objects are no longer used and deletes them for you. This can sometimes cause very short delays in your program whilst garbage collection occurs and happens more frequently the less memory that you have available.

Immutable Classes

All objects in Java fall into two categories: mutable and immutable. Mutable classes are able to modify and change. Immutable objects aren't and include classes like "`String`, `Boolean`, `Char`, `Double`, `Float`, `Integer`, `Long`". Immutable objects are extremely handy as it means you don't have to worry about them changed on you and therefore force you to create copies of them all of the time. This is especially noticeable with Strings. No string operation changes the contents of a string, they simply return a new one. Whilst this is extremely useful for passing them around and storing them in Vectors or HashMaps, this can mean that if you concatenate multiple strings this is quite inefficient and you are better off using a StringBuffer that is IS mutable.

Exceptions

Error handling within Java is taken care of through exception handling. An exception is a class that contains a bundle of information regarding the error that occurred. Exceptions are divided into two types: Exceptions that you must explicitly declare and handle, and runtime exceptions that don't. For example, a "NullPointerException" is a run-time exception occurs at run time when you try to use a reference that is null. You can also throw this exception yourself manually using "throw new NullPointerException("my message");" and you won't have to explicitly define that your function may throw this exception.

If your function throws a normal exception, then it must declare in its function definition that it may throw this exception. Any code that calls this function is required to surround this function call with a "try" statement and "catch" the exception, or define that it too can throw this type of exception.

Inner Classes

All Java code is generally contained in classes defined in their own `.java` file. Sometimes your code requires you to define a small class that is only relevant to the function that you are in. This is where "inner classes" or "anonymous inner classes" come in handy. Inner classes are defined inside of you class similar to an enumeration or a function, given a name, and are instantiated using this name when required. Anonymous inner classes are defined inline without a name and are the instantiation. You can see examples of each of these in the processMatchingRows example.

Java Versions

Different platforms or different environments support different versions of Java. When writing and compiling your code you need to be aware of this. When writing for Windows Mobile devices for example, you are limited to using version 1.1.8 that has much less functionality than the more current versions 5 or 6 (also known as 1.5 or 1.6) . This is particularly noticeable when operating with strings. There are many extremely useful string functions that are available in 5 or 6 that are just not available in 1.1.8. You are provided with some useful utility classes to assist here though.

Advanced Java Concepts

Object Oriented Programming

It is common to approach writing code as a sequence of steps that you must perform in order to achieve some objective. A consequence of this approach is that quite often extremely long, complicated functions are created that are very similar to other long and complicated functions. Whilst you can solve the problem of duplication by creating additional functions, it is better overall to adopt a different mindset when writing code and this is called "Object Oriented" or "OO" programming. Essentially OO involves "abstracting" or breaking up your code into separate classes for each role or object in your problem, and then concentrate on writing appropriate functions for each of these objects specifically. Your program flow then goes from object to object rather than being all contained in the one class. To assist in breaking your problem into objects, it is extremely useful to have a thorough understanding of "design patterns". Design patterns are essentially common problems and the ways that they are best represented by objects.

Synchronization

To prevent two different threads from accessing a class simultaneously and overwriting each others state, you can define that your functions or sections of code are "synchronized". This means that only one thread can call this function or section of code at a time, and a second thread pauses until the first is finished before it proceeds. You can additionally synchronize functions or sections of code on objects or locks. This can be any object (including "this", the object is accessed) and can mean that you can coordinate together different functions and sections of code.

Static variables

You can use static variables to simplify the interaction between multiple java exits, however can also be extremely dangerous if misunderstood. Static variables are contained in the class definition as opposed to being contained in the class instances. This means that if you define a variable as static, there is only ONE of them, and it is SHARED amongst all of the individual class instances. Additionally, if this variable is made public, or is retrieved by a public static function, then this variable is accessible to any piece of code anywhere.

For example, if you have a single VtiCustomSerialPortExit that continuously reads values in from a weighbridge, it could write this value into a static variable. You can access this variable by a VtiUserExit to display the most recent value on the screen. This works well, however, if you have two weighbridges connected to your server and create two separate instances of the same VtiCustomSerialPortExit, you immediately run into trouble. The single static variable is updated continuously from both weighbridges and the VtiUserExits are not able to know which weighbridge the variable is from. This new situation is best resolved using a separate weighbridge manager singleton class that stores a list of the weighbridge values paired with a corresponding name so that it can be requested by the VtiUserExits.

Singleton

Singleton is a design pattern that specifies a method to create a single statically accessible class. This is quite often used for manager classes or whenever you want to ensure that there is only ever one instance of this class instantiated. The use of this design pattern is much more preferable to create a class that consists solely of static variables and functions for two reasons. It gives you control over the instances life cycle, and it also allows you to easily upgrade to allow multiple instances in the future if necessary.

Singleton Example

```
Class MySingleton {
    private MySingleton me = new MySingleton();
    public MySingleton getInstance() {
        return me;
    }
    Public MySingleton() {
    }
}
```

This class instantiates itself immediately on class load, while the example below instantiates itself on first access.

```
Class MySingleton {
    private MySingleton me = null;
    public MySingleton getInstance() {
        if (me == null)
            me = new MySingleton();
```

```
        return me;
    }
    Public MySingleton() {
    }
}
```

JNI

The Java Native Interface (JNI) allows java code to call native C++, C or assembler functions. This is extremely useful specifically for calling faster performing code, or for accessing system functionality that is not available from within Java. However, writing JNI is non-trivial and small mistakes can cause the JVM to crash and /or be extremely difficult to track down. It is best to stick to passing primitive types if possible, as the interfacing of java objects in the JVM from native code gets complicated very quickly. It is possible though, to do most things that you can do in Java within the native code.

Within Java, you must define the native function, and the native library loaded. A static initializer is a good place to load the library as it loads it at the same time as the Java class.

JNI Example 1

```
Package com.company.project;
Class MyJniClass {
    Public static native double performLongCalculation(double value);
    Static {
        System.loadLibrary("MyJniLib");
    }
}
```

The C++ code must follow a specific structure, so it is best to generate the header file automatically using "javah".

JNI Example 2

```
javah -jni -o MyJniLib.h -classpath . com.company.project.MyJniClass
```

Then you can create your source file from the generated header.

JNI Example 3

```
#include <jni.h>
#include "MyJniLib.h"
#include <stdio.h>

JNIEXPORT jdouble JNICALL Java_com_company_project_MyJniLib_
performLongCalculation(
    JNIEnv *env, jobject obj, jdouble value)
{
    jdouble ret = value;
    // Perform the long calculation...
    return ret;
}
```

Important Design Considerations

The process of developing well-behaved Java code that works correctly in conjunction with the VTI exits API, requires conformance to a number of important conventions. Failure to observe these conventions may result in your code behaving incorrectly or erratically.

The VTI Server Must Instantiate VtiExit Classes

The first important convention to observe is that the VtiExit class, and all its subclasses, are always created by the server on an "as required" basis. In other words, at no point in your code you should instantiate a class that subclasses VtiExit or any of its subclasses. If you require access to any of the instance methods of the various VtiExit classes from within another class, you have two alternatives:

1. Create your class as an inner class of the main exit class. This approach allows your class to use any of the parent class methods, including those inherited from VtiExit; or
2. Pass the calling exit through as a reference into your own class. You can invoke any required instance methods against the reference to the calling exit.

Do Not Retain References To VtiExit Instances

These classes are intended to have a transient existence and you should not retain for reuse within your own code. In other words, don't store references to an exit inside a member variable or collection class where you may use it after its execution context lapsed.

Do Not Retain References To VtiExitLdbTable Or VtiExitLdbDataObject Classes Beyond The Scope Of The Current Exit

References to these objects should be acquired anew each time the exit execution content changes.

> ***Important:*** These conventions are important and you should observe in order to make your code robust.

4.11.13.3 Common Exit Functionality

Common exit functionality

Regardless of their type, all Java exits have certain common capabilities. These capabilities are available to your code as methods inherited from the VtiExit class. The most significant and commonly used of these capabilities are as follows:

Log Trace Messages
Configuration File
Working with LDBs
Working with DOBs
Print Templates
Number Ranges
The DateFormatter Class
String Utilities Classes
XML Parsing

Log Trace Messages

All types of exit are able to write messages to the log. The relevant methods are shown in the table below. Each of these methods write a message of the corresponding type to the log. Some of these (logAbort, logError, and logWarn) also accept a Throwable error or exception object as an additional parameter, for use in printing a stack trace to the log.

Method
logAbort()
logError()
logInfo()
logTrace()
logWarn()

The logAbort method is something of a special case, as in addition to writing the specified message to the log, it also causes the server to halt. Obviously, you should only perform this action in the event of extremely dire errors, since manual intervention is necessary to restart the server.

The logTrace method differs from the other methods of this type, in that it requires a trace level that you should specify (from 0-2, with 0 being the lowest level of detail and 2 being the highest). If the server current trace level equals or exceeds the figure specified in the method call, then the trace message is written to the log. However, nothing is written to the log if this is not the case. Trace messages are always accompanied by the name of the class and the method from which the trace call is made.

Configuration File

All types of exit are able to read parameter settings from the configuration file and are stored in the following format:

Example Configuration Parameters

```
SERVER.SECTION.PARAMETERBOOLEAN = true
SERVER.SECTION.PARAMETERINTEGER = 42
SERVER.SECTION.PARAMETERDOUBLE = 1.0045
SERVER.SECTION.PARAMETERSTRING = Value
```

> ***Important:*** You must store the parameters in the SERVER node and the parameter name is stored in uppercase, the same as all other configuration parameters.

And depending on what type of value is stored, may be read using the following functions:

Example

```
getConfigBoolean("SECTIONNAME", "PARAMETERBOOLEAN")
getConfigInteger("SECTIONNAME", "PARAMETERINTEGER")
getConfigDouble("SECTIONNAME", "PARAMETERDOUBLE")
getConfigString("SECTIONNAME", "PARAMETERSTRING")
```

These methods allow an exit to examine existing or additional configuration file parameters that you can use to alter its behaviour.

Additionally, the following functions are provided to interrogate the contents of a configuration file:

getConfigSectionNames() - Provides an array containing all of the section names specified in the config file.
getConfigKeywords("SectionName") - Provides an array containing all of the keyword or parameter names specified in the given section of the configuration file.

Working with LDBs

The server maintains a set of a local database tables for each SAP interface it is configured to be aware of. All Java exits, regardless of their type, are able to both read from, and write to, these tables. The initial point of access to local database tables is the getLdbTable() method. The exit code must specify the name of the SAP interface and the name of the desired table as input parameters. It then receives a VtiExitLdbTable object that you can then use in the following ways:

- Interrogation/Manipulation of data
- Interrogation/Manipulation of metadata
- LDB task scheduling

LDB Actions

Exits are also able to directly interrogate and / or update data stored on local database tables, using a variety of methods on the VtiExitLdbTable object. The table below lists the main methods that are available.

Method
countRows()
countMatchingRows()
deleteRow()
deleteMatchingRows()
getAllRows()
getMatchingRows()
newRow()
processAllRows()
processMatchingRows()
saveRow()

4.11.13.4 MatchingRows

Where a query is required to perform on the database to select a set of matching rows, you must specify the query using a VtiExitLdbSelectCriterion object. This criterion object can either be a single VtiExitLdbSelectCondition, or it can be a heirachy of other criterions contained in a

VtiExitLdbSelectConditionGroup. You can specify the Group object as an "AND" (matchAll = true) for ensuring that all conditions in the group are met, or an "OR" (matchAll = false) for ensuring that at least one of the conditions in the group are met. This means that you are able to define quite complicated selection criteria to select the matching rows.

4.11.13.5 getMatchingRows Versus processMatchingRows

getMatchingRows returns an array of matching row objects from the database. These rows are all loaded into memory simultaneously and is useful to retrieve small recordsets (Less than 50 records). Whilst this is simple to use and understand, this method is far less efficient than calling processMatchingRows. Additionaly, processMatchingRows allows you to exit the query prematurely if desired, without processing any remaining rows. This makes it ideal for queries where you want to perform your own search criterion or perhaps retrieve the first 10 matching rows. In some ways processMatchingRows are analogous to the manner in which a database cursor functions. When you call the function, you must provide a handler function that is called for each record.

getMatchingRows Example

The following example code shows a user exit that reads all records from a table called "MYTABLE" that have a "STATUS" field set to "X". It then updates these records with a new value of "A" in the same field and then writes them back to the local database.

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

public VtiUserExitResult execute() throws VtiExitException {

VtiExitLdbTable table = getLdbTable("MYTABLE");

if (table == null)
throw new VtiExitException("LDB table MYTABLE not found!");

VtiExitLdbSelectCondition cond =
```

```
new VtiExitLdbSelectCondition("STATUS",
VtiExitLdbSelectCondition.EQ_OPERATOR, "X");

VtiExitLdbTableRow [] rows = table.getMatchingRows(cond);

for (int i = 0; i < rows.length; ++i) {
rows[i].setFieldValue("STATUS", "A");
table.saveRow(rows[i]);
}

return new VtiUserExitResult(0, "Success!");
}
}
```

processMatchingRows Example

The following example code shows a user exit which reads all records from a table called "MYTABLE" that have a "STATUS" field set to "X". It then updates these records with a new value of "A" in the same field and then writes them back to the local database. This example assumes that "MYTABLE" has a significant number of rows and is significantly more efficient than calling getMatchingRows as the number of records on the table increases.

Example 1

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit implements
VtiExitLdbRowProcessor {

public VtiUserExitResult execute() throws VtiExitException {

VtiExitLdbTable table = getLdbTable("MYTABLE");

if (table == null)
throw new VtiExitException("LDB table MYTABLE not found!");
```

```
VtiExitLdbSelectCondition cond =
new VtiExitLdbSelectCondition("STATUS",
VtiExitLdbSelectCondition.EQ_OPERATOR, "X");

table.processMatchingRows(cond, this);

return new VtiUserExitResult(0, "Success!");
}

public boolean processRow(VtiExitLdbTable table, VtiExitLdbTableRow
row) {

row.setFieldValue("STATUS", "A");
try {
table.saveRow(row);
} catch (VtiExitException e) {
logError("Unable to save record");
return false;
}

return true;
}
}
```

Example 2

The same example again, however this time using an inner class for the row processor.

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

public VtiUserExitResult execute() throws VtiExitException {
```

```
VtiExitLdbTable table = getLdbTable("MYTABLE");
if (table == null)
throw new VtiExitException("LDB table MYTABLE not found!");

VtiExitLdbSelectCondition cond =
new VtiExitLdbSelectCondition("STATUS",
VtiExitLdbSelectCondition.EQ_OPERATOR, "X");

table.processMatchingRows(cond, new MyRowProcessor());

return new VtiUserExitResult(0, "Success!");
}

public class MyRowProcessor implements VtiExitLdbRowProcessor {

public boolean processRow(VtiExitLdbTable table, VtiExitLdbTableRow
row) {

row.setFieldValue("STATUS", "A");
try {
table.saveRow(row);
} catch (VtiExitException e) {
logError("Unable to save record");
return false;
}
return true;
}
}
}
```

Example 3

The same example again, however this time using an anonymous inner class for the row processor.

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

public VtiUserExitResult execute() throws VtiExitException {

VtiExitLdbTable table = getLdbTable("MYTABLE");
if (table == null)
throw new VtiExitException("LDB table MYTABLE not found!");

VtiExitLdbSelectCondition cond =
new VtiExitLdbSelectCondition("STATUS",
VtiExitLdbSelectCondition.EQ_OPERATOR, "X");

table.processMatchingRows(cond, new VtiExitLdbRowProcessor() {

public boolean processRow(VtiExitLdbTable table, VtiExitLdbTableRow
row) {

row.setFieldValue("STATUS", "A");
try {
table.saveRow(row);
} catch (VtiExitException e) {
logError("Unable to save record");
return false;
}
return true;
}
});
return new VtiUserExitResult(0, "Success!");
}
}
```

Screen Table Example

The following example code shows a user exit that reads all records from a table called "MYTABLE" that have a "STATUS" field set to "X". It then puts all of the CUST_NAME's into a screen table called "LIST".

Updating a screen table example

```
import au.com.skytechnologies.vti.*;

public class MyExit extends VtiUserExit {

public VtiUserExitResult execute() throws VtiExitException {

// Retrieve table
VtiExitLdbTable table = getLdbTable("MYTABLE");
if (table == null)
throw new VtiExitException("LDB table MYTABLE not found!");

// Retrieve screen table (list)
final VtiUserExitScreenTable list = getScreenTable("LIST");
if (list == null)
throw new VtiExitException("Screen Table LIST not found!");

// Create selection criteria
VtiExitLdbSelectCondition cond =
new VtiExitLdbSelectCondition("STATUS",
VtiExitLdbSelectCondition.EQ_OPERATOR, "X");

// Clear the list and fill it with customer names from the table
list.clear();
table.processMatchingRows(cond, new VtiExitLdbRowProcessor() {

public boolean processRow(VtiExitLdbTable table, VtiExitLdbTableRow
row) {
```

```
// Retrieve the customer name and add it if it has a value
String customerName = row.getFieldValue("CUST_NAME");
if (customerName != null && customerName.length() > 0)
{
VtiUserExitScreenTableRow listRow = list.getNewRow();
listRow.setFieldValue("CUST_NAME", customerName);
list.appendRow(listRow);
}
return true;
}
});
return new VtiUserExitResult(0, "Success!");
}
}
```

LDB Metadata Actions

Exits are able to interrogate, and in some limited cases directly alter, the metadata associated with a local database table. You are provided the following methods for this type of activity:

Method
getDescription()
getErrorFlag()
getField()
getFieldAt()
getFieldCount()
getFieldIndex()
getFields()
getLastRefreshTime()
getLastUploadTime()
getName()
getNextRefreshTime()
getNextUploadTime()
setErrorFlag()

4.11.13.6 LDB Task Scheduling

Exits are able to explicitly take control of local database table synchronisation tasks that normally take place in background mode (downloads, refreshes, and uploads, of table data). This is achieved by using a reference to the target table (in the form of a VtiExitLdbTable object) to construct a VtiExitLdbRequest.

The following example code shows a user exit that forces an upload to occur on a table called "MYTABLE". This particular exit returns control to the user immediately rather than waiting for the upload to complete.

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

    public VtiUserExitResult execute() throws VtiExitException {

        VtiExitLdbTable table = getLdbTable("MYTABLE");

        if (table == null)
            throw new VtiExitException("LDB table MYTABLE not
found!");

        VtiExitLdbRequest request = new VtiExitLdbRequest
            (table, VtiExitLdbRequest.UPLOAD);

        request.submit(false);

        return new VtiUserExitResult(0, "Success!");
    }

}
```

LDB Task Scheduling

Exits are able to explicitly take control of local database table synchronisation tasks that normally take place in background mode (downloads, refreshes, and uploads, of table data). You can achieve this by using a reference to the target table (in the form of a VtiExitLdbTable object) to construct a VtiExitLdbRequest.

The following example code shows a user exit that forces an upload to occur on a table called "MYTABLE". This particular exit returns control to the user immediately rather than waiting for the upload to complete.

Example 1

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

    public VtiUserExitResult execute() throws VtiExitException {

        VtiExitLdbTable table = getLdbTable("MYTABLE");

        if (table == null)
            throw new VtiExitException("LDB table MYTABLE not
found!");

        VtiExitLdbRequest request = new VtiExitLdbRequest
            (table, VtiExitLdbRequest.UPLOAD);

        request.submit(false);

        return new VtiUserExitResult(0, "Success!");
    }

}
```

It is also possible to schedule arbitrary code to run on completion of an LDB task. You can do this by assigning a callback handler to the LDB request before it is submitted. The callback handler is invoked (in a separate thread) once the LDB task completes (regardless of whether or not the request completed successfully).

The following code builds on the previous example by also writing a message to the VTI log file once the upload is completed.

Example 2

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

    public VtiUserExitResult execute() throws VtiExitException {

        VtiExitLdbTable table = getLdbTable("MYTABLE");

        if (table == null)
            throw new VtiExitException("LDB table MYTABLE not
found!");

        final VtiExitLdbRequest request = new VtiExitLdbRequest
            (table, VtiExitLdbRequest.UPLOAD);

        Runnable callbackHandler = new Runnable() {
            public void run() {
                if (request.getSuccessfulFlag()) {
                    logInfo("Upload of LDB table MYTABLE was
successful.");
                else
                    logError("Upload of LDB table MYTABLE failed!");
            }
        };
```

```
        request.setCallbackHandler(callbackHandler);
        request.submit(false);

        return new VtiUserExitResult(0, "Success!");
    }
}
```

It should also be noted that it is possible to force a VTI heartbeat to occur before an LDB request is processed. This can be particularly important in a scenario where the SAP host needs to be notified of identity changes (for example, in the user ID or work group) in order to send down the correct data set via a download or refresh. The following code shows how this scenario can be handled.

Example 3

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyUserExit extends VtiUserExit {

    public VtiUserExitResult execute() throws VtiExitException {

        // Change the user ID to "FRED".

        VtiUserExitHeaderInfo headerInfo = getHeaderInfo();
        headerInfo.setUserId("FRED");

        // Force a download of the MYTABLE table under the new user
ID.

        VtiExitLdbTable table = getLdbTable("MYTABLE");

        if (table == null)
            throw new VtiExitException("LDB table MYTABLE not
found!");

        VtiExitLdbRequest request = new VtiExitLdbRequest
            (table, VtiExitLdbRequest.DOWNLOAD);

        request.setForceHeartbeatFlag(true); // To register new user
ID.
        request.submit(false);

        return new VtiUserExitResult(0, "Success!");
    }
}
```

Working with DOBs

Data objects represent a group of local database tables that are aggregated together for purposes of synchronisation scheduling. All Java exits, regardless of their type, are able to interrogate and manipulate data objects. The initial point of entry to this capability is the getLdbDataObject() method that returns a reference to a VtiExitLdbDataObject instance. In a manner similar to a VtiExitLdbTable reference, you can use this to interrogate / manipulate the metadata of the data objects, or alter its download / refresh / upload schedule. However, unlike a local database table, there is no facility for direct manipulation of data. An exit wishing to do this must instead obtain references to the relevant local database table(s) and proceed as described in section 1.3.3.3.

- Interrogation/Manipulation of metadata
- Task Scheduling

DOB Metadata Actions

Exits are able to interrogate, and in some limited cases directly alter, the metadata associated with a data object. You are provided the following methods for this type of activity:

Method
getDescription()
getErrorFlag()
getLastRefreshTime()
getLastUploadTime()
getName()
getNextRefreshTime()
getNextUploadTime()
getTableNames()
setErrorFlag()

DOB Task Scheduling

Exits are able to explicitly take control of local database data object synchronisation tasks that normally take place in background mode (downloads, refreshes, and uploads, of table data). You can achieve this by using a reference to the target data object (in the form of a VtiExitLdbDataObject object) to construct a VtiExitLdbRequest.

The following example code shows a user exit that forces a refresh to occur on a data object called "MYDATAOBJECT". This particular exit returns control to the user immediately rather than waiting for the refresh to complete.

```
package com.mycompany;
import au.com.skytechnologies.vti.*;
public class MyUserExit extends VtiUserExit {
    public VtiUserExitResult execute() throws VtiExitException {
        VtiExitLdbDataObject dataObject = getLdbDataObject
("MYDATAOBJECT");
        if (dataObject == null)
            throw new VtiExitException("LDB data object MYTDATAOBJECT
not found!");
        VtiExitLdbRequest request = new VtiExitLdbRequest
            (dataObject, VtiExitLdbRequest.REFRESH);
        request.submit(false);
        return new VtiUserExitResult(0, "Success!");
    }
}
```

4.11.13.7 Binary Files

You can directly manage binary files and how they are synchronized with SAP through functions provided in the exits. When adding files, you can schedule whether you can upload this file to the server or not, and whether this synchronization is to happen immediately. If you do not set to upload, the binary file can be queued for upload at a later time.

The classification of a file associates it with a particular data object as specified by the SAP workbench. This classification is able to be set or modified by the user exits and when set to be part of a data object, the uploading and synchronization of the file is controlled by the data object.

You are provided the following methods:

Method
addBinaryFile()
classifyBinaryFile()
deleteBinaryFile()
deleteBinaryFilesByClassification()
uploadBinaryFile()

4.11.13.8 Print Templates

The print template capability is provided to help facilitate the printing of simple reports and documents from within the Java Server. Print templates are configured within the configuration file, but they only really take effect when they are invoked from within a Java exit.

Print templates, as the name suggests, work from a template file. A template file is a text file that contains the body of the document to be printed, interspersed with what are known as "substitution variables". The server replaces the substitution variable values at run-time, and the resulting output is written to a work file located within a designated directory. The Java Server then invokes a system command specified in the configuration file. Typically, this command takes the work file and either sends it to a print spooler, or pipe it directly to a local printer through a serial or parallel port.

You can find an example to configure and use a print template here.

Generating PCL Print Templates using Word

If you are generating forms for a laser printer for example, invoice, then an easy way to do this is to use Microsoft Word to design the layout, embedding the substitution variables. In this way, you can design complex forms with embedded images.

An example procedure is outlined here.

Print Template Example

You want to write a print template that creates a simple docket and then sends it to a docket printer device connected through a serial port on COM1. You may set the configuration file entry for this print template as follows:

PrintTemplate

```
[PrintTemplate:Docket]
TemplateFileName = docket.txt
WorkDirectory = c:\temp
ShellCommand = cmd.exe /c
PrintCommand = type &filepath& > COM1
PrintCommandTimeout = 10
LogPrintCommands = true
DeleteWorkFiles = false
WorkFileExpiryDays = 7
```

The `docket.txt` file contains the "raw" document interspersed with substitution variables. It may look something like this:

```
docket.txt
Docket Number: &DOCKETNO:8ZR& Operator: &OPERATOR&
Docket Date: &DOCKETDATE:10L& Register: &REGISTER&
Docket Amount: &DOCKETAMT&
```

Substitution variables always begin and end with an ampersand (&). The `docket.txt` print template shown incorporates 5 substitution variables, called DOCKETNO, DOCKETDATE, DOCKETAMT, OPERATOR, and REGISTER. Substitution variables can optionally incorporate a formatting code that is separated from the name of the variable through a colon (:-) . In the above example, you can see such formatting codes on the DOCKETNO and DOCKETDATE variables. The formatting code provides a convenient method of ensuring that the variable value always conforms to a certain format. As you see shortly, you can achieve the same result by massaging the values passed in by the exit code, but format codes are a simple way of accomplishing the same goal. Format codes take the following form:

```
<mandatory length> <optional pad character> <optional alignment>
```

Where:

```
<mandatory length> is an integer value indicating the length of the
field.
<optional pad character> is either "S" (indicating spacepadded) or
"Z" (indicating zeropadded). Default if not specified is "S".
<optional alignment> is either "L" (indicating the field is
leftaligned) or "R" (indicating the field is rightaligned). Default
if not specified is "L".
```

In our example shown, the formatting codes you used for the DOCKETNO and DOCKETDATE fields ensure that they are always padded to the same length every time, so that the fields on the righthand side line up correctly. In order to invoke this print template from within an exit, you provide the server with the name of the print template you wish to invoke, and a list of key / value pairs that give us the values of the substitution variables.

```
VtiExitKeyValuePair [] keyValuePairs = {
    new VtiExitKeyValuePair("DOCKETNO", "123456"),
    new VtiExitKeyValuePair("DOCKETDATE", "01/01/2003"),
    new VtiExitKeyValuePair("DOCKETAMT", "100.55"),
    new VtiExitKeyValuePair("OPERATOR", "JOE"),
    new VtiExitKeyValuePair("REGISTER", "111")
};

invokePrintTemplate("Docket", keyValuePairs);
```

Obviously, this is a fairly simplified example. In reality the hardcoded values you passed for the each of the substitution variables normally derive from variables within the exit. However, if you want to execute the code shown above, the server builds a work file that looks like this:

Docket

```
Docket Number:00123456 Operator: JOE
Docket Date: 01/01/2003 Register: 111
Docket Amount:100.55
```

The server then executes the print command specified in the VTI configuration file for the print template that in this case is "type &filepath& > COM1". As is apparent from the "&filepath&" within the command, the Java Server supports the concept of substitution variables within the print command itself. The substitution variables that you can use in this context are listed in the configuration file documentation. The &filepath& substitution variable is used to denote the full pathname to the work file. This has the effect of typing the contents of the work file to the COM1 port that (all being well) results in our docket printer device printing our docket.

Generating PCL Print Templates using Word

> ***Important:*** This solution is only viable if the target printer supports PCL.

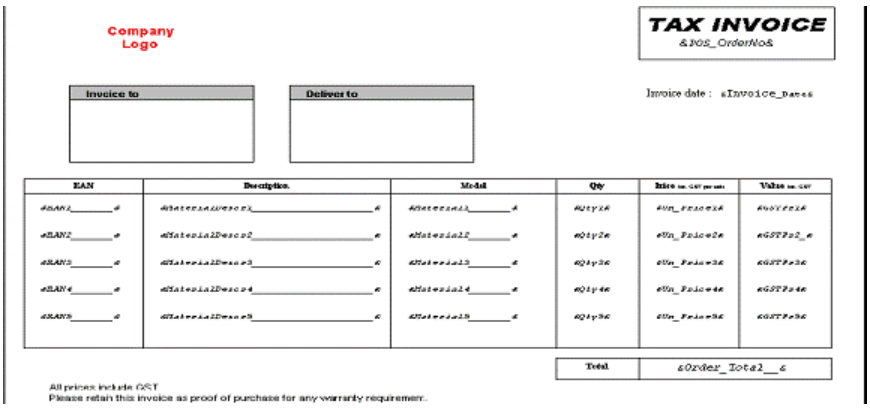
Basic Process

1. Install a Postscript printer driver, for example, Adobe Generic Postscript Driver, to print output to file as PCL `.prn` file.
2. Implement a Windows printer definition using the Postscript driver.
3. Design the form in Word (or equivalent).
4. Print the Word document to file, using the postscript printer. This generates a PCL template.
5. Implement a Java print exit to dynamically substitute any variables. See the Java programming guide section / Printing through print templates for details.
6. Direct the generated print file to a PCL compatible printer in the Java server configuration.

> ***Note:*** You can use any software that can output the print template to a file using the Postscript Driver.

Designing the Print Template

Using the word processing software or equivalent, layout the design of the document as required (see sample Invoice with embedded substitution variables below).



Special Formatting

You can use different fonts, colour and shading to highlight the design. You can also place these inside boxes as shown in the sample. You can import and insert company logo (stored as jpeg, bitmap or tiff format) as part of the document. However, the imported object increases the PCL print file size. This can result in an increase in data traffic across the network, dependant on the print volume.

Substitution Variables

You need to enclose substitution variables on the print template within an 'ampersand (&)'. You should use the 'courier' or 'courier new' fonts or best results. You can format the substitution variable as required with font size, bolding, italics, and underline.

To ensure that the substitution variable is substituted correctly, the substitution variable must be the same size as the data to be printed. If the substitution variable is shorter than the data, you can pad it with 'underscore'. The following is an example to substitute an 18 character material number:

&Material_____&

> ***Note:*** The ampersand is included in the character count.

Verifying the PCL Print File (important)

Prior to using the print file, view the print file and verify the content for correctness. You can view the print file using 'Notepad' or 'Wordpad'. Find all the substitution variables to ensure that they are enclosed within the ampersand, for example, &variable&.

If you do not enclose the substitution variable with ampersand or do not split into two separate lines, return to the template and delete the variable and re-key. After you save the changes, print the output to file using the Postscript Print Driver.

Repeat the substitution variables verification using the new PCL print file.

Using Number Ranges

All Java exits, regardless of their type, are able to use number ranges to obtain unique integer values. You must first define the number range within the SAP workbench. Once it is defined, you can access within an exit code by invoking either the getNextNumberFromNumberRange() method (for a single number) or the getNextNumbersFromNumberRange() method (for multiple contiguous numbers).

The following obtains the next available number from the "INVOICE_NOS" number range.

```
long invoiceNumber = getNextNumberFromNumberRange
    ("ProductionHostInterface", "INVOICE_NOS");
```

4.11.13.9 Using Number Ranges

All Java exits, regardless of their type, are able to use number ranges to obtain unique integer values. The number range must first be defined within the SAP workbench. Once it has been defined, it can be accessed within an exit's code by invoking either the getNextNumberFromNumberRange() method (for a single number) or the getNextNumbersFromNumberRange() method (for multiple contiguous numbers).

The following obtains the next available number from the "INVOICE_NOS" number range.

```
long invoiceNumber = getNextNumberFromNumberRange
    ("ProductionHostInterface", "INVOICE_NOS");
```

4.11.13.10 Date Formatter

SkyMobile uses separate custom date and time formats that are easy to store in either integer or text form in a database. Dates are stored as "YYYYMMDD" and times are stored as "HHMMSS". The DateFormatter class facilitates the conversion between this internal format and the java Date format. The DateFormatter class is in the package "au.com.skytechnologies.ecssdk.util".

To convert to SkyMobile format:

- getDateAsInt()
- getTimeAsInt()

The default value when instantiating the java "Date" class is the current time and date. To retrieve the current date or time use:

- getDateAsInt(new Date())
- getTimeAsInt(new Date())

To convert to date format from SkyMobile format:

- getDateFromInt()
- getTimeFromInt()
- parseDate()
- parseTime()

> ***Note:*** The parseDate and parseTime require you to first call initialiseDateParsing or initialiseTimeParsing respectively or else an exception occurs.

4.11.13.11 String Utilities

Java 1.1.8, is the version supported by CrEme on Windows Mobile, has extremely little in the way of useful string formatting utilities. To assist with this, SkyMobile includes a number of classes in the package "au.com.skytechnologies.ecssdk.util" to provide some of the more common functions.

- Convertor - provides an additional numeric to string conversions.
- StringUtil - provides a number of tools for trimming, replacing, and padding text.
- StringTokenizer – a version of the java.util.StringTokenizer class

4.11.13.12 XML Parsing

Frequently when interfacing with data retrieved from third parties, the data is formatted in XML. There are several libraries that are freely available for use, however many are quite big and unnecessarily complicated for simple tasks. A simple SAX-like parser is provided to assist with decoding simple XML data. To use the simple XML parser, you need to create a class that implements the SimpleXmlHandler interface and is provided to a new instance of a SimpleXMLParser class. When you call the SimpleXMLParser "parse" function, events are triggered in your handler as each element or attribute is retrieved. It is up to your handler to keep track of the traversal through the element hierarchy. In the event of an error in the XML, an XmlException occurs that you should catch by your code.

These classes are located in the package "au.com.skytechnologies.ecssdk.xml".

The following example demonstrates how you can potentially parse a stream into a hierarchy of XmlElements:

```
import java.io.InputStream;
import java.util.HashMap;
import java.util.Vector;

import au.com.skytechnologies.ecssdk.xml.SimpleXmlHandler;
import au.com.skytechnologies.ecssdk.xml.SimpleXmlParser;
import au.com.skytechnologies.ecssdk.xml.XmlException;

public class MyXmlHandler implements SimpleXmlHandler {

	private Vector stack = new Vector();
	private Vector rootElements = new Vector();

	public class XmlElement {
		public String name = null;
		public String value = null;
		public HashMap elements = new HashMap();
		public HashMap attributes = new HashMap();
	}
```

```
	/**
	 * Entry point for parsing function
	 * @param is The stream containing the xml to be parsed
	 * @return A vector containing root XmlElement objects
	 * @throws Exception On parsing error
	 */
	public static Vector parseXml(InputStream is) throws Exception
{
		MyXmlHandler handler = new MyXmlHandler();
		SimpleXmlParser parser = new SimpleXmlParser(handler);
		parser.parse(is);
		return handler.rootElements;
	}

	/**
	 * Event signalling that the current element has an attribute
	 */
	public void attribute(String name, String value) throws
Exception {
		XmlElement element = (XmlElement)stack.elementAt
(stack.size() - 1);
		element.attributes.put(name, value);
	}

	/**
	 * Event signalling that we've finished parsing an element
	 */
	public void endElement(String name) throws Exception {
		stack.remove(stack.size() - 1);
	}

	/**
	 * Error handler
	 */
```

```
	public void error(String message, int line) throws Exception {
		throw new XmlException(message, line);
	}

	/**
	 * Event signalling that we're starting parsing of an element
	 */
	public void startElement(String name) throws Exception {
		XmlElement element = new XmlElement();
		element.name = name;
		stack.add(element);
		if (stack.size() == 0)
			rootElements.add(element);
	}

	/**
	 * Event signalling that the current element has a value
assigned
	 */
	public void value(String name, String value) throws Exception {
		XmlElement element = (XmlElement)stack.elementAt
(stack.size() - 1);
		element.value = value;
	}
}
```

4.11.13.13 Java Exit Types

Java exits may be provided as either "screen exits" or "server exits".

Screen Exits

Java screen exits are also called "User Exits" and are specified within the SAP workbench and allow java code to be executed alongside or in place of ABAP screen exits. One of the most common reasons for writing java code for your screen exits is that it allows to perform functionality when the java server is offline and unable to communicate with SAP.

Java Server Exits

Java server exits are defined in the Secure Containers configuration file (`skymobile.cfg`) and consist of a number of different types:

- Server exits - Invoked by the server whenever certain system events take place (for example, at start up or shut down, or when SAP connectivity is lost). Java server exits are not associated with any particular user and are the primary way in which you can implement custom "background tasks".
- Poll directory exits - Invoked when an incoming file is detected in the configured poll directory.
- Custom port exits - Automatically invoked when the server detects an incoming message on a connection established through a configured network port.
- Custom serial device exits - Automatically invoked when the server detects an incoming message from the configured serial device.
- Identity service exits - Permit custom functionality to introduce into the SkyMobile Access Gateway, to cater for specific requirements during verification of a user identity.

Each of the major types of Java exit is implemented as a separate Java class (VtiUserExit, VtiServerExit, VtiPollDirectoryExit, VtiCustomPortExit, VtiCustomSerialDeviceExit, and VtiIdentityServiceExit respectively). These classes all extend a base class called VtiExit that describes the basic functionality common to all types of exit.

User Exits

User (screen) exits are probably the most common type of Java exit. They are similar to ABAP exits in that they are invoked in response to specific actions that the user takes while interacting with an application.

> ***Important:*** Java exits are only called by the Java server, and are ignored by the SAPGUI emulator.

Configuration

In Java, user exits are defined by extending the "VtiUserExit" class and implementing its "Execute" function. In SAP, you configure within the SAP workbench using exit definitions, in the same way as ABAP exits. Click here to see the exit definition screen. You may specify the Java exit in three different modes:

- None – There is no Java exit associated with this exit definition
- Equivalent – Use the Java exit in place of the ABAP exit; or
- Complimentary – Call the Java exit as well as the ABAP exit.

If you do not specify the class name, the default name is used. When deriving the name of the Java user exit class to invoke for a given exit, the server appends the name of the class to the exit package name specified on the version definition. Thus, in the example screen, if you set the exit package name to "com.mycompany", the server looks for a Java user exit class called "com.mycompany.ContactEditFormat".

Specific Functionality

User exits are able to get and set values from a user's screen. You can obtain a reference to a user's screen field by invoking the getScreenField() method that returns a VtiUserExitScreenField object (or null if the specified screen field does not exist). You can also use the getScreenFields() method to obtain an array of all screen fields for the user session. Click here to see an example exit that modifies screen fields.

Because you already know the SAP interface that a user is logged into, the VtiUserExit class also provides some additional functions that no longer require you to specify this:

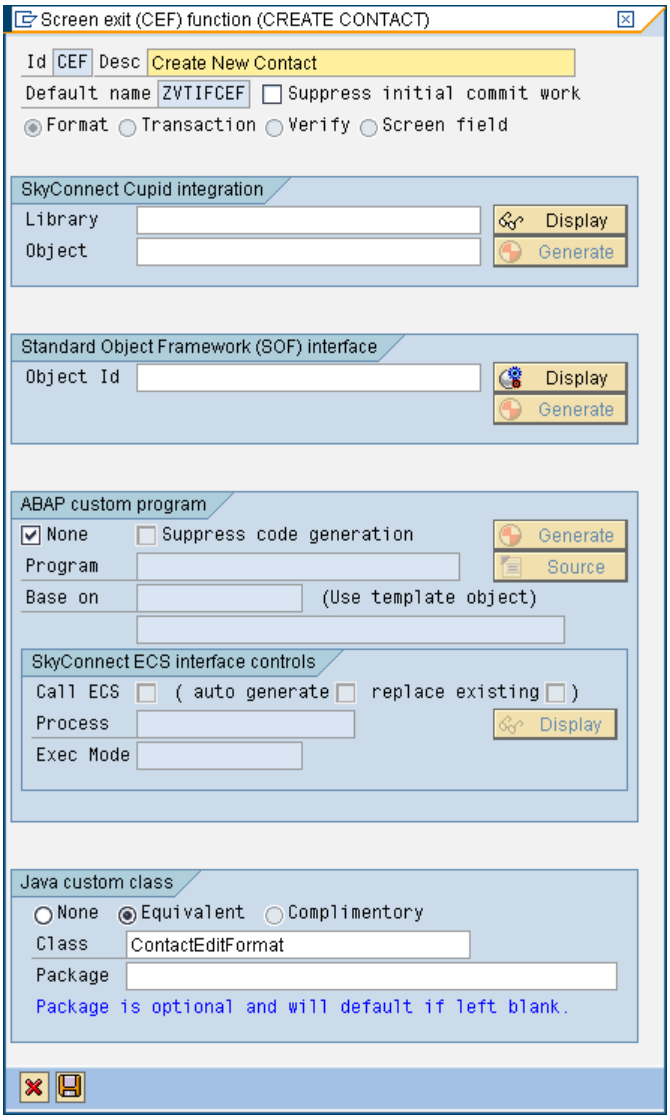
- getNextNumberFromNumberRange()
- getNextNumbersFromNumberRange()
- getLdbTable()

Error Handling

Once a user exit finished processing, it must notify the server whether or not processing is successful. It does this by either throwing a VtiExitException, or by returning a VtiUserExitResult once processing is completed. A user exit is treated as having encountered an error if it throws a VtiExitException, or returns a VtiUserExitResult with a nonzero return code. It is considered to have completed successfully if it returns a VtiUserExitResult with a zero return code, or returns null.

Unsuccessful completion of a user exit causes the server to roll back any uncommitted database changes and returns control to the user. Any error message returned, is displayed to the user. Successful completion of a user exit causes you to commit any uncommitted database changes and allows processing to continue normally.

User Exit Definition



Field	Description
No Java exit	There is no Java exit associated with this exit definition
Equivalent Java exit	The Java exit is invoked instead of an ABAP exit (if specified)
Complimentary Java exit	The Java exit is invoked as well as the ABAP exit (both are called).

Field	Description
Class	The class name. If you do not specify, the default name is used.

User Exit Example

The following user exit takes the value from a screen field called "CODE", attempts to find a corresponding record on a local database table called "MYTABLE", and then returns a description back to a screen field called "DESC".

```
package com.mycompany;
import au.com.skytechnologies.vti.*;
public class MyUserExit extends VtiUserExit {

   public void execute() throws VtiExitException {
        VtiUserExitScreenField codeField = getScreenField("CODE");
        if (codeField == null)
            throw new VtiExitException
                ("Screen field CODE not found!");
        String code = codeField.getFieldValue();
        VtiExitLdbTable table = getLdbTable("MYTABLE");
        if (table == null)
            throw new VtiExitException("LDB table MYTABLE not
found!");

       VtiExitLdbSelectCondition cond =
            new VtiExitLdbSelectCondition("CODE",
                VtiExitLdbSelectCondition.EQ_OPERATOR, code);

       VtiExitLdbTableRow [] rows = table.getMatchingRows(cond);

       if (rows == null || rows.length == 0)
            throw new VtiExitException("Code not found!");

       if (rows.length > 1)
            throw new VtiExitException("Multiple rows found!");

       String desc = rows[0].getFieldValue("DESC");
        if (desc == null)
            throw new VtiExitException
```

```
                ("Field DESC not found on LDB table MYTABLE!");

        VtiUserExitScreenField descField = getScreenField("DESC");

       if (descField == null)
            throw new VtiExitException
                ("Screen field DESC not found!");

       descField.setFieldValue(desc);
        return new VtiUserExitResult();
    }
}
```

Server Exits

Server exits are not associated with any individual user or user session but exist for the duration of the SkyMobile Secure Container on the device. Server exits extend the VtiServerExit class and are invoked by the server in response to the occurrence of certain predefined system events:

Start
Stop
Connected
Disconnected
BackupRestoreStart
BackupRestoreFinish

It is critical that these function calls are kept as quick as possible as delays can trigger timeouts in the communications between the presentation client and the server. If long running processing is required, it is advisable to launch a separate thread and return immediately. This exit is quite often used to launch and manage additional threads to control custom processes that run for the life of the server. (See "Implementing a background task")

Configuration

Server exits are configured in the configuration file. There is a section called "SERVEREXITS" in this file that tells the Container the name of the server exits that you should load. An example is shown below.

```
SERVER.SERVEREXITS.SERVEREXIT1 = com.mycompany.MyServerExit
SERVER.SERVEREXITS.SERVEREXIT2 = com.mycompany.MyOtherServerExit
```

In this example, the two Java classes specified are "com.mycompany.MyServerExit" and "com.mycompany.MyOtherServerExit". The configuration parameter key values are not important, provided they are unique within the "SERVEREXITS" section.

Error Handling

In most situations there is no sensible generic response that the server can undertake when application errors arise inside a server exit. Any such error conditions that may occur therefore you need to handle by the code that performs the task. Often this involves writing detailed messages to the log through methods such as logWarn(), logError(), or even (in the event of a catastrophic failure) logAbort().

Many server exits launch a background thread that loops continuously, waiting for some event to occur in order to begin processing. When writing such threads, it is important to consider error handling behaviour. A thread that polls for some condition to occur 5 times a second very quickly creates an enormous log file if it keeps encountering the same error, and writes a log message each time it does so. A more graceful way to handle this type of situation is to place the background thread in a sleep or wait state for some arbitrary error retry interval before trying again.

Unexpected exceptions that a server exit may throw at runtime is automatically caught and written to the log, along with a stack trace.

Implementing a Background Task

In the normal course of events, a server exit is invoked when a system event takes place. Its code is executed and it then releases control back to the server once its job is done. However, if there is a need for your custom code to remain active once an event comes and goes, the best way to accomplish this is to start a background task.

Background tasks are usually linked to the start() and stop() methods for a Java server exit. A new thread starts when the server starts up, and stops when the server shuts down.

Class "com.mycompany.MyServerExit"

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public MyServerExit extends VtiServerExit {

    MyBackgroundThread thread = null;

    public void start() {

        thread = new MyBackgroundThread();
        thread.start();

    }

    public void stop() {

        if (thread != null)
            thread.halt();
    }

}
```

Class "MyBackgroundThread"

```
package com.mycompany;

public MyBackgroundThread extends Thread {

    private Boolean halted = false;

    public void run() {

        while (!halted) {

            // Insert some code to do a background task here.

        }

    }

    public void halt() {

        halted = true;

    }

}
```

Poll Directory Exits

Poll directory exits extend the VtiPollDirectory class. When a file arrives in the configured `incoming directory`, the Java server invokes the execute() method in the associated Java class. This method is responsible for taking whatever action is necessary to process the file. You can obtain a handle to the file in question by invoking the getFile() method.

> ***Important:*** A common error when writing poll directory exits is to forget to close input streams and / or readers that reference the input file. This effectively leaves the file temporarily "locked", and the server is unable to move or delete it, resulting in errors written to the log. In order to avoid this problem, take care to ensure that your exit always closes any input streams and / or readers that reference the input file, before returning control to the server. For example, in the code shown above, the BufferedReader ("br") used to read the input file is closed before the exit returns.

You can find an example poll directory exit here.

Configuration

Poll directory exits are configured in the Kony for SAP configuration file. This file can contain one or more "POLLDIRECTORY:xxx" sections that each relate to a directory that the server is expected to monitor for incoming files. An example poll directory configuration file section is shown here, however details of the specific parameters are specified in Configuration options.

Error Handling

Once an exit finishes processing a file, it must notify the server whether or not the file is successfully processed. It does this by either throwing a VtiExitException, or by returning a VtiPollDirectoryExitResult once processing is completed. A poll directory exit is normally treated as having encountered an error if it throws a VtiExitException, or returns a VtiPollDirectoryExitResult with a nonzero return code. It is normally considered to have completed successfully if it returns a VtiPollDirectoryExitResult with a zero return code, or returns null. You can override the default behaviour as specified in the configuration file by code. Refer Overriding the default behaviour for details.

Example configuration - poll directory exit

Example Configuration for Poll Directory "MYPOLLDIRECTORY"

```
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.INCOMINGDIRECTORY =
c:\mypolldir\incoming
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.ARCHIVEDIRECTORY =
c:\mypolldir\archive
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.ERRORDIRECTORY =
c:\mypolldir\error
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.INCOMINGFILEPATTERN = .
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.POLLFREQUENCY = 30
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.LOGINCOMINGFILES = true
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.STOPPOLLINGONERROR = false
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.EXITCLASSNAME =
com.mycompany.MyPollDirectoryExit
```

In this example, the Java class that is invoked whenever an incoming file is detected (in the directory `c:\mypolldir\incoming`) is called "com.mycompany.MyPollDirectoryExit". The section name may be anything that uniquely identifies the poll directory, but must begin with the prefix "POLLDIRECTORY:"

Poll Directory Exit Example

The following sample code implements a poll directory exit that reads an incoming tabdelimited file containing two fields, a code and a description. The incoming codes and descriptions are then written to a local database table called "MYTABLE".

```
package com.mycompany;

import au.com.skytechnologies.vti.*;
import java.io.*;
import java.util.*;

public class MyPollDirectoryExit extends VtiPollDirectoryExit {

    public VtiPollDirectoryExitResult execute() throws
VtiExitException {

        File file = getFile();
        FileReader fr;
```

```
        try {
            fr = new FileReader(file);
        }
        catch (FileNotFoundException fnfe) {
            throw new VtiExitException("Incoming file not found!");
        }

        BufferedReader br = new BufferedReader(fr);

        VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYTABLE");

        try {
            beginTransaction();

            String line;
            int lineNo = 0;

            while ((line = br.readLine()) != null) {
                ++lineNo;

                StringTokenizer strTok = new StringTokenizer(line,
"\t");

                if (strTok.countTokens() != 2)
                    throw new VtiExitException
                        ("Bad field count on line " + lineNo);

                VtiExitLdbTableRow row = table.newRow();
                row.setFieldValue("CODE", strTok.nextToken());
                row.setFieldValue("DESC", strTok.nextToken());

                try {
                    table.saveRow(row);
```

```
                }
                catch (VtiExitException ee) {
                    rollbackTransaction();
                    br.close();
                    throw ee;
                }
            }

            commitTransaction();
            br.close();
        }
        catch (IOException ioe) {
            rollbackTransaction();
            br.close();
            throw new VtiExitException("I/O error!");
        }

        return new VtiPollDirectoryExitResult(0, "Success!");
    }
}
```

```
package com.mycompany;

import au.com.skytechnologies.vti.*;
import java.io.*;
import java.util.*;

public class MyPollDirectoryExit extends VtiPollDirectoryExit {

    public VtiPollDirectoryExitResult execute() throws
VtiExitException {

        File file = getFile();
        FileReader fr;
```

```
        try {
            fr = new FileReader(file);
        }
        catch (FileNotFoundException fnfe) {
            throw new VtiExitException("Incoming file not found!");
        }

        BufferedReader br = new BufferedReader(fr);

        VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYTABLE");

        try {
            beginTransaction();

            String line;
            int lineNo = 0;

            while ((line = br.readLine()) != null) {
                ++lineNo;

                StringTokenizer strTok = new StringTokenizer(line,
"\t");

                if (strTok.countTokens() != 2)
                    throw new VtiExitException
                        ("Bad field count on line " + lineNo);

                VtiExitLdbTableRow row = table.newRow();
                row.setFieldValue("CODE", strTok.nextToken());
                row.setFieldValue("DESC", strTok.nextToken());

                try {
                    table.saveRow(row);
```

```
                }
                catch (VtiExitException ee) {
                    rollbackTransaction();
                    br.close();
                    throw ee;
                }
            }

            commitTransaction();
            br.close();
        }
        catch (IOException ioe) {
            rollbackTransaction();
            br.close();
            throw new VtiExitException("I/O error!");
        }

        return new VtiPollDirectoryExitResult(0, "Success!");
    }
}
```

Example Configuration - Poll Directory Exit

Example Configuration for Poll Directory `MYPOLLDIRECTORY`

```
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.INCOMINGDIRECTORY
=c:\mypolldir\incomingSERVER.POLLDIRECTORY:MYPOLLDIRECTORY.ARCHIVEDI
RECTORY = c:\mypolldir\archive
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.ERRORDIRECTORY =
c:\mypolldir\error
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.INCOMINGFILEPATTERN = .
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.POLLFREQUENCY = 30
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.LOGINCOMINGFILES = true
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.STOPPOLLINGONERROR = false
SERVER.POLLDIRECTORY:MYPOLLDIRECTORY.EXITCLASSNAME =
com.mycompany.MyPollDirectoryExit
```

In this example, the Java class that is invoked whenever an incoming file is detected (in the directory `c:\mypolldir\incoming`) is called "com.mycompany.MyPollDirectoryExit". The section name may be anything that uniquely identifies the poll directory, but must begin with the prefix "POLLDIRECTORY:"

4.11.14 Poll Directory Exit Example

The following sample code implements a poll directory exit that reads an incoming tabdelimited file containing two fields, a code and a description. The incoming codes and descriptions are then written to a local database table called "MYTABLE".

Poll Directory Exit Example

```
package com.mycompany;

import au.com.skytechnologies.vti.*;
import java.io.*;
import java.util.*;

public class MyPollDirectoryExit extends VtiPollDirectoryExit {

    public VtiPollDirectoryExitResult execute() throws
VtiExitException {

        File file = getFile();
        FileReader fr;

        try {
            fr = new FileReader(file);
        }
        catch (FileNotFoundException fnfe) {
            throw new VtiExitException("Incoming file not found!");
        }

        BufferedReader br = new BufferedReader(fr);

        VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYTABLE");

        try {
```

```
            beginTransaction();

            String line;
            int lineNo = 0;

            while ((line = br.readLine()) != null) {
                ++lineNo;

                StringTokenizer strTok = new StringTokenizer(line,
"\t");

                if (strTok.countTokens() != 2)
                    throw new VtiExitException
                        ("Bad field count on line " + lineNo);

                VtiExitLdbTableRow row = table.newRow();
                row.setFieldValue("CODE", strTok.nextToken());
                row.setFieldValue("DESC", strTok.nextToken());

                try {
                    table.saveRow(row);
                }
                catch (VtiExitException ee) {
                    rollbackTransaction();
                    br.close();
                    throw ee;
                }
            }

            commitTransaction();
            br.close();
        }
        catch (IOException ioe) {
            rollbackTransaction();
            br.close();
```

```
            throw new VtiExitException("I/O error!");
        }

        return new VtiPollDirectoryExitResult(0, "Success!");
    }
}
```

4.11.15 Overriding the Default Behavior

Poll directories have default behavior that they manifests unless the code inside the exit takes explicit control. For example, if you set the "StopPollingOnError" configuration item in the relevant configuration file section, this tells the server that if the exit encounters an error, the default behavior is to stop polling the directory. However, an exit can override this default behavior if it were to execute the following code:

```
return new VtiPollDirectoryExitResult(999,
    "Non-fatal error encountered, polling will continue",
    DONT_STOP_POLLING);
```

There are a number of different aspects of the server behavior that you can control in this way. These are listed in the following table.

Behavior	Flags	Default behavior
Error status (whether or not the server treats the result as an error, regardless of the return code sent back).	TREAT_AS_ERROR DONT_TREAT_AS_ERROR	Default behavior is to treat the result as a success if it has a zero return code, or an error if it has a non-zero return code.
Message logging (whether or not any message returned by the exit is written to the log).	LOG_MESSAGE DONT_LOG_MESSAGE	Default behavior is set in the configuration file through the **LogIncomingFiles** option.

Behavior	Flags	Default behavior
Return code logging (whether or not any return code sent back by the exit is written to the log).	LOG_ RETURN_ CODE DONT_ LOG_ RETURN_ CODE	Default behavior is set in the configuration file through the **LogIncomingFiles** option.
Database rollback (whether or not any outstanding database changes are rolled back).	PERFORM_ ROLLBACK DONT_ PERFORM_ ROLLBACK	Default behavior is to only perform a rollback if an error occurs.
Polling control (whether or not polling should be halted until manual intervention takes place).	STOP_ POLLING DONT_ STOP_ POLLING	Polling is normally not stopped unless an error occurs. Default behavior when an error occurs is set in the configuration file through the **StopPollingOnError** option.

Custom Port Exits

Java custom port exits extend the VtiCustomPortExit class. They are automatically invoked when the server detects an incoming message on a connection established through a custom port defined in the configuration file. If multiple connections are made to the port, a new instance of the java exit is instantiated for each connection. No connection or disconnection events are generated by the server other than instances of the exit being created and garbage collected.

When setting up a custom port, it is important to consider the nature of the incoming data to ensure that the Java Server definition of what a "message" consists of, matches the desired outcome. Incoming data received over a connection to the port should conform to some type of logical structure. Examples might include messages of a fixed length, or messages terminated by a delimiter character (such as a carriage return or line feed) that signifies the incoming data is now ready for processing. The server permits custom ports to be configured in a variety of different ways, to cater for these different situations.

You can find an example custom port exit here.

Configuration

Java custom port exits are configured in the configuration file. This file can contain one or more "CUSTOMPORT:xxx" sections that each relate to a custom port that the server is expected to monitor for incoming messages. An example custom port configuration file section is shown here, however you can find the details of the specific parameters specified in the "Installation Guide" under "The Java Server Configuration File".

Specific Functionality

In a situation where it is necessary to write a response or acknowledgement back to the connection over which a message is received, you can accomplish this by using the write() method. This method can take either a Java string or a byte array as possible sources for the information to be written.

Error Handling

The server provides a facility for reprocessing of input received on a custom port connection. The integer value returned from the processInput() method specifies the number of bytes left unprocessed from the message just sent. The server automatically adds this number of bytes back to the input stream to be reprocessed. Returning a value of inputBytes.length effectively forces the server to reprocess the entire message.

> *Important:* Use this option with great care, as it is possible to create a never ending loop if you cannot resolve the error during subsequent processing attempts.

The server automatically traps any uncaught exceptions occurring inside a custom port exit, and writes them to the log file, along with a stack trace.

Example Configuration - Custom Poll Exits

Example Configuration for Custom Port "PORT5075"

```
SERVER.CUSTOMPORT:PORT5075.PORT = 5075
SERVER.CUSTOMPORT:PORT5075.INPUTMESSAGETYPE = F
SERVER.CUSTOMPORT:PORT5075.INPUTFIXEDLENGTH = 150
SERVER.CUSTOMPORT:PORT5075.CUSTOMPORTEXIT =
com.mycompany.MyCustomPortExit
```

In this example, the custom port is configured to expect fixedlength input messages that are 150 characters in length. The Java class that is invoked to handle any such messages is called "com.mycompany.MyCustomPortExit". The section name may be anything that uniquely identifies the custom port, but must begin with the prefix "CUSTOMPORT:"

Custom Port Exit Example

When a message arrives on a connection opened on a custom port, the Java server invokes the processInput() method in the associated Java custom port exit class. This method is responsible to take whatever action is necessary to process the message. The content of the message is passed to the processInput() method through two parameters, inputString (that contains the message encoded as a Java string), and inputBytes (that contains the raw bytes from which the message is composed). In most cases inputString is more useful of the two parameters. However, there are some situations (particularly when dealing with incoming characters that have the high-order bit set) in which the raw bytes may be useful.

The following code extracts a material number and a quantity from arbitrary positions within an incoming message, then writes these fields into a local database table called "MYSTOCK".

Example 1

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyCustomPortExit extends VtiCustomPortExit {

   public int processInput(String inputString, byte [] inputBytes) {

        String matnr = inputString.substring(10,15);
        String qtyStr = inputString.substring(55,65);
        int qtyInt;

        try {
            qtyInt = Integer.parseInt(qtyStr);
        }
        catch (NumberFormatException nfe) {
            logError("Non-numeric quantity (" + qtyStr +
                ") for material code " + matnr + ".", nfe);
            return 0;
        }
```

Example 2

```
VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYSTOCK");

        beginTransaction();

        VtiExitLdbTableRow row = table.newRow();
        row.setFieldValue("MATNR", matnr);
        row.setFieldValue("QTY", qtyInt);

        try {
            table.saveRow(row);
        }
        catch (VtiExitException ee) {
            rollbackTransaction();
            logError("Error saving row for material code " + matnr +
                ", quantity " + qtyInt + ".", ee);
            return 0;
        }

        commitTransaction();
        return 0;
    }
}
```

Custom Serial Device Exits

Java custom serial device exits extend the VtiCustomSerialDeviceExit class. They are automatically invoked when the server detects an incoming message from serial devices (such as bar code scanners, smart card readers, credit card readers, cheque readers, weighbridges, and other such devices that generate input).

When setting up a custom port, it is important to consider the nature of the incoming data to ensure that the Java Server definition of what a "message" consists of, matches the desired outcome. Incoming data received over a connection to the port should conform to some type of logical structure. Examples may include messages of a fixed length, or messages terminated by a delimiter character (such as a carriage return or line feed) that signifies the incoming data is now ready for processing. The server permits custom ports to configure in a variety of different ways, to cater for these different situations.

The server has two different ways in which it can communicate with serial devices that are as follows:

- Through the Java Comm Extension. The Java Comm Extension is a standard extension for Java that you can download from http://java.sun.com. Once installed (as per its installation instructions), it enables Java programs to communicate with serial devices. The server uses this method by default. The Java Comm Extension is not available on all platforms, but it does run under many flavours of Unix as well as Windows.
- Through a Custom Native Library. An alternative method for serial device communications is the native library that comes standard. This method is only available on the Windows platform, but does get around the need to install the Java Comm Extension.

The thing controlling which of these two methods is used is the "UseNativeComm" parameter in the configuration file section dealing with the custom serial device. By default, the Java Comm Extension is used. However, if "UseNativeComm" is set to *true*, the native serial device communications library is used instead.

Communication with serial devices is a notoriously problematic area. If you experience difficulty, refer to the information included under the Troubleshooting section in the General Installation Guide.

When a message arrives from a custom serial device, the Java server invokes the processInput() method in the associated Java custom serial device exit class. This method is responsible to take whatever action is necessary to process the message. The content of the message is passed to the processInput() method through two parameters, inputString (that contains the message encoded as a Java string), and inputBytes (that contains the raw bytes from which the message is composed). In most cases inputString is more useful of the two parameters. However, there are some situations (particularly when dealing with incoming bytes that have the highorder bit set) in which the raw bytes may be useful.
You can find an example custom port exit here.

Configuration

Java custom serial device exits are configured in the configuration file. This file can contain one or more "CUSTOMSERIALDEVICE:xxx" sections, which each relate to a custom serial device that the server is expected to monitor for incoming messages. An example custom serial device configuration file section is shown here, however details of the specific parameters are specified in the "Installation Guide" under "The Java Server Configuration File".

Specific Functionality

In a situation where it is necessary to write a response or acknowledgement back to the serial device from which a message has been received, this can be accomplished by using the write() method. This method can take either a Java string or a byte array as possible sources for the information to be written.

Error Handling

As with custom port exits, the server provides a facility for reprocessing of input received on a custom serial device connection, through the integer parameter returned from the processInput() method. The server automatically adds this number of bytes back to the input stream to reprocess.

> ***Important:*** As with custom ports, you should use this option with great care as it is possible to create a neverending loop if the error cannot be resolved during subsequent processing attempts.

The server automatically traps any uncaught exceptions occurring inside a custom serial device exit, and writes them to the log file along with a stack trace.

Example Configuration - Custom Serial Device Exit

Example Configuration for Custom Serial Device "CARDSCANNER"

```
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.COMMPORT = COM1
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.SPEED = 9600
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.DATABITS = 8
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.STOPBITS = 1
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.PARITY = None
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.FLOWCONTROL = None
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INITIALCONNECT = true
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER. RECONNECTINTERVAL = 60
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.USENATIVECOMM = false
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTMESSAGETYPE = D
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTDELIMITERS = \010
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTSTRIPDELIM = true
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.CUSTOMSERIALDEVICEEXIT =
com.mycompany.MyCustomSerialDeviceExit
```

In this example, the custom port is configured to expect an input message that terminates with a newline character (ASCII code 010). The Java class that is invoked to handle any such messages is called "com.mycompany.MyCustomSerialDeviceExit". The section name may be anything that uniquely identifies the custom serial device, but must begin with the prefix "CUSTOMSERIALDEVICE:"

Custom Serial Device Exit Example

The following code extracts a card number from an arbitrary position within an incoming message, and then writes it into a local database table called "MYSCANS".

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyCustomSerialDeviceExit extends
```

```
VtiCustomSerialDeviceExit {

    public int processInput(String inputString, byte [] inputBytes)
{

        String cardNo = inputString.substring(0, 10);

        VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYSCANS");

        beginTransaction();

        VtiExitLdbTableRow row = table.newRow();
        row.setFieldValue("CARDNO", cardNo);

        try {
            table.saveRow(row);
        }
        catch (VtiExitException ee) {
            rollbackTransaction();
            logError("Error saving row for card number " + cardNo +
"."", ee);
            return 0;
        }

        commitTransaction();
        return 0;
    }
}
```

Example Configuration - Custom Serial Device Exit

Example Configuration for Custom Serial Device "CARDSCANNER"

```
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.COMMPORT = COM1
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.SPEED = 9600
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.DATABITS = 8
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.STOPBITS = 1
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.PARITY = None
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.FLOWCONTROL = None
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INITIALCONNECT = true
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER. RECONNECTINTERVAL = 60
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.USENATIVECOMM = false
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTMESSAGETYPE = D
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTDELIMITERS = \010
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.INPUTSTRIPDELIM = true
SERVER.CUSTOMSERIALDEVICE:CARDSCANNER.CUSTOMSERIALDEVICEEXIT =
com.mycompany.MyCustomSerialDeviceExit
```

In this example, the custom port is configured to expect an input message that is terminated with a newline character (ASCII code 010). The Java class that is invoked to handle any such messages is called "com.mycompany.MyCustomSerialDeviceExit". The section name may be anything that uniquely identifies the custom serial device, but must begin with the prefix "CUSTOMSERIALDEVICE:"

Custom Serial Device Exit Example

The following code extracts a card number from an arbitrary position within an incoming message, and then writes it into a local database table called "MYSCANS".

```
package com.mycompany;

import au.com.skytechnologies.vti.*;

public class MyCustomSerialDeviceExit extends
VtiCustomSerialDeviceExit {

    public int processInput(String inputString, byte [] inputBytes) {

        String cardNo = inputString.substring(0, 10);

        VtiExitLdbTable table = getLdbTable
                ("ProductionHostInterface", "MYSCANS");

        beginTransaction();

        VtiExitLdbTableRow row = table.newRow();
        row.setFieldValue("CARDNO", cardNo);

        try {
            table.saveRow(row);
        }
        catch (VtiExitException ee) {
            rollbackTransaction();
            logError("Error saving row for card number " + cardNo +
"."", ee);
            return 0;
        }
```

```
        commitTransaction();
        return 0;
    }
}
```

Identity Service Exits

Identity service exits are the means through which you can insert customized "glue" code into the identity management process. They allow for specifically tailored solutions to be devised that suits organizations with individual needs in relation to identity management.

An identity service exit runs within the environment of the SkyMobile Access Gateway, and manages the dialogue with the Secure Container during the process of identity checking.

Guidelines and conventions for coding identity service exits are described here.

You can find an example identity service exit here.

Configuration

Identity service exits are configured in the configuration file. Gateway ports and identity ports are associated with an identity service, that you can share between ports as required. You must configure each identity service to use a particular type of user validation, for example, SAP, Sky, or LDAP. When the type of identity service is set to "Custom", this indicates that you should invoke custom Java class to handle the necessary processing.

Some configuration file entries that demonstrate how you may configure an identity service exit are shown here.

Error Handling

The SkyMobile Access Gateway automatically traps any uncaught exceptions occurring inside an identity service exit, and writes them to the log file along with a stack trace. An error is also returned to the user before terminating the identity management process (that is, the user is logged out of the SkyMobile application once the error is acknowledged).

Example Configuration - Identity Service Exit

Example Configuration for Custom Identity Service "MYIDENTITYSERVICE"

```
SERVER.XMLGATEWAYPORT:PORT15094.PORT = 15094
SERVER.XMLGATEWAYPORT:PORT15094.DEFAULTIDENTITYSERVICE =
MYIDENTITYSERVICE
SERVER.XMLGATEWAYPORT:PORT15094.COMPRESSION = GZIP
SERVER.XMLGATEWAYPORT:PORT15094.USEHTTP = true
SERVER.XMLGATEWAYPORT:PORT15094.SECURE = true
SERVER.XMLGATEWAYPORT:PORT15094.KEYSTORE = .keystore
SERVER.XMLGATEWAYPORT:PORT15094.KEYSTOREPASSWORD = mykeystorepassword
SERVER.IDENTITYPORT:PORT15095.PORT = 15095
SERVER.IDENTITYPORT:PORT15095.DEFAULTIDENTITYSERVICE =
MYIDENTITYSERVICE
SERVER.IDENTITYPORT:PORT15095.COMPRESSION = GZIP
SERVER.IDENTITYPORT:PORT15095.USEHTTP = true
SERVER.IDENTITYPORT:PORT15095.SECURE = true
SERVER.IDENTITYPORT:PORT15095.KEYSTORE = .keystore
SERVER.IDENTITYPORT:PORT15095.KEYSTOREPASSWORD = mykeystorepassword
SERVER.IDENTITYSERVICE:MYIDENTITYSERVICE.TYPE = Custom
SERVER.IDENTITYSERVICE:MYIDENTITYSERVICE.EXITCLASSNAME =
com.mycompany.MyIdentityServiceExit
SERVER.IDENTITYSERVICE:MYIDENTITYSERVICE.LOGFAILURES = true
SERVER.IDENTITYSERVICE:MYIDENTITYSERVICE.LOGSUCCESSES = true
```

In this example, the identity service called "MYIDENTITYSERVICE" is configured to invoke by default for incoming identity requests received on ports 15094 (an XML gateway port) and 15095 (an identity management port). Because the identity service is specified as of type "Custom", the Java class "com.mycompany.MyIdentityServiceExit"is invoked to process any such incoming requests.

Identity Service Exit Coding Guidelines

Identity service exits must extend the VtiIdentityServiceExit class. They have a single point of entry, an abstract method called execute(), that is invoked whenever the SkyMobile Access Gateway detects an incoming identity management request from a Secure Container. The Secure Container's role in the identity management process is merely to render the fields sent to it by the identity service exit, and capture input from the user where applicable. The identity service exit manages everything else, and effectively carries responsibility for overall control of the identity management process.

The mechanism used for communication between the Secure Container and the identity service exit module is a collection of named parameters, known as identity parameters. Identity parameters are key-value type pairs that are exchanged between the identity service module and the Secure Container attempting to obtain authorization. The VtiIdentityServiceExit class contains a number of methods for accessing and manipulating identity parameters (getIdParam(), getIdParamAt(), getIdParamValue(), addIdParam(), removeIdParam()) These methods and more are all detailed in the Java exits API documentation that you can find in the `doc` directory, under the SkyMobile installation directory.

Controlling The Identity Management Process

The Secure Container initiates the identity management process by submitting an empty request (i.e. one that contains no identity parameters). The identity service exit is expected to respond in the first instance with a list of fields to present to the user. These are sent as identity parameters. Thereafter, the Secure Container and the identity service exit engage in a dialogue, passing the identity parameters backwards and forwards until either the authorization process is successful, or fails in such a way that the user cannot retry.

The execute() method of the VtiIdentityServiceExit class returns an instance of the VtiIdentityServiceExitResult class. The nature of the returned VtiIdentityServiceExitResult object ultimately instructs the Secure Container on how to proceed. A VtiIdentityServiceExitResult contains three important attributes that are:

- The return code;
- The error code; and

- The error message.

Internally, the identity management infrastructure transports these pieces of information as special internal identity parameters called (respectively) RESULT, ERRORCODE, and ERRORMESSAGE. You should avoid using any of these keys for your own purposes as their values are likely to be overwritten.

Probably the most important item here is the return code. You may set it to any of the following values:

OK - The authorization process is considered successful, and the Secure Container proceeds to run the user's first application.

RETRY - The user is re-prompted to enter any fields currently in the identity parameter set. Any error code and /or error message returned in the result is displayed.

ERROR - The authorization process is considered to have failed. Any error code and / or error message returned in the result is displayed. Once the error is acknowledged, the Secure Container exits (that is, the SkyMobile application shuts down).

Understanding the different ways in which you can construct a VtiIdentityServiceExitResult object and is returned is an essential step in comprehending how the identity service exit maintains control over the identity management process.

Identity Parameter Characteristics

Identity parameters passed between the identity service exit and the Secure container have a number of characteristics that are as follows.

Key - This is the "name" or "ID" of the identity parameter, and should be unique. Certain key names are reserved for internal use; these are described in the section below entitled Special Identity Parameters.

Label - This is the text label that appears on the screen when the user is prompted to enter a value for the identity parameter.

Value - This is the current value of the identity parameter that the identity service exit or upon entry of some input by the user can change.

Length - This is the maximum length of the identity parameter that is used as a guide when sizing text fields and as a hard limit when accepting input. If set to 0 or lower, no maximum length is enforced.

Target Attribute - This is a value that indicates where the Secure Container should store the identity parameter's value when identity management processing is completed. See the section below entitled Identity Parameter Target Attributes.

Flags - This is a set of binary (true / false) flags that switch various aspects of the identity parameters behavior on or off. See the section below entitled Identity Parameter Flags.

Special Identity Parameters

The following special identity parameters are reserved for internal use and have specialized functionality.

ERRORCODE - Used internally to store any error code sent to the Secure Container.

ERRORMESSAGE - Used internally to store any error message sent to the Secure Container.

FOOTERTEXT - The value (or if no value, the label) of this identity parameter is placed immediately below any input fields displayed to the user. It can be used to display legal notices and / or welcome messages.

OFFLINELEASEHOURS - This is the number of hours for which any login credentials, once cached, remain valid. If a user attempts to log in whilst offline, the values they enter are checked against any cached credentials and, if a match is found, the user is allowed to proceed. If this parameter is not set or set to a value of *0* or lower, then no time limit on use of cached credentials for offline logins is enforced.

RESULT - This parameter is used internally to control the process flow of the identity management processing, as described earlier.

SCREENTITLE - The value (or if no value, the label) of this identity parameter is used as the screen title when the user is prompted for input during the identity management process.

SUBMITBUTTON - The value (or if no value, the label) of this identity parameter is used as the label for the **Submit** button when the user is prompted for input during the identity management process.

Also refer to Identity Parameter Fetch Values, below.

Identity Parameter Target Attributes

When an identity parameter is assigned a target attribute, this indicates that it should have a lifetime that extends beyond the identity management process. Once the identity management process succeeds, the value from the parameter is passed into the desired location within the user session.

It makes little sense to have more than one identity parameter with the same target attribute, as you can store only one value ultimately in each target location. Avoid doing this as the results vary randomly depending on which identity parameter is processed first.

However, a single identity parameter can target more than one field. If you have a need to do this, simply add or use a bitwise "or" to combine the relevant values (for example, TARGET_ATTR_USER + TARGET_ATTR_PARAMETER3, or TARGET_ATTR_USER | TARGET_ATTR_ PARAMETER3).

The list of recognized target attribute values and their meanings is as follows:

TARGET_ATTR_NONE - The value for the parameter is not stored for future use.

TARGET_ATTR_PARAMETER1 - The value for the parameter is stored in user-defined parameter field #1.

TARGET_ATTR_PARAMETER2 - The value for the parameter is stored in user-defined parameter field #2.

TARGET_ATTR_PARAMETER3 - The value for the parameter is stored in user-defined parameter field #3.

TARGET_ATTR_PARAMETER4 - The value for the parameter is stored in user-defined parameter field #4.

TARGET_ATTR_PARAMETER5 - The value for the parameter is stored in user-defined parameter field #5.

TARGET_ATTR_PARAMETER6 - The value for the parameter is stored in user-defined parameter field #6.

TARGET_ATTR_PARAMETER7 - The value for the parameter is stored in user-defined parameter field #7.

TARGET_ATTR_PARAMETER8 - The value for the parameter is stored in user-defined parameter field #8.

TARGET_ATTR_LANGUAGE - The value for the parameter is stored as the user's language.

TARGET_ATTR_PASSWORD - The value for the parameter is stored as the user's password.

TARGET_ATTR_USER - The value for the parameter is stored as the user's username.

TARGET_ATTR_WORKAREA - The value for the parameter is stored as the user's work area.

TARGET_ATTR_WORKGROUP - The value for the parameter is stored as the user's work group.

Identity Parameter Flags

Identity parameter flags switch various behavioral characteristics on or off. The default state for all flags is OFF, so if you wish to turn on any of the following behavior for an identity parameter you need to set the flags appropriately.

CACHEKEY - When *true*, this flag indicates that the identity parameter should be cached for use in validating offline login attempts. It differs from the CACHEATTR flag in that it also denotes a field used to uniquely ascertain which user you are talking about. Examples of fields that may attract use of this flag may include the user name. In a Windows (LDAP / AD) environment, the attribute containing the Windows domain name may also form part of the cache key.

CACHEATTR - When *true*, this flag indicates that the identity parameter should be cached for use in validating offline login attempts. It differs from the CACHEKEY flag in that it denotes a field that does not play a part in identifying which user you are talking about. Examples of fields that may attract use of this flag includes a user's password or PIN.

MANDATORY - When true, this flag indicates that a value for the identity parameter is required. The Secure Container does not permit the user to attempt a login until they have entered a value for the field.

MASK - When *true*, this flag indicates that the identity parameter should be secured during input (masked with asterisks).

PROMPT - This flag indicates that the user should be prompted to enter / change the value of the identity parameter. If this flag is not set then the identity parameter is passed backwards and forwards as part of the identity parameter bundle, but it is not visible to the user.

FETCH - When *true*, this flag indicates that the identity parameter should be populated by the Secure Container without the user's intervention. You can use this flag to obtain information about the Secure Container and perform logic branching if desired. The key of the identity parameter determines the value sought as shown in the section below.

Identity Parameter Fetch Values

Identity parameters flagged with the FETCH flag is assigned a value by the Secure Container prior to user input. In order to retrieve the correct value, the key of the identity parameter must assume a specific value. Currently, the following values are supported.

DEVICE_IDENTIFIER - The Secure Container configures the parameter to the device unique identifier.

DEVICE_MAKE - The Secure Container configures the parameter to the device make (often the name of the manufacturer).

DEVICE_MODEL - The Secure Container configures the parameter to the device model.

DEVICE_OS_NAME - The Secure Container configures the parameter to the device operating system name.

DEVICE_OS_VERSION - The Secure Container configures the parameter to the device operating system version.

HOST_NAME - The Secure Container configures the parameter to the host name.

IP_ADDRESS - The Secure Container configures the parameter to the IP address.

SCREEN_HEIGHT - The Secure Container configures the parameter to the screen height.

SCREEN_WIDTH - The Secure Container configures the parameter to the screen width.

TENANT - The Secure Container configures the parameter to the tenant identifier.

> ***Note:*** If a set of identity parameters are sent to the Secure Container containing one or more FETCH flagged parameters, and no PROMPT flagged parameters, the Secure Container populates all FETCH parameters and sends back the result to the identity service exit, without any information displayed to the user.

Identity Service Exit Example

The following sample code implements an identity service exit that prompts the user for a username and password, then checks to see that they match a hard-coded value. In a real-world scenario, the validation process typically interfaces to another system for validation of the user credentials.

```
package com.mycompany;
import au.com.skytechnologies.vti.*;
public class MyIdentityServiceExit extends VtiIdentityServiceExit {
    public final static String HARDCODED_VALID_USER = "MYUSER";
    public final static String HARDCODED_VALID_PASSWORD =
"MYPASSWORD";
    public final static int MAXIMUM_ATTEMPTS = 3;

    public VtiIdentityServiceExitResult execute() throws
VtiExitException {
        // First time through, send back a response requesting user ID
and
        // password.
        if (getIdParamCount() == 0) {
            addIdParam(new VtiIdentityParameter("USER", "", "User ID",
                20, TARGET_ATTR_USER, CACHEKEY | MANDATORY | PROMPT));
            addIdParam(new VtiIdentityParameter("PASSWORD", "",
"Password",
                20, TARGET_ATTR_PASSWORD, CACHEATTR | MANDATORY | MASK
| PROMPT));
            addIdParam(new VtiIdentityParameter("ATTEMPTCOUNT", "0",
"",
                10, TARGET_ATTR_NONE, 0);
            return new VtiIdentityServiceExitResult(RETRY);
        }
        // If not the first time through, validate the incoming
parameter values.
        else {
            // Retrieve the user and password identity parameter
```

```
values.
            String user = getIdParamValue("USER");
            String password = getIdParamValue("PASSWORD");
            // Validate the user and password against our hard-coded
values. If all
            // is well then signal a success.
            if (HARDCODED_VALID_USER.equalsIgnoreCase(user)
            && HARDCODED_VALID_PASSWORD.equalsIgnoreCase(password))
                return new VtiIdentityServiceExitResult(OK);
            // Fetch the value of the attempt count parameter as an
integer.
            int attemptCount = 0;
            try {
                String attemptCountStr = getIdParamValue
("ATTEMPTCOUNT");
                attemptCount = Integer.parseInt(attemptCountStr);
            }
            catch (Exception e) { }
            // Increment the attempt count and update the
corresponding identity
            // parameter with the new value.
            ++attemptCount;
            setIdParam("ATTEMPTCOUNT", Integer.toString
(attemptCount));
            // If the number of attempts is less than our limit, let
the user retry.
            // Otherwise, boot them out.

            if (attemptCount < MAXIMUM_ATTEMPTS) {
                return new VtiIdentityServiceExitResult(RETRY,
                    "BAD_LOGON_CREDENTIALS",
                    "The logon credentials are not valid.");
            }
            else {
                return new VtiIdentityServiceExitResult(ERROR,
```

```
                    "TOO_MANY_ATTEMPTS",
                    "The logon credentials are not valid; maximum
attempts exceeded.");
            }
        }
    }
}
```

4.11.16 Utilities and Device Integration

4.11.16.1 File Polling

The Application Server has a built-in mechanism to automatically poll a directory for files and then invoke a program to process the contents. This feature is especially useful to intercept messages from other applications and / or devices to prompt SkyMobile to perform an action, for example, a request from another customer production line to deliver or manufacture materials.

4.11.16.2 Printer Interface

The Application Server has a built-in printer interface that supports any type of pre-defined template, for example, laser, and thermal label. It works by reading in the template, automatically substituting variables and then issuing the print to a designated printer. The prompt to print is associated with a screen function and a Java exit may perform a complex formatting.

4.11.16.3 Connectivity to Serial and Networked Devices

The SkyMobile Java SDK contains built-in functions to communicate with and control external devices, for example, scan card reader, weighbridge, PLC, cash-drawers, debit / credit scanners. These devices may be either serially or network connected. Using this feature, a Application Server may receive and / or send data to any number of connected devices, without the need for additional software / middleware.

4.12 Change Management and Implementation

4.12.1 About

The following sections document how to effectively manage the changing of Kony for SAP applications and their dependencies and their deployment out to remote Secure Containers. For more detailed information on how to install Kony for SAP onto the supported devices and platforms, refer to the installation section.

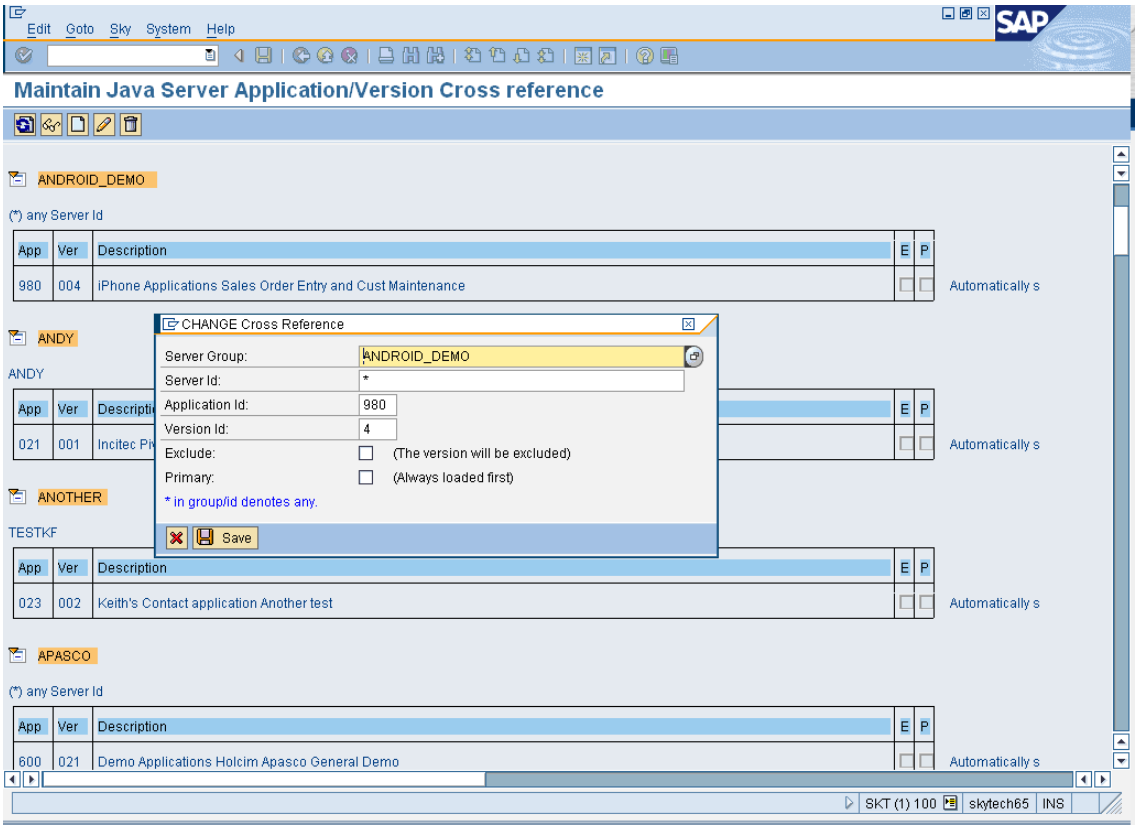
Key Topics

Tracking Versions
Locking Versions
Manual Synchronization
Export Utility
Packaging Utility

4.12.2 Manually Linking Applications to Containers (Servers)

> ***Important:*** When you use the server and user application profiling, this cross reference is maintained automatically for you when Secure Containers are provisioned. You should only manually maintain these entries if you are not using the SkyMobile provisioning process.

Before a Secure Container can use any application, you must link it to it through a cross reference entry. These entries may be generic, for example, any server in a specific group to avoid tedious re-entry. The container application cross reference option is selected from the main workbench **Utility** -> **Application Server** menu or through transaction code YVTX. It is used to specify / restrict the applications that you may download to a specific container. You can do this by specifying a cross reference of container groups and / or ids and application and versions. You may use a selection screen to restrict the cross reference display.



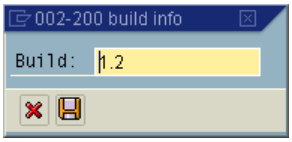
You may create and remove entries by positioning your cursor on the appropriate level and clicking the application toolbar option.

> ***Important:*** If you do not configure specific application versions, you cannot download any applications to the Secure Container.

4.12.3 Tracking Development Versions

When you develop SkyMobile applications and test them using a Application Server, it is sometimes hard to track whether a screen change is downloaded and that is the one you are testing against. To help track changes, you can configure a free format 'build' against the application version. You can update this build information at any time whilst maintaining a screen function, or directly from the workbench hierarchy screen. The build information appears against the version hierarchy node and in the screen painter as a hot link.

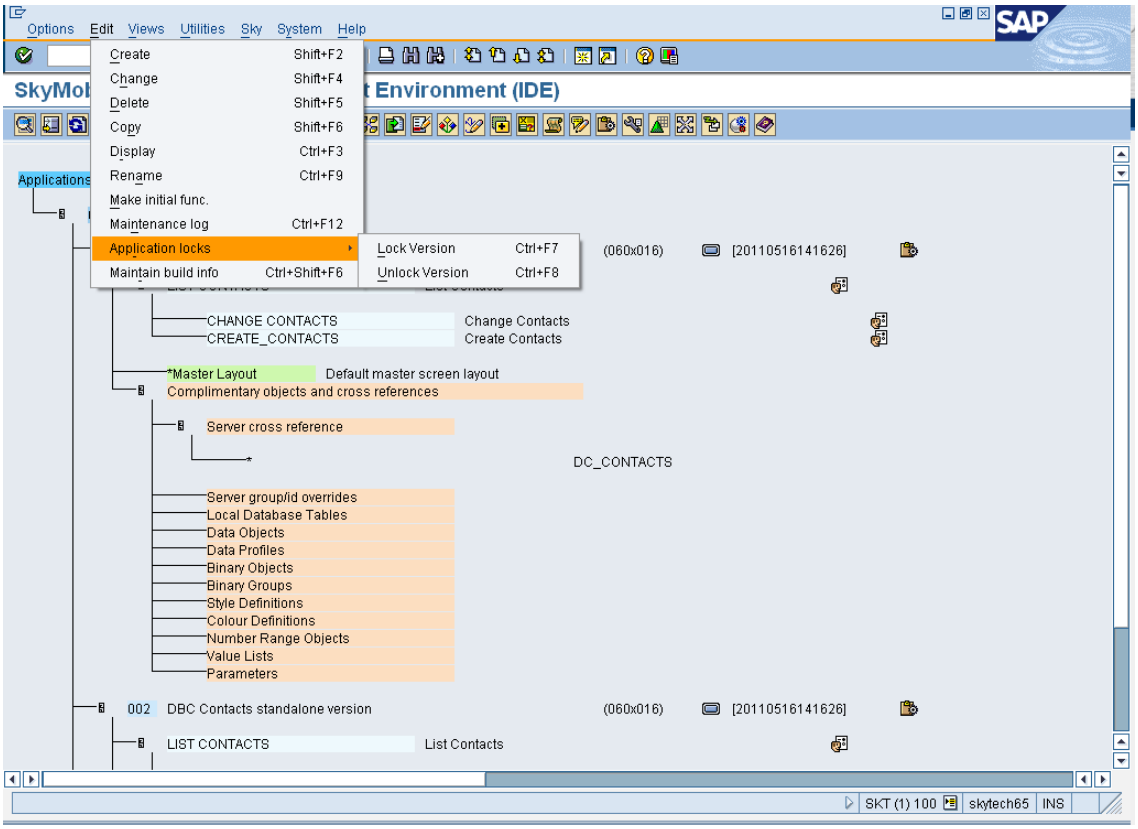
Launch the build maintenance screen using the icon and the following pop-up appears:



The build information is also transferred to Application Servers and appears on the web status page and in the 'about' information in the presentation clients (if supported).

4.12.4 Password Protection of Versions

You may apply a password or pin against an application version to protect it from unauthorised update. Once locked, you need to unlock it each time you wish to maintain it, and export it. You need to do this only once within the context of the workbench session. To lock and unlock a version, position your cursor on the application version node in the workbench hierarchy and select **Edit -> Application locks** menu option. Once locked, a 'padlock' icon appears on the right hand side of the hierarchy node as a hot link that indicates the current locked / unlocked status.



Once locked, unauthorised maintenance, copying and exporting / importing of the definition is prevented. You must unlock the version by clicking the padlock or using the menu option and then entering the PIN.

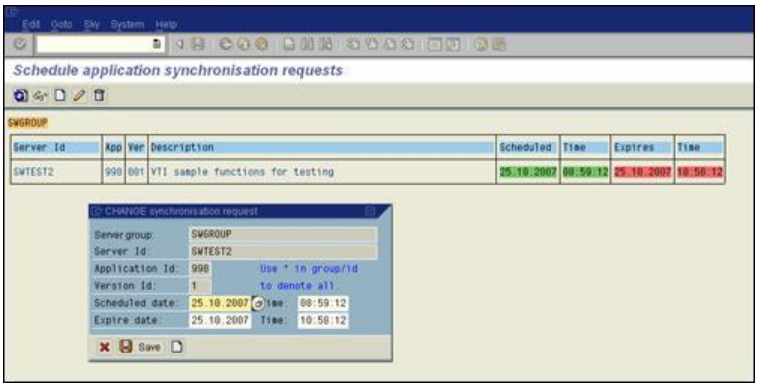
> ***Important:*** If you forget the PIN, a mechanism is provided for a administrator to reset the lock.

4.12.5 Manual Application Synchronization

You can select this option from the main workbench utility / Application Server menu or through transaction code YVETTE. It is used to schedule an application synchronisation with one or more Application Servers. You can do this by referencing a specific Application Server id. You may use a selection screen to restrict the cross reference display.

> ***Note:*** This functionality is only relevant if the application version is configured for manual synchronisation. By default, the connected Application Servers with synchronise automatically whenever an application version timestamp changes.

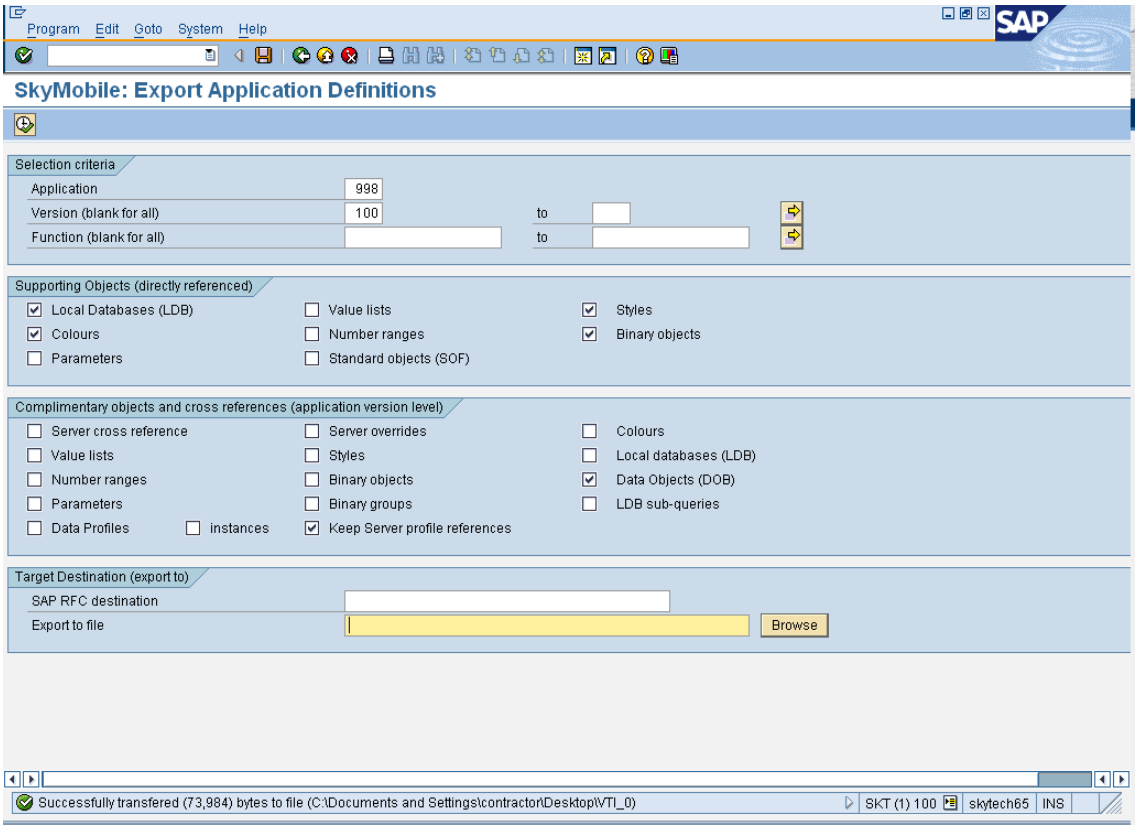
You may also trigger an application synchronisation from the Application Server web status page.



You basically schedule when the synchronisation is to occur (that is from / to date and time), and the next time the specified Application Server connects to SAP, it evaluates the schedule and performs the synchronisation. If the scheduled entry is still active, the timestamp is used to determine if the application changes until the expiry date / time is reached, after which the schedule is ignored.

4.12.6 Exporting Application Definitions

The export facility is available from the main MEAP workbench screen. Select the transport function from either the application toolbar (truck) or through the utilities menu. Alternatively you can run transaction YVTT. A selection screen appears. The lowest level for the export is an application version, or you may select a specific (or range of) functions.



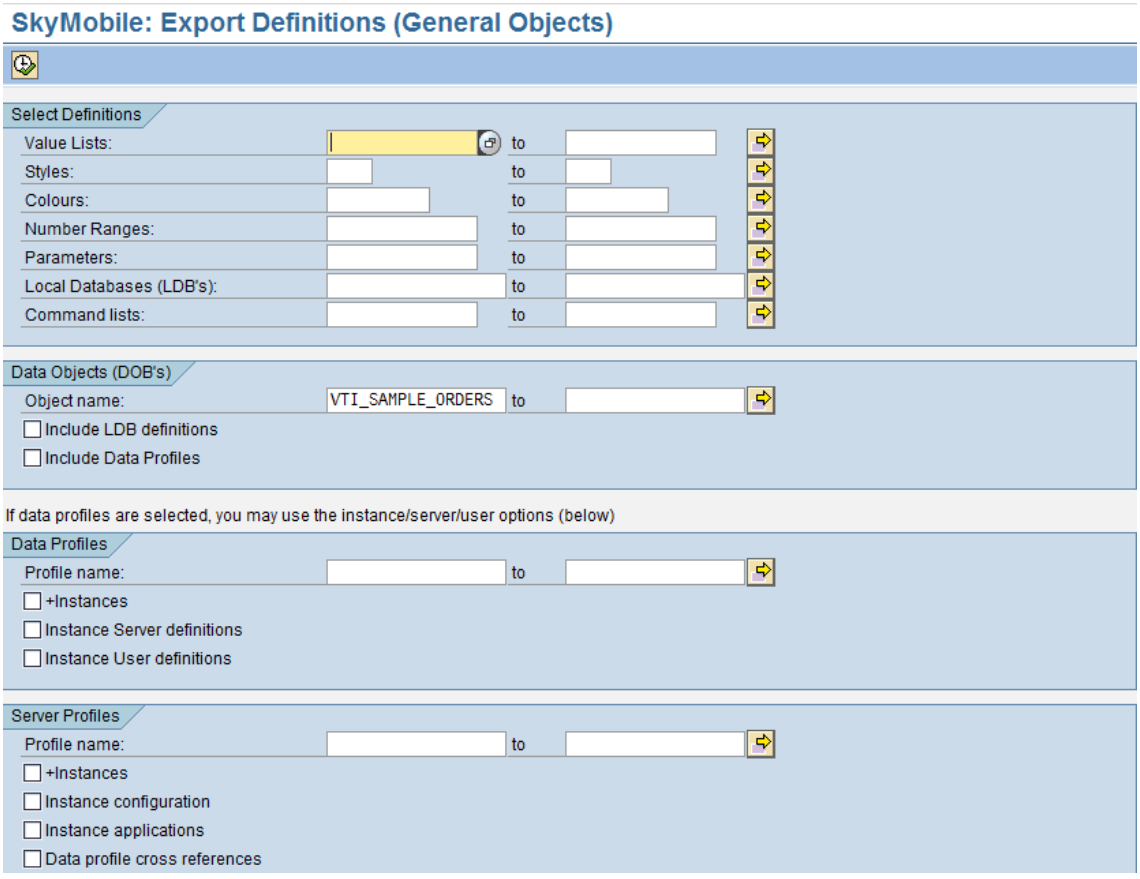
Field	Description
Selection criteria	The application version or specific functions to transport. You may leave version or function blank to select all.
Supporting Objects	Select additional objects directly associated with the screen functions selected.

Field	Description
Complimentary Objects and cross references	Transport the complimentary objects and cross reference entries associated with the selected application versions.
Target Destination	There are two modes; either through a direct RFC connection to another SAP system, or create an export file that you can import using the import function utility from the main workbench screen. The RFC transport happens immediately and requires a RFC destination to set up.

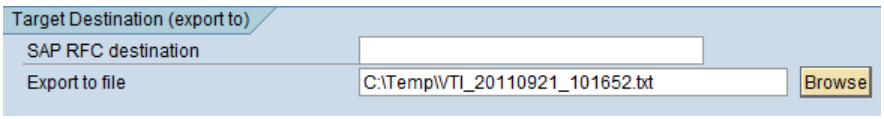
Note: You can export definitions to a text file and then import into remote MEAP system. You can perform the export to file as an option on the normal transport function. You may select the corresponding import function from the Utility menu on the main workbench screen. This option is useful to not only transport between two completely unrelated MEAP systems, but also to back up versions of an application version.

4.12.7 Generic Export Utility

Aside from SkyMobile applications, there are many other object types that you can individually change and require to migrate to another system. You can invoke this utility from many of the different workbench screens through the icon to provide the ability to export definitions to another MEAP server or out to a file. The selection screen is broken up into sections that you may specify both individually, or in groups (that is, multiple object types). It is too large to list here in its entirety, but the example below gives you the basic idea.

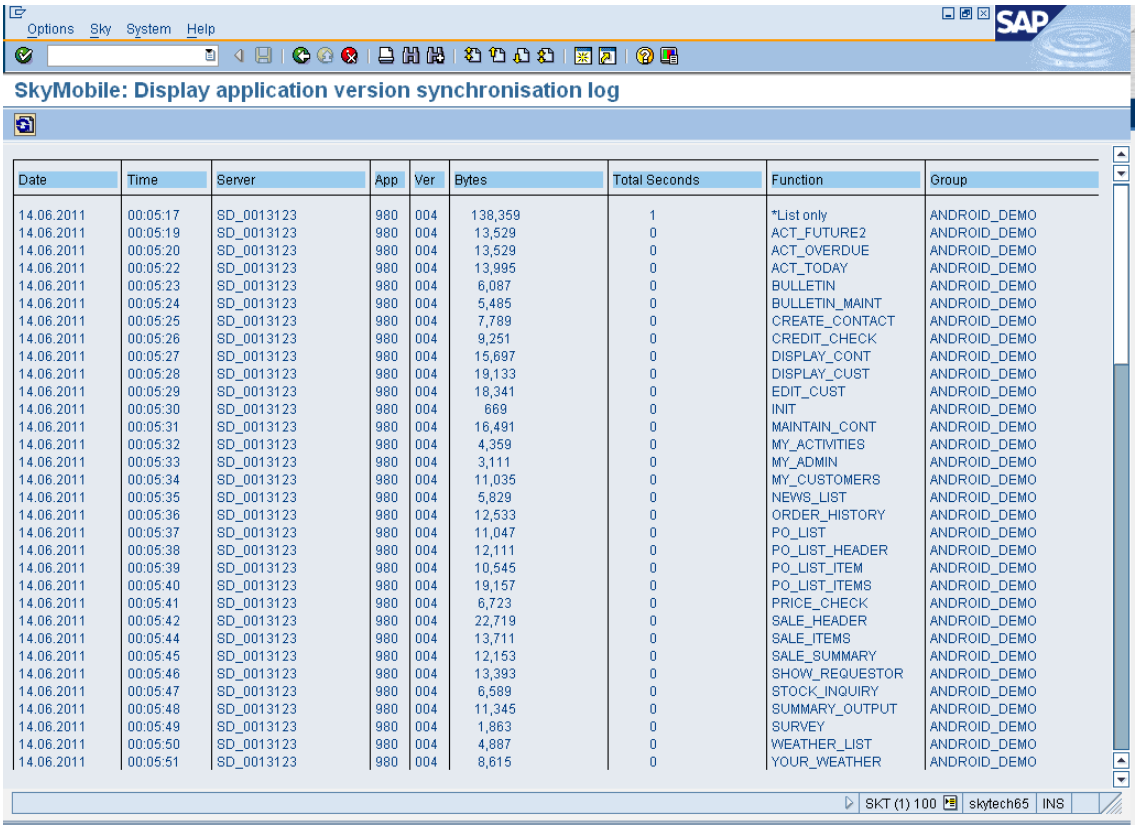


At the very bottom, you have the option to export / import directly to a remote MEAP Server, for example, development to system test; or you may export out to a file on your desktop and then use the import definitions option under the utilities menu in the appropriate workbench.



4.12.8 Tracking Application Downloads

This log lists all the application downloads that took place. You may use a selection screen to filter the data. You invoke the log from the workbench hierarchy by positioning your cursor on the application or version level and selecting the icon. The following screen appears:



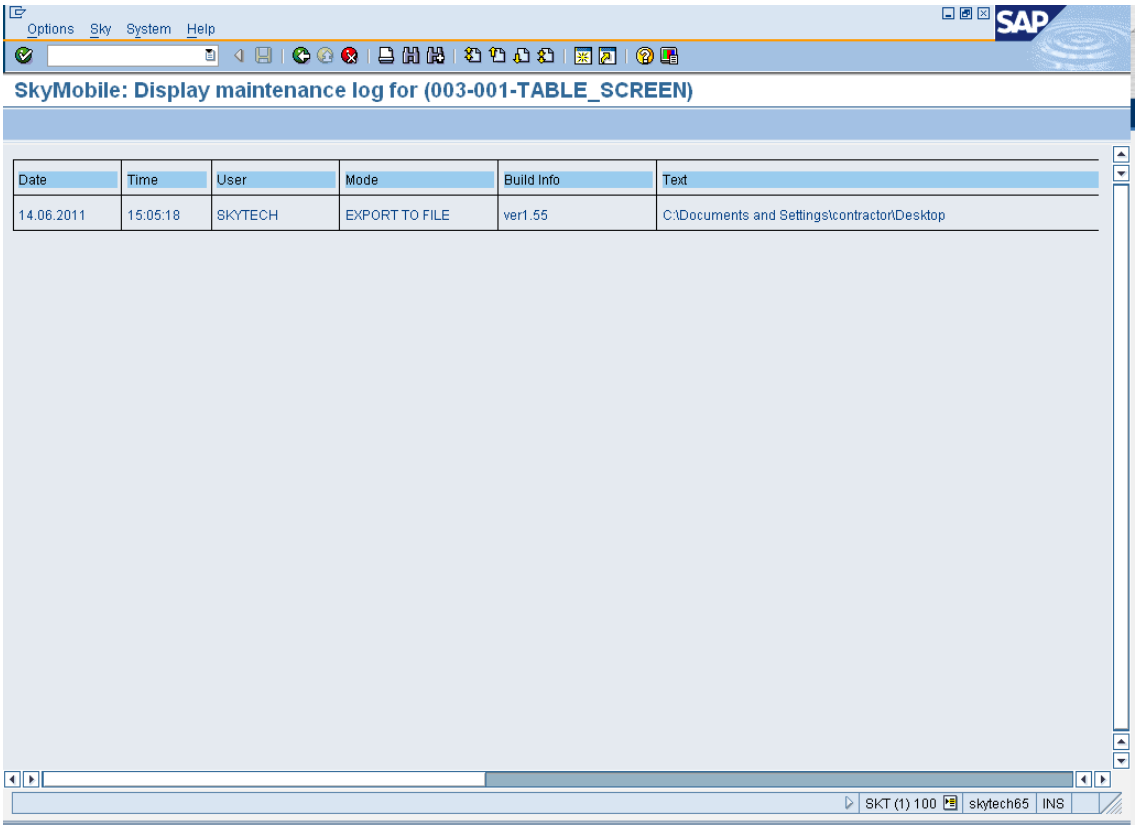
Date	Time	Server	App	Ver	Bytes	Total Seconds	Function	Group
14.06.2011	00:05:17	SD_0013123	980	004	138,359	1	*List only	ANDROID_DEMO
14.06.2011	00:05:19	SD_0013123	980	004	13,529	0	ACT_FUTURE2	ANDROID_DEMO
14.06.2011	00:05:20	SD_0013123	980	004	13,529	0	ACT_OVERDUE	ANDROID_DEMO
14.06.2011	00:05:22	SD_0013123	980	004	13,995	0	ACT_TODAY	ANDROID_DEMO
14.06.2011	00:05:23	SD_0013123	980	004	6,087	0	BULLETIN	ANDROID_DEMO
14.06.2011	00:05:24	SD_0013123	980	004	5,485	0	BULLETIN_MAINT	ANDROID_DEMO
14.06.2011	00:05:25	SD_0013123	980	004	7,789	0	CREATE_CONTACT	ANDROID_DEMO
14.06.2011	00:05:26	SD_0013123	980	004	9,251	0	CREDIT_CHECK	ANDROID_DEMO
14.06.2011	00:05:27	SD_0013123	980	004	15,697	0	DISPLAY_CONT	ANDROID_DEMO
14.06.2011	00:05:28	SD_0013123	980	004	19,133	0	DISPLAY_CUST	ANDROID_DEMO
14.06.2011	00:05:29	SD_0013123	980	004	18,341	0	EDIT_CUST	ANDROID_DEMO
14.06.2011	00:05:30	SD_0013123	980	004	669	0	INIT	ANDROID_DEMO
14.06.2011	00:05:31	SD_0013123	980	004	16,491	0	MAINTAIN_CONT	ANDROID_DEMO
14.06.2011	00:05:32	SD_0013123	980	004	4,359	0	MY_ACTIVITIES	ANDROID_DEMO
14.06.2011	00:05:33	SD_0013123	980	004	3,111	0	MY_ADMIN	ANDROID_DEMO
14.06.2011	00:05:34	SD_0013123	980	004	11,035	0	MY_CUSTOMERS	ANDROID_DEMO
14.06.2011	00:05:35	SD_0013123	980	004	5,829	0	NEWS_LIST	ANDROID_DEMO
14.06.2011	00:05:36	SD_0013123	980	004	12,533	0	ORDER_HISTORY	ANDROID_DEMO
14.06.2011	00:05:37	SD_0013123	980	004	11,047	0	PO_LIST	ANDROID_DEMO
14.06.2011	00:05:38	SD_0013123	980	004	12,111	0	PO_LIST_HEADER	ANDROID_DEMO
14.06.2011	00:05:39	SD_0013123	980	004	10,545	0	PO_LIST_ITEM	ANDROID_DEMO
14.06.2011	00:05:40	SD_0013123	980	004	19,157	0	PO_LIST_ITEMS	ANDROID_DEMO
14.06.2011	00:05:41	SD_0013123	980	004	6,723	0	PRICE_CHECK	ANDROID_DEMO
14.06.2011	00:05:42	SD_0013123	980	004	22,719	0	SALE_HEADER	ANDROID_DEMO
14.06.2011	00:05:44	SD_0013123	980	004	13,711	0	SALE_ITEMS	ANDROID_DEMO
14.06.2011	00:05:45	SD_0013123	980	004	12,153	0	SALE_SUMMARY	ANDROID_DEMO
14.06.2011	00:05:46	SD_0013123	980	004	13,393	0	SHOW_REQUESTOR	ANDROID_DEMO
14.06.2011	00:05:47	SD_0013123	980	004	6,589	0	STOCK_INQUIRY	ANDROID_DEMO
14.06.2011	00:05:48	SD_0013123	980	004	11,345	0	SUMMARY_OUTPUT	ANDROID_DEMO
14.06.2011	00:05:49	SD_0013123	980	004	1,863	0	SURVEY	ANDROID_DEMO
14.06.2011	00:05:50	SD_0013123	980	004	4,887	0	WEATHER_LIST	ANDROID_DEMO
14.06.2011	00:05:51	SD_0013123	980	004	8,615	0	YOUR_WEATHER	ANDROID_DEMO

Application downloads can occur either function by function or "*All", that is, entire application. This log is useful to see if and when a download takes place and approximately how large (uncompressed) the transfer is.

4.12.9 Screen Function Change Log

This log lists all the changes made to a screen function. It is useful to track when definitions changed and by whom. You may use a selection screen to filter the data. You invoke the log from the workbench hierarchy by positioning your cursor on a screen function level and selecting the icon.

The following screen appears:



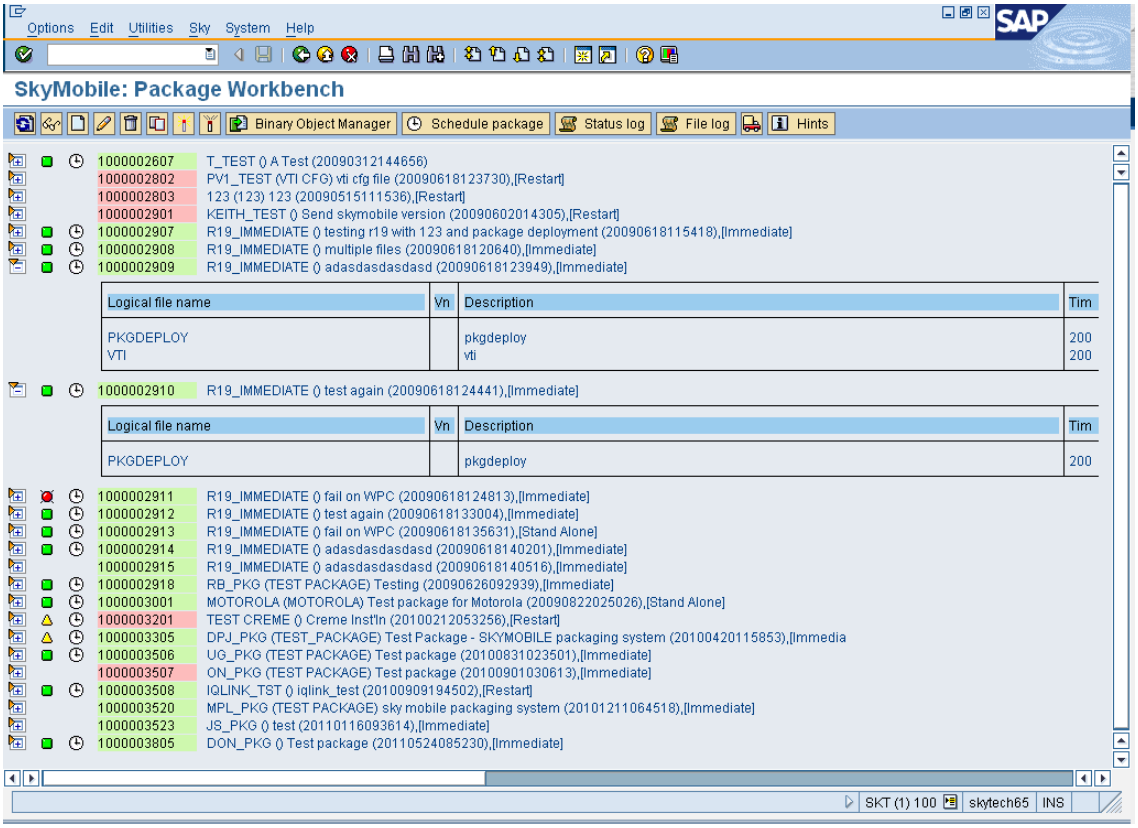
The log lists the mode of change, when it happened and by whom.

> ***Note:*** The log also shows if the function is copied, renamed, exported and / or transported.

4.12.10 Binary Object Packages

An extension of the binary object manager is a packaging utility that enables you to batch up one or more files into a package and schedule the package to download to one or more Secure Containers. In this way, you may easily apply system and application upgrades and fixes. The advantage of the packaging utility over the other transfer mechanisms is that it has a more controlled procedure to cut over to the new files, restart the Container, and back out changes. You may start the binary package workbench directly using transaction YVTP, from the workbench through the **Utilities** > **Application Server** > **Package Workbench** menu or from the binary file manager through the "packaging" application toolbar option. Each package definition is uniquely identified by a number that is automatically assigned when the package is created.

> ***Note:*** See also the remote software upgrades section in the management guide.



The main package workbench display lists all the existing package definitions, the files that are assigned and the current status. From the main display, you may create, change, display, delete, activate, deactivate and schedule packages.

4.12.10.1 Packaging Overview

The basic concept is to define a package containing all the files to update. You should have already uploaded (imported) the files centrally into the MEAP server using the binary file manager. Once you define the package with all the file updates, it is then activated and then scheduled to download to a list of Application Servers. While a package is active and has scheduled dependencies, you may not change. You must remove these first and then deactivate the package.

Each Secure Container periodically queries the MEAP server for any package updates. When a package is selected, it then requests the list of associated files from the MEAP. These files are then downloaded from the MEAP binary file manager onto the device. The files are stored in a temporary location on the device. If a reboot is required, the files are copied and the originals replaced when the Application Server is rebooted, otherwise the files are copied immediately. A special command file is created in the Containers primary directory that contains the copy instructions and a log is also written, describing the progress. Messages on the package processing are also written to the Containers system log.

4.12.10.2 Troubleshooting

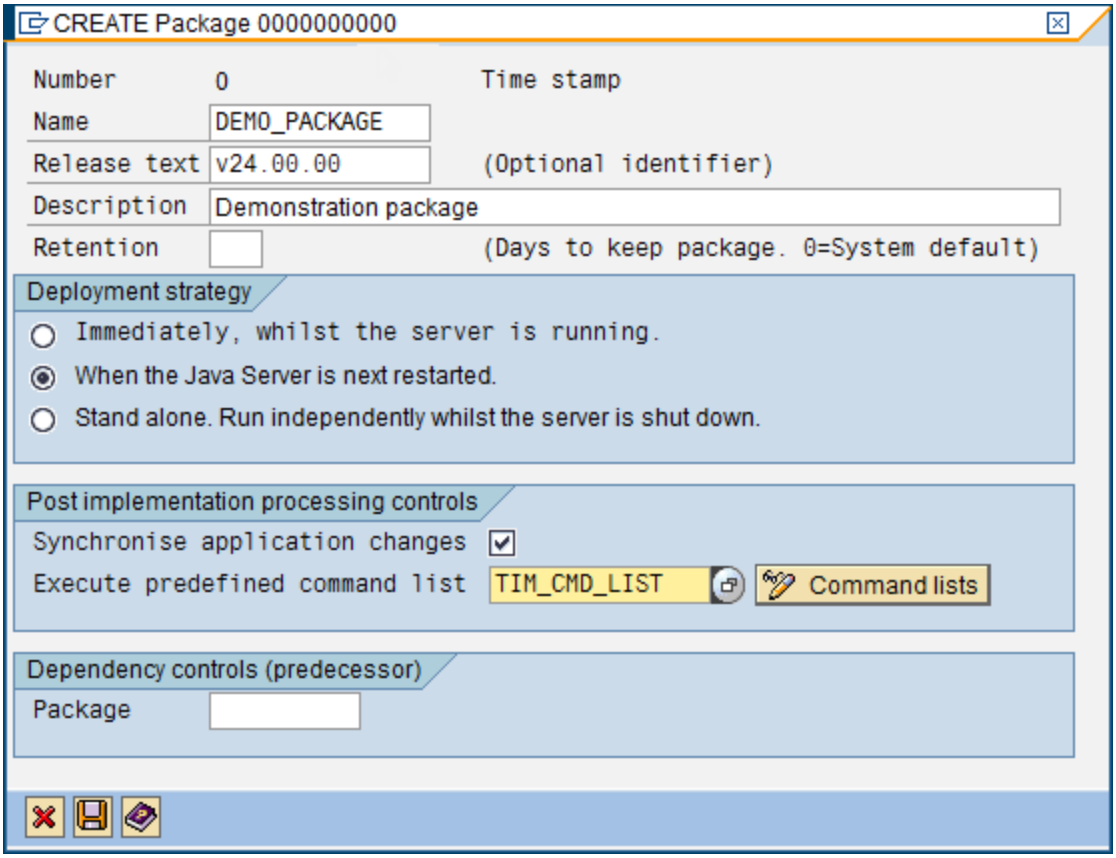
If you cannot copy a file, the package update is flagged as in error and the Container is not started. This is because you may start it in an unstable state, that is, the package is half implemented.

In this case, review the log message and determine the cause of the problem. In cases where you cannot rectify the problem or copy the file manually, you can edit the package command file in the VTI primary directory and make the necessary changes. In more extreme cases, you may remove the command file and stop the package schedule definition, held or deleted centrally in the MEAP Server.

4.12.10.3 Maintaining a Package

Maintaining the Package Header

Launch the package workbench (YVTP). If you create a new package, click the create icon in the menu bar, otherwise to maintain an existing package, double-click the package header. Define the package header by providing the package name, release text, description and delivery option.

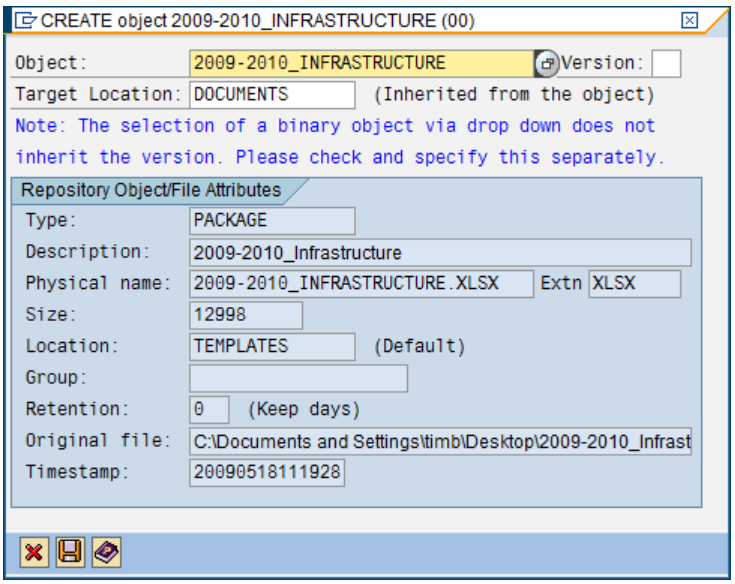


Field	Description
Name	Unique name to identify the package
Release text	Short description
Description	Long description

Field	Description
Retention	Number of days to retain the package definition
Deployment strategy	• Immediately (no restart required) • On restart • Stand alone
Synchronise application changes	Optional. Indicates an "appcheck" heartbeat command, need to issue after the package delivery
Execute command list	Optional. Provide a pre-defined heartbeat command list that is issued after the package delivery
Dependency control package	If this package is dependent on another package, provide the package number.

Maintaining the Package Contents

Once you add the package definition, you expand the definition, position your cursor in the file section and click create. You specify the logical name of the file definition in the binary object manager. The version number is used to control a specific version of the file (multiple versioning is supported). The target location usually defaults. This indicates the physical location that you are going to copy the file to. The drop-down lists the valid entries.



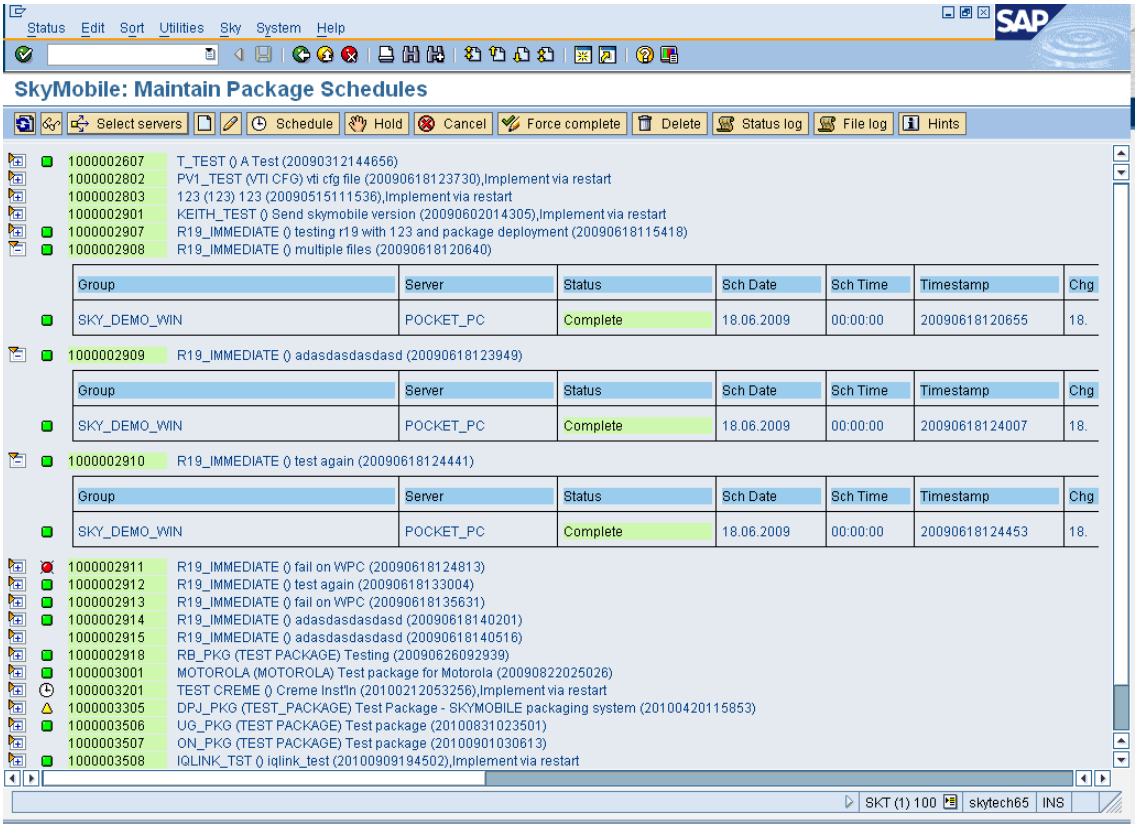
Activation

The final step before scheduling the package is to activate it. To activate the package, select it from the list and click the activate icon in the menu bar. Once a package is activated, the package number is shaded green rather than red.

> ***Important:*** You cannot modify an active or scheduled package. To change an active package (for example, to add an extra file) you must first deactivate the package.

4.12.10.4 Scheduling a Package

Once you define the package and its attributes, it is activated. This indicates that it is ready to schedule. Position your cursor on the package and click the schedule icon in the application toolbar. The package schedule maintenance screen is invoked. This utility is used to specify all the Application Servers that should receive the package update.

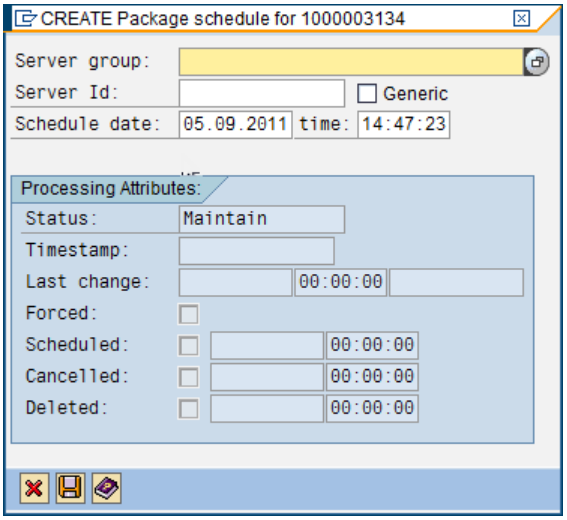


Package Scheduling Overview

The basic concept is that you define the schedule entries. These are listed with a status of "Maintain", meaning that you may change them. When you are ready, you position your cursor on the schedule entry and click the schedule icon in the application toolbar, the status then changes to "scheduled", meaning that next time that the Application Server connects, it detects the package, download it, schedule a reboot (if required) and finally once the package is implemented, update the status to complete.

Manually Maintaining Schedule Entries

You may create the schedule definition manually using the toolbar options. In this case, the following pop-up appears:

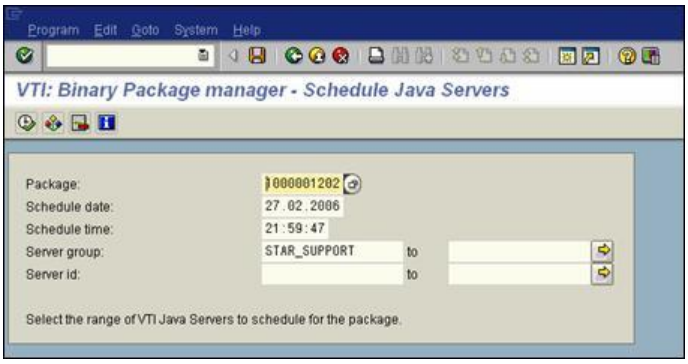


Field	Description
Server Group	Server group of the application server
Server ID	Server ID of the application server
Generic	Tick to make the package available to all server ids and groups
Schedule date	Date to make the package available from
Schedule time	Time of the specified date to make the package available

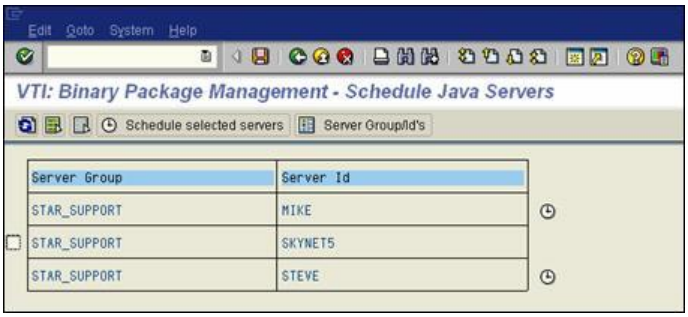
Important: You must have already registered the server itself with the MEAP server. If you did not, you cannot add the manual schedule entry.

Automatically Adding Schedule Entries

It can become very tedious to add large numbers of servers that receive a package update. To select multiple servers to add to the list, position your cursor in the list section for the package and click the **select servers** button in the application toolbar. The following selection screen appears. You can use this selection to restrict the list of available servers from which to select.



Specify the selection criteria and the following selection list appears:

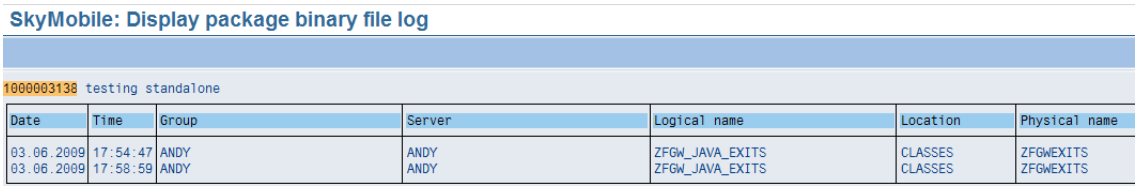


If the server is already in the schedule list, no selection check box appears; otherwise you select the servers to add to the schedule list.

4.12.10.5 The Package Binary Object Log

This report provides a audit trail of binary object downloads to Secure Containers. It is invoked from the main packaging and scheduling workbenches through the File log icon in the application tool bar.

The following list appears:



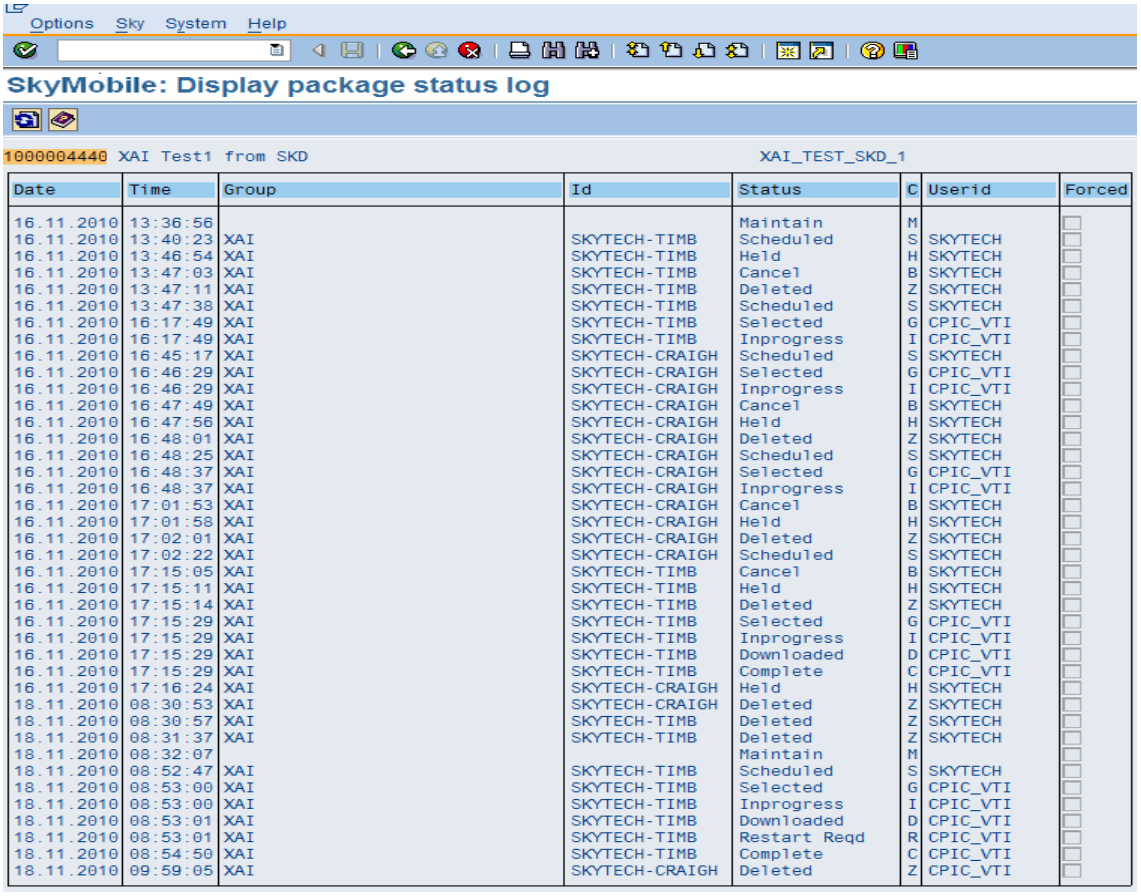
SkyMobile: Display package binary file log

1000003138 testing standalone

Date	Time	Group	Server	Logical name	Location	Physical name
03.06.2009	17:54:47	ANDY	ANDY	ZFGW_JAVA_EXITS	CLASSES	ZFGWEXITS
03.06.2009	17:58:59	ANDY	ANDY	ZFGW_JAVA_EXITS	CLASSES	ZFGWEXITS

4.12.10.6 The Package Status Log

The package manager records the status of a package for each server it is scheduled to deliver to. You can access the status log from either the package workbench, or package schedule workbench through the Status icon in the menu bar.



Date	Time	Group	Id	Status	C	Userid	Forced
16.11.2010	13:36:56			Maintain	M		
16.11.2010	13:40:23	XAI	SKYTECH-TIMB	Scheduled	S	SKYTECH	
16.11.2010	13:46:54	XAI	SKYTECH-TIMB	Held	H	SKYTECH	
16.11.2010	13:47:03	XAI	SKYTECH-TIMB	Cancel	B	SKYTECH	
16.11.2010	13:47:11	XAI	SKYTECH-TIMB	Deleted	Z	SKYTECH	
16.11.2010	13:47:38	XAI	SKYTECH-TIMB	Scheduled	S	SKYTECH	
16.11.2010	16:17:49	XAI	SKYTECH-TIMB	Selected	G	CPIC_VTI	
16.11.2010	16:17:49	XAI	SKYTECH-TIMB	Inprogress	I	CPIC_VTI	
16.11.2010	16:45:17	XAI	SKYTECH-CRAIGH	Scheduled	S	SKYTECH	
16.11.2010	16:46:29	XAI	SKYTECH-CRAIGH	Selected	G	CPIC_VTI	
16.11.2010	16:46:29	XAI	SKYTECH-CRAIGH	Inprogress	I	CPIC_VTI	
16.11.2010	16:47:49	XAI	SKYTECH-CRAIGH	Cancel	B	SKYTECH	
16.11.2010	16:47:56	XAI	SKYTECH-CRAIGH	Held	H	SKYTECH	
16.11.2010	16:48:01	XAI	SKYTECH-CRAIGH	Deleted	Z	SKYTECH	
16.11.2010	16:48:25	XAI	SKYTECH-CRAIGH	Scheduled	S	SKYTECH	
16.11.2010	16:48:37	XAI	SKYTECH-CRAIGH	Selected	G	CPIC_VTI	
16.11.2010	16:48:37	XAI	SKYTECH-CRAIGH	Inprogress	I	CPIC_VTI	
16.11.2010	17:01:53	XAI	SKYTECH-CRAIGH	Cancel	B	SKYTECH	
16.11.2010	17:01:58	XAI	SKYTECH-CRAIGH	Held	H	SKYTECH	
16.11.2010	17:02:01	XAI	SKYTECH-CRAIGH	Deleted	Z	SKYTECH	
16.11.2010	17:02:22	XAI	SKYTECH-CRAIGH	Scheduled	S	SKYTECH	
16.11.2010	17:15:05	XAI	SKYTECH-TIMB	Cancel	B	SKYTECH	
16.11.2010	17:15:11	XAI	SKYTECH-TIMB	Held	H	SKYTECH	
16.11.2010	17:15:14	XAI	SKYTECH-TIMB	Deleted	Z	SKYTECH	
16.11.2010	17:15:29	XAI	SKYTECH-TIMB	Selected	G	CPIC_VTI	
16.11.2010	17:15:29	XAI	SKYTECH-TIMB	Inprogress	I	CPIC_VTI	
16.11.2010	17:15:29	XAI	SKYTECH-TIMB	Downloaded	D	CPIC_VTI	
16.11.2010	17:15:29	XAI	SKYTECH-TIMB	Complete	C	CPIC_VTI	
16.11.2010	17:16:24	XAI	SKYTECH-CRAIGH	Held	H	SKYTECH	
18.11.2010	08:30:53	XAI	SKYTECH-CRAIGH	Deleted	Z	SKYTECH	
18.11.2010	08:30:57	XAI	SKYTECH-TIMB	Deleted	Z	SKYTECH	
18.11.2010	08:31:37	XAI	SKYTECH-TIMB	Deleted	Z	SKYTECH	
18.11.2010	08:32:07			Maintain	M		
18.11.2010	08:52:47	XAI	SKYTECH-TIMB	Scheduled	S	SKYTECH	
18.11.2010	08:53:00	XAI	SKYTECH-TIMB	Selected	G	CPIC_VTI	
18.11.2010	08:53:00	XAI	SKYTECH-TIMB	Inprogress	I	CPIC_VTI	
18.11.2010	08:53:01	XAI	SKYTECH-TIMB	Downloaded	D	CPIC_VTI	
18.11.2010	08:53:01	XAI	SKYTECH-TIMB	Restart Reqd	R	CPIC_VTI	
18.11.2010	08:54:50	XAI	SKYTECH-TIMB	Complete	C	CPIC_VTI	
18.11.2010	09:59:05	XAI	SKYTECH-CRAIGH	Deleted	Z	CPIC_VTI	

4.13 External Application Integration (XAI)

4.13.1 About

External application integration is all about how to:

- Invoke the Secure Container from third party applications
- Manage distributed back end SAP applications.

4.13.2 Key Topics

XAI Secure Container functionality
XAI distributed SAP applications

4.13.3 XAI Secure Container Functionality

This section describes the XAI architecture that enables bi-directional integration of external applications and programs with SkyMobile application screens, that is, the ability to launch and be launched from either a "native" application, third party application or a programming language. Some of this capability is possible through the use of function procedures and / or SOF (Standard Object Framework) program modules. For examples of how to use this functionality, click here.

For example:

- Launch and position on a SkyMobile screen with data from an email link.
- Launch the native calendar application from a SkyMobile screen
- Start SkyMobile from a C++ program, VBScript, and WSH.

The benefits and capabilities of XAI are extensive. Some common examples include:

- In workflow applications alerting users of items through an email that includes a direct link to the SkyMobile screen for actioning the workflow
- In field service / PM applications allowing users to use GIS tools to select assets trigger creation of fault and /or asset update requests through SkyMobile
- Use common office / productivity / collaboration tools (for example, MS Office, Notes, Share point) as entry points to SkyMobile functions

Sky plans to incorporate extensive XAI functionality to all its supported platforms (where possible) so that you can deploy 'feature rich' and productive applications easily to mobile users.

4.13.3.1 XAI Secure Container Architecture

XAI is a collection of components built for the Windows and Smart phone platforms (for example, Blackberry). The architecture of XAI is very simple and is shown in the diagram below. XAI is bi-directional , that is, it provides the mechanisms to both call external applications and be called by external applications.

Third party systems call XAI with a message. The message includes the SkyMobile function to call and optionally the field / value pairs to pass in. You can configure XAI to automatically launch the SkyMobile presentation client (for example, WPC, and BPC) if it is not running or bring an existing instance to the foreground.

You can make XAI calls to both SAP host (real-time) and local (stand alone) screens.

XAI Container Components - Windows Components

XAI Windows Components

XAI components are delivered with the SkyMobile installation. If you are using the 123 SkyMobile Installer, you must use the **Advanced** option when you install on Windows 32 / 64 to include and automatically configure XAI. The following sections describe the various components required for each platform.

The table below summarizes the various XAI components for Windows environments.

	Windows 32	Windows
GUI	WPC	WPC
DLL	Yes	Yes
COM Server	Yes - built into xai.dll	Not currently
Configuration File	Yes	Yes
Registry Entry	Yes	No

XAI Windows Configuration File

The XAI DLL for Windows32 and Windows Mobile platforms uses a configuration file `xai.ini` to control certain actions. The configuration file must be present in the same directory as the `xai.dll` file. By default, both the configuration and dll file get installed in the `VTI` directory.

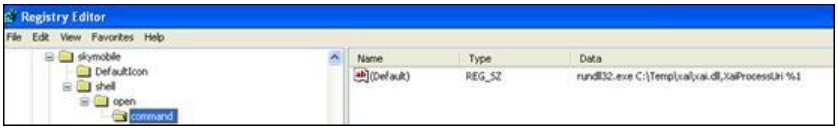
The table below details the configuration options present in the configuration file and their default values.

Option	Description	Default
WPCLaunchImage	The name of the WPC executable file.	wpc.exe
WPCLaunchDirectory	The path where the WPC executable file is located.	The directory that the `xai.dll` file is located in.
WPCLaunchAllowed	Whether XAI should launch WPC if it is not running.	True
WPCLaunchWait	If you have to start WPC, the delay in seconds after starting the gui before the message is passed in. This is important for slower platforms like Windows Mobile particularly when the WPC is configured to start the Java server.	2
DebugToScreen	Whether XAI should show write debug information to the screen as it processes a message.	False
FileTransferDeleteFile	If a message is sent to XAI through the file message format, whether XAI should delete the file after successful processing.	True

Windows Registry Entry

On the Windows32 platform, XAI uses a registry entry for managing the capture of URI messages (skymobile://function/?fiel1=value1). The registry key is installed automatically by the 123SkyMobile and basic installers.

The registry key is located in the hive `HKEY_CLASSES_ROOT\skymobile`.



The following is a sample registry file that inserts the key.

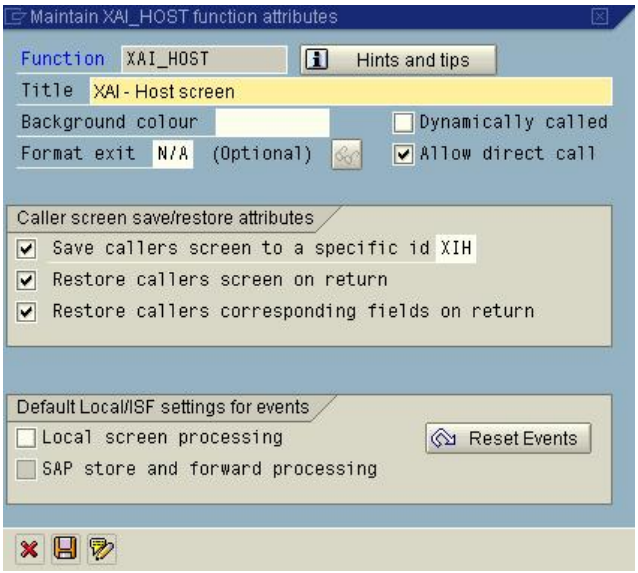
> ***Note:*** You need to customize the path of `xai.dll` to match your installation.

Windows Registry

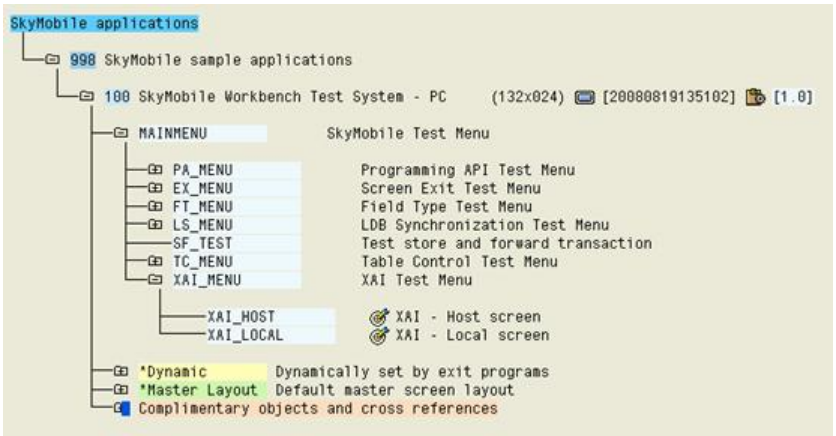
```
[HKEY_CLASSES_ROOT\skymobile]
@="URL:SkyMobile XAI Protocol"
"URL Protocol"=""

[HKEY_CLASSES_ROOT\skymobile\DefaultIcon]
@="shell32.dll,103"

[HKEY_CLASSES_ROOT\skymobile\shell]

[HKEY_CLASSES_ROOT\skymobile\shell\open]

[HKEY_CLASSES_ROOT\skymobile\shell\open\command]
@="rundll32.exe Unknown macro: 'highlight',XaiProcessUri %1"
```

4.13.3.2 XAI SAP Workbench Configuration

When an XAI call is made, WPC passes the request to the SkyMobile Java server. The Java server checks that the function name provided is valid and also that the function is configured to allow direct calls (see "allow direct call" in the screen shot below). A special internal event $GOTO is generated for the screen. It is this special event, and not the normal initial formatting controls, that is invoked when the screen function is first requested. You configure the function to allow direct calls in the screen function header attributes.



A "dart board" icon highlights the direct calls in the application hierarchy.

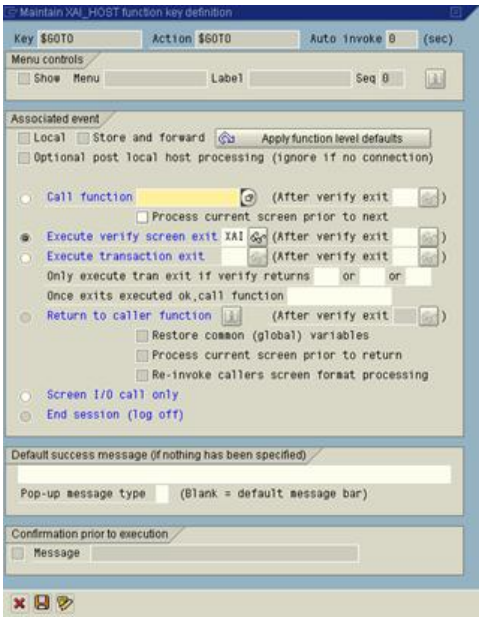


If a screen function allows direct calls, the screen map is loaded, field values provided by XAI are copied and the special internal $GOTO event is invoked. The $GOTO event allows for special LDB / Exit processing to be configured for direct calls.

> ***Note:*** Any normal format processing, LDB, and exits are not invoked for XAI direct calls.

After configuring the screen function to allow direct calls, you may configure the $GOTO event in the same way as any other screen event.

> ***Note:*** Certain attributes and options, such as return to caller and logoff are suppressed.



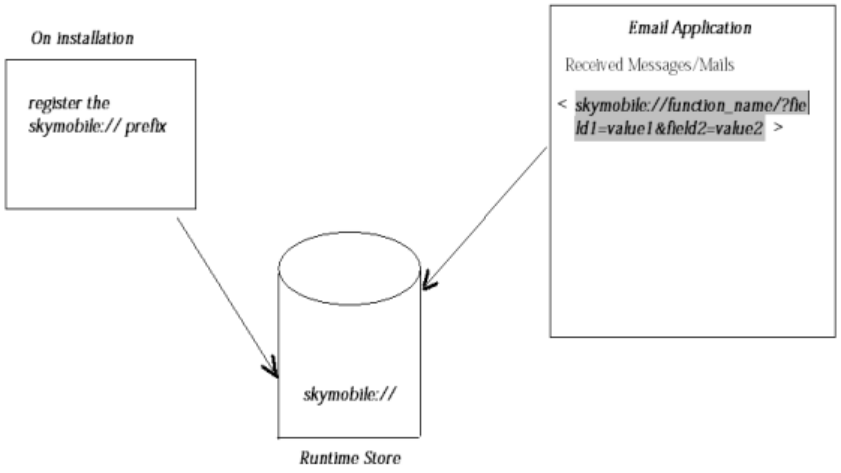
4.13.3.3 XAI Blackberry Components

Launching SkyMobile functions from email works differently from other native applications, for example, Contacts, and Calendar. Thus the components are described in separate sections.

Email Integration

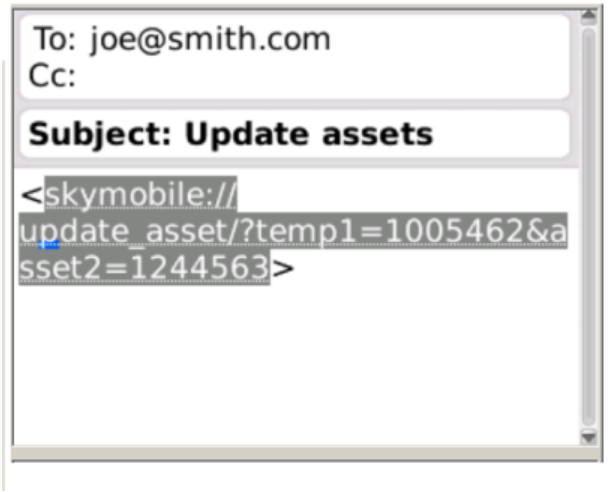
The SkyMobile application is able to recognize certain patterns in received emails that is of the form specified.

On installing the SkyMobile software to the BlackBerry device, the software registers the "skymobile://" pattern in a BlackBerry internal, conceptual store, called as a Runtime Store that is then automatically associated with the email application. When an email with this URI is received, the BlackBerry operating system scans the email message and provides a capability to highlight this URI and launch SkyMobile from within the email application. In cases where SkyMobile runs in the background, launching it from the email application effectively brings the SkyMobile to the foreground. Conceptually, this is how it is installed and configured to run on a BlackBerry.



The function that you specify in this URI has to be set up on the workbench as being able to receive direct calls, without which, SkyMobile gives an error. The parameters required by SkyMobile, such as the function name to go to, and the screen fields and values, are embedded in the URI, and is understood by SkyMobile.

Consider the example on a BlackBerry device. On viewing a mail with a URI format prefixed with the 'skymobile://' pattern in the BlackBerry email inbox, the BlackBerry operating system recognizes it and provides the capability to highlight the entire URI.

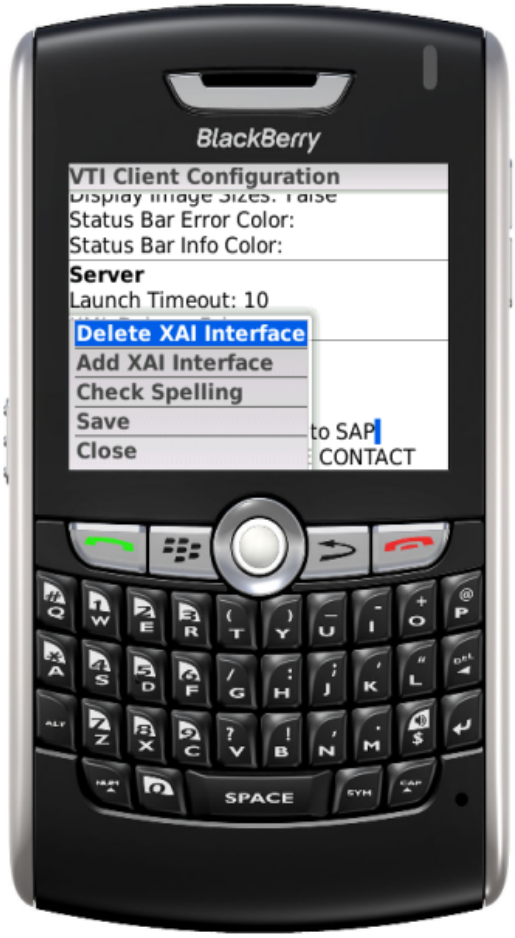


On selecting the menu button on the screen, you have an option to launch the SkyMobile application as shown.

Important: Ensure that the server and client configurations are set up to point to the correct application and version. SkyMobile then uses these configuration options to download the application, and position on the function name with the parameters embedded in the email URI.

Client Configuration Options

The XAI configuration option is provided when installing the SkyMobile software to your BlackBerry device. This option is provided as part of the Client Configuration, under the heading XAI. To dynamically add or delete sections, when viewing the client configuration on the BlackBerry, scroll down, and highlight either the XAI heading, or any section, if present. This screen should look like the following



In addition to this, you can dynamically add or delete menu items to applications. In other words, when you add new interfaces or delete from the client configuration screen, the changes take place

immediately, and you do not have to restart your device. You also have the capacity to specify multiple configuration sections, some even having the same "Application" option. This causes multiple menu items to appear for that native application that launches the SkyMobile application with that function name.

4.13.3.4 XAI Container Calls

All calls to XAI must pass in two pieces of the information:

- The message format being used
- The message data

XAI provides a variety of message formats:

- XML message
- Array
- URI
- File

The table below summarizes the methods supported currently on each platform.

Type	Windows 32	Windows Mobile	Blackberry
XML	Yes	Yes	No
Array	Yes	Yes	No
URI	Yes	Yes	Yes - emails only at present
File	Yes	Yes	No

XML Message - Type 0

XML messages to XAI are a straight string (no white space or CRLF formatting). The messages take the following format:

```
<XAI FUNCTION = "Function_Name"> <FIELD NAME="Field1" VALUE="Field1_
Value"/> <FIELD NAME="Field2" VALUE="Field2_Value"/></XAI>
```

Array Message - Type 1

Array messages pass in a two dimension array. The first field in the first array record must contain the required function. Subsequent array records contain the field and value.

For example, the array may look like the following:

Array Message

```
[FUNCTION_NAME,]
[FIELD1,VALUE1]
[FIELD2,VALUE2]
```

File Message - Type 2

File messages pass in the path to a file containing the required function and field values. The first line in the file must contain the name of the screen function. Subsequent lines contain the field and values to pass in to the function.

File Message

```
FUNCTION_NAME
FIELD1=VALUE1
FIELD2=VALUE2
```

URI Message - Type 3

Uniform Resource Identifier (URI) messages are a straight, case insensitive string. If the field values include white spaces, then you need to wrap the string in quotes. The URI takes the following format:

URI Message

```
skymobile://function_name/?field1=value1&field2=value2
```

4.13.3.5 XAI Secure Container Examples

The following section provides a number of XAI examples:

Launching Kony for SAP from Visual Basic

Launching Kony for SAP from Windows Scripting

Launching Kony for SAP from ESRI ArcPad

Launching Kony for SAP from BlackBerry

Blackberry Contacts Example

The purpose of this object is to provide the ability to launch a SkyMobile application from the native contacts application (for example, BlackBerry Contacts). In this way, you can seamlessly integrate SkyMobile functions into device applications.

BlackBerry Example

The XAI client configuration is specified to indicate what SkyMobile function to invoke and describe the option that is to appear on the contacts menu. Refer to BlackBerry XAI Client Configuration options for more information.

Once you are positioned on a BlackBerry contact, the SkyMobile menu option automatically appears.

> ***Note:*** You may specify whatever text you want.

When you select the option, the SkyMobile XAI monitor generates internally a SkyMobile URI:

skymobile://CREATE CONTACT/?name="Joe Smith"&address="6/16 Test Street Google Home California USA"&email_address=joe.smith@google.com&url=google.com&phone_number=95586211&mobile_number=048875621&company=Google

The screen function is then launched with the field values.

Windows Scripting Example

Windows Script Host (WSH) and VBScript rely upon the xai.dll COM server component. When SkyMobile is installed using either the 123SkyMobile or standard NSIS installer, the xai.dll COM server registers automatically. See the previous section on XAI components for details on how to manually register the xai.dll COM server.

This example is expecting that a properly formatted XAI URI is provided. For details on the XAI URI format, see the previous sections.

```
''''''''''''''''''''''''''''''''''''''''
'
' XAI VBScript Example
'
''''''''''''''''''''''''''''''''''''''''

Option Explicit

'Setup variables
Dim xaiObj
Dim msgType, msg

'Check that one & only one argument provided
If WScript.Arguments.Count <> 1 then
    MsgBox("Must have only one argument provided.")
    WScript.Quit 999
End If

'Setup call for XAI
set xaiObj = WScript.CreateObject("XAI.skyxai")
msgType = 3
msg = WScript.Arguments(0)

'Call XAI
xaiObj.CALLXAI msgType, msg
```

Visual Basic Example

This example uses the Visual Basic (VB) capabilities present in Microsoft Excel. Note that VB is available in all MS Office components. The example pulls data from a number of cells and then formats the data into the XAI XML format. For details on the XAI XML format see the previous sections.

> ***Note:*** Unlike VBScript, Visual Basic can access the `xai.dll` directly with the COM server being registered.

```
''''''''''''''''''''''''''''''''''''''''''
'
' XAI MS Excel VB Example
'
''''''''''''''''''''''''''''''''''''''''''

Private Declare Function XAILaunchWPC Lib "xai.dll" (param1 As
Variant, param2 As Variant) As Integer

Private Sub XmlMsg_Click()
   Dim funcName
   Dim fields As Variant
   Dim values As Variant
   Dim msgType As Variant
   Dim msg As Variant
   Dim ret As Integer

   'Get SkyMobile Function
   funcName = getFuncName()
   If funcName = "" Then
      Exit Sub
   End If
      \\
   'Get XAI field values
   fields = getFields()
```

```
   values = getValues()
          \\
   \\
   ' Build XML message
   msg = "<XAI FUNCTION = " & Chr$(34) & funcName & Chr$(34) & ">"
   For i = LBound(fields) To UBound(fields)
       msgTag = "<FIELD NAME=" & Chr$(34) & fields\(i) & Chr$(34)
       msgTag = msgTag & " VALUE=" & Chr$(34) & values\(i) + Chr$(34)
& "/>"
       msg = msg + msgTag
   Next
   msg = msg & "</XAI>"
   \\
   \\
   ' Call XAI
   msgType = 0
   ret = XAILaunchWPC(msgType, msg)
End Sub

Private Function getFuncName() As String
   'Get SkyMobile Function
   Range("B4").Select
   If ActiveCell.Value = "" Then
      MsgBox ("Function name not provided")
   End If
   getFuncName = ActiveCell.Value
End Function

Private Function getFields() As Variant
   Dim fields(3) As String
   \\
   'Get XAI Fields & Values
   Range("C7:C10").Select
   counter = 0
   num = Selection.Rows.count
```

```
   ActiveCell.Offset(0, 0).Select
   For i = 1 To num
       If ActiveCell.Value <> "" Then
          fields(counter) = ActiveCell.Value
          counter = counter + 1
       End If
       ActiveCell.Offset(1, 0).Select
   Next i
   \\
   getFields = fields
End Function

Private Function getValues() As Variant
   Dim values(3) As String

   'Get XAI Fields & Values
   Range("C7:C10").Select
   counter = 0
   num = Selection.Rows.count
   ActiveCell.Offset(0, 0).Select
   For i = 1 To num
       If ActiveCell.Value <> "" Then
          values(counter) = ActiveCell.Offset(0, 1).Value
          counter = counter + 1
       End If
       ActiveCell.Offset(1, 0).Select
   Next i
   \\
   getValues = values
End Function
```

ESRI ArcPad Example

This example uses the ArcPad GIS tool to capture GPS coordinates and passes them in to XAI in a field called "FIELD1". ArcPad uses a customised VBScript method to access XAI.

You require the following three components to achieve this:

- ArcPad applet file
- ArcPad VBScript file
- An icon file
- XAI dll and configuration file

The steps are:

1. Check that the configuration option **WPCLaunchDirectory** in the `XAI ini` file points to the correct directory.
2. Copy the XAI dll and ini file into the `ArcPad extensions` subdirectory.
3. Copy the ArcPad applet, VBScript file and icon file into the `ArcPad Applets` subdirectory.
4. Restart ArcPad if it is already running.

ArcPad Applet File - XAI.apa

The applet file must have the extension `.apa`. You need to modify the section highlighted in yellow bellow to match the name of your icon file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ArcPad>
   <APPLET>
 <TOOLBARS>
    <TOOLBAR name="tblXAI" caption="XAI" visible="true" image=""
buttonsize="1">
  <TOOLBUTTON onpointerup="XAI Message:" name="btn1" shortcut=""
```

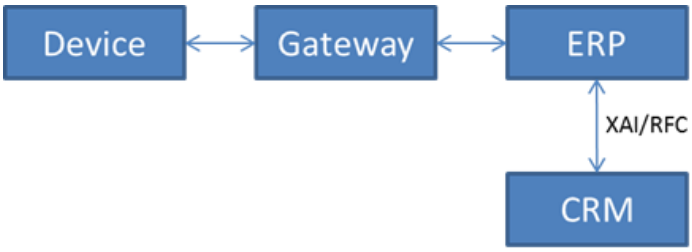
```
image="*{_}iconA.ico{_}*"/>
    </TOOLBAR>
 </TOOLBARS>
 <FORMS/>
 <SYSTEMOBJECTS/>
 </APPLET>
 <SCRIPT src="XAI.vbs" language="vbscript"/>
</ArcPad>
```

ArcPad VBScript File - XAI.vbs

```
Sub XmlMsg

  Dim objClickPt, strXaiMsg, strStartTag, strEndTag
  Dim E, T
  Set objClickPt = Application.CreateAppObject ("point")
  objClickPt.X = Map.PointerX
  objClickPt.Y = Map.PointerY
  MsgBox CStr(objClickPt.X)&","&CStr(objClickPt.Y)
  strStartTag = "<XAI FUNCTION=""FUNCTION_NAME"">"
  strEndTag = "</XAI>"

  strXaiMsg = strStartTag + "<FIELD NAME=""FIELD1"" VALUE=""" + CStr
(objClickPt.X)  + "/" + CStr(objClickPt.y)  +"""/>" + "<FIELD
NAME=""FIELD2"" VALUE=""VALUE2" +"""/>"  + strEndTag

  'MsgBox strXaiMsg

  Set E = Extensions("SKY")
  T = E.Escape(0,strXaiMsg)

End Sub
```

4.13.4 XAI Distributed SAP Applications

This functionality allows you to effectively integrate mobile applications that are run on different SAP environments through a central "master" MEAP system. As an example, you can use a SAP ERP system as the master and route requests for the CRM application, data objects to the SAP CRM system as below:



Note: The SkyMobile Gateway is connected to the SAP ERP system as the “master” SkyMobile system and any requests for the SkyMobile CRM application and related data objects are automatically routed to the SAP CRM system; the CRM application definition does not have to reside in ERP. Any requests to SkyMobile ERP applications are performed locally in ERP.

To use this functionality, perform the following number of configurations:

1. Select which SAP environment is the master system
2. Create an RFC destination in the master system for the target system(s)
3. Configure RFC destination security
4. Configure an XAI destination
5. Enable XAI

Note: To use XAI, you must install the SkyMobile MEAP server add-in in every participating SAP environment.

4.13.4.1 Selecting a SAP Master System

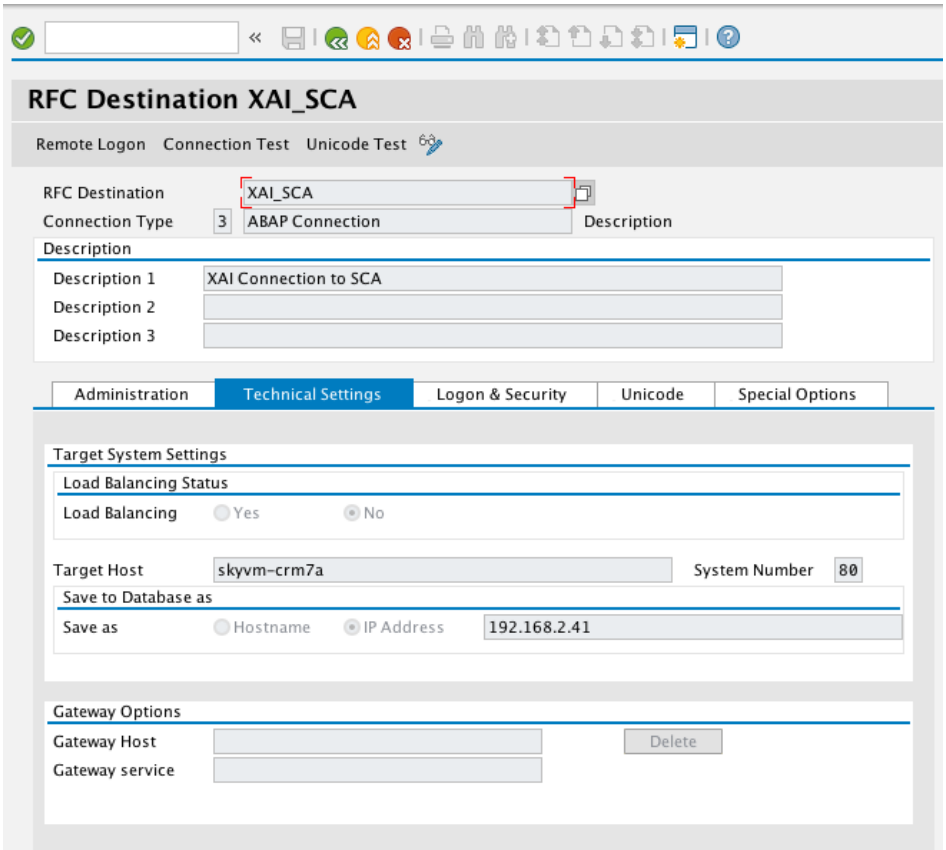
The first thing you must decide is which SAP environment is going to act as the “master” system; the central SkyMobile MEAP that is going to receive all SkyMobile requests and then re-route them to the appropriate SAP system. The master system is often the busiest SkyMobile system in respect to users and volume. This can be an existing SAP environment, for example, ERP or CRM, or a totally stand-alone SAP NetWeaver instance that acts as a central broker. This is an architectural decision that is Customer specific. In all cases, you must install a SkyMobile SAP MEAP in every SAP environment where you wish to use SkyMobile applications and functionality.

> ***Note:*** The SkyMobile gateway(s) point to the allocated master system. The re-routing is performed within SAP.

Once you decide which SAP system is the “master”, you are now ready to configure XAI.

4.13.4.2 Creating RFC Destinations

You must create RFC destinations in the master SAP system for each of the remote SAP systems that you wish to route SkyMobile requests to. Your SAP Basis team usually does this by using transaction SM59.



4.13.4.3 Configuring RFC Destination Security

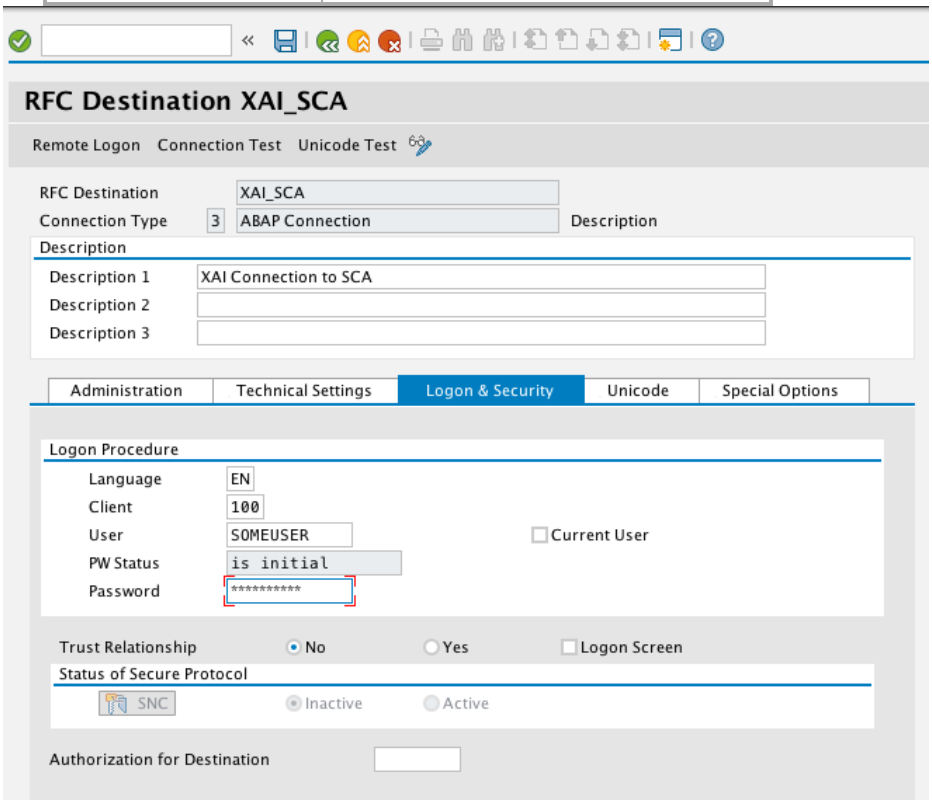
The RFC destination needs to have a logon configured to allow the master system to access the target system. This configuration is made on the **Logon & Security** tab of the RFC destination. There are two options available:

- Static logon
- Trusted logon

Static Logon

This is the simplest option, as it only requires a logon for the target system. The details required are shown in the table below:

Field	Value Required
Language	Logon language of the SAP User
Client	SAP client in the target system
User	SAP User ID in the target system
Password	Password for the SAP User ID
Current User	Unchecked
Trust Relationship	No



Trusted Logon

In the static logon option mentioned above, a user session in the master system connects to the target system as a different user. For example, the Sky gateway logs on to the master SAP system using a Service / Communications user and then jumps across to the target SAP system using the static user configured on the RFC destination.

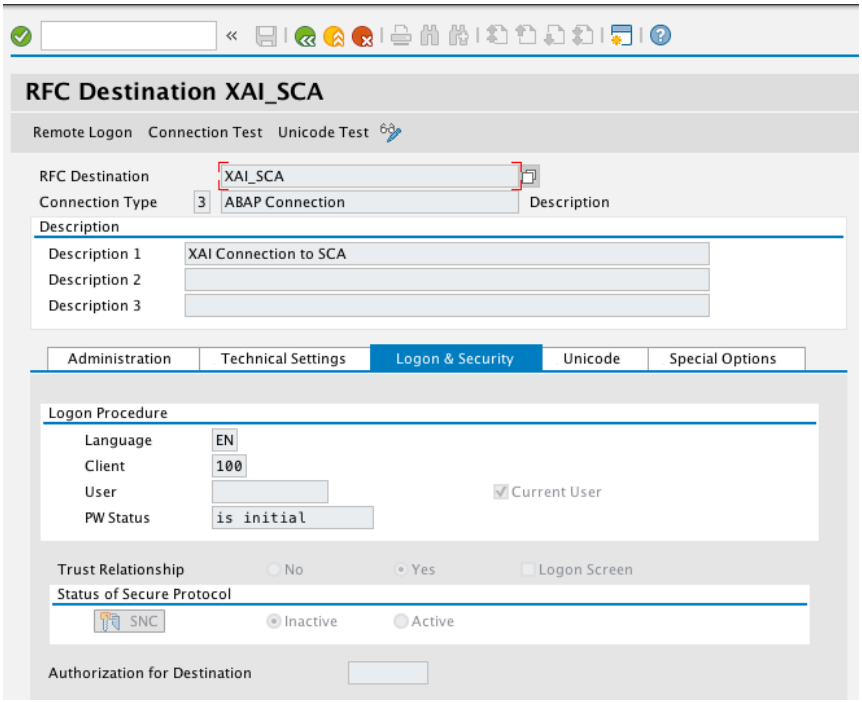
This may not be appropriate when your mobile infrastructure uses SAP SSO Identity Management. In these scenarios, the users on the mobile devices logg on to the SAP system directly for posting of transactions – not the Service / Communication user of the Sky Gateway. In this scenario, you most likely need the user session in the master SAP system to extend to the target SAP system. This scenario requires the use of a trusted logon.

This is a slightly more complex configuration, as it requires:

- Configuration of the trust option on the master SAP system RFC destination
- Configuration of the trust relationship in the target SAP system
- Assignment of specific authorization to logons in the target SAP system

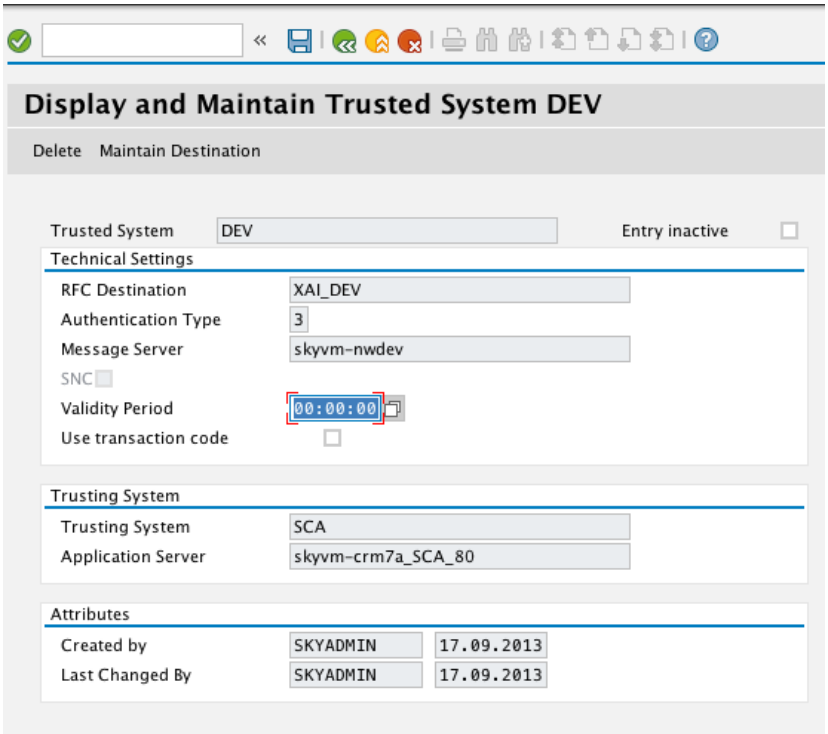
Firstly, to configure the trust option in the master system, select the **Logon & Security** tab on the RFC destination, and provide the following details:

Field	Value Required
Language	Logon language of the SAP User
Client	SAP client in the target system
User	Leave blank
Password	Leave blank
Current User	Checked
Trust Relationship	Yes

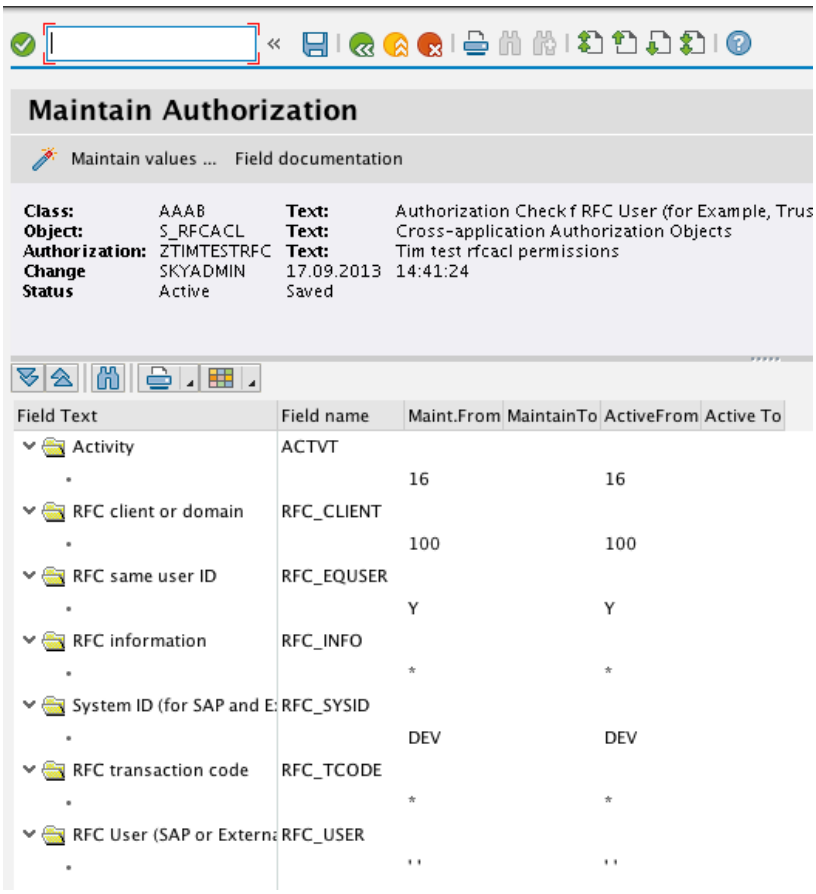


Secondly, you need to create a trust relationship entry in the target system for the master system. This involves:

- Creating an RFC destination in the target system for the master SAP system (similar to what is done earlier)
- Create the trusted system entry based on this RFC destination



Finally, users who access the target SAP system from the master SAP system through the trusted connection need to have a specific authorization in the target system. This authorization object is called S_RFCACL. The following is a screenshot that shows an example profile with this authorization:



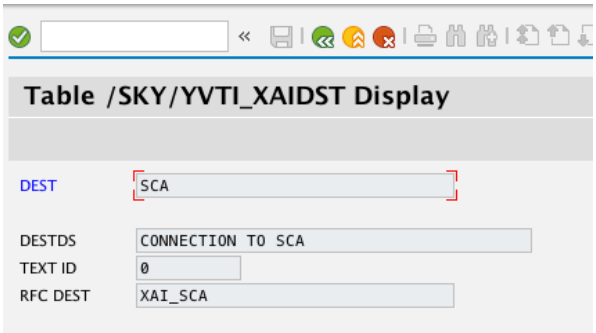
> ***Note:*** All users accessing the target SAP system through the trusted connection - including the Service / Communication user, require this authorization.

4.13.4.4 Configuring XAI Destinations

Once you configure your SAP RFC destinations, you need to create an XAI cross reference in table /SKY/YVTI_XAIDST. There is no standard maintenance utility at this stage, so you need to use SM16 (or equivalent) to create an XAI destination for each of your RFC destinations. For example:

Field	Value Required
DEST	Destination name (for example, Target SAP System SID)
DESTDS	Description

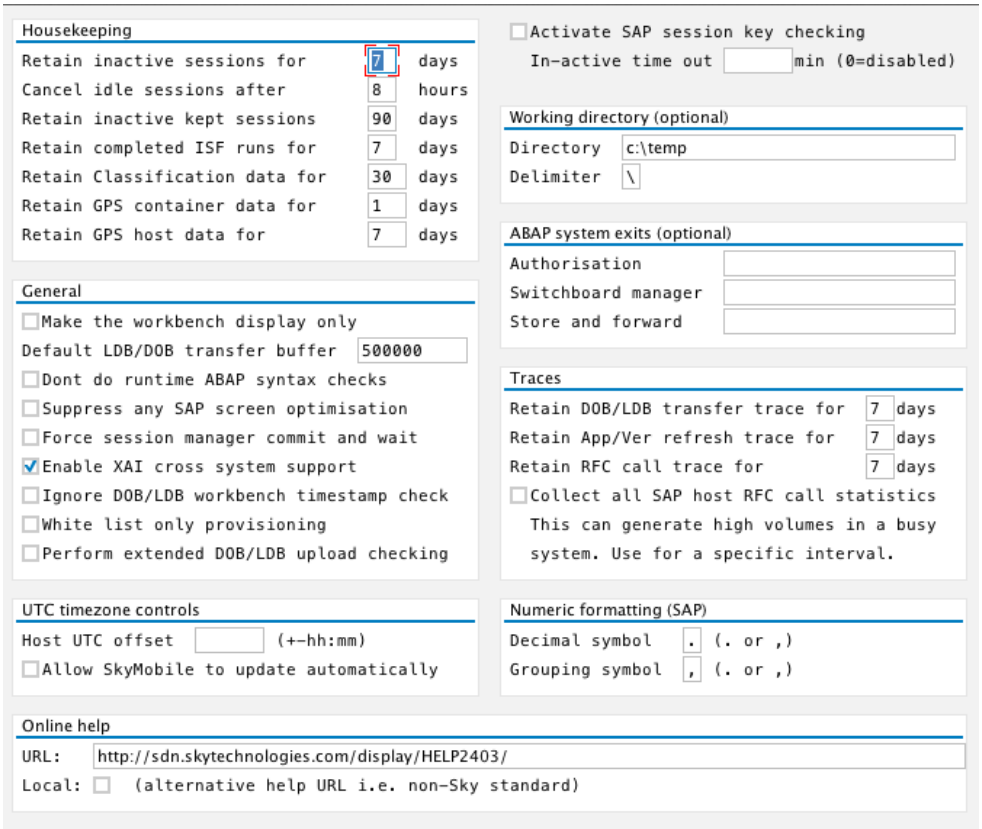
Field	Value Required
RFC_DEST	RFC Destination name for target SAP system



Once this is done, SkyMobile automatically analyzes all incoming requests: heartbeats, host session calls, and data object / LDB requests to see if they are local (exist on the master system) or exist on any of the configured XAI destinations. A cross reference of objects and where they exist is automatically maintained and the requests are routed to the appropriate SAP system through the associated RFC destination.

4.13.4.5 Enabling XAI

The final step after you configure your XAI and RFC destinations is to enable XAI in the master system. You do this in the SkyMobile system configuration. You do this in SkyMobile system management (YVTI) by clicking configuration and selecting the **Enable XAI cross system support** check box. XAI is now activated and you can now develop and test your cross SAP system applications and data synchronization.



Note: The following are the important considerations and restrictions when using XAI:

1. User attributes (work group, parameters 1-8) are defaulted from the master system first, and then any others that are not populated are then merged in from the remote XAI destinations (if the user definition is duplicated). So if you are using user parameter data for dynamic substitution into data profile selection criteria then you must segregate the parameter data effectively for applications; you must not confuse &P1 in ERP for application A with &P1 in CRM for application B; use &P2 for application B. It is advised that you maintain the user profile definitions centrally in the master XAI system only.
2. If for whatever reason, you need to forcably re-generate component cross reference, you need to clear the **WB_TIMESTAMP** field in the /SKY/YVTI_XAIAPP table for the relevant application (s).

5. System Management and Administration

5.1 About

Kony for SAP provides a comprehensive suite of tools and utilities to effectively manage mobile applications. These provide options to manage the SAP host, Access Gateways and remote Secure Containers, diagnose problems, optimize performance, provide system security, and high availability.

5.2 Key Topics

Networking
Provisioning
MEAP Server Management
Secure Container Management
Diagnostics and Tracing
High Availability and DRP Mechanisms
Performance Tuning and Optimization
Security
Mind Over Mobile
Utilities

5.3 Networking

As explained in the summary of components in the installation section, the Secure Container on the mobile device communicates with one or more Access Gateways over TCP / IP using either proprietary or HTTP / HTTPS protocols. The Access Gateway then communicates with the SAP MEAP server using the SAP standard RFC protocol. Message content transmitted between the secure container and SAP is in a proprietary internal format. This section provides a central reference of how networking works and is configured. Wherever possible, links are made to existing content to avoid duplication. The following diagram highlights the network communication points between the primary components, that is, the Secure Container on the device, the Access Gateway and the MEAP server.

1. The Secure Container client component communicates with its server component locally on the device using TCP / IP sockets (this is totally transparent to the user)
2. The Secure Container communicates with the Access Gateway through TCP / IP using either sockets using a nominated IP address and gateway port, or through HTTP using a URL reference. These gateway configurations are specified in the Secure Container configuration file.
3. The Access Gateway communicates with the MEAP Server using the standard SAP RFC protocol. The configuration for this, known as a host interface, is defined in the gateway configuration file.

5.3.1 Key Topics

TCP/IP
Communications using HTTP and HTTPS
Access Gateway Communications
When to Implement Data Encryption
Diagnosing Network Problems

5.3.2 TCP IP Ports

Kony for SAP makes extensive use of TCP / IP sockets to communicate between applications, hosts and devices. The process of configuring a socket for use in TCP / IP communications involves one side opening a socket on a specified port number, binding to it, then listening for incoming connections on that socket. The other side then sends a request to connect to that port. All going well, a new socket for communication between the two respective sides is then opened.

The Application Server and Access Gateway rely on port configurations to configure listening sockets for incoming connections. Different types of ports are used to categorize incoming connections based on their characteristics. Thus, for example, incoming connections from a graphical presentation client such as WPC targets a specific port. Incoming connections from a browser that attempt to view the web status page targets another port. Connections from a VT220 client may target yet another port. The Application Server and Access Gateway can both potentially listen on a number of different ports for incoming requests of different types, and the port configuration entries in their configuration data drive this.

Port numbering is a somewhat arbitrary process. In theory, you can use any port number between 1 and 65535. In practice, however, convention dictates that there are certain port numbers commonly reserved for use by the operating system or the other services that run on the host machine. For example, ports 20 and 21 are commonly used by FTP servers, port 23 by telnet services, port 80 for HTTP web traffic. On some operating systems, any port below 1024 is reserved for use by the operating system.

Generally speaking, you have less chance of causing a conflict with existing services if you choose a higher port number. It is also a good idea to segregate development, Quality Assurance (QA), and production environments using port numbering conventions. For example, you may use 10xxx for development, 12xxx for QA, and 13xxx for production. If a port number is already in use by another service running on the machine, unpredictable behavior can result. It is therefore important to select port numbers that are not already allocated.

Sky reserves ports in the range 5070 to 5079 for default installation and internal system ports. Refer types of ports for more information.

> ***Important:*** If you run more than one Secure Container on the same computer, you need to allocate a new range of port numbers for each instance. If you attempt to use the same port, the Secure Container normally detects this and stops. However, in some instances, unpredictable behavior may occur.

The Application Server communicates with all user interfaces, its web status page and its command interface through designated TCP / IP ports. Different types of ports are configured in `skymobile.cfg`. These entries are described in detail in the Configuration options section.

Click here to go back to the top.

5.3.2.1 Types of Ports

Port Type	Function	Default
Command port	Used to operate the Application Server from a telnet connection or used internally to stop, restart, and list active threads.	5070
Web status page	Used to view the status of a designated Application Server from a Web browser. The Application Server status page allows users to list current connections stop and restart and view / maintain local databases.	5071
Web HTTP port	Used to invoke applications from within a Web browser.	5073
GUI	Used to communicate with the specialized Presentation Clients, for example, WPC, BPC, APC, and IOS.	5074
Access Gateway	Used to facilitate communication with the back end SAP system.	5075
Text Terminal	Used to connect to and display application screens on a VT220 text terminal	5072

Port Type	Function	Default
Relay port	Used to monitor devices that are not otherwise accessible due to factors such as NAT (network address translation), and firewalls.	5076
Custom port	Used to implement customized behavior.	5077

5.3.2.2 Recommendations

The following conventions are recommended for port numbering:

- Ports 5070-5076 are reserved and you should not use them
- Check operating system services file to see the ports that are already declared for the host
- Allocate an odd number. Even numbers, for example, 10000 or 5000 tend to get allocated first
- Sky recommends to have application ports in the range 15072-15079 and 25072-25079
- Document your port allocations in the OS services file so that other future applications are aware.

Click here to go back to the top.

5.3.3 Communications using HTTP and HTTPS

Traditionally, communications between the secure container on the device and the access gateway took place over TCP / IP, using a proprietary internal protocol based around compressed / encrypted XML messages. In more recent years, it became more common for our customers to funnel all access to corporate backend systems through an HTTP proxy. To cater for this scenario, Kony for SAP now also supports communication between the secure container and the access gateway using the HTTP and HTTPS protocols.

For those who may be unfamiliar with some of the terminology involved, a brief explanation follows. HTTP is a protocol developed for use in transmitting data over TCP / IP. A complementary technology to HTML, it forms the basis for communications over the World Wide Web, and its explosive growth and very wide use has come about as a result of this. HTTPS (sometimes known as HTTP Secure or Secure HTTP) combines the HTTP protocol with an encryption protocol called SSL / TLS (Secure Sockets Layer / Transport Layer Security). Like HTTP, HTTPS is widely used on the World Wide Web. Its primary use is as a means to secure sensitive information, including such things as payment transactions and private or confidential data. In essence, HTTPS traffic is HTTP messaging secured by SSL / TLS.

SSL / TLS is really a group of related protocols that evolved and changed over time. Developed by Netscape, the first incarnation is originally known as SSL. It went through three versions, SSL1.0, SSL2.0, and SSL3.0. TLS was then proposed as an upgrade to SSL 3.0. At the time of writing, TLS also went through three iterations, TLS1.0, TLS1.1, and most recently TLS1.2.

The encryption model that the SSL / TLS used to secure information (a session key exchange using an asymmetrically encrypted handshake, with symmetrically encrypted communications thereafter) is also the same model that the Kony for SAP internal encryption used when using its own XML-based protocol. However, because HTTPS is widely known and well understood, many security professionals in large corporate environments may tend to prefer the "known quantity" of HTTPS.

> ***Note:*** One limitation to be aware of when using HTTP or HTTPS, is that the relay port connection feature no longer works. This effectively prevents a direct connection being made to an individual device to check on its status from (say) an authorized browser. This may be a factor to consider when deciding between use of HTTP/HTTPS and more traditional communications.

Kony does plan to introduce support for relay connections over HTTP / HTTPS at some point in the future. The problem is technical in nature, and lies in the fact that most proxies do not allow inbound connections to remain open for longer than a couple of seconds.

5.3.3.1 Underlying Mechanism

Communications are actually quite well suited to the HTTP model, as they use strict request / response pairing and the client (that is, the Secure Container) always initiates them. When using HTTP, the traditional communications messages are wrapped in an HTTP “envelope” and then simply transmitted in this format.

HTTP allows for two major ways for a client to talk to a server - GET and POST. An HTTP GET is usually a fairly simple request for the data associated with a specified URL. This is what browsers issue when a URL is typed into their address bar. It can become more complex through use of such things as query strings, but it is too limited for purposes of communications. Consequently, when using HTTP, the secure container sends all messages as HTTP POST requests. This is analogous to submitting a form or (even more accurately) uploading a file from a browser.

An example of POST request:

```
POST / HTTP/1.0
User-Agent: Secure Container/24.01.00 (iPhone OS 5.0.1)
Content-Type: multipart/form-data; boundary=--------7828186548
Content-Length: 519
Connection: Keep-Alive

----------7828186548
Content-Disposition: form-data; name="Content"; filename="Content"
Content-Type: application/octet-stream

<request message goes here>
----------7828186548
```

The response sent back from the Access Gateway to the Secure Container comes back as an HTTP-wrapped response message.

```
HTTP/1.0 200 Success
Date: Fri, 30 Mar 2012 00:39:22 GMT
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
```

```
Content-Length: 2687
Server: Secure Container/24.01.00 (Windows 2003 5.2)

<response message goes here>
```

Use of HTTPS applies SSL / TLS encryption over the top of these HTTP messages, resulting in (to the human eye) encrypted binary gibberish.

> ***Note:*** Because the underlying request/response pairs are essentially left unchanged other than by being wrapped as HTTP messages, you can still use compression and even encryption. However, it is typically unnecessary to also apply encryption if HTTPS were in use, as this results in the information getting "double encrypted" - once by Kony for SAP and then a second time during HTTPS transmission.

5.3.3.2 Levels of Security when using HTTPs

Before one can embark on configuring Kony for SAP for use with HTTPS, it is important to know some of the concepts underlying SSL / TLS.

SSL / TLS is complex, with many possible configurations and a multitude of different setup options. However, when one looks at the ways in which it is typically used, it is possible to identify three basic scenarios. In order of least to most secure, these are as follows.

Encryption Only. In this scenario, the "server" (that in the case of Kony for SAP, is either the Access Gateway or the HTTP proxy) sends the "client" (in case of Kony for SAP, the secure container) a certificate. The client performs limited or no checking of the server credentials as stated on the certificate. This scenario has the benefit of providing encrypted communications, but the client has no guarantee that the server is authentic. The fact that communications are encrypted may deter opportunistic snooping, but this does not protect against a knowledgeable attacker who may be able to masquerade as ("spoof") the server.

Encryption + Server Authentication. This scenario is similar to the last, except that the client performs strong checking of the server credentials in the certificate. This includes things such as:

- Ensuring that the certificate is trusted (that is, that it has a chain of trust back to a known certificate authority)
- Ensuring that it specifies a domain name matching the domain name that the client connects to
- Ensuring that the certificate is active and is not expired.

Provided if you do these steps correctly, the client can be reasonably assured that the server is genuine (that is, authenticated). However, in this scenario the server has no guarantee that it is talking to an appropriately authorized client.

Encryption + Server / Client Authentication. This scenario is similar to the previous one, except that the client also presents a certificate to the server that then checks its credentials as well. This is the most secure of the three scenarios, since both the server and the client are checked for authenticity, and all communications are securely encrypted. However, it raises the problem of how to manage the client certificate, so that it you can safely transmit and the client can store in a usable location.

To be clear, Kony for SAP does not support “Encryption Only” type scenarios that are felt to be of limited value to our customers. The Secure Container always checks the certificate that the SSL / TLS server sends. Although it is theoretically possible to make such checking switchable through a configuration option, this is a potential security weakness. Hence, the checks always take place.

5.3.3.3 Configuration and Setup

The precise nature of the configuration required to configure Kony for SAP for use with HTTP or HTTPS greatly depends on the desired deployment model. There are a number of possible scenarios. The following possible scenarios are in the table, along with the configuration tasks required for each.

Scenario	Secure Container Configuration	Secure Container Certificate Setup	Gateway Configuration	Gateway Certificate Setup	Proxy Configuration
Direct HTTP with no proxy	X		X		
Direct HTTPS with no proxy	X		X	X	
HTTP using a proxy	X		X		X
HTTPS using a proxy (HTTP to the gateway)	X		X		X

Scenario	Secure Container Configuration	Secure Container Certificate Setup	Gateway Configuration	Gateway Certificate Setup	Proxy Configuration
HTTPS using a proxy (HTTPS to the gateway)	X		X	X	X
HTTPS using a proxy (HTTP to the gateway), with client certificate validation	X	X	X		X

Scenario	Secure Container Configuration	Secure Container Certificate Setup	Gateway Configuration	Gateway Certificate Setup	Proxy Configuration
HTTPS using a proxy (HTTPS to the gateway), with client certificate validation	X	X	X	X	

Because there are many potential pitfalls involved in configuring some of the more complex scenarios, the recommended approach is to attempt to get things working one step at a time, working from the simpler scenarios through to the more complex ones, rather than trying to implement everything at once.

In a development environment, it is comparatively straightforward to implement direct HTTP communications between a secure container and the gateway. It is then possible to progressively introduce further elements into the scenario (for example, the HTTP proxy may come next). This provides a number of “checkpoints” along the path to a final solution.

> ***Important:*** It is not uncommon to run into configuration / setup difficulties, particularly when HTTPS is involved. The nature of SSL / TLS that involves an exchange of credentials in a layer with limited visibility to the calling application, often complicates the diagnosis of problems. Many HTTP proxies have logging modes that permit detailed examination of the handshaking process. This can often provide valuable information on exactly what is going on.

Secure Container Configuration

Use of HTTP/HTTPS communication influences the Secure Container in two main areas.

Provisioning

The first change is in the provisioning process. When using traditional communications, a provisioning server and port number is normally entered. If HTTP or HTTPS communications are used, you need to enter a URL instead of a hostname. For example, if HTTP is in use, rather than entering a provisioning host of "skytest" and a port number of 15076, the user enters "http://skytest:15076" in the hostname field (that is now labelled "Host or URL"). The fact that the URL begins with "http://" tells Kony for SAP that HTTP communications are used. If HTTPS communications are used instead, the user instead enters "https://skytest:15076".

> ***Note:*** Because a URL can contain the port number, if a URL is used, the port number field on the provisioning screen becomes "greyed out" and you cannot use.

If not specified, the port number defaults to 80 when using HTTP, and 443 when using HTTPS. These are the default ports for the respective protocols.

Application Server Configuration

The next area of impact is in the Application Server host interface configuration. In order to use HTTP or HTTPS communications, you must change the Application Server configuration entries from something that looks like this:

```
SERVER.HOSTINTERFACE:<name>.XMLGATEWAYHOSTS = skytest
SERVER.HOSTINTERFACE:<name>.XMLGATEWAYPORT = 15076
```

To something more like this:

```
SERVER.HOSTINTERFACE:<name>.XMLGATEWAYHOSTS = http://skytest:15076
```

Once again, note that because the port number is part of the URL, you no longer require the XMLGATEWAYPORT entry when you specify a URL.

Secure Container Certificate Configuration

Configuring a certificate on the Secure Container is only necessary in situations where the customer wishes to implement client certificate validation (the most secure of the levels of HTTPS security discussed here). The HTTP proxy is typically configured to only accept incoming connections from clients that present a valid certificate. Here we must grapple with the problem of how to get the necessary certificate onto the target device and make it available for use by Kony for SAP.

Firstly, it is necessary to point out that at this point in time, only the iPhone client currently supports client certificate validation.

In order to be able to present a certificate to the server, the Secure Container first needs to obtain a copy the certificate to be presented. On the iPhone, you can typically accomplish this by opening a document with the file extension skymobile-p12.

The `.p12` file extension normally indicates a PKCS #12 export file that is a widely used file format used to transport security credentials. You can protect the credentials by a passphrase. The `.skymobile-p12` file extension simply denotes a `.p12` file that is intended for use with ony for SAP. So, for example, you can make a file called credentials.p12 suitable for use with by simply renaming it to credentials.skymobile-p12.

You can make the `skymobile-p12` file available to the device on which the Secure Container is installed in a number of ways. The most common would be to mail the actual file in question to the end user, or alternatively a URL link to the file.

When the `skymobile-p12` document is launched, the secure container is invoked and it attempts to import the security credentials in the file for use with HTTPS. If the file has an associated passphrase, the user is prompted to enter the passphrase to unlock the file. The user is advised of the success or otherwise of the operation.

Once a certificate is imported, it is always presented to the server during HTTPS handshaking. The only way to purge the certificate is to perform a reset of the Secure Container (that deletes all associated data, including the stored certificate).

Gateway Configuration

In order to ensure that the Access Gateway has a port open that is able to converse in HTTP or HTTPS, you must configure the following configuration option:

```
SERVER.XMLGATEWAYPORT:<name>.USEHTTP = true
```

It is not possible to intermix normal communications with HTTP traffic on the same port, so if you plan to use both modes of communication you should configure multiple XML gateway ports on the Access Gateway - one for regular traffic and one for HTTP traffic. You are recommended to do this whilst attempting to implement HTTP communications in a development environment.

Gateway Certificate Configuration

If the Access Gateway is required to accept inbound HTTPS connections, further configuration is required. In the first instance, you must set the following configuration option:

```
SERVER.XMLGATEWAYPORT:<name>.SECURE = true
```

This tells the Access Gateway that the port is to be secured through HTTPS. However, in order to do this, it also requires installation of a certificate that it can use to send to incoming connections to verify its authenticity. As was mentioned earlier, the secure container performs stringent tests on this certificate, so a a trusted certificate authority must issue it, and must match the server domain name.

Installation of the certificate on the gateway server requires use of the Java keytool utility. This utility has its own documentation, but some simplified instructions follow.

Firstly, obtain a copy of the certificate to install in the form of a file (in either X.509 v1, v2, v3, or PKCS#7 format). Then, invoke the keytool utility as follows:

keytool -import -file <cert file name>

The keytool utility prompts for a password to protect the newly created keystore. It then requires re-entry of the password to verify that it is correct. If all goes well, you should create a new file called `.keystore` that contains the imported certificate. Once you create the keystore file, copy it to the same directory as the Access Gateway. You then need to configure the Access Gateway to use it, by configuring the following directives:

```
SERVER.XMLGATEWAYPORT:<name>.SSLKEYSTORE = .keystore
SERVER.XMLGATEWAYPORT:<name>.SSLKEYSTOREPASSWORD = <your password>
```

Proxy Configuration

There are a multitude of different products and devices available that can function as an HTTP proxy. Because of this, it is difficult to offer guidance in exactly how you may need to configure a proxy. Generally speaking, the proxy is already in place on any given customer site, and has administrator(s) responsible for its correct configuration. You generally require liaising with these staff members in order to effect any configuration change.

5.3.4 Access Gateway SAP RFC Communication

The Access Gateway uses standard SAP RFC libraries and functionality to communicate with the MEAP "inside SAP". Instead of creating a unique connection per user or constantly opening and closing connections with SAP that are inefficient, the gateway manages a connection pool that it establishes using a specific SAP CPIC or service type user id (you cant use to logon directly to SAP). A minimum and maximum number of connections is configured, that is, if idle for a period of time, then the number of RFC connections is scaled back to the minimum setting. You may monitor the Access Gateway RFC connections using the standard SAP gateway trace and diagnostic tools (for example, SMGW). You may also use the standard SAP RFC trace if required.

A typical Access Gateway SAP host configuration section looks like this:

```
SERVER.HOSTINTERFACE:SAP.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SAP.JNILOGGING = false
SERVER.HOSTINTERFACE:SAP.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SAP.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SAP.MINIMUMNUMBERHANDLERS = 2
SERVER.HOSTINTERFACE:SAP.SAPCLIENT = client number
SERVER.HOSTINTERFACE:SAP.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SAP.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SAP.SAPHOST = SAP Host IP Address
SERVER.HOSTINTERFACE:SAP.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SAP.SAPLBGROUP =
SERVER.HOSTINTERFACE:SAP.SAPLBHOST =
SERVER.HOSTINTERFACE:SAP.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SAP.SAPPASSWORD = SAP CPIC/Service user
password
SERVER.HOSTINTERFACE:SAP.SAPSYSTEM = SAP system number
SERVER.HOSTINTERFACE:SAP.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SAP.SAPUSELB = 0
SERVER.HOSTINTERFACE:SAP.SAPUSER = SAP CPIC/Service user id
```

5.3.5 When to Implement Data Encryption

Data encryption, in essence, is all about making sure that unauthorized third parties can not get access to your application confidential information. Encryption technology is complex and there are many different ways in which you can use. The most common ways to implement secure / encrypted communications in an implementation are as follows.

1. Virtual Private Networking. A Virtual Private Network (VPN) is an encrypted tunnel established over a public (or semi-public) network. Communications that are routed through a VPN tunnel are encrypted and hence are normally considered secure. If your organization makes use of a VPN, you can hide the Access Gateway from the larger Internet, with access only possible through the VPN. The downside to this is that you must somehow establish the VPN in order for the Secure Container to talk to the Access Gateway. Some VPN products have "on demand" features that permit them to automatically launch whenever a request is made to communicate with the Access Gateway. Others, however, the user may launch manually.

2. Using an HTTPS Proxy. If you decide to use HTTPS communications, the Secure Container can communicate with the Access Gateway through a HTTP proxy, using HTTPS. HTTPS is a secure, encrypted protocol designed specifically to safeguard information during transmission over public networks.

3. Encryption. Kony for SAP also has an encryption layer of its own that permits strong encryption of information transmitted between the Secure Container and the Access Gateway. Encryption is highly customizable, and allows you to select from a variety of different encryption algorithms and key strength options.

If you transmit information over a public network (which in the vast majority of implementations are definitely the case), you must think to ensure the security of your data using one or many of the above technologies. Even if you have no financial information of any kind passing back and forth, privacy legislation in many countries dictates that third party information should not be made freely available to casual snooping.

5.3.6 Diagnosing Network Problems

Networking is a very deep field of knowledge, with many different fields of specialization. Consequently, it is not possible to give an exhaustive or definitive dissertation on the way in which you can diagnose all networking problems. However, there are some useful tools and techniques that you can use in a variety of situations involving network problems. A brief introduction to some of these is below.

ping

The ping command is available in both Windows (from the command line) and UNIX. It sends a series of single data packets to the target host, and then awaits responses. The results appear back to you. Here is an example of a healthy result.

```
$ ping 192.168.2.62
PING 192.168.2.62 (192.168.2.62): 56 data bytes
64 bytes from 192.168.2.62: icmp_seq=0 ttl=127 time=0.304 ms
64 bytes from 192.168.2.62: icmp_seq=1 ttl=127 time=0.399 ms
64 bytes from 192.168.2.62: icmp_seq=2 ttl=127 time=0.374 ms
64 bytes from 192.168.2.62: icmp_seq=3 ttl=127 time=0.641 ms
```

If connectivity between the two network points is poor, you generally see errors or timeouts in the ping results. Failure is, however, not conclusive proof of networking problems. Some hosts are configured not to respond to ping traffic. Interposing firewalls between the two network nodes can also be a factor, as a firewall may prevent the ping traffic from even reaching its target.

traceroute

The traceroute command is available in both Windows (from the command line, where it is known as "tracert") and UNIX (where it is known as "traceroute"). It works in a similar manner to the ping command, but gives a more detailed breakdown of the route taken to reach the target host. This can sometimes be extremely useful in pinpointing which network node may cause a particular problem.

```
$ traceroute crateandbarrel.com
traceroute to crateandbarrel.com (63.73.227.20), 64 hops max, 40 byte packets
1 192.168.1.254 (192.168.1.254) 3.811 ms 1.183 ms 1.878 ms
2 bras4-l0.rcsntx.sbcglobal.net (151.164.182.19) 13.565 ms 12.481 ms 12.897 ms
3 dist1-vlan120.rcsntx.sbcglobal.net (151.164.162.66) 13.191 ms 49.095 ms
11.045 ms
4 bb1.10g5-0.rcsntx.sbcglobal.net (151.164.243.183) 36.636 ms 14.752 ms 12.251
```

```
ms
5 151.164.40.29 (151.164.40.29) 209.134 ms 10.710 ms 10.689 ms
6 gar6.dlstx.ip.att.net (12.122.79.65) 11.948 ms 11.528 ms 11.110 ms
7 tbr2.dlstx.ip.att.net (12.123.16.222) 38.891 ms 38.852 ms 40.728 ms
8 tbr1.sl9mo.ip.att.net (12.122.10.89) 92.264 ms 41.760 ms 40.105 ms
9 cr1.sl9mo.ip.att.net (12.122.18.33) 38.605 ms 37.589 ms 34.865 ms
10 cr2.sl9mo.ip.att.net (12.122.2.218) 35.912 ms 35.823 ms 34.482 ms
11 cr2.cgcil.ip.att.net (12.122.2.21) 38.396 ms 42.149 ms 67.544 ms
12 tbr1.cgcil.ip.att.net (12.122.17.158) 43.427 ms 43.544 ms 39.890 ms
13 gar1.chail.ip.att.net (12.123.4.69) 34.402 ms 47.905 ms 36.671 ms
14 * * *
15 border6.pc1-bbnet1.chg.pnap.net (64.94.32.11) 146.279 ms 70.568 ms *
16 fry-6.border6.chg.pnap.net (69.25.240.10) 36.692 ms 37.844 ms 36.965 ms
17 64.95.84.253 (64.95.84.253) 38.493 ms 47.781 ms 50.404 ms
18 * * *
19 * * *
20 63.73.227.20 (63.73.227.20) 98.401 ms * *
```

The output from the traceroute command shows the various network "hops" between the current host and the target host. It also outlines the network latency (time taken to transmit data between) each of the nodes along the way. Once again, if connectivity between the two network points is poor, you will generally see errors or timeouts in the traceroute results. The output shown above clearly indicates some kind of issue from hop 14 onwards, with the asterisks indicating data packets sent with no response.

The traceroute command is, however, subject to the same limitations as the ping command. The target host may be configured not to respond to traceroute (ICMP) traffic, and once again an interposing firewall may potentially cause it to fail.

telnet

The telnet command is available in earlier versions of Windows (prior to Windows 7), and UNIX. It is designed to open a simple socket for communication with the target server on a specified port. Most telnet tools target port 23 by default, but you can in fact specify any port. The value of the telnet command as a network diagnostic tool is often overlooked. If you can successfully open a telnet session to a target server on a given port, then you have an open communication channel (that is, network connectivity) between the two nodes. Its value lies in the fact that, unlike ping or traceroute, the telnet command can target a specific port, meaning that it often bypasses other sources of interference (such as firewalls).

For example, if you want to check that port 80 is open on host 192.168.2.62, you can use the telnet command to produce any of the following results.

```
$ telnet 192.168.2.62 80
Trying 192.168.2.62...
Connected to 192.168.2.62.
Escape character is '^]'.
^]
telnet> quit
Connection closed.
```

This tells us that you have successfully established a connection to the host on port 80. It suggests that if there are in fact networking issues, they are sporadic and / or transient.

Note that, as per the introductory message, '^]' (control-right square bracket) is the key combination normally used to break out of a telnet session and drop-down to the telnet command prompt. From there, type 'quit' to normally end the telnet session.

```
$telnet 192.168.2.62 80
Trying 192.168.2.62...
telnet: connect to address 192.168.2.62: Connection refused
telnet: Unable to connect to remote host
```

This tells us that the target host refused the connection that means that you have connectivity to the host, but that no process is listening on the specified port.

```
$ telnet 192.168.2.62 80
Trying 192.168.2.62...
telnet: connect to address 192.168.2.62: Operation timed out
telnet: Unable to connect to remote host
```

This tells us that you have no connection to the remote host at all, and it can be symptomatic of network connectivity issues.

Packet Sniffers

When all else fails, the "big guns" in network diagnostics are programs known as "packet sniffers". A packet sniffer is able to capture and record every network packet making its way through a particular node. They normally capture enormous volumes of data, much more than can normally be analyzed in real time. Typically they are used to capture an instance of problematic behavior that you can then painstakingly analyze after the fact. Packet sniffers are often difficult and confusing for the novice, as they generally assume a fairly high

degree of networking knowledge on part of the user. However, as a substitute for diagnosing network problems, they have no equal.

Wireshark is one freely available and widely used packet sniffing tool that you often use internally within Sky Technologies.

5.4 Provisioning

In the past, you manually maintained configuration and application cross reference entries to control how the server ran on the device and what applications you can download. As of release 23+, Sky introduced an automated provisioning model where you can maintain centrally all the requirements in the MEAP Server and then push out to the Secure Container when it is first starts on the device. This capability is also a necessity for the newer IOS and Android platforms that are more "black box", that is, maintaining application components on the device is more restricted. This process of automatically deploying the Secure Container on the device is known as provisioning. The objective is that users can download the Secure Container from either a public or private application store and then start it and it automatically configures itself up with minimum or no user input.

In Kony for SAP, there are two levels of provisioning:

- The Secure Container run time environment on the device that is called Container provisioning
- The User applications and data that they are authorized to use, is called Role based provisioning

A summary of the recommended on-premise provisioning process is:

- Configure the network encryption parameters and generate a key (or use HTTPS networking)
- Configure the Identity management interface on the Access Gateway
- Use the Server Profile workbench to define a profile
- Use the User Profile workbench to define users and roles
- Implement a re-direction profile in Kony for SAP Cloud (optional)
- Install the Secure Container onto the device
- Start the Secure Container:
    - Follow the provisioning prompts (if not white listed in which case no input is required)
    - Associate the user with the device using identity management

- Start the Application Console to launch and manage the on demand applications.

5.4.1 Key Topics

Secure Container Provisioning
Role based Provisioning
The Application Console

5.4.2 Secure Container Provisioning

Secure Container provisioning is the process by which a newly installed container is automatically configured with:

- appropriate run time configuration options
- any required encryption objects
- access to required application(s) at the server level (not user)
- access to required data objects at the server level (not user)

All rules and configurations are defined centrally in the MEAP Server using the Server Profile Workbench and are linked to a unique profile name. Each profile may have different instances to describe variances depending on the platforms technical attributes, for example, specific configuration for iOS versus BlackBerry. So when a newly installed container first starts, it then goes through a initial provisioning process that selects what profile instance to apply and sets up the container environment for use.

> ***Note:*** The Container provisioning process is ignored if there is any existing configuration object for example, `SkyMobile.cfg` file in Windows.

5.4.2.1 Container Provisioning Process

First you download the Secure Container and install on the mobile device. For information on how to download and install the container onto the device, refer Installation and Upgrade. The first time, you invoke the container and there is no existing configuration on the device, the automatic provisioning process is activated. The first step is to check for a explicit "white list" entry in Kony for SAP Cloud for the unique device ID. If one exists, the device is automatically configured and the application server starts without the user having to enter any information. If no white list entry is found, then the user is guided through a manual provisioning process as per the steps documented in the table below.

When the application server first runs, it detects that it has no local configuration and automatically prompts the user with the following screen. There are two choices, that is, "Kony Cloud" or "On Premise". Kony Cloud is Kony own system providing demonstration and hosted applications. On Premise is to connect with a Gateway installed on the customers own system. You may also have what is called a re-direction profile that is used to re-route the provisioning process to another MEAP Server. For example, the user selects Kony Cloud and enters 'MY COMPANY' as the profile and a secure PIN code. The provisioning process automatically disconnects from the Kony Cloud and starts the provision process again against the specified Access Gateway, that is, the Companies own MEAP Server.	
On Premise	
When you select this option, the following screen appears. You are prompted for the Gateway Host address and port to connect to, and the provisioning profile to use (PIN pass code and User are optional). When you submit the request, a provisioning request is made through to the MEAP Server Profiler to generate all the requirements for the device. Once the Container is provisioned, it goes through its initialization and the primary application starts.	
Kony Cloud	

Kony Cloud Access Gateway is known, therefore you need to specify only the profile and optional PIN pass code and User ID.

As above, when you submit the request, a call is made through to Kony Cloud MEAP Server Profiler to generate all the requirements for the device. You may use a redirection profile in Kony cloud to re-route the provisioning process to another MEAP Server.

Once the Container is provisioned, it goes through its initialization and the primary application starts.

5.4.2.2 Redirection Profile

A redirection profile is a special profile definition created using the server profile workbench that is used to re-route provisioning from one MEAP Server to another. A typical example of this is to create a redirection profile in the Sky Cloud to a Customers on-premise MEAP server to help simplify provisioning.

> ***Note:*** The entire provisioning process starts again from scratch against the target system, that is white / black lists.

To create a redirection profile, you create a profile definition and flag it as a redirection profile. You then configure a gateway reference pointing to the target Access Gateway and an optional encryption object to apply. Once the profile is activated, any Secure Container provisioning request against this profile is re-routed to the other MEAP Server through the configured Access Gateway.

5.4.3 Role based Provisioning

Role based provisioning is where a user only receives the applications and data that they are authorized to use. It only applies to known users does not apply at the container level. All the user and role definitions are defined in the MEAP Server using the User Profiler workbench. This functionality is designed to use in conjunction with the Sky Application Console that is used as a launch pad in the Secure Container on the device.

To use role based provisioning, you must know the user ID either through identity management before the Secure Container starts or through your own logon process by populating the session user. In this way, the user is directly associated with the unique Container group / id and subsequently can be derived by the MEAP Server for application and data cross referencing and synchronization.

Sky recommends that you use the Secure Container provisioning model in conjunction with identity management and user profiling to fully automate the management of mobile applications and eliminate the need for mobile device management software.

5.4.4 The Application Console

Kony provides a default Application Console application that you may use as a launch pad for applications. The console provides the framework for "mobile applications on demand", that is, you can add and remove applications from an authorized list. The user may also perform some basic customization (if allowed), such as change the background image, assign different colors, manage a favorites list, and toggle the list view. You can fix all or part of the user interface, for example, the company logo and standard color scheme.

> ***Note:*** You may replace the default Application Console (910/001) as part of upgrade. If you wish to customize it, make sure that you copy it to another application version (do not use app IDs 900-999 that are generally used for Kony for SAP internal applications).

The idea is that the user logs onto the console, using Kony for SAP standard Identity Management functionality. The user id is then used to determine the available applications through roles defined in the user profiler. The first console screen shows a list of available applications in either a tiled or list view. You may also configure a "folder" of related applications in the user profiler. You may then expand the folder to list just those applications. The application console only supports a single level of folders.

The following is covered in this section:

Configuring the Application Console in the Server Profiler

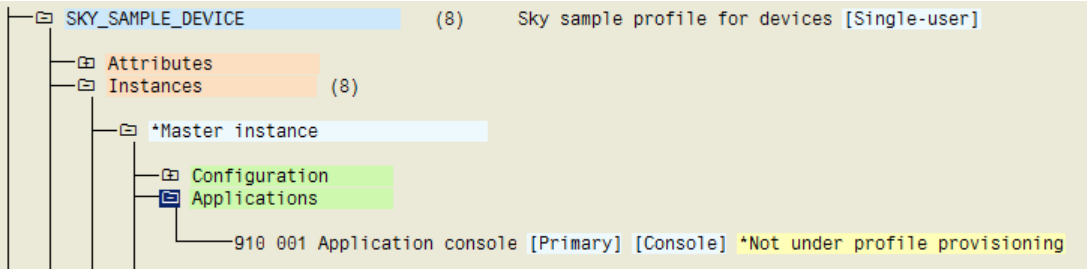
Assigning Applications to a User Console

Using the Application Console

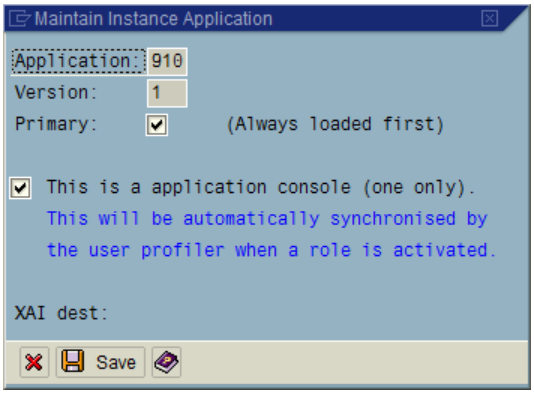
5.4.4.1 Configuring the Application Console in the Server Profiler

The standard Kony for SAP application 910 version 001 is implemented as part of the MEAP Server check installation process. Typically this application version is defined as the base application in a Server Profile, flagged as the application console and primary application. User role based

provisioning is then defined using the User Profiler, that is, what on-demand applications the user can use through their associated roles. If you took your own copy of the Console, then you should configure this in your server profile as a application console and the primary application. An example of this is in profile SKY_SAMPLE_DEVICE that refers to the default Application Console 910/001:

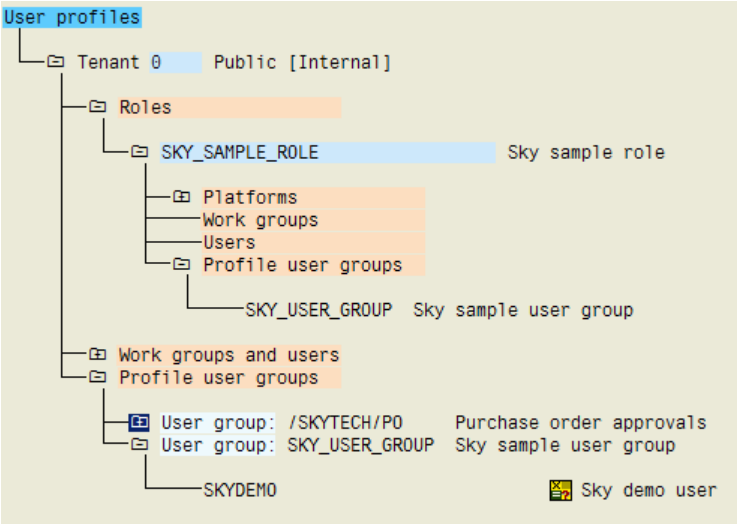


> ***Note:*** This is configured at the "Master instance" level; the same console is used for all the different logical platforms.



5.4.4.2 Assigning Applications to a Users Console

The user profiler is used to assign Kony for SAP applications and URL references to roles. The roles are then associated with users through special profile user groups, work groups or directly. Kony recommends using the "user group" feature because it is much more flexible and has no limitations. A user may be associated with many roles and groups. In the following example, user group SKY_USER_GROUP is assigned to the role SKY_SAMPLE_ROLE that contains the user SKYDEMO.



So when user SKYDEMO logs on and the Secure Container does a "application check", the user profiler is checked to determine which applications should be returned to the device. If there are no applications associated by either the server or user profiler, then the direct applications server cross reference is used.

Using folders:

You can group one or more applications into a "folder" that is defined at the role level, that is, as soon as you define an icon at the role level, it becomes a folder. When you then configure the applications, you can decide whether they are under the folder (not stand alone) or on the main screen (stand alone). Folders are useful to logically group related applications, for example, you may have a folder `Sales Automation` that has applications: Account management , and Opportunities. The role icon appears on the main list and the sub-applications are listed when you activate it. You return to the main overall list using the refresh icon in the tool bar.

On demand (optional) applications:

If you configure an application as "on demand", this means that the user has a choice whether to download it or not, that is, it is optional. If the console detects that there are on-demand applications, a special "add applications" icon appears in the console tool bar. This displays a list of on-demand applications that are available and when you select them, they are automatically downloaded and made available to the user. The user may also remove a on demand application, that is, the application and its data are physically deleted and is made available again.

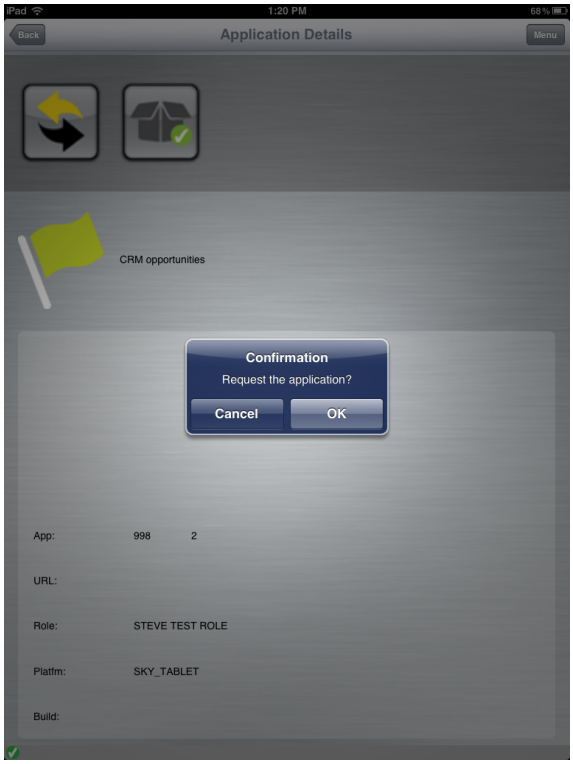
5.4.4.3 Using the Application Console

Once you provision your secure container and logs on using identity management (or your own custom screen), the application console automatically launches and displays the applications and folders that the user is authorized to. The console has a basic tool bar at the top of the screen that is used to:

- Toggle the view between tiled and a list
- Refresh the display with the host (synchronize)
- Add on-demand applications.

Screen	Description
	The default console layout is a tiled view of the available applications and folders. The tiled view shows five applications across. You activate the application icon to launch it. If the icon is a folder and you activate it, the list re-appears with only the applications in the folder. To return to the main list, activate the refresh icon.

Screen	Description
	The list view shows one application per row. You launch the application by activating its icon. If you activate the row, you are transferred to a screen that displays the applications details. For on-demand applications, this is how you remove them from the list, that is, from the application details screen. You activate the application icon to launch it. If the icon is a folder and you activate it, the list re-appears with only the applications in the folder. To return to the main list, activate the refresh icon.

Screen	Description
	If there are applications on-demand, then the "add application" icon appears in the tool bar. When you activate this, a list of on-demand applications appear. When you select one, the application details appear and you may request the application from here.

5.5 MEAP Server Management

The MEAP Server provides a range of utilities and tracing options. These are all available from the System Management Console (YVTI) and in the workbench to a limited extent. The systems management console (YVTI) provides a central point from which you may invoke basic diagnostic tests and the various available utilities.

> ***Note:*** You can automatically provision Application Servers from the central Server Profiler. Click here for more details.

5.5.1 Key Topics

Standard Interface and Schedules
Systems Management Console
Housekeeping
Automatically Monitoring Heartbeat Connections
Remote Software Upgrades
Overriding the Server Group and ID

5.5.2 Standard Interfaces and Schedules

The check install procedure automatically creates ECS process and schedules to execute standard Sky functions, such as daily housekeeping. These are all created under the SPI_SERVICES, ECS_SERVICES and VTI_SERVICES groups. Normally, you never need to maintain these internal definitions. But if you do, you may use standard ECS workbench functionality to review and maintain the ECS process and schedules. The following is a list of the generated processes and schedules:

Process	Description
ECS_CHECK_HBEAT	Check Secure Container heartbeats
ECS_ESCALATE_PROCESS	ECS notification: Escalate process
ECS_ESCALATE_SCHED ECS	ECS notification: Escalate schedule
ECS_HOUSEKEEPING	ECS daily housekeeping routine (Workdays)
ECS_HOUSEKEEPING_SUN	ECS weekend housekeeping routine (Sunday)
ECS_IDOC_MONITOR	ECS IDOC monitor
ECS_PHASE_TIMEOUT	Check phase timeout
ECS_RFC_MONITOR	ECS: SAP RFC destination monitor
ECS_SCHEDULE_CHECK	ECS: Check activated schedule definitions
ECS_SERVER	ECS: Background server
ECS_SERVER_CHECK	ECS: Check active background servers
ECS_START_SCHEDULE	ECS: Start nested schedule definition

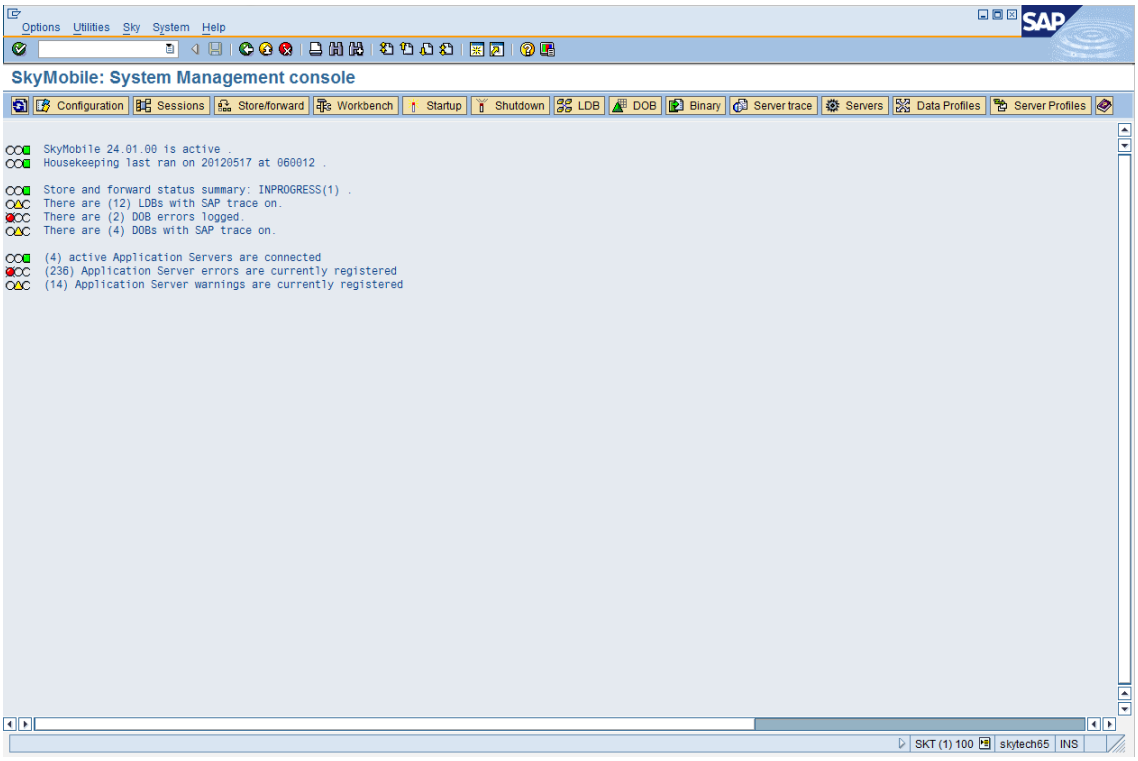
Process	Description
ECS_START_SERVER	Start background server
SPI_HOUSEKEEPING	SPI: Daily housekeeping routine
TSX_CHECK_HBEAT	TSX: Check Java server connections
TSX_LOG_DATA	TSX: Log unknown SAP connections
TSX_PING	TSX: Ping remote Java server
VTI_EXAMPLE_DOBUPL	Example DOB upload process
VTI_EXAMPLE_UPLOAD	Example LDB upload process
VTI_HOUSEKEEPING	Periodic housekeeping routine
VTI_ISF_TRAN	Execute stored ISF transaction
VTI_USER_APPS	Maintain user applications

Schedule	Frequency
ECS_HOUSEKEEPING	EVERY WORK DAY AT 060000
ECS_HOUSEKEEPING_SUN	EVERY SUN FROM 20120101 AT 060000
ECS_SCHEDULE_CHECK	EVERY DAY, EVERY 01 HOURS
ECS_SERVER_CHECK	EVERY DAY, EVERY 01 HOURS
ECS_START_SERVER	WHEN /SKY/START_PROCESS_#CLIENT IS TRIGGERED

Schedule	Frequency
SPI_HOUSEKEEPING	EVERY DAY AT 040000
VTI_HOUSEKEEPING	EVERY DAY AT 050000

5.5.3 Systems Management Console

You invoke this through transaction YVTI. On invocation, it performs some basic system checks and highlights any problems encountered. It is also the central point from which you maintain the MEAP Server configuration.



> ***Note:*** The list of checks is context sensitive and so you just need to click the relevant line and you are taken to the associated detail display (drill down).

From the console, you can perform a range of tasks including:

- Shutdown and Start Up the SAP add-in
- Maintain Kony for SAP configuration
- Perform a check install
- Reset application / version passwords

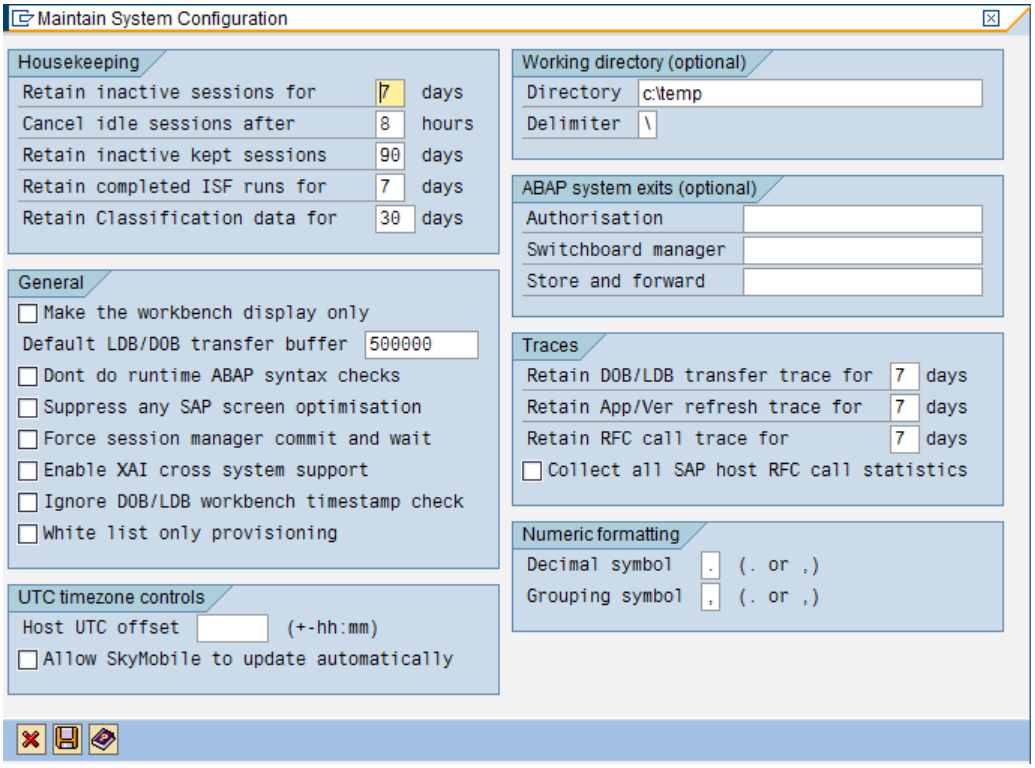
In addition, there are buttons to access the following:

- MEAP Session Manager
- Store and Forward Manager
- Workbench
- LDB Workbench
- DOB Workbench
- Binary Object Workbench
- Server Trace Facility
- Container Management Console

5.5.3.1 MEAP System Configuration

The system configuration is client independent, that is, you only need to perform it once in any client. Log onto any SAP client where Kony for SAP is used. Execute transaction, YVTI (Systems Management Console) and select the configuration option from the toolbar.

> ***Important:*** An important configuration is to make the entire Workbench in SAP 'display only' thereby globally preventing the maintenance of any definitions, that is, production environment.



The check installation process creates a default configuration definition. The above options govern how Kony for SAP house keeps definitions, where working files are written, the installation of specialized management ABAP exits and other run-time behavior. Generally the defaults suffice, but you may tailor as required.

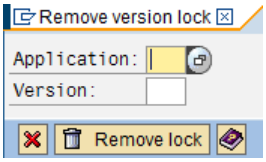
Option	Function
Retain inactive sessions	SAP session manager records and tracks automatically the SAP user connections (sessions). This option governs how long in-active sessions can remain in the system before the housekeeping deletes them automatically.
Cancel idle sessions	This option automatically cancels active sessions that are not used for a specified number of hours.
Retain inactive kept sessions	Kept sessions are permanently saved session definitions that the Switchboard definitions uses. This option governs how long in-active "kept" sessions can remain in the system before the housekeeping deletes them automatically.
Retain completed ISF runs	ISF stands for Intelligent store and forward. Store and forward requests to SAP are stored in a queue and processed either automatically or manually. This option governs how long successfully completed store and forward requests are kept for reference, before the housekeeping automatically deletes them.
Retain classification	LDB classification data is stored centrally in a SAP table. This option determines how long the system is to keep historical data.
Working directory and delimiter	Data may be written to host files during processing, for example, diagnostic tracing. These options identify the host directory and delimiter to use for external file processing.
ABAP System Exits	You may install custom ABAP exits. The precise usage of these exists is described in the system management section.
Retain LDB transfer trace	LDB stands for local database definitions. You can track download, refresh and upload requests of database data. This option governs how long the trace options are kept for reference, before the housekeeping deletes them automatically.

Option	Function
App / Version Refresh trace	A trace is kept automatically by the system of all application version refreshes. This option determines how long the system is to keep historical data.
Global RFC call trace	You may configure the RFC Call trace either **system wide** through the **Collect all RFC call statistics to host** option here or specifically through the Application Server trace facility in the workbench. In either case, this option determines how long the system keeps historical data.
Make the workbench display only	This option is designed for production systems to 'lock down' the ability to change definitions using the workbench. If specified, you may only import changes using the transport facility in conjunction with change management procedures.
Default LDB / DOB transfer buffer	Populate with a default transfer buffer size in bytes. If a LDB or DOB download or refresh synchronization request is received in SAP with no transfer buffer, this value is automatically defaulted. This is to prevent possible situations where 'no transfer buffer' requests of extremely large data volumes can potentially crash the receiving server or gateway.
Do not do run-time ABAP syntax checks	By default, Kony for SAP performs ABAP syntax checks on all custom exits and function modules to provide more graceful error processing, that is, avoid short dumps. This is optimized to do only once per session, still can incur overhead. This option suppresses all syntax checking and is recommended for production environments.
Suppress any SAP screen optimization	By default, screen definitions between the server and SAP add-in are optimized, that is, only essential elements are transmitted and then reconstructed by the SAP session manager. This is done to optimize network and gateway performance. This global option effectively turns off screen optimization, that is, the entire screen definition is transmitted. You should also do this only if there are problems with the optimization facility.

Option	Function
Force session manager commit and wait	This option forces the SAP session manager (host screen processing) to wait for all its commit works to confirm before continuing with processing. The default is perform as much asynchronous processing as possible, that is client / server. In some heavy SAP environments, it is possible for slow SAP database processing to cause internal session errors as it uses the SAP database to persist session information. You require this only in extreme circumstances that Kony support recommends.
Enable XAI cross system support	Cross system services enable you to transparently invoke Kony for SAP functionality across multiple back end SAP systems, for example, ERP, CRM, and BW. You should never use this option without consulting Kony.
Ignore DOB / LDB workbench timestamp check	This system wide option disables the data object and LDB table workbench timestamp checking that makes sure that the definition on the application server is the same as in the central MEAP server. If there are any new field or changes detected, the application server automatically drops or converts the data. You may also use this option selectively at the individual DOB or LDB levels.
White list only provisioning	This option enforces that all Application Servers and gateways must have a provisioning white list entry or else they are automatically "blacklisted" and reset, that is, configuration and data is removed. Use this with caution.
UTC timezone controls	Use these configurations to define the UTC time zone for the MEAP Server, that is, its time offset from GMT. The configurations can either be a fixed value, or Kony for SAP can configure automatically, that is, derived. This MEAP server UTC offset is made available to functions to perform UTC type time calculations.

5.5.3.2 Reset Version Lock

If an application version is locked in the IDE workbench and if you forget the PIN, you may use this function to reset the lock, that is, remove the lock altogether. You may re-apply the lock in the IDE workbench if required. Select the **Remove version lock** option from the **Utilities** menu and the following pop-up appears:



Type in the application version and click the **Remove lock** button in the application tool bar.

5.5.3.3 Session Manager

This facility displays all connections to the host session manager. These are typically real-time host application calls or "post local host" calls, for example, host procedures. The host session manager is invoked using the **Workbench** > **Utilities** > **Host** >**Session Manager** menu option or transaction YVTS. A selection screen is used to restrict the display. By default, only active sessions are listed.

? Unknown Attachment

From this screen, you can perform the following:

- Display session attributes (double-click or display detail)
- Deactivate and re-activate sessions
- Set the session manager trace (refer to the session manager call trace section)
- Delete trace information
- Test host session manager calls

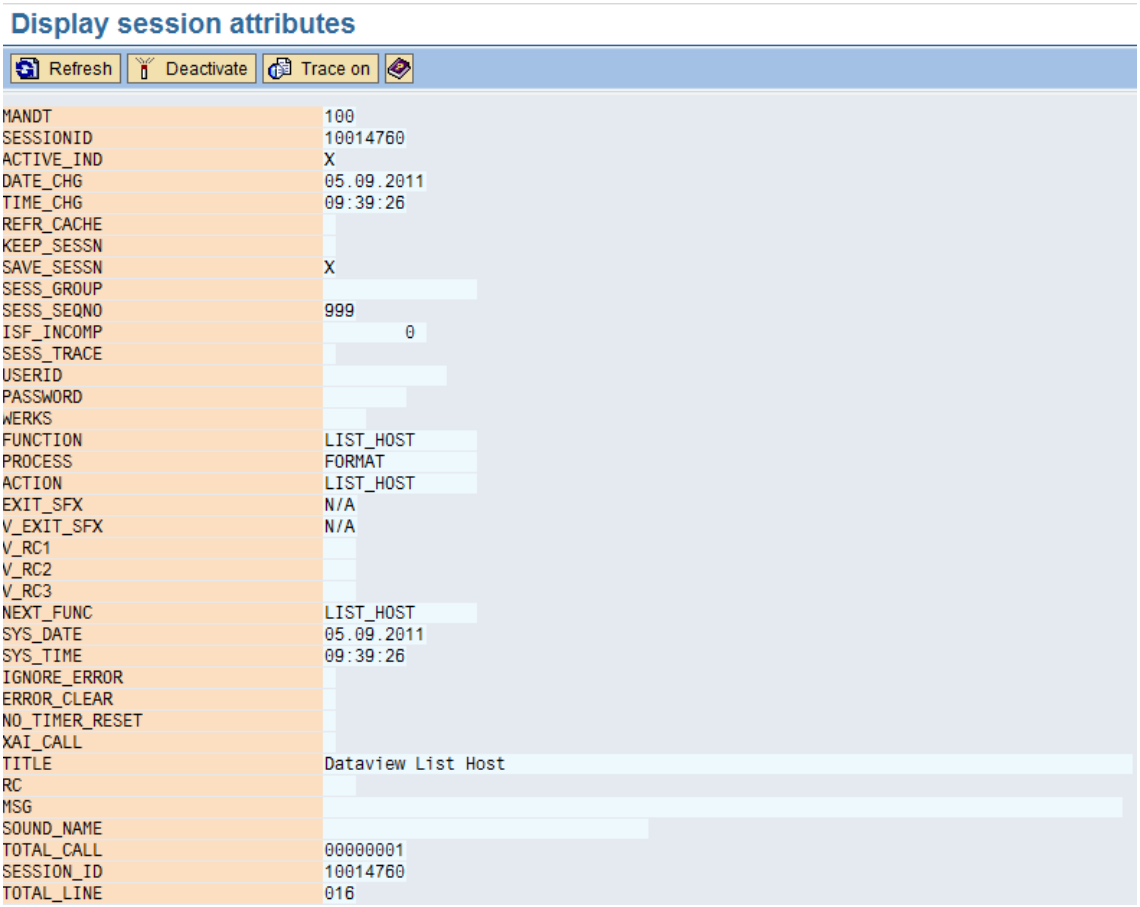
Test Session Manager Calls

This is a useful facility that enables you to replay session manager trace files, that is, help reproduce a problem. To do this, you must have already captured a session trace. Position your cursor on the session (with the trace) and select the test host call from the **Utilities** menu. The utility positions on the first call in the trace and then prompts you to issue the session manager call, shows the result, and prompts for the next call.

> *Important:* This utility is designed to be used by Kony for SAP support to diagnose internal problems. Do not attempt to use without first consulting Sky support.

Session Attribute List

When you display a session from the host session manager, all the attributes in the session header appear. These settings may be useful to Kony support to help diagnose session manager call issues.



Test Session Manager Calls

This is a useful facility that enables you to replay session manager trace files, that is, help reproduce a problem. To do this, you must have already captured a session trace. Position your cursor on the session (with the trace) and select the test host call from the **Utilities** menu. The utility positions on the first call in the trace and then prompts you to issue the session manager call, shows the result, prompts for the next call.

> ***Important:*** This utility is designed so that Sky support uses to diagnose internal problems. Do not attempt to use without first consulting Sky support.

Shutdown and Startup

You may shut down the SAP add-in components to prevent remote sessions and transactions from processing. This is useful during SAP backup windows or if serious problems occur with the SAP system. Both options are processed from the Administration Console (YVTI).

- Manual shut-down
- Start the Administration console (YVTI)
- Select the Shut-down SkyMobile option from the system management menu

> ***Note:*** Upon Shutdown, all heartbeat commands, LDB processing and internal system RFC calls are rejected. This has the same effect as the host not available. The Java Server and Gateway writes error messages to their respective system logs to this effect. Reconnection, error retry and store and forward configuration all come into effect.

Manual start-up

- Start the Administration console (YVTI)
- Select the Start-up SkyMobile option from the system management menu

Function Modules

You may also shut-down and re-start SkyMobile using SAP function modules. You may call these from ABAP or execute externally through RFC (remote function call).

Command	Description
/SKY/VTI_SHUTDOWN	Shut Down
/SKY/VTI_STARTUP	Start Up

5.5.3.4 Shutdown and Startup

You may shut down the SAP add-in components to prevent remote sessions and transactions from processing. This is useful during SAP backup windows or if serious problems occur with the SAP system. Both options are processed from the Administration Console (YVTI).

- Manual shut-down
- Start the Administration console (YVTI)
- Select the shut-down option from the system management menu

> ***Note:*** Upon Shutdown, all heartbeat commands, LDB processing and internal system RFC calls are rejected. This has the same effect as the host not available. The Java Server and Gateway writes error messages to their respective system logs to this effect. Reconnection, error retry and store and forward configuration all come into effect.

Manual start-up

- Start the Administration console (YVTI)
- Select the Start-up option from the system management menu

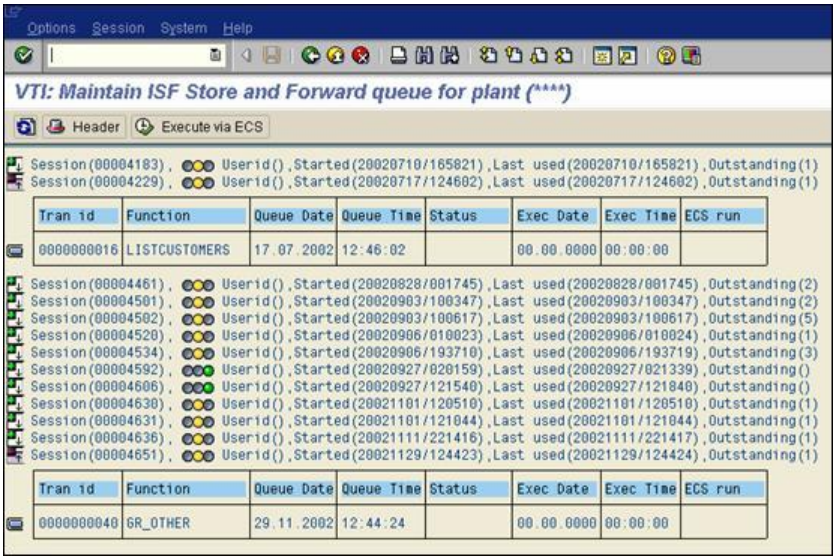
5.5.3.5 Function Modules

You may also shut-down and re-start Kony for SAP using SAP function modules. You may call these from ABAP or execute externally through RFC (remote function call).

Command	Description
/SKY/VTI_SHUTDOWN	Shut Down
/SKY/VTI_STARTUP	Start Up

5.5.3.6 Store and Forward Manager

The store and forward manager views the status of the MEAP store and forward queue. It is started from the **Workbench** > **Utilities** > **SAP Host** > **Store** and **forward** menu option, or through the YVTI system management console. The store and forward transactions are executed in SAP by the ECS Process VTI_ISF_TRAN. This is configured by the check-install process. The SAP session manager automatically starts this Process if 'manual start' is not configured on the application version. Otherwise, you need to review and process all outstanding transactions using this workbench.



This main list groups all store and forward requests by session ID. The traffic lights indicate the overall status of the requests and if expanded, each request in sequence.

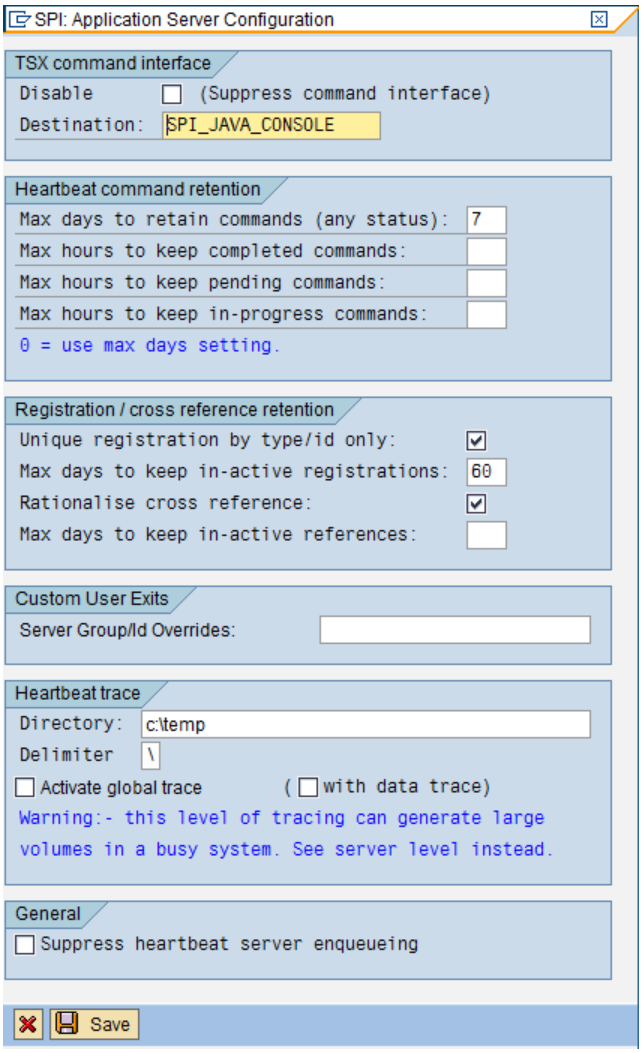
From this screen, administrators may:

- Display the transaction details
- Display processing errors
- Restart failed processing

5.5.4 Housekeeping

You should run the SAP add-in housekeeping routine daily to maintain the system. Housekeeping deletes unused sessions, system trace information and other internal tables. The retention periods are specified in the System Management Console configurations. The installation procedure automatically implements a ECS VTI_HOUSEKEEPING process and schedule that defaults to run each day at 6:00 AM. You may change this as required. The housekeeping program produces a report documenting what is analyzed and processed.

It is also worth noting that the Secure Container Console also has housekeeping configuration to control the retention of heartbeat registrations, and heartbeat commands. See the **Utilities** > **Configuration** option for more details. You should schedule the SPI_HOUSEKEEPING ECS process on a regular basis to keep the system clean.

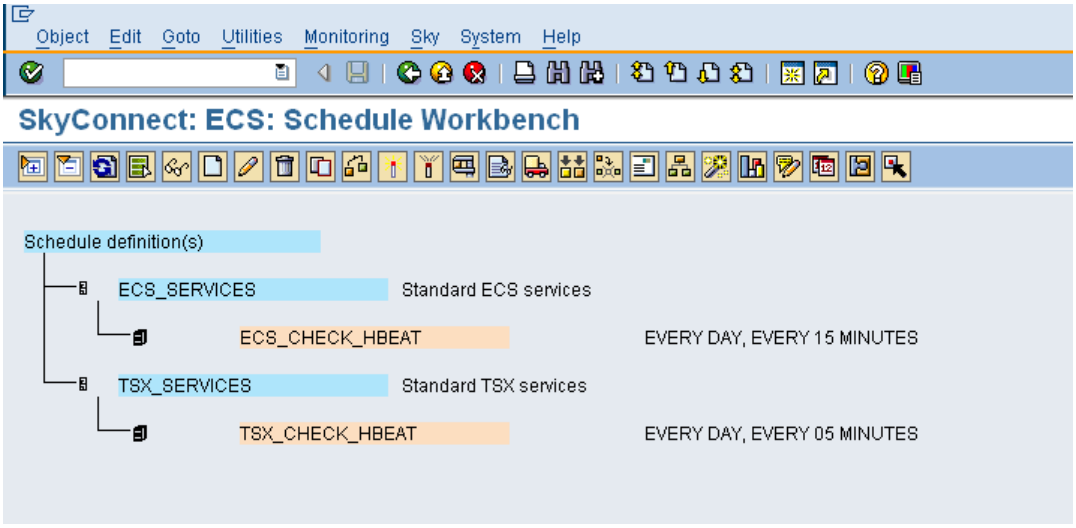


5.5.5 Automatically Monitoring Heartbeat Connections

You may create a scheduled task to automatically monitor Java Server connections to a particular SAP system. This runs as an SkyConnect ECS Schedule that monitors the heartbeats and reports any errors, that is, those connections that missed two or more consecutive heartbeats. The ECS Check install procedure should have already created the ECS ECS_CHECK_HBEAT process and schedule. If not, you need to create these manually (see Installation / ECS processes for details). You need to configure that Java servers to monitor through the configuration table /SKY/YECS_SRVMN. Maintain entries using SE16. You may specify wild card entries using "*", for example,

Server id	Server type
WM*	VTI
Mercury	VTI

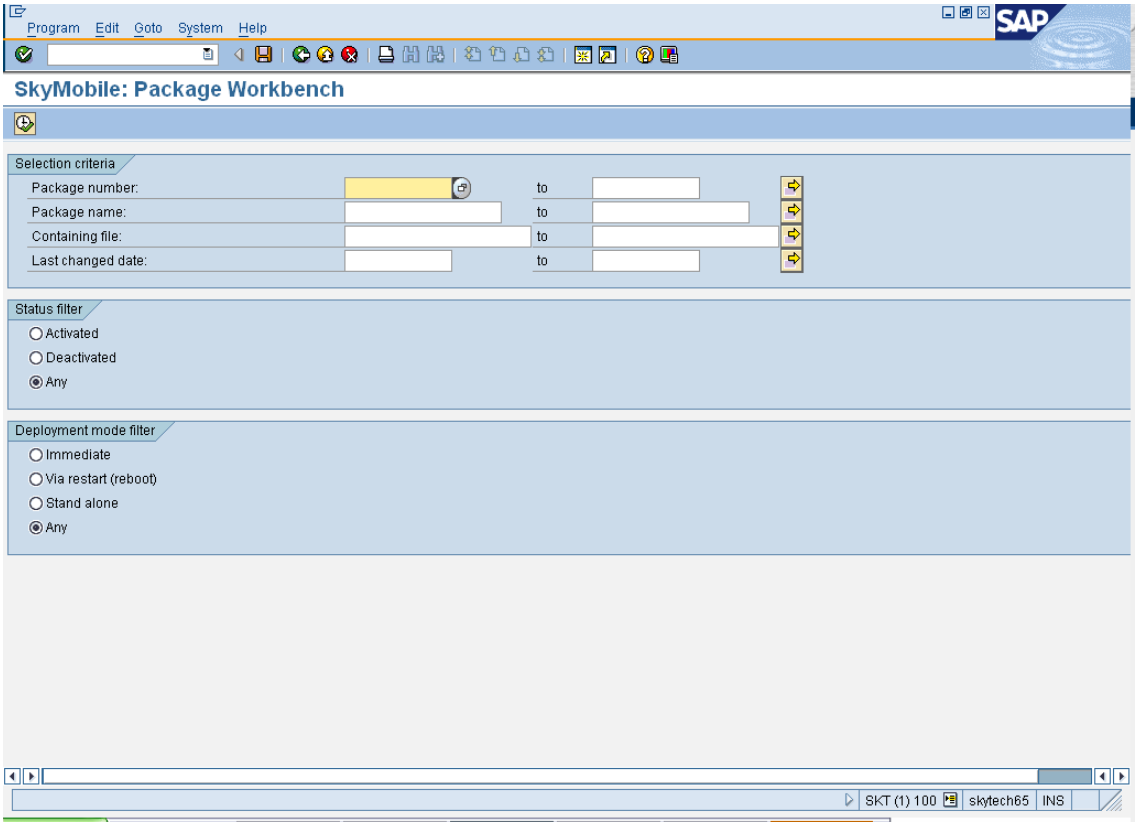
If a problem is detected, the ECS run fails and the reason is logged in the ECS annotation log. You may then use the ECS notification facility to automatically escalate potential problems.



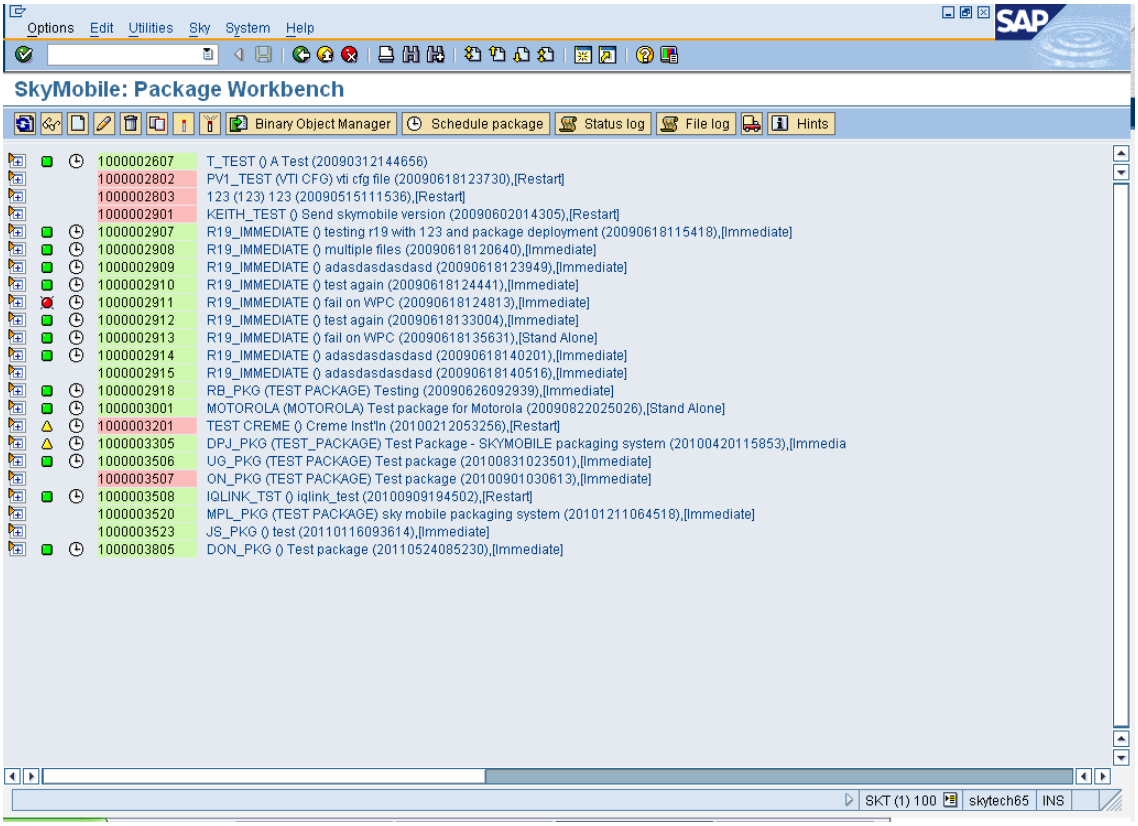
> ***Important:*** Schedule for this ECS process is not enabled by default. If you want to use this form of monitoring, you need to populate the /SKY/YECS_SRVMN table and then activate the schedule.

5.5.6 Remote Software Upgrades

Apart from automatic application refreshes that the workbench initiates, Kony for SAP has a software packaging tool that you may use to define components (files, binaries, and jar files) centrally in the MEAP Server. You may then schedule these updates to perform to a select group of Secure Containers that reference the unique individual server IDs and groups. You may start the package workbench directly using transaction YVTP, from the workbench through the **Utilities** > **Java Server** > **Package Workbench** menu or from the binary file manager through the **packaging** application toolbar option. Each package definition is uniquely identified by a number that is automatically assigned when the package is created. Refer SkyMobile Workbench for more information on the binary object manager and packaging.



The package manager selection screen allows you to restrict the list of packages that appear.



The main package workbench display lists all the existing package definitions, the files that are assigned and the current status. From the main display, you may create, change, display, delete, activate, deactivate and schedule packages.

5.5.6.1 The Concept

Packages are a mechanism to distribute binary objects to remote Java Servers in a controlled manner. Each package definition comprises of a header, uniquely identified by a number, and one or more associated binary objects to implement. Once the package is defined, it is activated and then you may schedule to one or more Java Servers. Active Java Servers regularly poll the SAP host for scheduled packages to implement. You may also use a push mechanism through the heartbeat command queue. Once the Java Server selects the package, the package is implemented and its status is regularly reported back to the SAP host. There are three server deployment strategies: immediate while the

server runs, the next time it starts (restarts) or stands alone while the server stops. An obvious consideration is to avoid objects (files) that are currently in use (locked) and cannot replace. When an error occurs with an implementation, the Java Server automatically "rolls back" the updates to the last stable point, where it may re-apply again, manually fixed or aborted.

5.5.6.2 Binary Objects

You must have uploaded all binary objects into the Binary Repository. Each object is assigned a logical location that determines where you need to implement the object, that is, directory, `SDCARD`. You may also flag a binary object (file) as redundant in which case it is physically deleted if it exists. You may implement any type of object from application Java packages, graphics, configuration files, and internal software.

5.5.6.3 Synchronizing Applications and using Heartbeat Commands

Once you successfully implement the binary objects, it is often required to synchronize application changes and perform other tasks. There are options in the package definition to synchronize the existing applications and execute a command list (heartbeat commands). Commands are useful to change configuration options and perform other low level tasks on the server. It is important to note that these are performed "after" any binary objects are implemented and the server "restarted" (if applicable), thus it may be necessary to restart the server again.

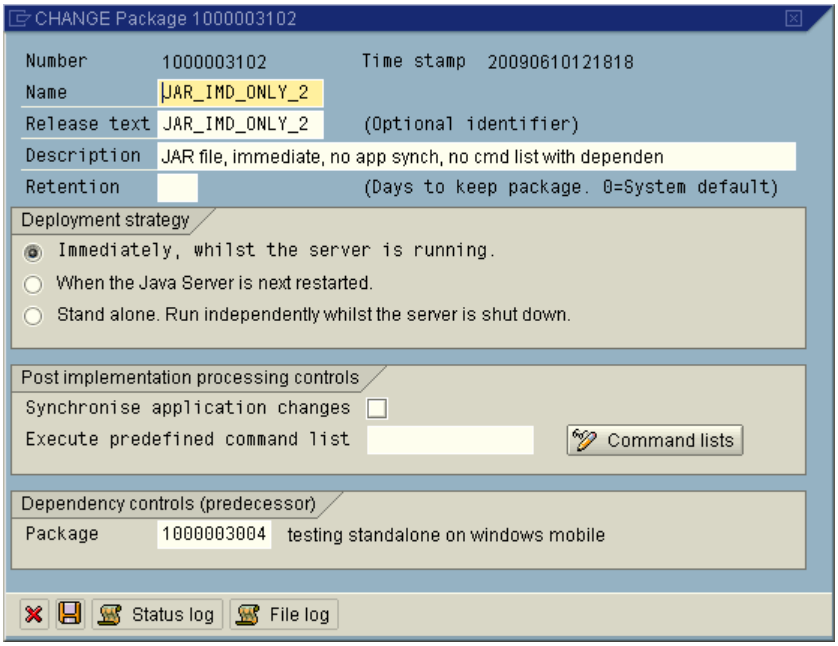
5.5.6.4 Scheduling Changes

Once you define the package definition and activate, you may schedule it for one or more Java Servers. The package schedule manager is used to maintain, manage and monitor all package schedule entries. For more information on the scheduling process, refer to the schedule workbench hints and tips. You cannot maintain the package definition whilst schedule entries exist. You must remove them first.

5.5.6.5 Dependencies

You may define a simple dependency chain, that is, only allow a package to be "scheduled" for implementation when another package is completed. The dependent schedule entry remains in a status of **queued** until the system changes to **scheduled** automatically.

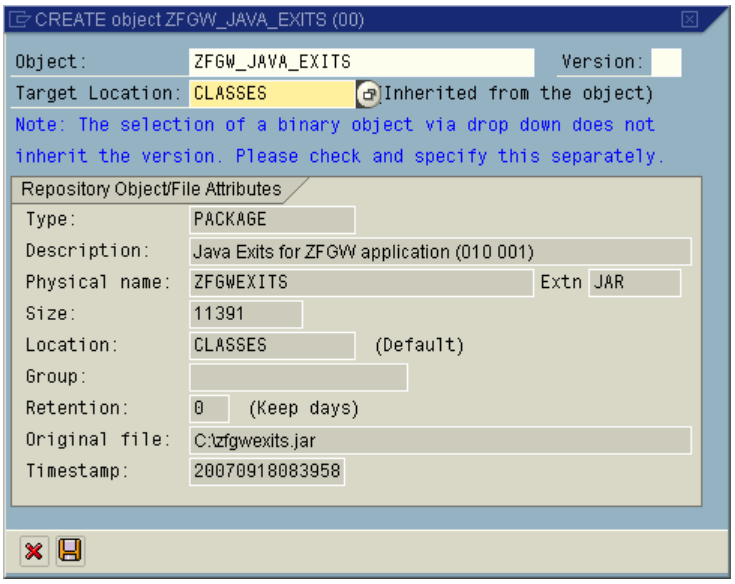
5.5.6.6 Binary Package Definition



You give a package a unique name. The release text and description are used to further identify the package. The deployment strategy determines how to implement the package, that is, while the server is running, upon server restart or through the "stand alone" utility. The one to use depends on what binaries are updated, that is, dependencies / files in use. You may also configure application synchronization and / or a command list to execute once the binary objects are successfully implemented, that is, in a status of complete.

Maintaining the File List

Once you create the package definition, you now associate one or more binary objects with it. To do this, you expand the definition, position your cursor in the file section and click create.

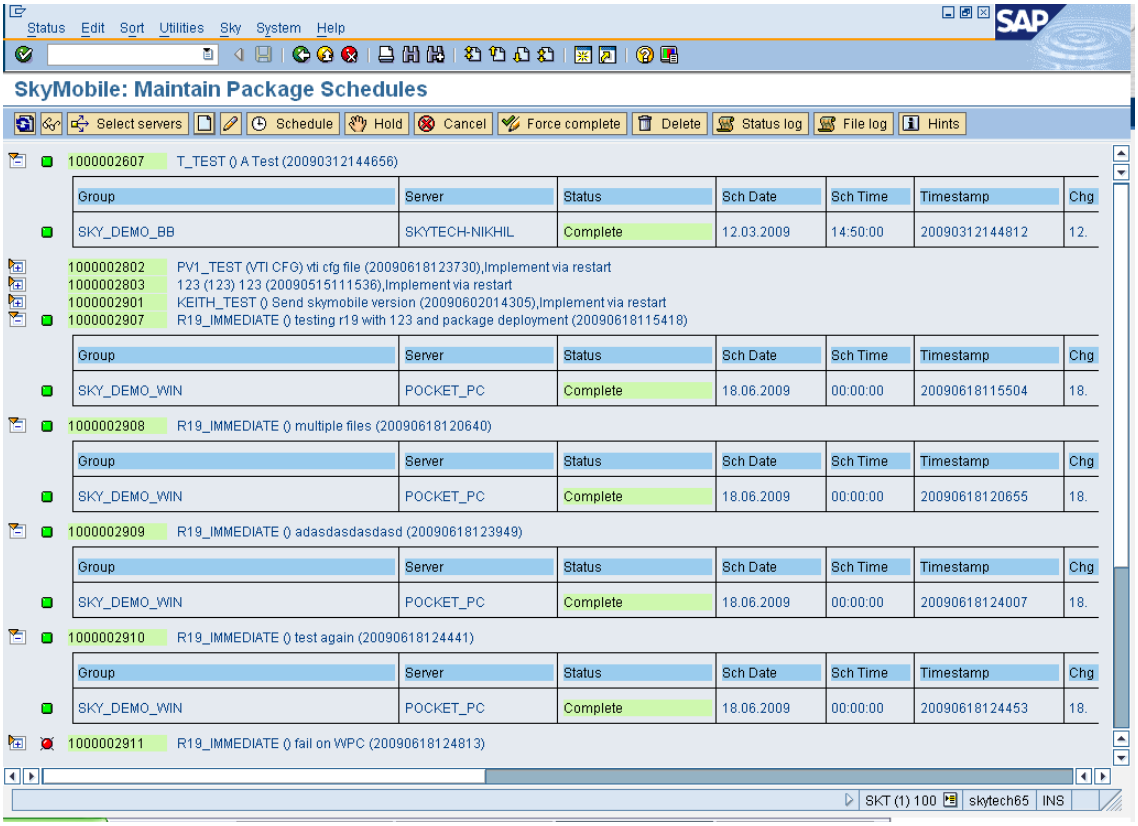


You specify the logical name of the file definition in the binary object manager. The version number is used to control a specific version of the file (multiple versioning is supported). The target location usually defaults. This indicates the physical location that the file is going to be copied to. The drop-down lists the valid entries.

5.5.6.7 Scheduling the Package Definition

Once the package and its attributes are defined, it is activated. This indicates that it is ready to schedule. Position your cursor on the package and click the schedule icon in the application toolbar. The package schedule maintenance screen is invoked. This utility is used to specify all the Java Servers that should receive the package update.

As well as scheduling packages to specific Java Servers, you may configure generic schedules that get automatically scheduled when the server connects and satisfies the wild card server ID/group criteria.



The Concept

Package schedules are used to distribute binary objects (files) to remote Java Servers. You may also request that the applications be synchronized and/or a heartbeat command list is executed after the binary objects are updated. Once the package definition is created and activated, you may then schedule it for distribution. There are two types of schedule entry, namely a specific entry for a named Java server group/ID and a generic schedule entry that dynamically generates specific schedule entries for servers that qualify. There is also a utility to "mass add" specific schedule entries to avoid tedious entry.

You may deliver the package in two modes: immediately while the server is still running or when the server is next re-started (recommended). This is specified at the package level (see package hints & tips).

How the Java Server Package Scheduling Works

The Java Server regularly polls the SAP host to select any scheduled, canceled or deleted packages. If one or more package definitions are returned, the Java server updates the status to *selected* and then starts downloading the binary objects from SAP. Once all the associated objects are downloaded, the package status is updated to *downloaded* and is ready to implement. The deployment strategy; immediate, through restart or stand alone dictates when the package is processed. Immediate should only be used for "on the fly" changes that have no run time dependencies, for example print templates, maps, so on. "Via restart" processes the package when the server is next restarted and "stand alone" is processed when the server is shut down (stopped). Once the package is processed, its status is updated to either *complete* or *error*. If the implementation fails, all changes are backed out, saved versions are restored if a cancel or delete is issued from the package workbench, then the local package definition and all associated binary objects are deleted from the server. If a application synchronization and/or command list is configured, these are only processed once the binary objects are successfully implemented. If you perform configuration file updates, you may need to restart the server again for these to take effect. You may view the log information and package details from the server web status page and/or SAP package workbench.

SAP Workbench Package Process

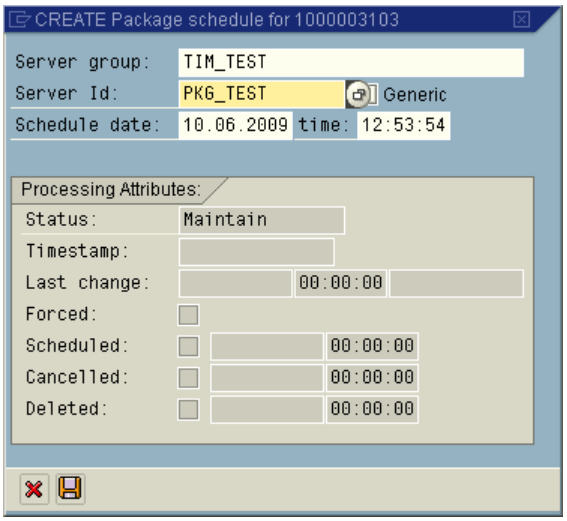
You may only create schedule definitions if the package definition is activated. When you create a schedule entry, it is in a of *maintained* status. When you schedule the entry, its status changes to *scheduled* and can now be selected by any qualifying Java Server. The Java server then updates the status (as per above) and you can monitor this in the workbench. If a package implementation fails or is corrupted in some way, you may remove the entire package definition using the *CANCEL/DELETE* options. Once the Java Server processes a *cancel* , the status of the schedule entry is reset to *stopped*. Instead of creating a schedule entry for every server, you may create a generic schedule definition, if a Java Server qualifies for the wild card group/ID, a specific schedule entry is automatically generated by the system for the server. In addition, you can dynamically select and build schedule entries from a list using the "selected servers" functionality. The workbench checks and controls the status workflow; the allowed sequence of processing as follows:

Status Summary

M Maintained (definition currently being edited)
H Held (temporarily held)
Q Queued (waiting on another package dependency to successfully implement)
S Scheduled (available for download)
G Selected (The package is selected and the data is downloaded)
D Downloaded (the package is fully downloaded but not implemented)
R Restart required (the package is implemented when the server restarts)
E Error (the download or implementation failed, check server log)
B Cancel and backout the package implementation on the server
X Stop (the package implementation is successfully backed out)
C Complete (implemented successfully or force completed)
Z Logically deleted (physically deleted based on retention period)

Manually Maintaining Schedule Entries

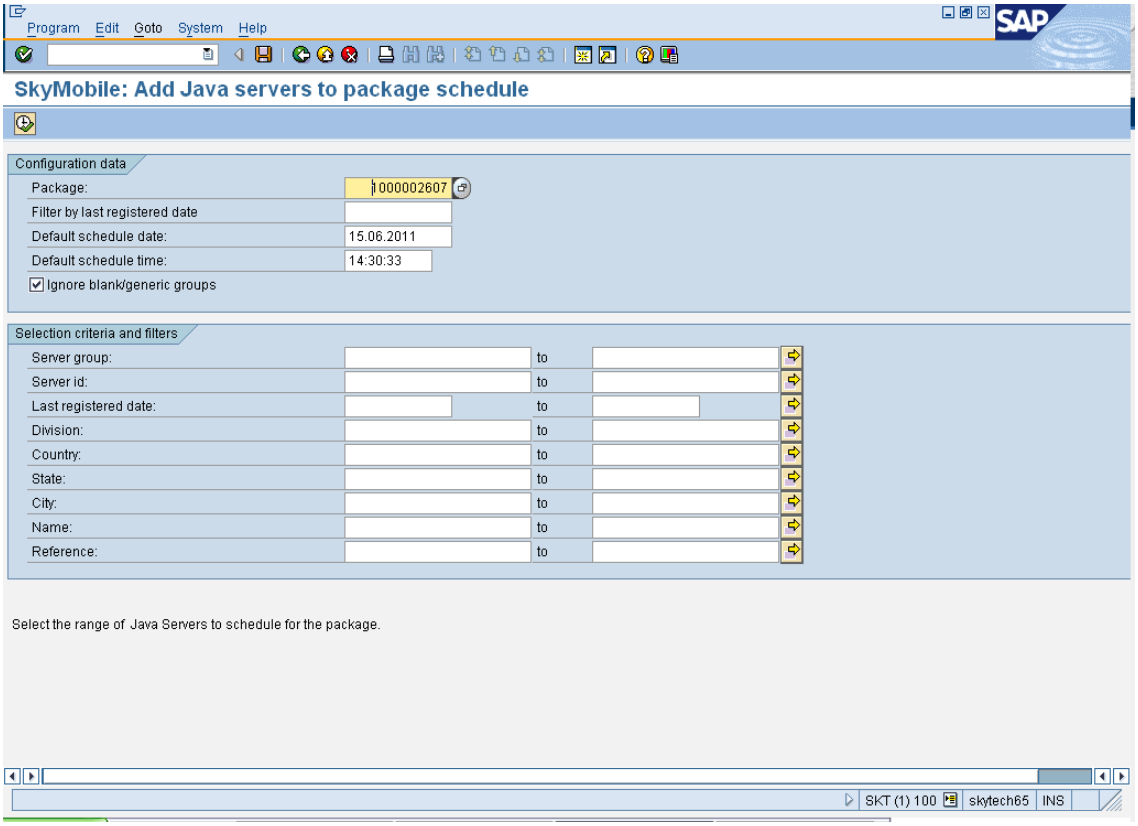
You may create the schedule definition manually using the toolbar options. In this case, the following pop-up appears:



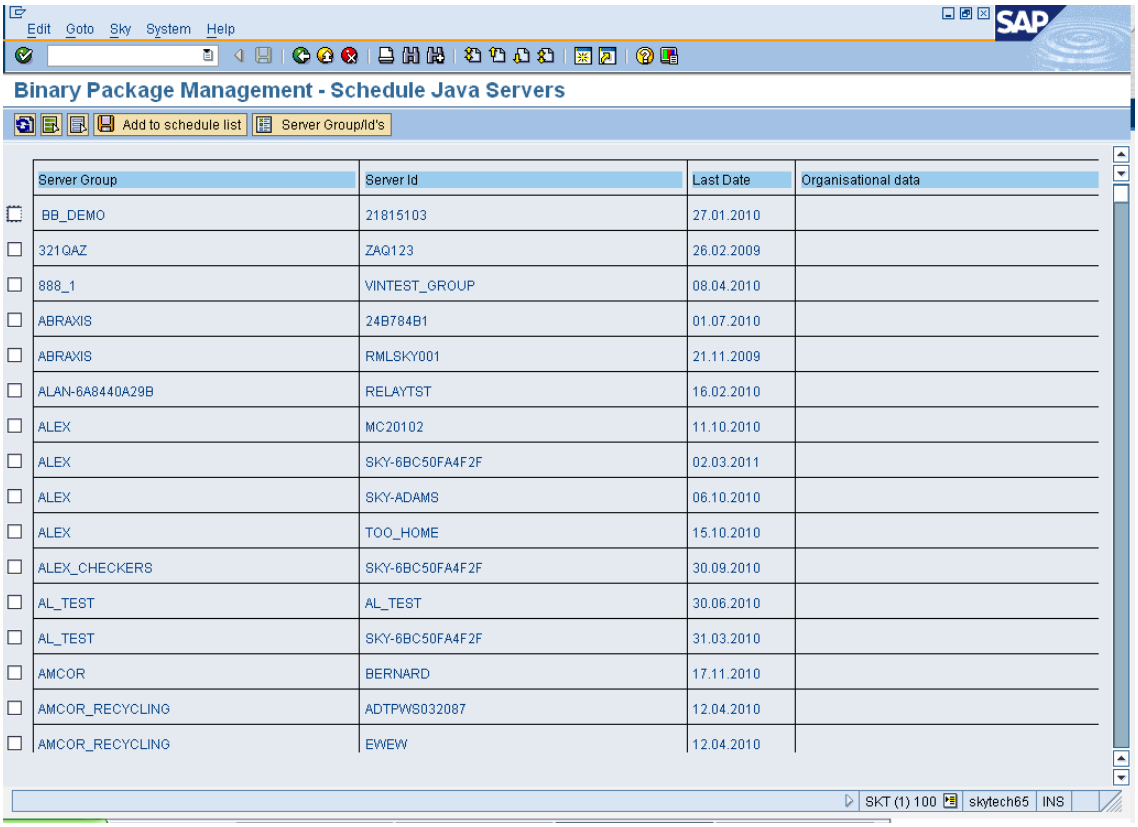
You specify the target Java server to receive he package. The server should have already registered itself with SAP.

Automatically Generating Schedule Entries

It can become very tedious to add large numbers of servers that receive a package update. To select multiple servers to add to the list, position your cursor in the list section for the package and click the **select servers** button on the application toolbar. The following selection screen is appears. This selection is used to restrict the list of available servers from which to select.



Specify the selection criteria and the following selection list appears:



If the server is already in the schedule list, no selection check box appears; otherwise you select the servers to add to the schedule list.

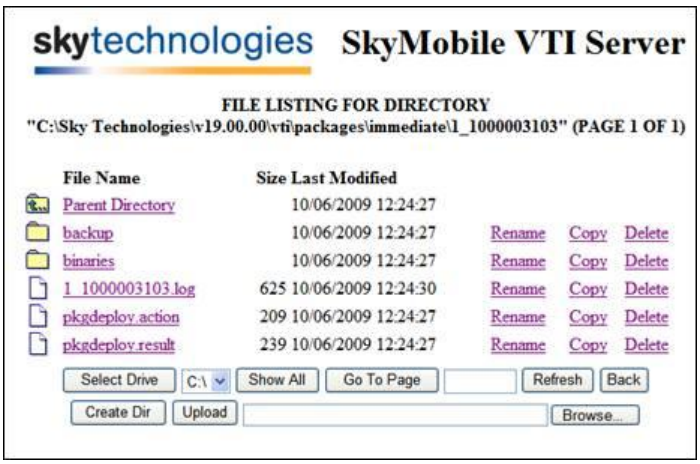
5.5.6.8 How is the Package Updated?

Each Java Server periodically checks for package updates. When one is selected, it then requests the list of associated files. These files are then downloaded from the SAP binary file manager onto the device. They are stored in the package directory location on the device. The package is then implemented depending on the deployment strategy, that is, immediately, through server restart or "stand alone" utility with the server down.

Troubleshooting

If any problem occurs, for example, file in use, the package is backed out and the package definition is flagged as in an error status. All processing is logged and you may view the package log from the server web status page. The log contains the reason for the failure.

All package details are stored locally in the package directory on the device. In extreme cases, you may maintain the binary objects and / or `pkgdeploy.action` file manually using the web status page file explorer (or something similar). However, you are recommended that the failed package is flagged as deleted and re-scheduled to address the error.



Each unique package directory entry contains the following:

- The repository of binary objects (binaries)
- A backup of any existing objects (backup)
- The processing log (.log)
- The action list (pkgdeploy.action) is the implementation script, that is, what you need to do
- The result file, that is, what happened. This is analyzed from any errors.

Housekeeping

Scheduled packages are only physically removed from the Java Server when they are in a status of 'deleted' or 'canceled'.

You may allocate each package definition a retention period (keep days) that is used to automatically remove activated package definitions with no outstanding schedule entries, that is, must all be completed or logically deleted.

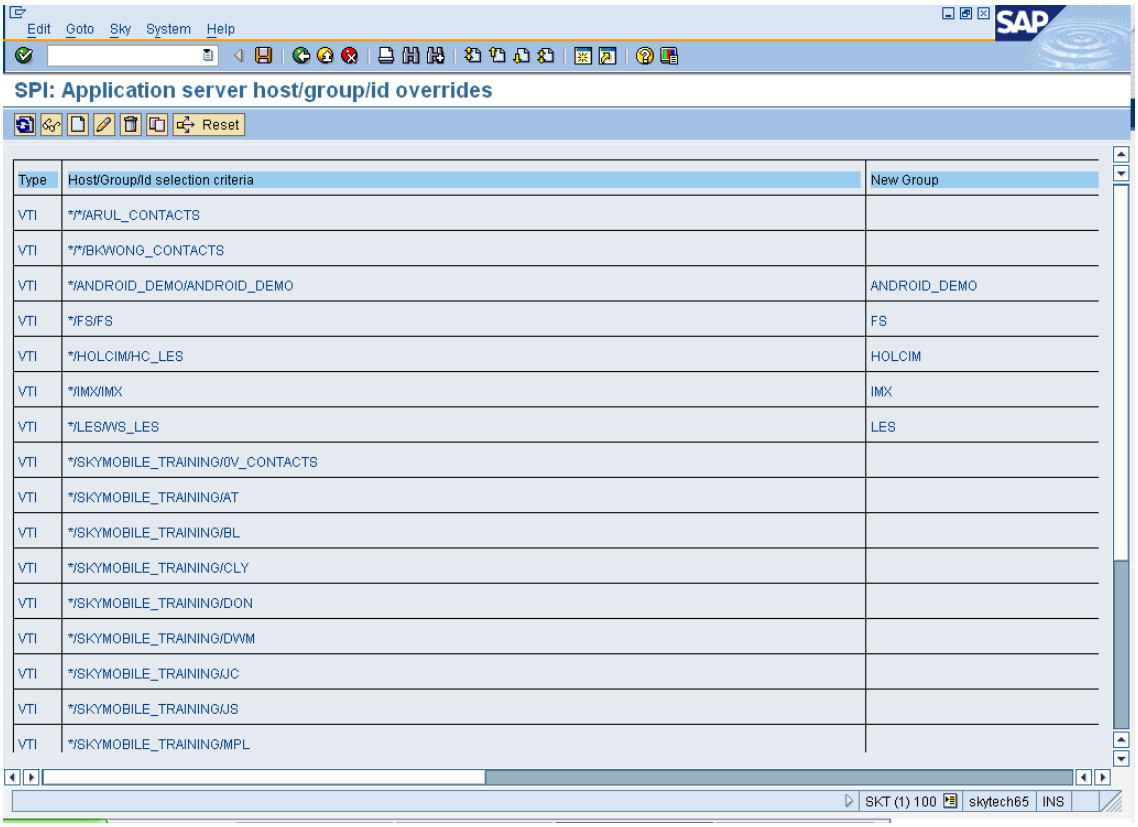
Logically deleted package schedules are automatically removed after 30 days.

5.5.7 Overriding the Server Group and ID

In cases where you deploy a generic configuration file (`skymobile.cfg`), you may use this facility to automatically assign a unique server group and /or Id, when the Secure Container connects to the SAP host for the very first time through its heartbeat. The obvious advantage of this is that you can package a generic configuration file (`skymobile.cfg`) that has no dependency on the unique server group and ID. The concept is that each device that Kony for SAP runs on has a unique host identifier or device ID. The Secure Container defaults this as the server ID, if one is not explicitly specified in `skymobile.cfg`. A cross reference is then maintained in the SAP host that translates this input host name to another server group / ID to use.

5.5.7.1 Maintaining Host Overrides

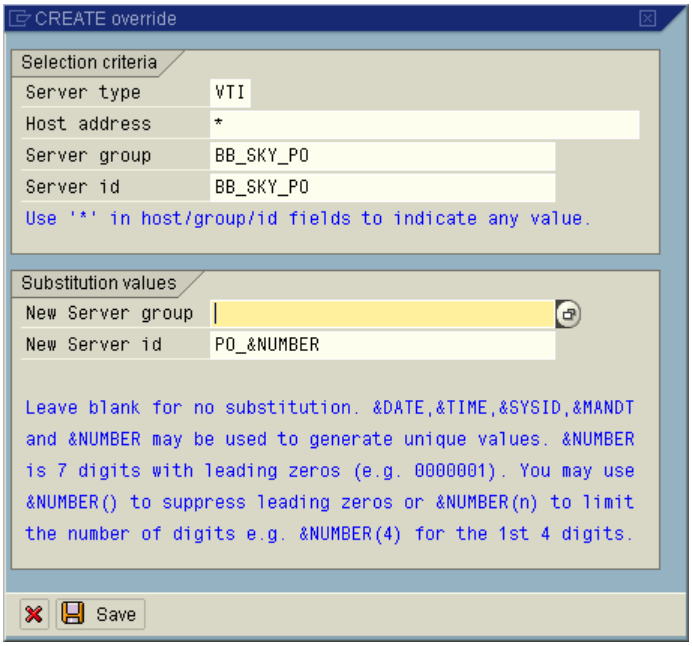
The host/group/ID overrides are maintained in SAP through the **Utilities** > **Java Server** > **Host/Group/ID** overrides menu option from the workbench. You may use a selection screen to restrict the list. The following list appears. The list is essentially a translation table. From this screen you may create, change, delete and copy override definitions.



> ***Note:*** The override is only performed once for a Secure Container on its first heartbeat call to the host, after which it is flagged as processed. If you wish to re-activate the rule, then position your cursor on the rule and click the **reset** button on the application toolbar.

5.5.7.2 Creating an Override Rule

Click the **create** icon in the application toolbar. If you wish to inherit the attributes of an existing definition, position your cursor on the definition and then click the create icon. The following pop-up appears:



The first group of criteria (Selection criteria) determines the match, that is, the incoming criteria to scan for. This may be specific or generic. A '*' is configured for generic values (anything). You leave the "new" criteria blank to ignore any substitution. You may automatically generate unique server IDs by configuring substitution variables in the new name as documented in the text of the pop-up.

You may also implement a custom user exit to generate your own group and ID values. This is configured using the Container Management Console. The custom exit is called after all substitution values are processed and only if an override rule is selected for processing.

5.6 Secure Container Management

5.6.1 Application Server Management

You may deploy containers over a variety of devices and platforms, however the management and configuration of the servers is the same. This section provides information on how the containers communicate and register themselves with the back end host, may manage through their web status pages, the different port configurations, basic operations (start, and stop), back up / recovery mechanisms. For details on the configuration file options, and installation refer to the Installation Guide for details.

> ***Note:*** You can automatically provision Secure Containers from the central Server Profiler. Click here for more details.

5.6.2 Key Topics

The Heartbeat Processing
The Web Status Page
Container Management Console
Command Ports
Relay Ports
Operation of Containers and Gateways
Backup and Recovery (Windows only)
Gateway Switchover

5.6.3 The Heartbeat Concept

Every application server automatically reports its status back to the MEAP server system using a 'heartbeat' call. The heartbeat has a number of functions:

- Indicates the last known connection status to the SAP Host
- Determines if real time functionality is possible (local connection status)
- Enables commands to be 'pushed' down to the application server
- Shares system information between the application server and MEAP server

The attributes and frequency of the heartbeat is specified in the application server configuration file. Refer the configuration section for more information.

Heartbeat

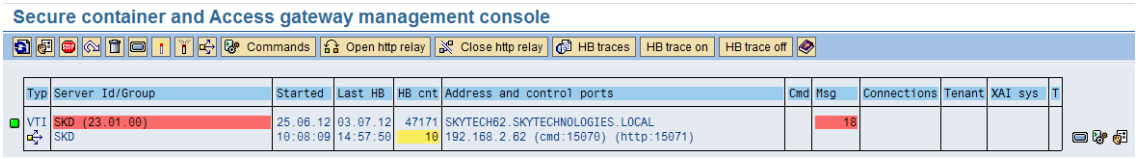
```
SERVER.HEARTBEAT.HEARTBEATCONNECTEDINTERVAL = 180
SERVER.HEARTBEAT.HEARTBEATINTERVAL = 60
SERVER.HEARTBEAT.HEARTBEATLOSSSEVERITY = I
SERVER.HEARTBEAT.HEARTBEATHOSTNAME =
SERVER.HEARTBEAT.HEARTBEATHOSTADDRESS =
```

The HeartbeatInterval and HeartbeatConnectedInterval options determine the frequency of the heartbeat calls. The concept is that once a heartbeat is successfully established, a "lazy heartbeat" (HeartbeatConnectedInterval) kicks in. If an interruption occurs, that is, host interface is not available, the system resorts back to the original HeartbeatInterval.

> ***Note:*** A call every 15 seconds can be excessive. Sky recommends setting the HeartbeatInterval to 60 seconds (one minute) and HeartbeatConnectedInterval to 180 seconds (three minutes). See the section on tuning and optimization for more details.

5.6.4 The Web Status Page

Use the Web status page to connect to any Access Gateway or Secure Container on the network and display internal information and its status. In addition, options are available to stop, restart and manage local databases. The Web status page is the primary mechanism to manage gateway and container instances. The most common way to invoke the web status page is directly from the Secure Container and Gateway Management Console (YECJ) in SAP. The use of relay port functionality is discussed later in this section. You may secure the web status page and restrict / allow certain functionality, refer The main page and configuration.



5.6.4.1 Launching the Web Status Page from SAP

You perform this through the SAP Secure Container and Gateway Management Console. From the list of selected servers, click the icon adjacent to the server or position your cursor on the server and click the icon in the application toolbar. If the remote servers IP address is NAT'ed by a firewall or router than you must configure a relay port and open to enable a remote connection.

5.6.4.2 Connecting to a Gateway or Container Directly from a Browser:

To connect to a specific Gateway or Container instance, you need to know the IP address (or DNS name) where the server executes, and its HTTP control port that is configured in the VTI Java configuration (`skymobile.cfg`) file. If the server executes on the same computer as the browser, you may use 'localhost' as the IP address.

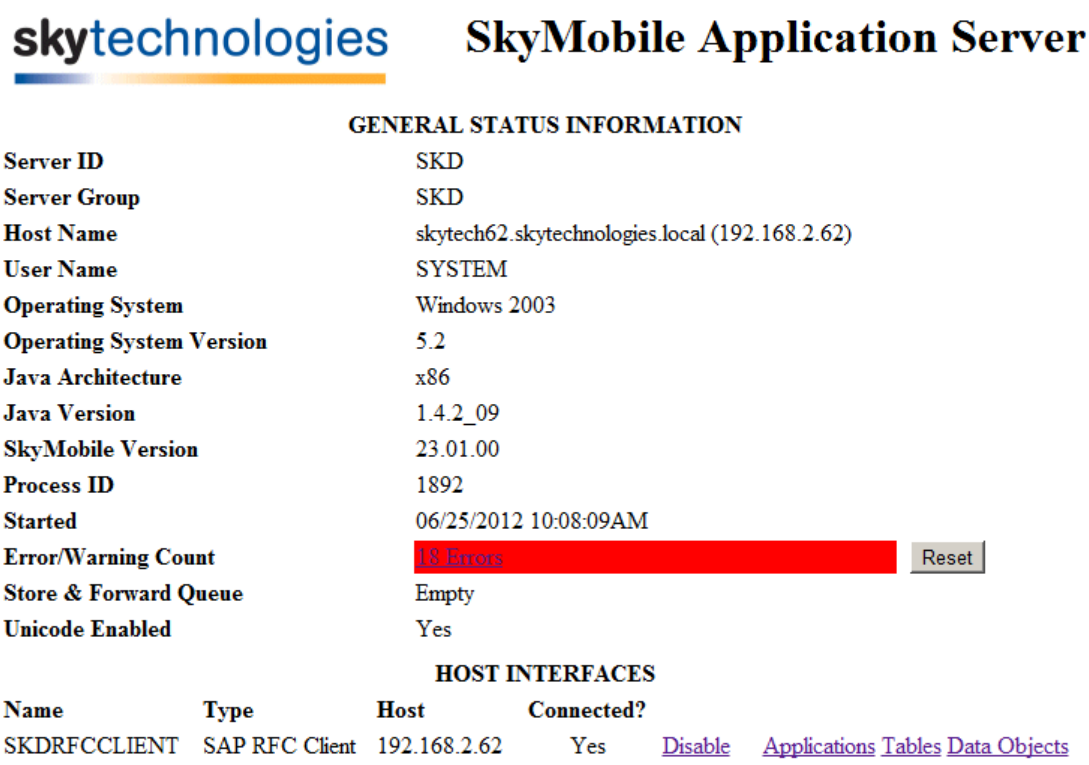
> ***Note:*** If you don't know the TCP/IP address of the target gateway or container, you may use the SAP YECJ transaction to find out (see the SAP Secure Container and Gateway manager documentation for details).

Start the Web browser and issue the following command:

Command

```
Direct:http://{IP address}:{http port}
Relay:http://{Gateway IP address}:{http port}/?relayTarget={type}%7C
{group}%7C{id)
```

Once a connection is established, the following main page appears:



5.6.4.3 The Main Page and Configuration

The main status page displays the system information, host connection status and the status of the ports that the Secure Container manages. Navigation is by the usual button and hyperlink mechanisms. Some customization of the layout, functionality and security is possible in the configuration file against the http control port definition.

Example HTTP Control Port Configuration

```
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWCONFIGCHANGESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWFILEENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWFILEUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.ALLOWLDBUPDATESFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT15071.AUTHORISATIONPASSWORD =
SERVER.HTTPCONTROLPORT:PORT15071.AUTHORISATIONUSERNAME =
SERVER.HTTPCONTROLPORT:PORT15071.HTMLBUTTONLOCATION =
SERVER.HTTPCONTROLPORT:PORT15071.HTMLPAGESIZE =
SERVER.HTTPCONTROLPORT:PORT15071.PORT = 15071
SERVER.HTTPCONTROLPORT:PORT15071.USEHTMLCOMPRESSION =
```

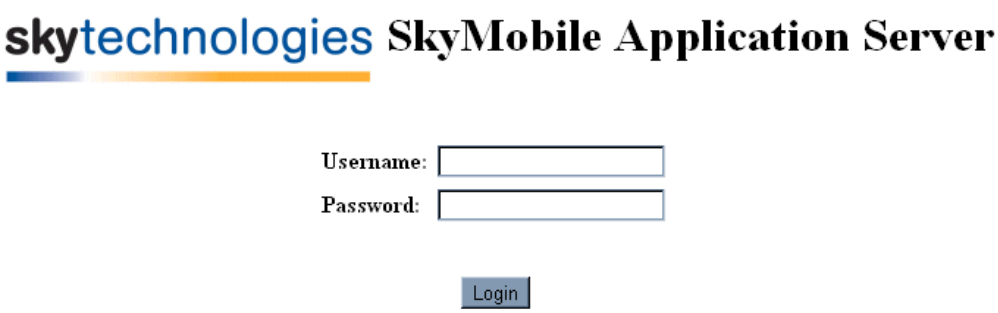
Refer the configuration section for more information on these options and their defaults.

The following functions are available

- Refresh the main display
- View / modify the configuration
- View / download the system log
- Remotely manage files on the host machine
- View the available applications
- View and maintain data objects and local database tables

- View and / or disconnect current connections on a given port
- Connect to a given port
- Restart / Stop the server
- Enable / disable host / gateway connections

> ***Note:*** In Microsoft IE, you may make the text size (font) smaller using the **View** >**Text size**. You may secure the status page using a user ID and password defined in the Secure Container configuration skymobile.cfg). Refer Configuration options for details. If this is done, a logon prompt is initiated. This is recommended to prevent unauthorized access.



5.6.4.4 Viewing and Maintaining the Configuration File

To view the configuration file, click the **Config** button on the main screen. The following screen appears.

> ***Note:*** Note the unencrypted/encrypted mode.

If configured, you can modify the configuration file and then save from this screen. You need to restart the server before any changes made take effect.

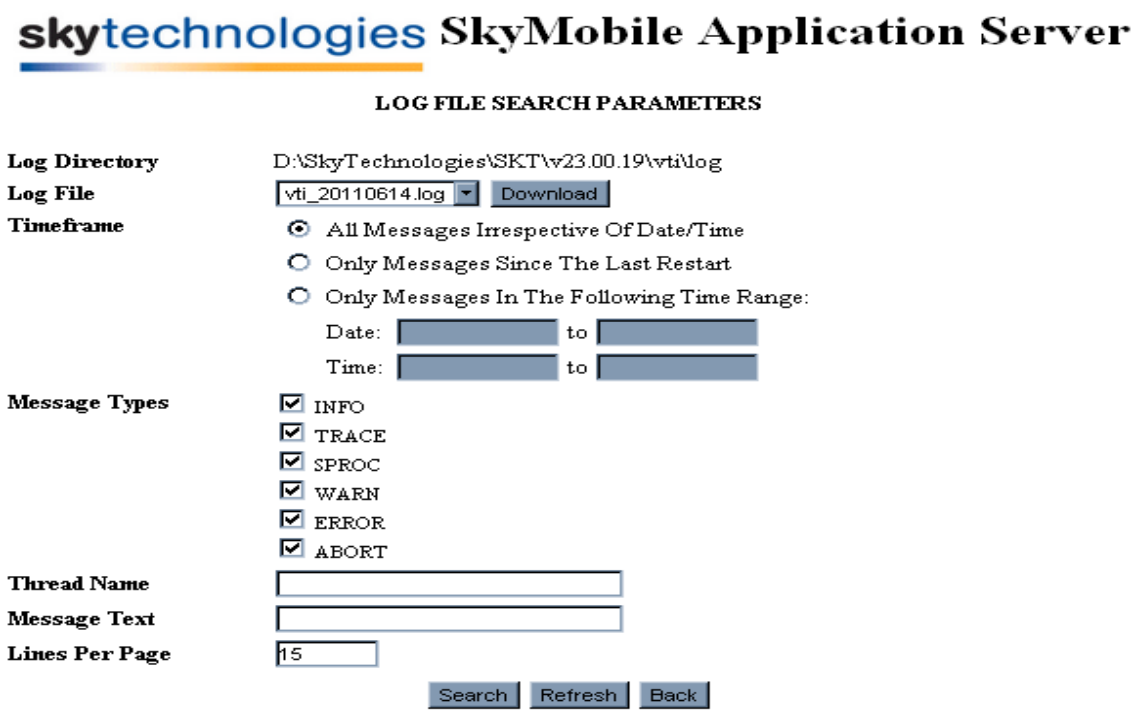
Encryption

As a security option, you may automatically encrypt the configuration file using the **Save Encrypted** button. The system automatically detects encrypted configuration files and so you may effectively deploy them with the application. You may deploy already encrypted configuration files to other servers as part of the application. Encryption allows to secure sensitive connection information and prevents unauthorized tampering.

5.6.4.5 Viewing the Log

Click the **Log** button from the main screen and the log search screen appears.

> ***Note:*** You may observe the download option to copy the specified log on your desktop.



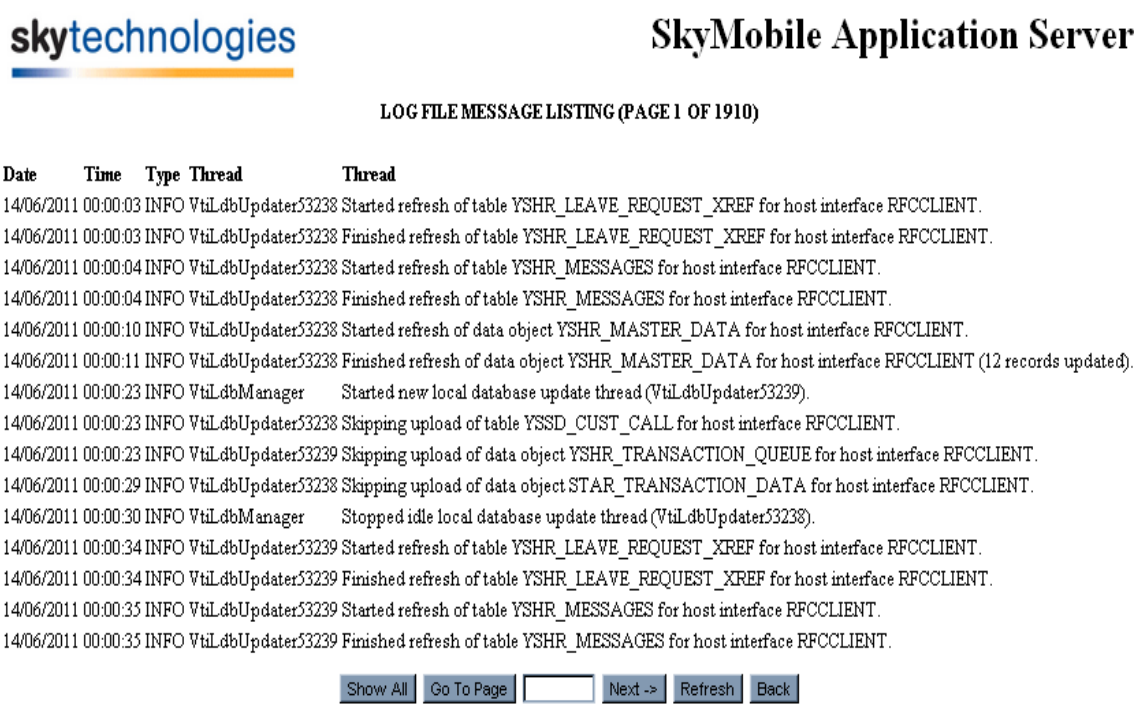
The screen permits selection of the log messages to appear based on a range of possible criteria. It is possible to go back through previous log files by selecting any of the available entries from the **Log File** dropdown. The SERVER.LOG section in the configuration file determines how the logs are managed.

Log

```
SERVER.LOG.EXCLUDELOGMESSAGESEVERITIES =
SERVER.LOG.EXTRALOGGINGOPTIONS =
SERVER.LOG.LOGFILEDIRECTORY = ./log
SERVER.LOG.LOGSTRATEGY = P
```

```
SERVER.LOG.MAXIMUMNUMBERLOGFILES = 5
SERVER.LOG.SHOWMILLISECONDS = false
SERVER.LOG.STOREDPROCDUMP = false
SERVER.LOG.STOREDPROCLOGGING = false
```

If you click the **Search** button, all the qualifying log messages appear:



skytechnologies

SkyMobile Application Server

LOG FILE MESSAGE LISTING (PAGE 1 OF 1910)

Date	Time	Type	Thread	Thread
14/06/2011	00:00:03	INFO	VtiLdbUpdater53238	Started refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
14/06/2011	00:00:03	INFO	VtiLdbUpdater53238	Finished refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
14/06/2011	00:00:04	INFO	VtiLdbUpdater53238	Started refresh of table YSHR_MESSAGES for host interface RFCCLIENT.
14/06/2011	00:00:04	INFO	VtiLdbUpdater53238	Finished refresh of table YSHR_MESSAGES for host interface RFCCLIENT.
14/06/2011	00:00:10	INFO	VtiLdbUpdater53238	Started refresh of data object YSHR_MASTER_DATA for host interface RFCCLIENT.
14/06/2011	00:00:11	INFO	VtiLdbUpdater53238	Finished refresh of data object YSHR_MASTER_DATA for host interface RFCCLIENT (12 records updated).
14/06/2011	00:00:23	INFO	VtiLdbManager	Started new local database update thread (VtiLdbUpdater53239).
14/06/2011	00:00:23	INFO	VtiLdbUpdater53238	Skipping upload of table YSSD_CUST_CALL for host interface RFCCLIENT.
14/06/2011	00:00:23	INFO	VtiLdbUpdater53239	Skipping upload of data object YSHR_TRANSACTION_QUEUE for host interface RFCCLIENT.
14/06/2011	00:00:29	INFO	VtiLdbUpdater53238	Skipping upload of data object STAR_TRANSACTION_DATA for host interface RFCCLIENT.
14/06/2011	00:00:30	INFO	VtiLdbManager	Stopped idle local database update thread (VtiLdbUpdater53238).
14/06/2011	00:00:34	INFO	VtiLdbUpdater53239	Started refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
14/06/2011	00:00:34	INFO	VtiLdbUpdater53239	Finished refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
14/06/2011	00:00:35	INFO	VtiLdbUpdater53239	Started refresh of table YSHR_MESSAGES for host interface RFCCLIENT.
14/06/2011	00:00:35	INFO	VtiLdbUpdater53239	Finished refresh of table YSHR_MESSAGES for host interface RFCCLIENT.

Show All | Go To Page | Next -> | Refresh | Back

> ***Note:*** You can efficiently skip to a specific page or to the end using the **Go To Page** button. You may also alter the number of lines that appear (global setting) in the http control port configuration. The log download option is useful to analyze the large log files.

5.6.4.6 Managing Files Remotely

In order to make use of remote file management facility, you must have the appropriate configuration in place. Refer Configuration options for further information.

Once activated, click the **Files** button on the main status screen to access the remote file management features of the Java Server. Upon doing so, the following screen appears.

From this screen, you can:

- Navigate up and down to different directory level
- Select different drives

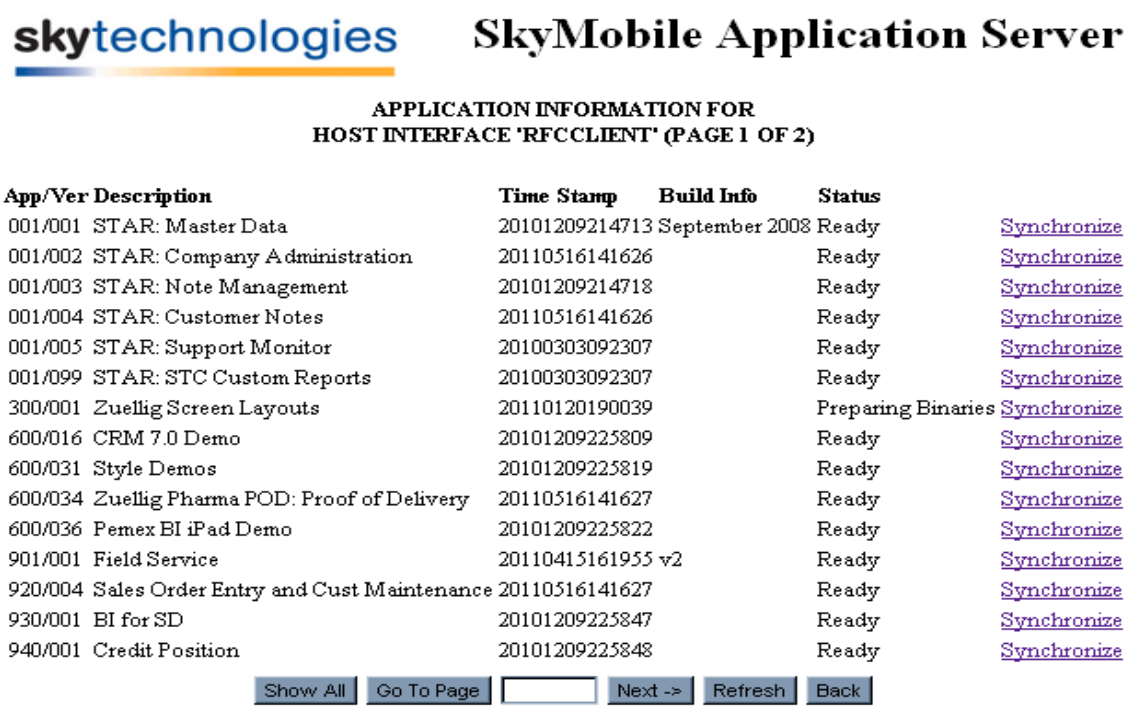
- Open a file to read or edit (depending on type)
- Rename files / directories
- Delete files / directories
- Create directories
- Upload and download files

Any updates you make take place immediately on the file system of the host upon which the Java Server runs.

> ***Important:*** You need to take extreme care with this facility, as it is possible to compromise files that you may need on the host system. If you use the facility, Sky Technologies strongly recommends that you secure access to the web status page by a user name and password. You should also configure the Java Server to run as a user with controlled or limited access to the host computer file system.

5.6.4.7 Application Details and Synchronization

A separate list of applications is held and managed for each host interface. Select the Applications link from the main page and a list of applications appear.



skytechnologies SkyMobile Application Server

APPLICATION INFORMATION FOR
HOST INTERFACE 'RFCCLIENT' (PAGE 1 OF 2)

App/Ver	Description	Time Stamp	Build Info	Status	
001/001	STAR: Master Data	20101209214713	September 2008	Ready	Synchronize
001/002	STAR: Company Administration	20110516141626		Ready	Synchronize
001/003	STAR: Note Management	20101209214718		Ready	Synchronize
001/004	STAR: Customer Notes	20110516141626		Ready	Synchronize
001/005	STAR: Support Monitor	20100303092307		Ready	Synchronize
001/099	STAR: STC Custom Reports	20100303092307		Ready	Synchronize
300/001	Zuellig Screen Layouts	20110120190039		Preparing Binaries	Synchronize
600/016	CRM 7.0 Demo	20101209225809		Ready	Synchronize
600/031	Style Demos	20101209225819		Ready	Synchronize
600/034	Zuellig Pharma POD: Proof of Delivery	20110516141627		Ready	Synchronize
600/036	Pemex BI iPad Demo	20101209225822		Ready	Synchronize
901/001	Field Service	20110415161955	v2	Ready	Synchronize
920/004	Sales Order Entry and Cust Maintenance	20110516141627		Ready	Synchronize
930/001	BI for SD	20101209225847		Ready	Synchronize
940/001	Credit Position	20101209225848		Ready	Synchronize

Show All | Go To Page | Next -> | Refresh | Back

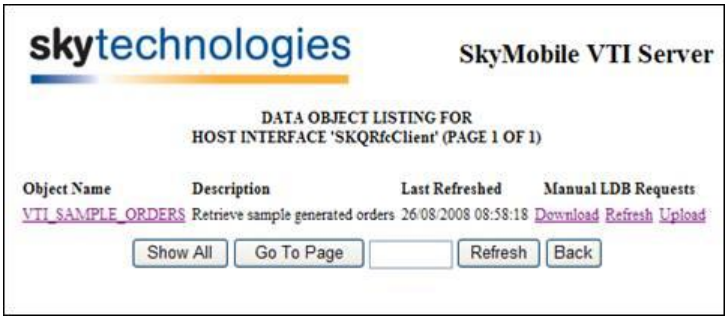
It is possible to force the immediate synchronization (refresh) of a given application from the SAP host by selecting the *Synchronize* link next to the application in question.

5.6.4.8 Viewing and Maintaining Data Objects

Data Objects contain one or more inter-related tables, that is, the concept is that data is synchronized as a 'whole transaction' thereby avoiding fragmented individual table synchronizations and guaranteeing complete data delivery. You configure the data objects in the application workbench and link to a application explicitly through a complimentary reference. The Java Server automatically maintains and creates locally all related underlying. Refer to Viewing and maintaining local database tables for more details.

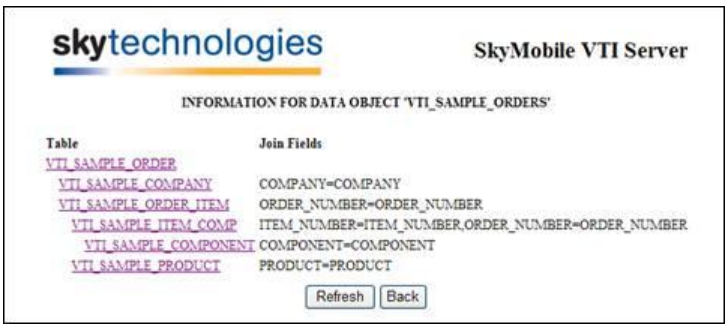
> ***Note:*** All tables related to a data object have their synchronization controls (download, refresh and upload) automatically nullified, that is, these you must perform these operations at the data object level.

This facility allows you to view and manage (if authorized) data objects. You may also export and import table data from here. Separate local databases are maintained for each host interface. Select the relevant Data objects (DOB) link from the main page and a list of Data Objects (DOBs) appear.



From this screen, you can:

- Trigger download, refresh, or upload data transfer operations
- Click the data object name to list the related tables and their dependencies.



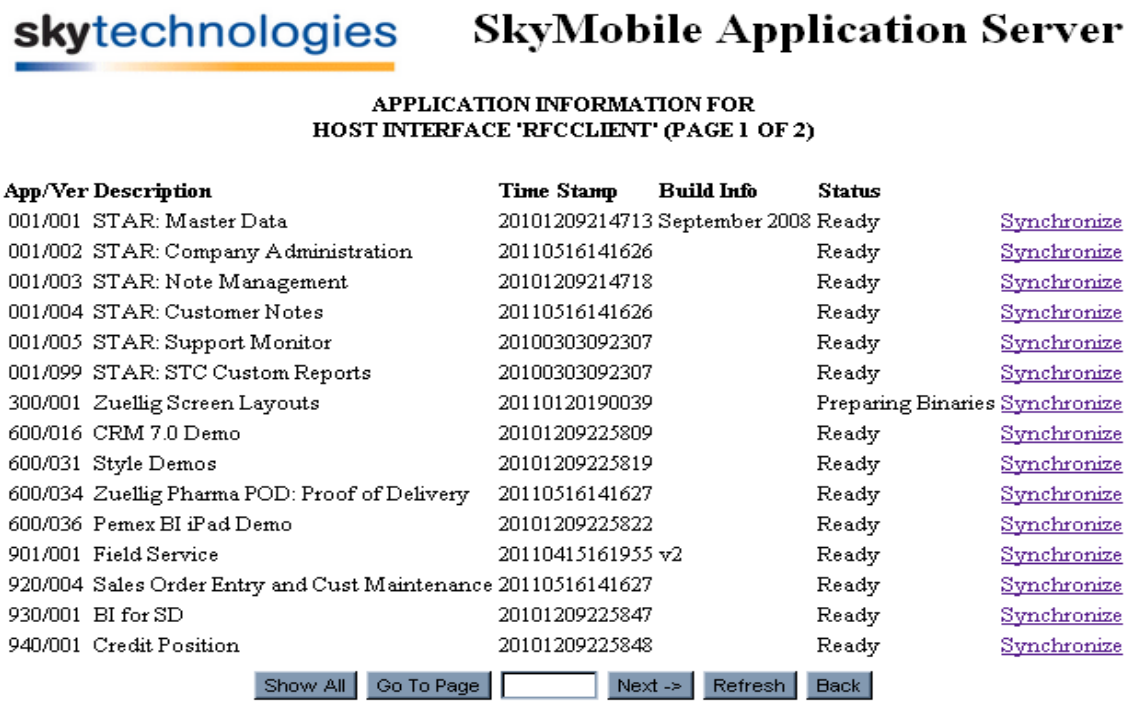
The above screen is an example of a data object table dependency list. The LDB tables are indented depending on their parent / child relationship. You may click the LDB name to view and manage the table definition and data. Refer to Viewing and maintaining local database tables.

5.6.4.9 Viewing and Maintaining Local Database Tables

Local database tables are configured in the application workbench and are either linked to a application either by association (used in a screen) or explicitly through a complimentary reference. In either case, the Java Server creates locally and automatically maintains these tables. By default, the built-in 'SkyDB' database is used, but you may implement your own JDBC compliant database, for example, SQL Server, and Oracle. See the database implementation guide for more information.

> ***Note:*** If a local database is part of a data object, its synchronisation controls (download, refresh and upload) are automatically nullified, that is, you must perform these operations at the data object level.

This facility allows you to view and manage (if authorised) database tables and data. You may also export and import table data from here. Separate local databases are maintained for each host interface. Select the relevant Tables link from the main page and a list of local database tables appears.

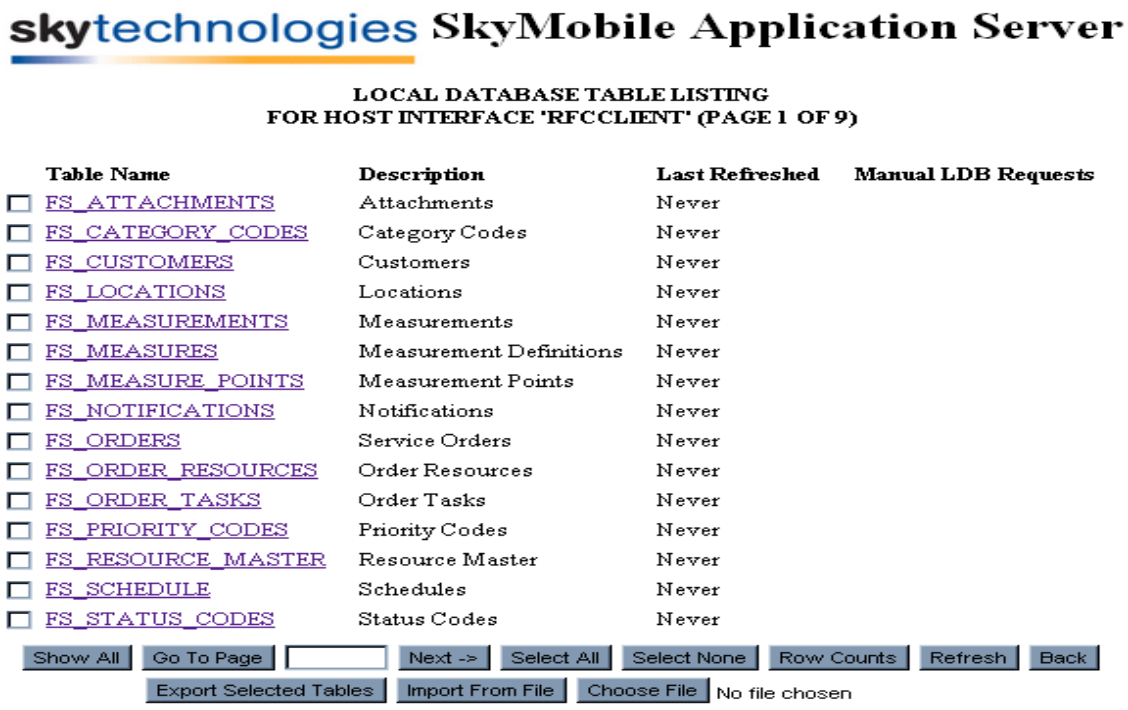


From this screen, you can:

- Export / Import table data (XML format)
- Trigger download, refresh, or upload data transfer operations
- View, and if configured, modify, table data

> *Important:* Separate instances of databases are stored for each host interface. If you change the name of the host interface or remove it, the Java Server automatically removes the associated database. Be aware of the consequences and any host interface change and the configuration to support it. Some functionality in this section may depend on the capability of JDBC drivers (check).

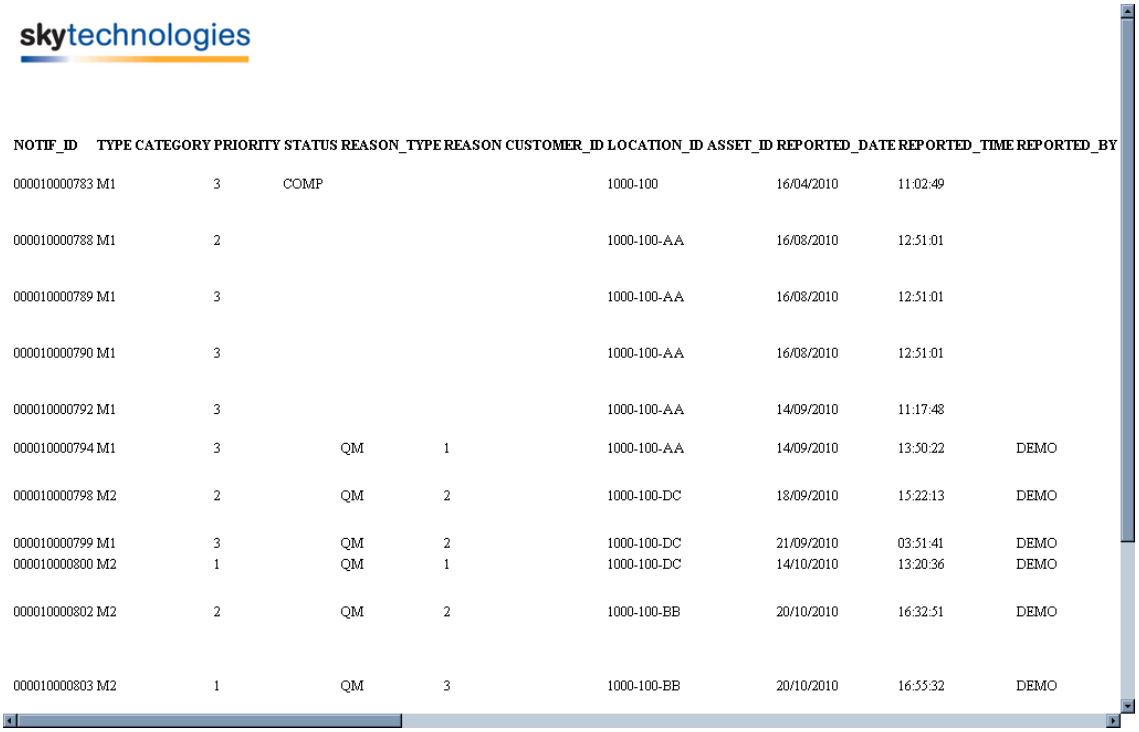
If you select a table for display / maintenance of data, a selection screen appears.



This screen is used to restrict the data appeared. The check box next to each field indicates whether you should include the field in question.

> *Note:* You may use the **Indices** button to display and rebuild secondary indexes on the table.

Press the search button, and the table data is displayed:



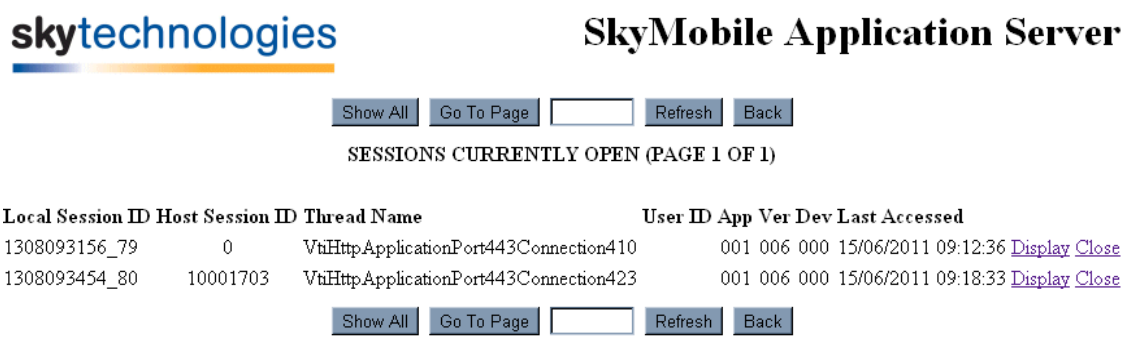
skytechnologies

NOTIF_ID	TYPE	CATEGORY	PRIORITY	STATUS	REASON_TYPE	REASON	CUSTOMER_ID	LOCATION_ID	ASSET_ID	REPORTED_DATE	REPORTED_TIME	REPORTED_BY
000010000783	M1		3	COMP				1000-100		16/04/2010	11:02:49	
000010000788	M1		2					1000-100-AA		16/08/2010	12:51:01	
000010000789	M1		3					1000-100-AA		16/08/2010	12:51:01	
000010000790	M1		3					1000-100-AA		16/08/2010	12:51:01	
000010000792	M1		3					1000-100-AA		14/09/2010	11:17:48	
000010000794	M1		3		QM		1	1000-100-AA		14/09/2010	13:50:22	DEMO
000010000798	M2		2		QM		2	1000-100-DC		18/09/2010	15:22:13	DEMO
000010000799	M1		3		QM		2	1000-100-DC		21/09/2010	03:51:41	DEMO
000010000800	M2		1		QM		1	1000-100-DC		14/10/2010	13:20:36	DEMO
000010000802	M2		2		QM		2	1000-100-BB		20/10/2010	16:32:51	DEMO
000010000803	M2		1		QM		3	1000-100-BB		20/10/2010	16:55:32	DEMO

This screen displays the selected data and, if you configure, allows you to insert, update or delete rows.

5.6.4.10 Sessions - Port Connections

To view and manage all active sessions for a Java Server, click the **Sessions** button.



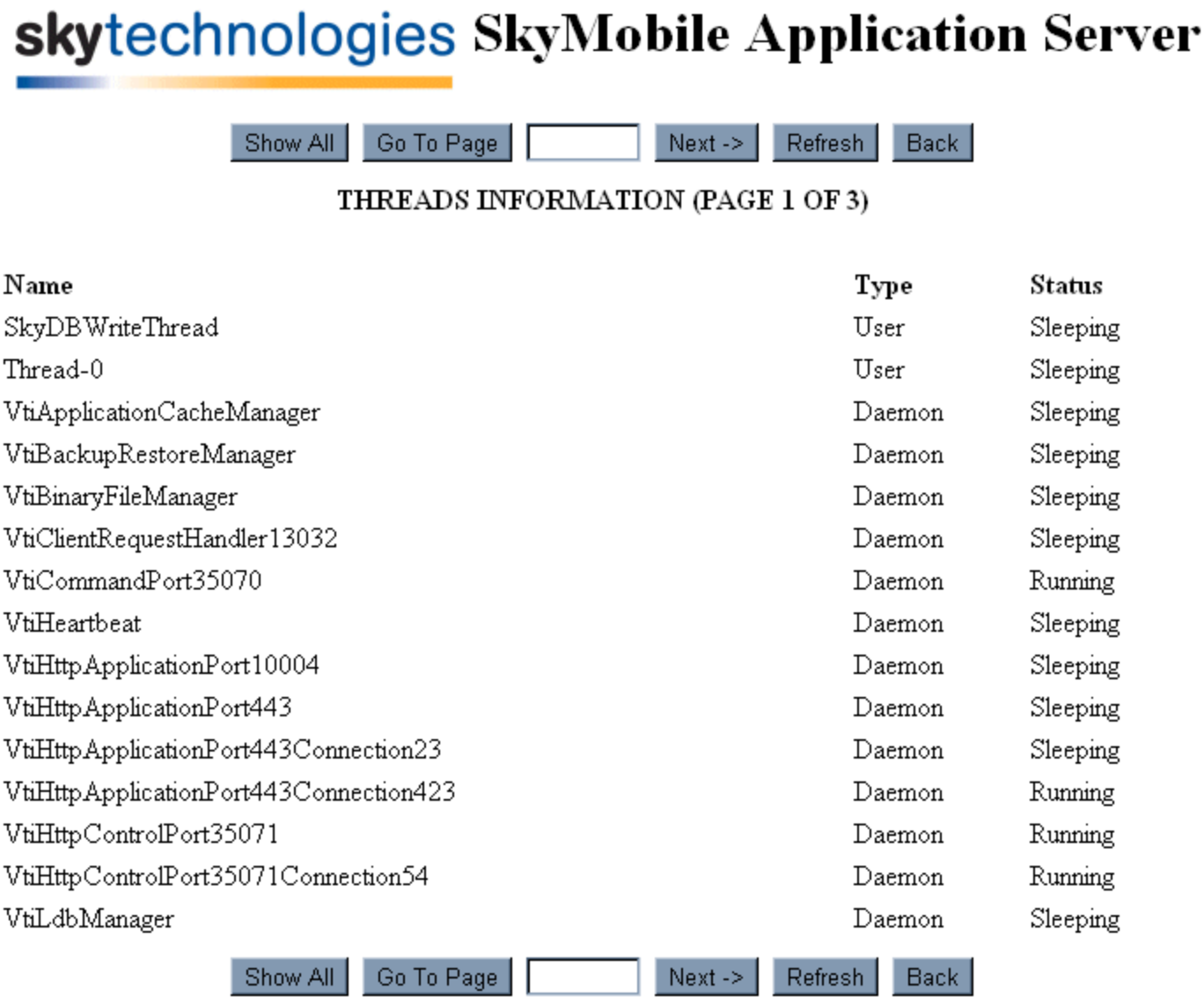
From this screen, you can:

- View what sessions are currently active
  - The thread name (connection) associated with the session
  - The session header user, application and version attributes
- Close active sessions

> ***Note:*** Each session has a local and host session ID. This is to uniquely identify the session on the local machine and on the central host. Use the host session ID with the SAP Session Manager display. Each active session is allocated a unique thread (internal process) that is responsible to process the application requests. This internal name is referenced in the log and technical tracing to identify the associated connection.

5.6.4.11 Threads - Internal Processes

This option displays all the multi-processing (threads) options that are active and their current status. It is useful to ascertain if certain processes have failed.

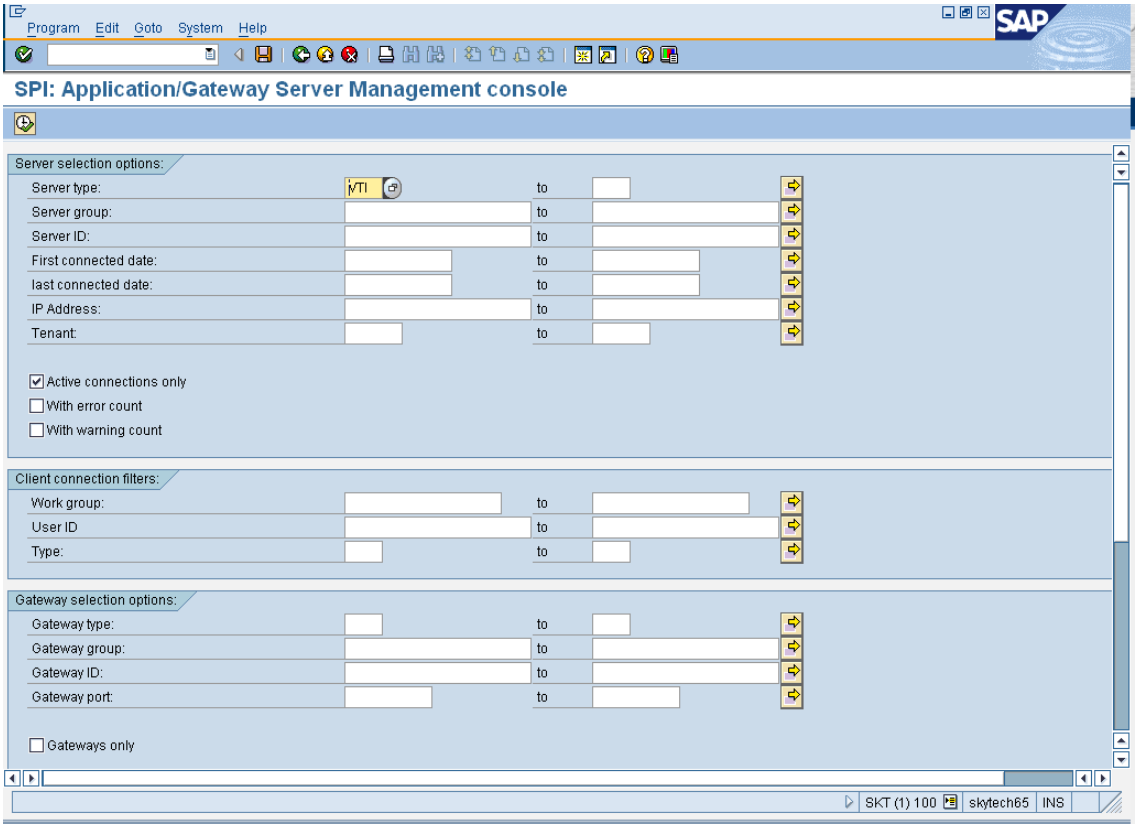


5.6.5 Container Management Console

A standard facility is provided in the MEAP server to view and manage all Secure Containers connected to the MEAP Server. There are effectively two views, a technical view providing IP connection / port details and an organizational view showing configured user / server data, for example, region / name / contact details.

5.6.5.1 Viewing Connected Containers

Logon to the MEAP and execute the YECJ transaction. A selection screen appears that provides various filter and list configuration options.



Once you enter the selection criteria, the following list appears.

The green or red light on the left hand side shows whether the server is still connected, that is, still has a heartbeat. Depending on the view selected, either technical or organizational data appears.

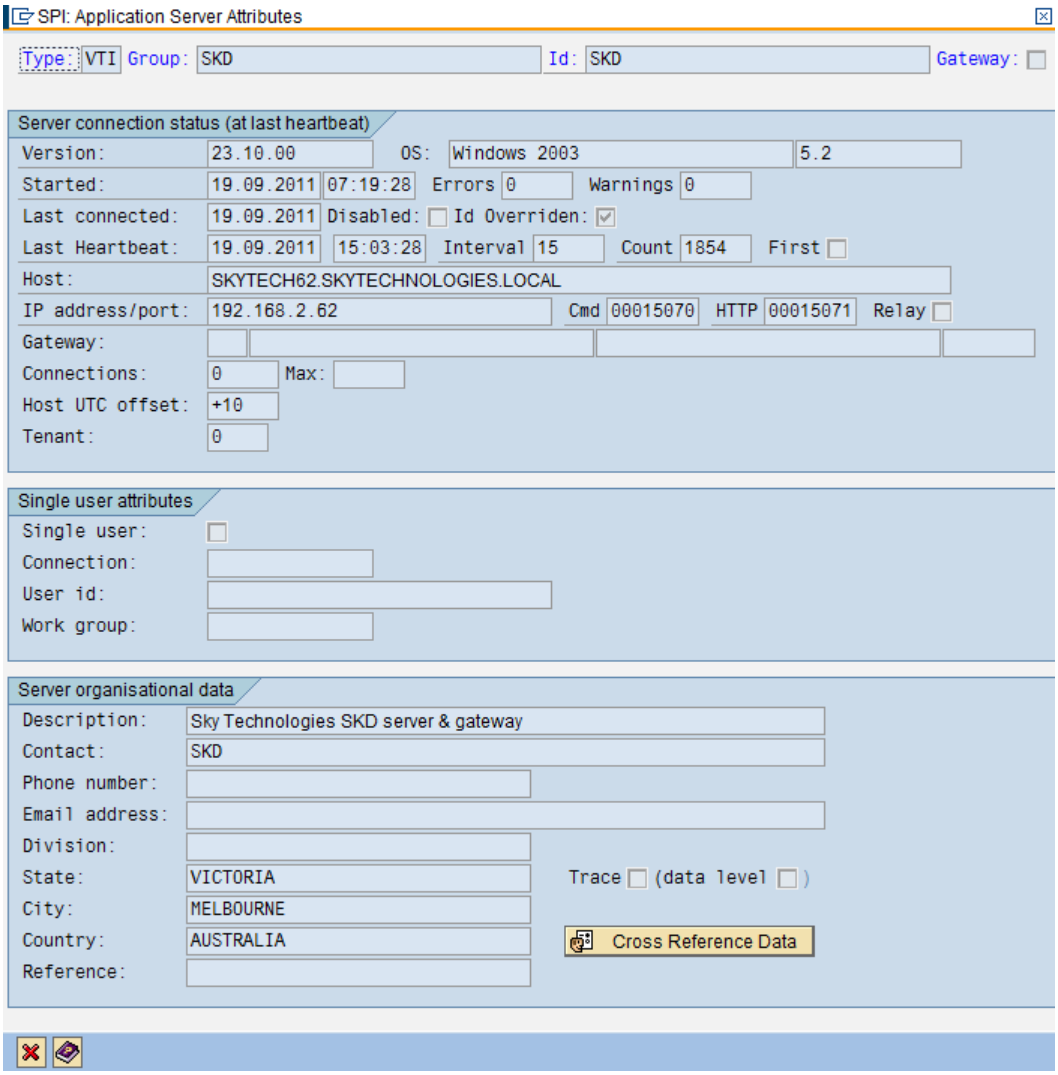
From this screen, you may:

- View container attributes and connections
- Connect to a servers web status page
- Stop or restart the server
- Toggle the trace level
- Issue heartbeat commands
- Open / Close Http relay port connections
- Maintain cross reference and organizational data.

Note: In order to issue direct 'real-time' stop / restart commands through the command port, you must implement the SkyConnect TSX component. TSX enables you to issue commands directly to Application Server command ports through TCP / IP sockets calls. Note that this is not the only option. Asynchronous heartbeat commands offer the same functionality.

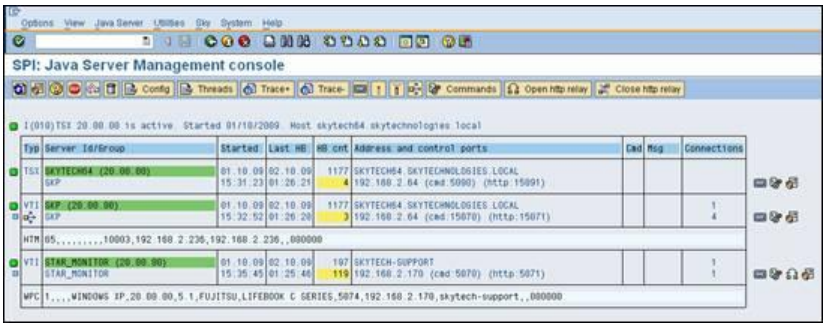
5.6.5.2 Container Technical Attributes

If you double-click any of the containers or gateways in the list, a pop-up of all its technical attributes appears. You can use this to diagnose the heartbeat details.

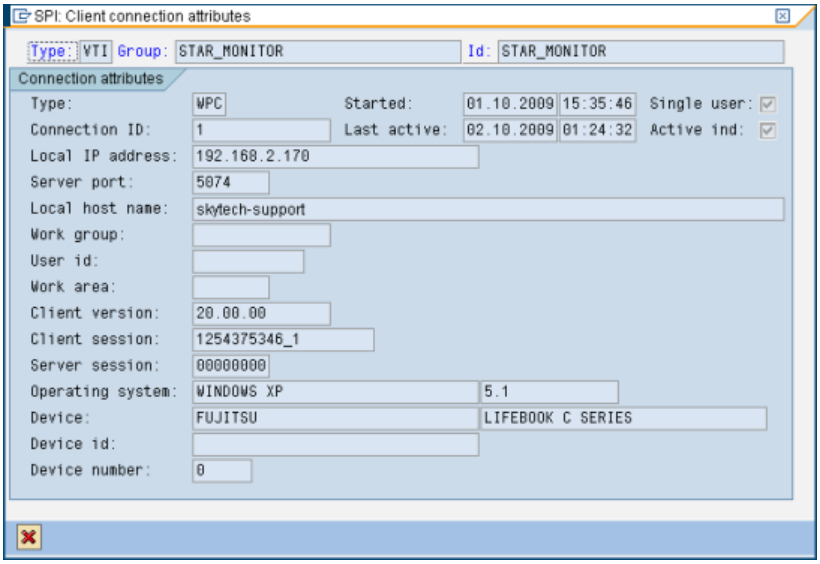


5.6.5.3 Current Client Connections

All the current client connection details (that is, port connections) are also passed up to SAP with each heartbeat request. A server / connection cross reference is automatically maintained and you can view this data in the console. If client connections are available, the total number appears under the connection column in the server list. In addition, '+' and '-' expansion symbols appear under the traffic light icon. You can click either of these to list the current client connections, as below:



In the above screen shot, the SKP server has a HTML (HTM) port connection and the STAR_ MONITOR server has a WPC (Windows Presentation Client) connection. You can double-click the line to display the details:



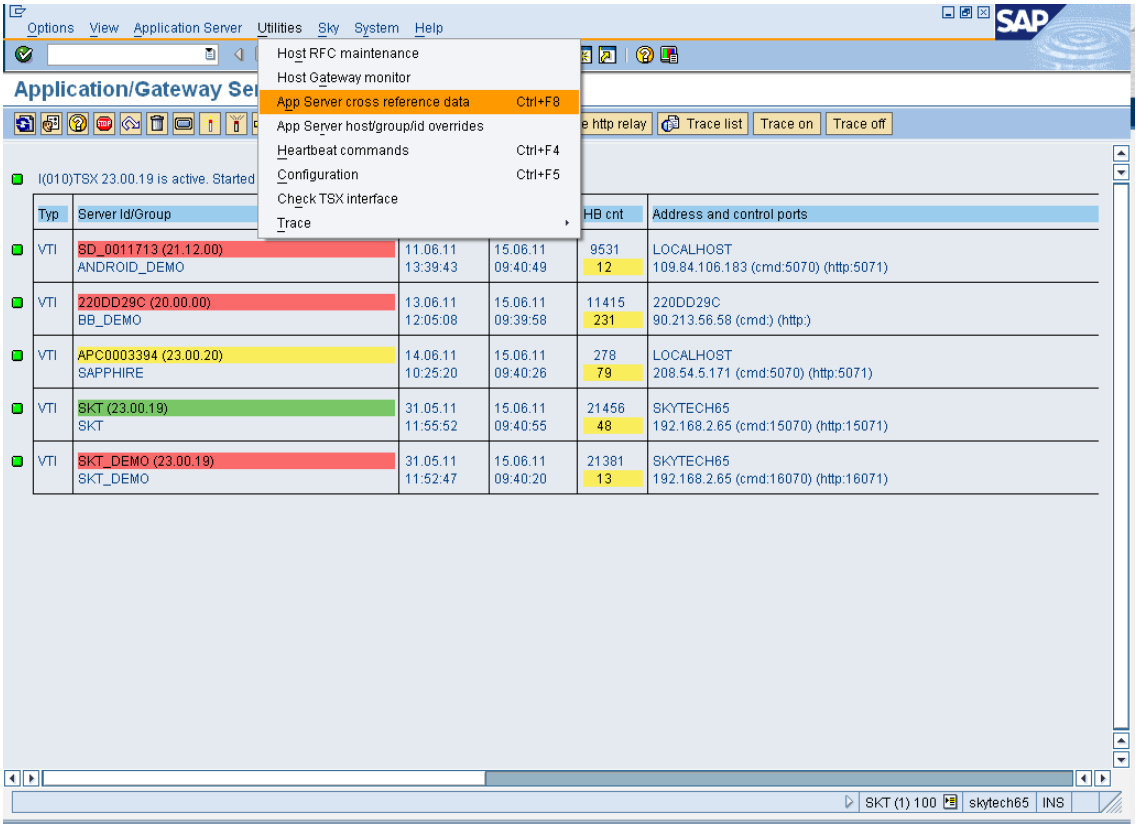
> ***Note:*** The technical connection and session details, the operating system, make and model details also appear.

5.6.5.4 Container Cross Reference Facility

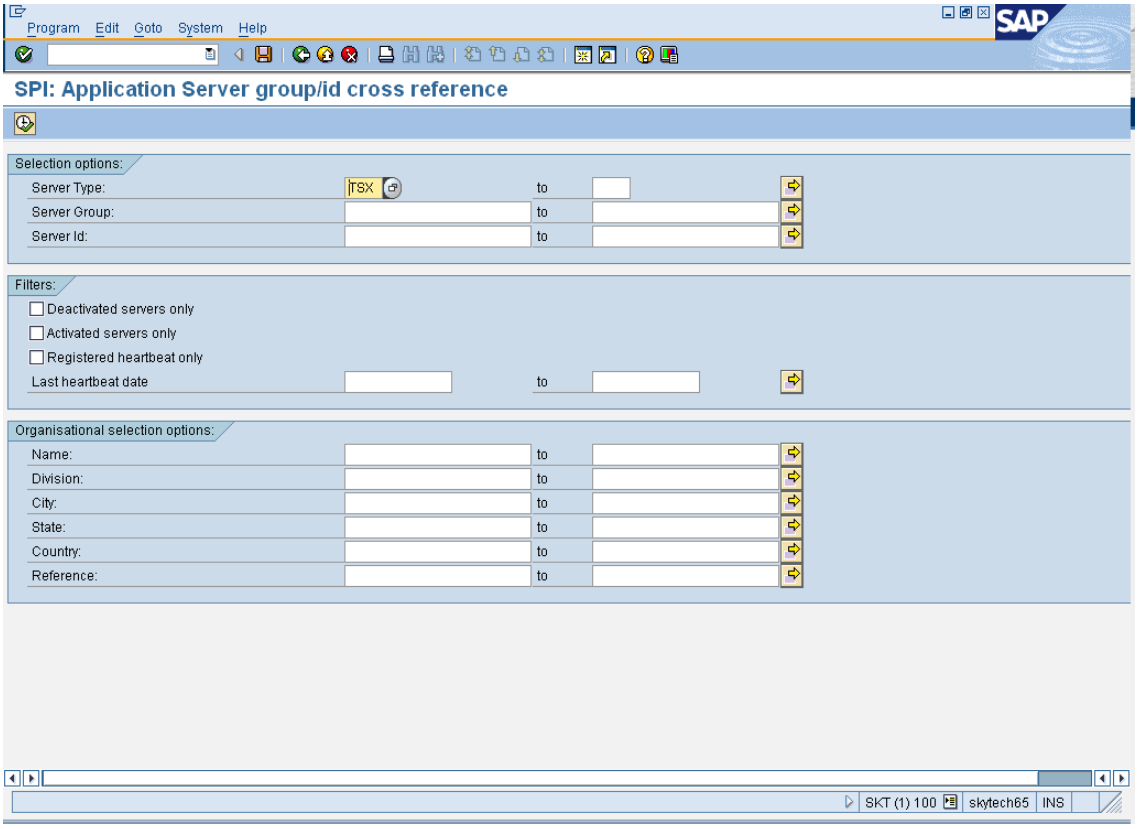
The system automatically maintains a list of 'registered' Secure Containers, but you can maintain the list and attributes manually using the cross reference facility.

> ***Note:*** You use this facility to disable / deactivate containers, that is, prevent them from connecting to the MEAP Server.

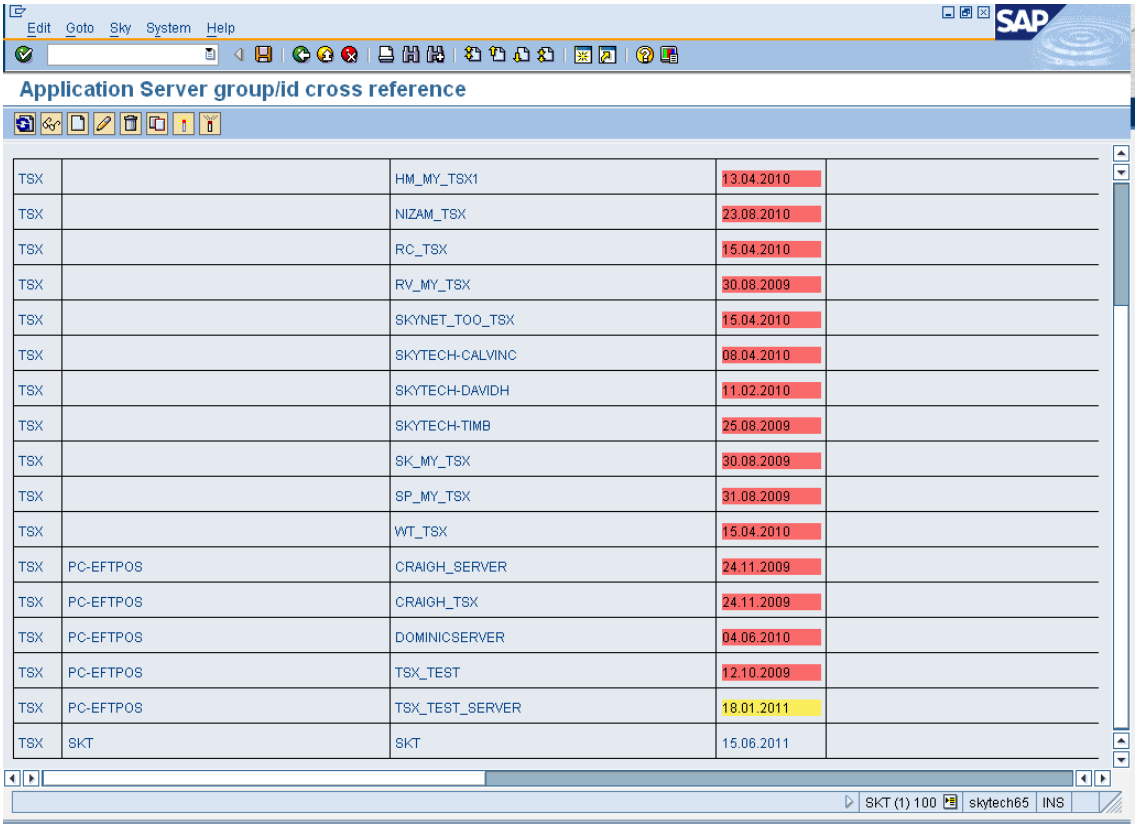
You invoke the utility from the management console **Utilities** menu.



The following selection screen appears that allows to restrict the list.



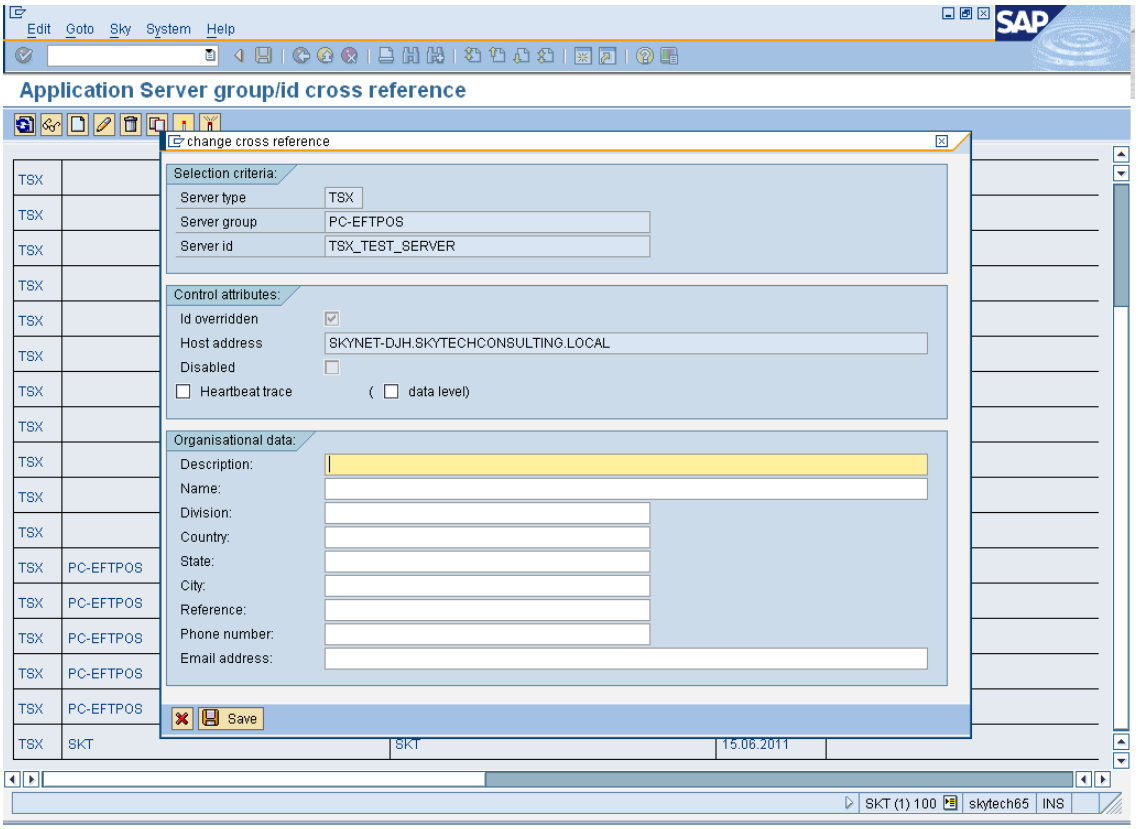
The following list appears:



From this screen you may:

- Create, maintain, copy or delete server entries
- Deactivate and activate servers

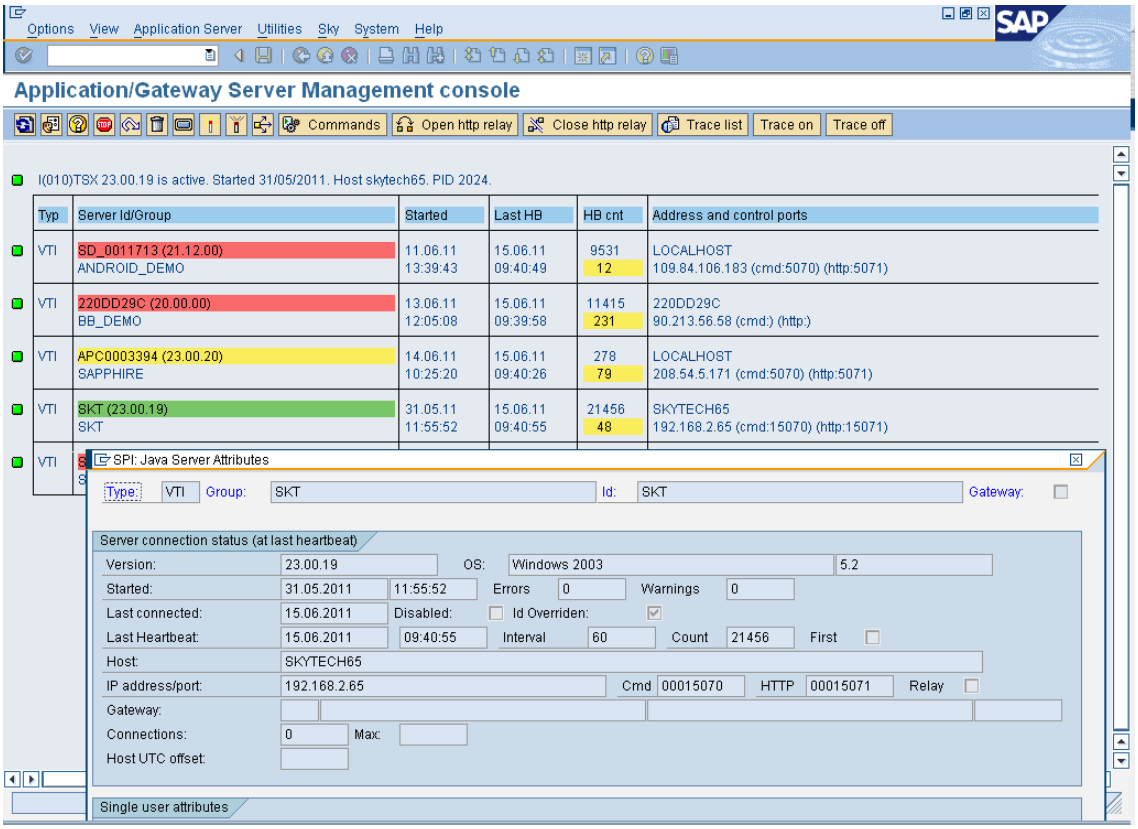
> ***Note:*** You may also just double-click an entry to maintain the organizational details. These are free format, so take care to maintain consistency.



5.6.5.5 Organizational Data Management

You can 'tag' a specific server ID / group with additional meta-data such as country / state / user / external reference. This is to help categorize and identify individual and / or groups of remote Java servers. This data is held as attributes of the Secure Container cross reference entry, refer to Container cross reference facility for more details. Easy access to organizational data maintenance is through the icon located on the server line or on the application toolbar.

In addition to maintaining entries directly through the cross reference facility, you may simply double-click a line to invoke a pop-up display.



You can then branch directly to the cross reference maintenance utility by clicking the **cross reference data** button.

Note: Once you define the organizational data, you may use it to select lists of containers and use the organizational data view on the console list. The list is simply a comma delimited string of values.

5.6.5.6 Working with Heartbeat Commands

Every Secure Container issues a heartbeat request to the MEAP server to advise its status. As part of the heartbeat processing, you may configure one or more commands and return to the secure container to execute. The range of options is quite extensive and allows you to manage all the major aspects of the server, for example, change the configuration and re-start the server. You may also use the Command lists facility to define a sequence of commands to deploy out to one or more mobile containers. You may either create commands manually using the heartbeat command manager or create them through a ABAP program / function using the ABAP SDK.

If the details of the work group and user ID are configured into the session header, then you may issue commands to current work group / user ID criteria.

> ***Important:*** It is highly recommended that you first test the commands on a 'test case' container in a production environment; some of these command options may cripple an active server from use if you do not use properly.

Use the following commands:

Command	Description	Number of parameters	Required parameters
APPCHECK	Check application definitions for changes	2	2
APPSYNCH	Synchronize application {appid} {verid}	0	0
BINARYCHECK	Check binary objects and packages for changes	1	1
BINARYDELETE	Delete a existing binary object {name}{version}	1	1

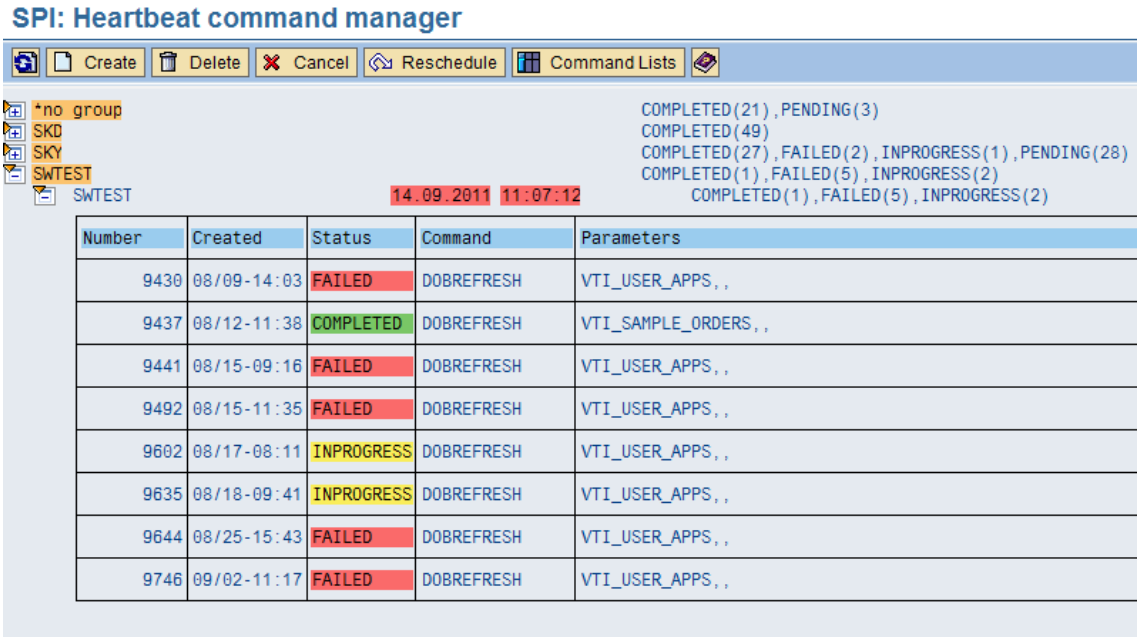
Command	Description	Number of parameters	Required parameters
BINARYDOWNLOAD	Download a binary object from the host {name}{version}	1	1
BINARYUPLOAD	Upload binary object to the host {name} {version}	3	3
CONFIGCHANGE	Change configuration {section} {option}{value}	0	0
CONFIGSAVE	Commit / apply configuration changes to disk	0	0
DISABLE	Temporarily disable the application server from use	1	1
DOBDOWNLOAD	Invoke download of DOB {name}	1	1
DOBREFRESH	Invoke refresh of DOB {name}	1	1
DOBUPLOAD	Invoke upload of DOB {name}	0	0
ENABLE	Re-enable the application server for use	0	0
KILL	Totally decommission the application server	1	1
LDBDOWNLOAD	Invoke download of LDB {name}	1	1
LDBREFRESH	Invoke refresh of LDB {name}	1	1
LDBUPLOAD	Invoke upload of LDB {name}	0	0

Command	Description	Number of parameters	Required parameters
PROVISIONCHECK	Perform a server provision check	0	0
PROVISIONNEW	Provision a existing server from scratch	0	0
PROVISIONRESET	Re-provision a existing server (keep data)	0	0
RELAYCLOSE	Close a HTTP relay port connection	0	0
RELAYOPEN	Open a HTTP relay port connection	0	0
RESET	Reset the data and configuration of a server	0	0
RESTART	Restart the server	0	0
STOP	Stop the server	0	0
TRACEMINUS	Decrease the trace level	0	0
TRACEPLUS	Increase the trace level	0	0

To maintain commands, select the **Heartbeat Command** option from the console **Utilities** menu or click the **Commands** icon on the server line or application tool bar. This invokes the heartbeat command manager that you use to issue and monitor heartbeat requests.

Heartbeat Command Manager

You use the heartbeat command manager to issue and monitor commands directed to one or more Secure Containers through the heartbeat facility. As well as manually, you can issue heartbeat commands from an ABAP program / function using the SkyMobile ABAP SDK. When the manager starts, a pop-up appears to restrict the list of servers / users to list commands for. Execute the selection and a list appears, broken down into server group and ID.



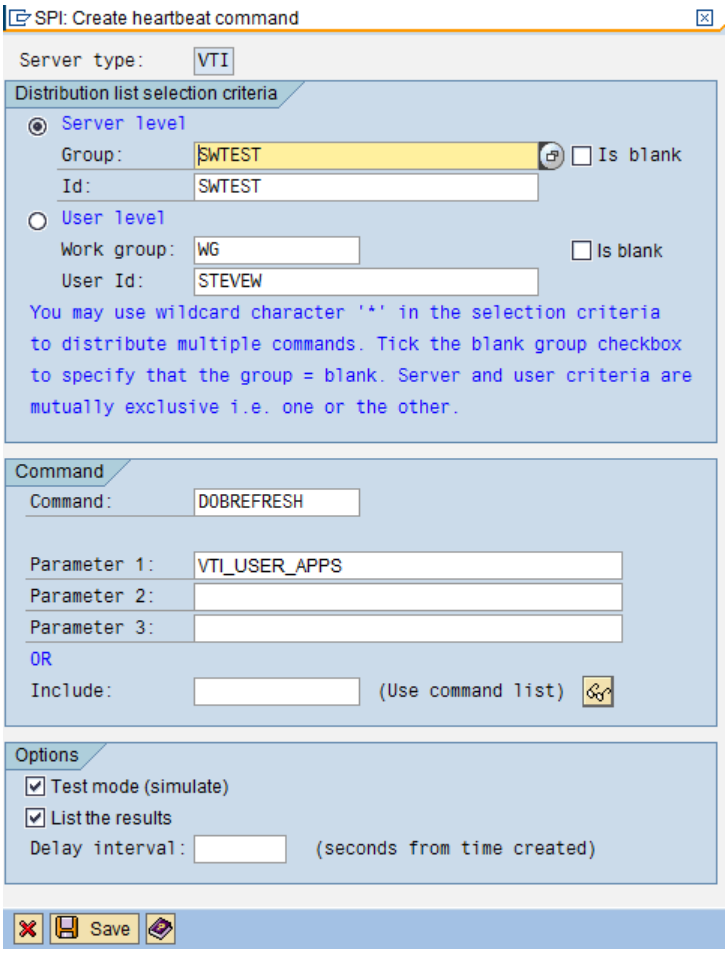
Number	Created	Status	Command	Parameters
9430	08/09-14:03	FAILED	DOBREFRESH	VTI_USER_APPS,,
9437	08/12-11:38	COMPLETED	DOBREFRESH	VTI_SAMPLE_ORDERS,,
9441	08/15-09:16	FAILED	DOBREFRESH	VTI_USER_APPS,,
9492	08/15-11:35	FAILED	DOBREFRESH	VTI_USER_APPS,,
9602	08/17-08:11	INPROGRESS	DOBREFRESH	VTI_USER_APPS,,
9635	08/18-09:41	INPROGRESS	DOBREFRESH	VTI_USER_APPS,,
9644	08/25-15:43	FAILED	DOBREFRESH	VTI_USER_APPS,,
9746	09/02-11:17	FAILED	DOBREFRESH	VTI_USER_APPS,,

Once the list appears, you may either create single commands or select a preconfigured command list. You may also schedule commands to only issue after a certain number of seconds (delayed). Once issued, you may monitor the command by its status, that is, *Pending*, *Issued*, *In-progress*, *Complete*, and *failed*. The command automatically fails if no status response is received back from the server in three subsequent heartbeat requests. The command utility is useful to monitor and manage commands. If any problems occurs, check the Secure Container log for details. The number of outstanding / failed commands also appear on the server line in the console listing.

To create a heartbeat command or allocate a command list, click the **Create** icon in the application tool bar. It is context sensitive, that is, place your cursor on an existing command, and server ID for it to copy. A pop-up appears to prompt for values.

Creating a Heartbeat Command

The following pop-up appears to create a heartbeat command. You may either use server group/ID or user / work group selection criteria. You may enter wild card entries using '*'.



A command may require one or more parameters that may be optional or mandatory. You may also issue a command to one or more application servers depending on the group and ID selected. You may also include a predefined command list. There are options to submit the heartbeat command request in test mode, that is, see what happens and also to list the result (recommended).

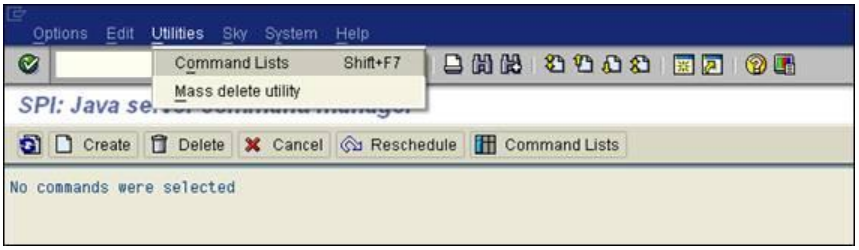
Important: A command list is copied in at the time of reference and so any subsequent changes to the command list has no effect. If you are unsure, a good idea is to execute the option in 'test mode' first. It is also a good idea to try out more complex command sequences on a test candidate first.

Heartbeat Command Results List

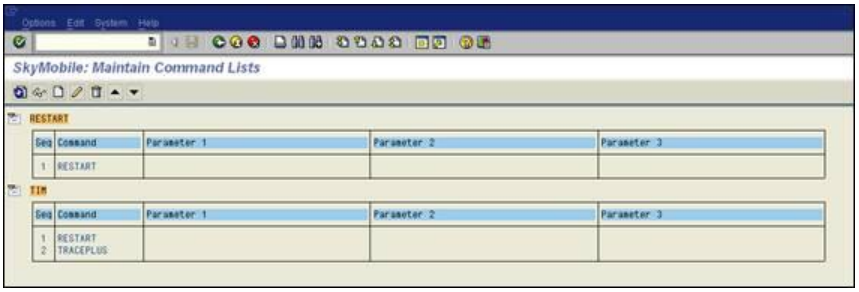
When a heartbeat command or command list is created and issued, a results list may appear (default). This displays a list of containers that the command is (or can be) issued to.

Command Lists

The Command lists compiles a list of commands that you may then issue to multiple Secure Containers. The command list is invoked from the heartbeat command manager either through the **Utilities** menu or the icon in the application toolbar.



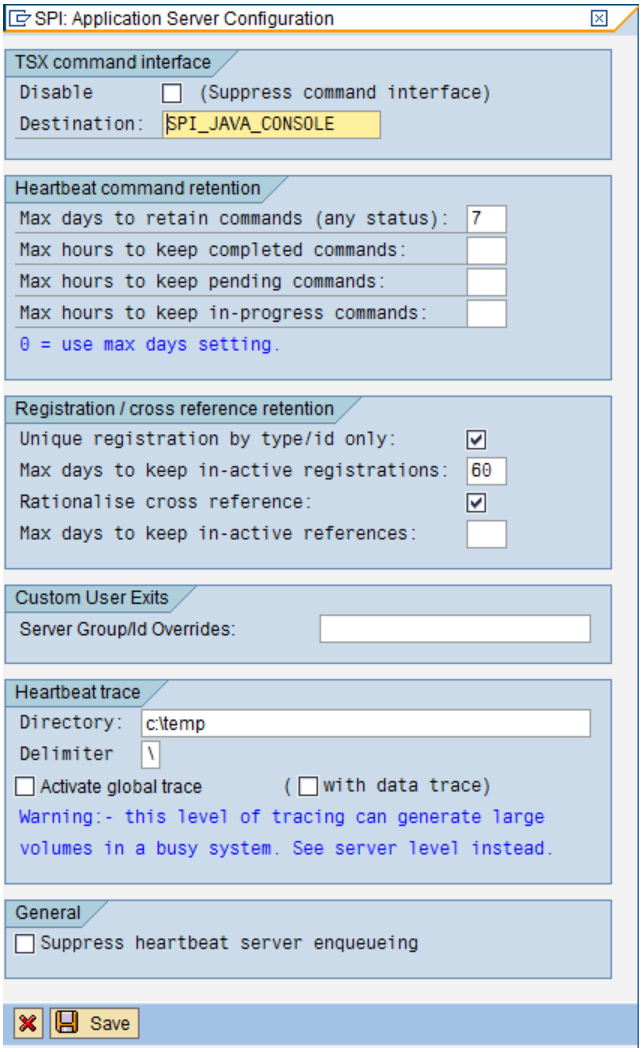
Each command list has a unique list ID. You position your cursor at the level you wish to maintain and use create, change, and delete icons in the application toolbar to maintain definitions.



5.6.5.7 Container Management Console Configuration

The Container management console configuration allows you to specify and override several console options. To access the configuration screen, choose the **Utilities** menu > **Configuration**.

The following pop-up appears:



Option	Description
TSX command interface	If you use a SkyConnect TSX Java Server to issue direct commands to remote Java Servers, then you must specify the TSX destination to use. A default destination (SPI_JAVA_CONSOLE) is created when TSX is installed. Refer to its destination and message configuration. If you are not using a TSX interface, then select the **Disable** check box.
Heartbeat command retention	The Sky SPI housekeeping routine and SkyConnect ECS SPI_HOUSEKEEPING) considers these options that are created automatically when the system is installed. They are used to automatically remove heartbeat commands. This becomes important if a large volume of DOB / LDB 'push' operations are processed.
Server registration / cross reference retention	These settings are important to automatically maintain the list of Java servers registered. The unique registration makes sure only the latest instance of a server (uniquely identified by its ID) is kept, that is, duplicate old entries with different IP addresses are dropped. If rationalize is selected, the housekeeping routine automatically removes any servers that are not registered in the system.
Custom User Exits	Server Group / ID overrides, dynamically generate and allocate the server group and / or ID to use on its first heartbeat call. This exit is called in conjunction with the "Host Override" facility and is called after all other configured processing. Refer example exit /SKY/YECS8198 for details.
Heartbeat trace	Use this section to configure the target trace directory and the trace globally. ***Important:*** Configuring the trace globally can generate a considerable amount of data. It is recommended that you use the server trace facility to selectively trace heartbeat calls.
Suppress heartbeat enqueue	By default, Kony for SAP enqueues internal data tables and cross references when a heartbeat is processed to ensure other processing is synchronized. Although, the heartbeat processing is efficient, this can cause delays in a high volume system and this option is used to switch it off. If its switched off, there is a slight risk that other processing may be performed off old server cross reference data while the heartbeat information is processed.

5.6.6 Command Port

Each Java Server has a command port that allows direct commands to issue to it. This command port is primarily for internal use (for example, SAP Server Management Console), but also accepts connections from any telnet client in either VT100 or VT200 modes or socket connection. Typically, a SkyConnect TSX server is used.

The following commands may be issued:

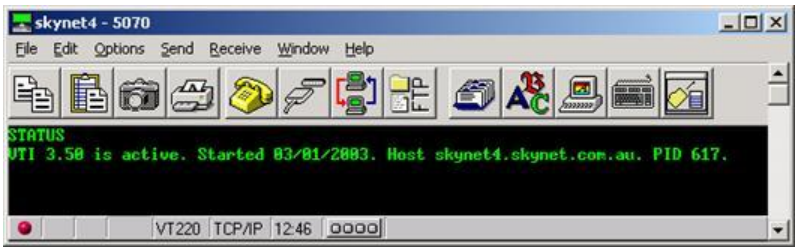
Command	Description
STATUS	Displays the current status of the server
STOP	Stops the server
RESTART	Restarts the server
RELAYOPEN	Open a HTTP relay connection
RELAYCLOSE	Close a HTTP relay connection
BACKUP	Initiate a backup
BACKUPINCR	Initiate a incremental backup
THREADS	Retrieves the active server threads
GETCONFIG	Retrieves the configuration file
PUTCONFIG	Updates the configuration file

5.6.6.1 Using a Telnet Connection

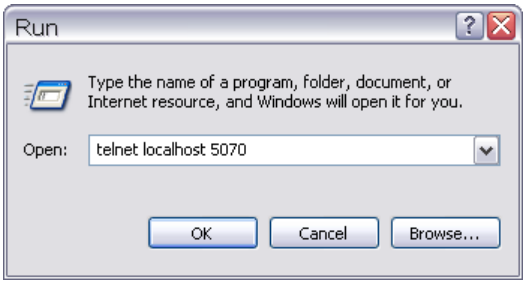
You may use any telnet client for example, Reflection, and Netterm. You must configure the telnet connection with echo on, so you can view the typed command. When the connection is made, the Java Server waits for an input command to be typed and sent by pressing return (enter). Each time a command is issued; the connection is closed and you must re-establish before you can issue an another command.

> ***Note:*** You may use the default Microsoft telnet client, but in some circumstances may not display the returned information (buffer) when the connection closes.

5.6.6.2 Example Status Command (using the Netterm Telnet Emulator)



To use the Windows telnet emulator, click start, then run and type in the following command with your specific IP address and command port:



5.6.7 Relay Ports

Relay ports are used in cases where the IP address of the remote Application Server is not reachable. This may be due to network address translation (NAT'ing) by a firewall or proxy server, or in the case of Blackberry devices where the operating system security does not support the opening of a remote socket. So instead of the central host attempting to open a connection to the remote device, the concept of a relay port is that the device opens a management connection to a gateway and this is used to 'relay' web status page displays and options.

In order to enable relay port functionality, you need to specify relay port configuration on both the Gateway server and on the Application Server. For example:

5.6.7.1 Gateway Relay Port

```
SERVER.RELAYPORT:PORT15076.IDLETIMEOUT =
SERVER.RELAYPORT:PORT15076.PINGINTERVAL =
SERVER.RELAYPORT:PORT15076.PORT = 15076
SERVER.RELAYPORT:PORT15076.RELAYREQUESTTIMEOUT =
```

5.6.7.2 Application Server Relay Connection

```
SERVER.RELAYCONNECTIONS.ALLOWCONFIGCHANGESFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWFILEENQUIRYFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWFILEUPDATESFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.RELAYCONNECTIONS.ALLOWLDBUPDATESFROMWEBPAGE = false
SERVER.RELAYCONNECTIONS.AUTHORISATIONPASSWORD =
SERVER.RELAYCONNECTIONS.AUTHORISATIONUSERNAME =
SERVER.RELAYCONNECTIONS.HTMLBUTTONLOCATION = Both
SERVER.RELAYCONNECTIONS.HTMLPAGESIZE = 15
SERVER.RELAYCONNECTIONS.RECONNECTSECONDS =
SERVER.RELAYCONNECTIONS.RELAYHOST = demo.skytechnologies.com
SERVER.RELAYCONNECTIONS.RELAYPORT = 40002
SERVER.RELAYCONNECTIONS.SHOWLDBROWCOUNTS = true
SERVER.RELAYCONNECTIONS.SOCKETCLOSEDELAY =
SERVER.RELAYCONNECTIONS.SUPPRESSLOGO =
SERVER.RELAYCONNECTIONS.USEHTMLCOMPRESSION =
```

> ***Note:*** Refer to the configuration section for more information on these options.

5.6.7.3 Relay Port Details

Application servers publish to the MEAP server that it is "relay capable" through its heartbeat call. In addition, gateway servers publish their open relay connections to the MEAP server whenever the list changes. The Application Server Management Console (YECJ) in the MEAP server uses this information to ascertain if relay ports are activated and their current status, thereby activating relay port functionality for an application server connections.

5.6.7.4 Opening a Relay Port

Before you can connect to an application server through a relay port, you must first open it. The  icon appears to the right of the connection. You either click this, or position your cursor on the line and click the **Open http relay** button on the application toolbar. This generates a RELAYOPEN heartbeat command that is queued to return to the application server on the next heartbeat call. Once the application server receives this, it attempts to open a relay connection to the gateway. When the gateway receives this connection, it notifies the MEAP server and the connection is deemed to open. This is indicated by the icon on the right hand side changing to a **Close relay** icon  .

5.6.7.5 Closing a Relay Port

A relay port remains open until it is explicitly closed, the application server shuts down or the connection is timed out (IdleTimeout parameter on the Gateway). To manually close the relay connection, either click the close **Relay** icon  or position your cursor on the line and click the **Close http relay** button on the application toolbar. You can close the relay connection from the Gateway web status page, see the connections link against the relay port definition.

5.6.7.6 Browser Connections

Once a relay port connection is established, you may now connect to the remote devices web status page. The Application Server Management console automatically detects this and generates the appropriate URL (with special query string) to the associated Gateway that controls the connection and then launches the browser in the standard way. For example,

> ***Note:*** You may also invoke the remote Java Servers web status page through the 'relay' connections page on the Gateway web status page.

5.6.8 Operation of Containers and Gateways

5.6.8.1 About

This section describes how to start, stop, cancel, restart and check the status of Secure Containers and Gateways. There are many mechanisms available depending on the platform. Remote connection to the web status page is performed either directly from a web browser or through the YECJ management console in the MEAP server.

5.6.8.2 Basic Operations

Starting a Container/Gateway
Stopping a Container/Gateway
Restarting a Container/Gateway
Checking the Status of a Container/Gateway

5.6.8.3 Starting a Container/Gateway

Platform	Method
Windows Service	Start service Sky Java Server (or given name). Refer the Windows Installation section for details on how to implement services. The services are usually configured as automatic and you may start through the services control panel. You may also start the Java Server through a DOS command window, as documented below.
Windows DOS Window	cd …\Sky Technologies\v17.00.00\vtivti.bat Or, double-click the `vti.bat` file in the installed directory. ***Note:*** This effectively starts the Java Server in a DOS window. Leave the window open. If you close it, the execution cancels.
Unix	cd …/<<Sky Home>>gcp.sh VTI start Or, change directory (cd) to the installed directory and execute: nohup ./vti.sh &In Unix, VTI starts a JVM that executes as a process. Use top or ps -ef to view. Refer monitoring Unix processes for details.

Platform	Method
SAP ECS Process (local host/Unix only)	In this case, you define a SkyConnect interface process (ECS) to periodically check and automatically start the Java Server that runs on at least one of the SAP instances; using a GCP command. If you are interested in this option, contact Kony for SAP support for information and instructions.
Smart Phones / PDAs: • Windows Mobile • Blackberry • Symbian	Click the application icon. If you configure to run locally, the presentation client (for example, WPC, BPC, and SPC) automatically launches the Java Server.

Important: You may not start the Java Server twice. If you require duplicate instances to run, they must execute out of separate libraries with unique command and HTTP ports. It is always a good idea to check the status of the Java Server once you start it. Refer checking the status of the Java Server for details.

5.6.8.4 Stopping a Container/Gateway

Platform	Method
Web Browser (Web status page)	Connect to the web status main page and click the **Stop** button.
Telnet Session	Connect to the Java Server command port and issue the STOP command.
Windows Service	You may implement the SkyMobile Java Servers as one or more Windows services. See the Windows Installation Guide for details. If the Java Server runs as a service, you may use the Services management screen to stop it. STOP service Sky Java Server.
Windows DOS window	Simply close / cancel the DOS window.
Unix	cd .../<<Sky Home>> gcp.sh VTI stop
SAP ECS Process (local host/Unix only)	In this case, you define a (ECS) process to issue a GCP stop command against a Java Server that runs on at least one of the SAP instances. If you are interested in this option, contact Kony for SAP support for information and instructions.

Platform	Method
SAP Java Server management console	• TSX direct connection - Position your cursor on the server you wish to cancel and select the **Stop** icon. This issues a direct call to the Java Servers command port. • Issue a STOP heartbeat command (refer Heartbeat Commands for more information).
Smart Phones / PDAs: • Windows Mobile • Blackberry • Symbian	• Select the **shutdown** option from the system menu. • Close the GUI window. The termination is intercepted and a shutdown of any associated Java Server (if launched) is performed.

5.6.8.5 Canceling a Container/Gateway

Important: You should cancel a container/gateway only as a last resort if the Java server and / or presentation client cannot stop gracefully.

Platform	Method
Windows	If running a service, the service may be stopped from the Windows service manager or ended from the Windows task manager. If you run as a DOS Window, close the window or cancel the application using the Windows task manager.
Unix	The JVM is started from a shell script (`vti.sh`). Therefore, there are two processes involved, the parent is the vti.sh process. Perform a "ps -ef|grep vti" to identify the parent process and "ps -ef|grep java" (or jre) to identify the child process. Perform a kill -9 {process id} on the parent process and ensure that the child process is canceled. If not, perform a kill -9 on that as well.
Smart Phones / PDA's: • Windows Mobile • Blackberry • Symbian	If you cannot stop the GUI gracefully, you need to use the OS system tools to cancel the "VTI" program that runs (if available). As a last resort, you should re-boot the phone or PDA.

5.6.8.6 Restarting a Java Server

Restart the Java Server after a configuration change or refresh connections. You can also configure the Java Server to auto-restart itself in the case of an unexpected failure (see the configuration section).

Platform	Method
Web Browser (Web status page)	Connect to the web status main page and click the restart button.
Telnet Session	Connect to the Java Server command port and issue the RESTART command.
Windows Service (stop / start)	You may implement SkyMobile Java Servers as one or more Windows services. See the Windows Installation Guide for details. If the Java Server runs as a service, you may use the Services management screen to stop it and then start it again.
Windows DOS window (cancel/start)	Simply close / cancel the DOS window and then re-execute `vti.bat`.
Unix	cd .../<<Sky Home>> gcp.sh VTI restart
SAP ECS Process (local host/Unix only)	In this case, you define a ECS process to issue a GCP restart command against a Java Server running on at least one of the SAP instances. If you are interested in this option, contact Kony for SAP support for information and instructions.

Platform	Method
SAP Java Server management console	• TSX direct connection - Position your cursor on the server you wish to restart and select the **Restart** icon. This issues a direct call to the Java Servers command port. • Issue a RESTART heartbeat command (see Heartbeat Commands for more information).
Smart Phones / PDAs: • Windows Mobile • Blackberry • Symbian	• Remotely, you may re-start the underlying Java Server through the SAP Java Server Management console, or the server web status page. If the server is launched by the GUI, the user is notified that the server connection is lost and is given the option to re-connect again at the first (initial) screen. • Perform a stop (shutdown) and then start the GUI again using the **Application** icon.

5.6.8.7 Checking the Status of Java Server

Platform	Method
Web Browser (Web status page)	Connect to the web status main page. The current status of the server and all its connections appear.
Telnet Session	Connect to the Java Server command port and issue the STATUS command.
Unix	cd .../<<Sky Home>> gcp.sh VTI status Typically each instance of a JVM runs as a separate Unix process. Type the Unix top command to view the current active processes. The JVMs shows up as either Java or JRE (if executing Java runtime environment). It is not possible from this display to differentiate between different instances of JVM and what is executed. Type ps -ef|grep java (or jre) to get the java process details and ps -ef|grep vti to associate the start up shell scripts with the java processes. Use the parent / child process IDs to match which java JVM processes were spawned. You may also use the top command to show the Unix processes and the resources used.
SAP ECS Process (local host/Unix only)	In this case, you define an ECS process to issue a GCP status command against a Java Server that runs on at least one of the SAP instances. If you are interested in this option, contact Kony for SAP support for information and instructions.
SAP Java Server management console	TSX direct connection - Position your cursor on the server you wish to restart and select the **Status** icon. This issues a direct call to the Java Servers command port.

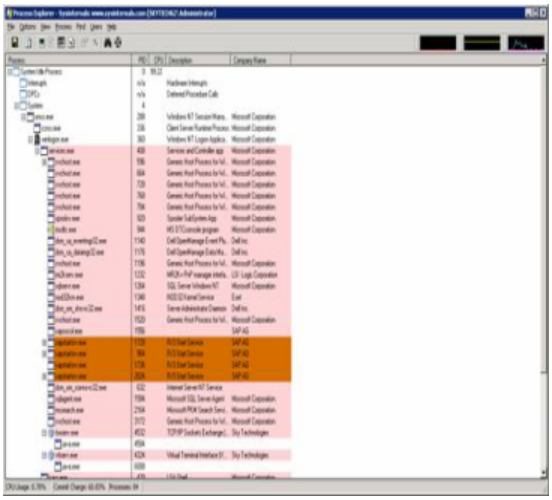
Platform	Method
Smart Phones / PDAs: • Windows Mobile • Blackberry • Symbian	Use either the "self test" facility or the "about" option to display the current connectivity and / or attributes of the server and / or presentation client.

5.6.8.8 Monitoring Java Server Processes

You may use the following techniques to monitor Java Server processes. This is OS specific.

Windows

The Sky Java Server applications run continuously as Java JVMs. Typically each instance of a JVM runs as a separate Windows process. You can view the processes using a tool such as Task Manager or Process Explorer.



The Java Virtual Machines show up as either "java" or "jre" (if executing Java runtime environment).

UNIX

The Sky Java Server applications run continuously as Java JVMs. Typically, each instance of a JVM runs as a separate UNIX process. (In some UNIX environments, this may not be the case). Type the UNIX top command to view the current active processes:

```
UNIX Processes Log
```

```
System: Skynet2 Fri Aug 10 16:53:33 2001
Load averages: 1.41, 1.67, 1.67
254 processes: 237 sleeping, 13 running, 4 zombies
Cpu states:
CPU LOAD USER NICE SYS IDLE BLOCK SWAIT INTR SSYS
0 1.39 27.4% 0.0% 10.9% 61.6% 0.0% 0.0% 0.0% 0.0%
```

```
1 1.57 5.4% 0.0% 3.4% 91.3% 0.0% 0.0% 0.0% 0.0%
2 1.23 39.6% 0.2% 22.3% 38.0% 0.0% 0.0% 0.0% 0.0%
3 1.46 46.3% 0.4% 15.1% 38.2% 0.0% 0.0% 0.0% 0.0%
— ---- ----- ----- ----- ----- ----- ----- ----- -----
avg 1.41 29.6% 0.0% 12.9% 57.5% 0.0% 0.0% 0.0% 0.0%
Memory: 915240K (488196K) real, 1111880K (657740K) virtual, 201904K
free Page#
1/29
CPU TTY PID USERNAME PRI NI SIZE RES STATE TIME %WCPU %CPU COMMAND
0 ? 23576 orasky 154 20 22720K 2428K sleep 21:04 35.83 35.77 oracleSky
3 ? 23563 skyadm 236 20 166M 152M run 13:04 29.00 28.95 dw.sapSky_D
3 ? 2434 root 154 20 64968K 5424K sleep 43:11 12.20 12.18 dm_fc_scsi_
1 ? 27793 orasky 154 20 22592K 1340K sleep 3:35 7.58 7.57 oracleSky
3 ? 26430 orasky 154 20 22720K 2428K sleep 1:31 6.45 6.44 oracleSky
3 ? 17179 orasky 154 20 28864K 5500K sleep 1:54 6.02 6.01 oracleSky
3 ? 18681 skyadm 155 20 37216K 23336K sleep 3:33 3.99 3.98 dw.sapSky_
```

The Java Virtual Machines show up as either "java" or "jre" (if executing Java runtime environment). It is not possible from this display to differentiate between the Sky Java servers. Type "ps -ef | grep java (or jre)" to get the Java process details and "ps ef | grep {product id} to associate the startup shell script with the corresponding Java process.

Example

```
+ $ ps -ef|grep java skyadm 14459 14958 11 Aug 7 ? 260:45
/opt/java1.2/jre/bin/../bin/PA- skytech 12828 11476 1 16:55:45 pts/tc
0:00 grep java
$ ps -ef|grep vti skytech 12937 11476 0 16:56:17 pts/tc 0:00 grep vti
skyadm 14958 1 0 Aug 7 ? 0:00 ./vti.sh ./vti.sh
```

Use the parent / child process IDs to match which java JVM processes were spawned by VTI. For example, Java process 14459 is spawned by process 14958 that is `./vti.sh`, that is, the VTI startup script.

5.6.9 Backup and Recovery (Windows only)

> *Important:* This option is only currently available for Windows platforms at this stage.

The in-built backup and recovery facility is a high performance general purpose tool designed to aid recovery in a situation where data is unexpectedly lost. The "BackupRestore" section drives it, within the configuration file. A typical entry may look something like this:

5.6.9.1 BackupRestore Configuration Section

```
Server.BackupRestore.BackupDirectory = backups
Server.BackupRestore.FullBackupItems = skymobile.cfg;storage
Server.BackupRestore.IncrementalBackupItems = storage/vti_number_range
Server.BackupRestore.AutoFullBackupTimes = 1:00:00
Server.BackupRestore.AutoIncrementalBackupInterval = 60
Server.BackupRestore.FullBackupRetentionCount = 2
Server.BackupRestore.UseCompression = true
Server.BackupRestore.CompressionAlgorithm = GZIP
```

When the Container performs a backup, it works through the list of relevant files and directories, writing their contents into a single backup file that is placed into the backup directory. There are two types of backups - full and incremental. A full backup always writes the contents of every file found to the backup file. Incremental backups only write files that changed since the last backup took place. A restore works in reverse. The Container looks for the most recent set of backup files and use them to reinstate the files and directories that are backed up, exactly as they were when the backup took place. In the above example configuration file section, the configuration file (`skymobile.cfg`) and the local database storage directory (`storage`) are backed up whenever a full backup takes place. The resulting backup files are written to a directory called `backups`. Incremental backups check the VTI_NUMBER_RANGE local database table (held in the `storage/vti_number_range` directory) for changes and then write an incremental backup file if any are found. In the normal course of events, a backup or restore is run by invoking `vti.bat` or `vti.exe` with a /backup or /restore option specified. You must stop the Container in order to do this. It is also possible to schedule the Container to automatically perform backups, either full or incremental. Obviously, it is undesirable to have a

backup that runs while data is altered, so if this feature is used, you should typically configure it to run at time(s) during which the Container is completely idle. In the above configuration file section example, a full backup runs daily at 1:00 am. An incremental backup is scheduled to take place every 60 seconds in the event of changes to the VTI_NUMBER_RANGE local database table. On the Windows Mobile platform, the command line is not readily accessible, so Start menu shortcuts are provided to facilitate this ("Backup" and "Restore"). The backup / restore facility is of particular relevance on this platform, since a "flat battery" condition commonly results in total loss of locally stored data. The usual strategy to overcome this risk is to ensure that backups are written to a directory on a storage card. In the event of a flat battery, Kony for SAP is reinstalled and then a restore is run, effectively recovering the data to the point at which the last backup is run.

5.6.10 Gateway Switchover

You may implement the Automatic gateway switchover where the host gateway connection may be switched to one or more secondary connections if the primary is not available. This is useful to automate disaster / recovery situations. Switch over is also useful to cater for load balancing where you may configure gateways to only accept a maximum number of connections; once this is reached, new connections are rejected and may be automatically switched over to an alternate gateway. The following `skymobile.cfg` configuration options control the chaining of primary / secondary gateway switchover options.

HostInterface:XmlClient

```
Server.HostInterface:XmlClient.HostInterfaceType = XmlClient
Server.HostInterface:XmlClient.XmlGatewayHosts =
192.168.2.62,192.168.2.133
Server.HostInterface:XmlClient.XmlGatewayHostSelection = Sequential
Server.HostInterface:XmlClient. XmlGatewayPort = 10000
Server.HostInterface:XmlClient.XmlGatewayResponseTimeout = 30
Server.HostInterface:XmlClient.XmlGatewayStallTimeout = 30
Server.HostInterface:XmlClient.UseCompression = true
Server.HostInterface:XmlClient.CompressionAlgorithm = GZIP
Server.HostInterface:XmlClient.UseEncryption = true
Server.HostInterface:XmlClient.EncryptionAlgorithm = RSA
Server.HostInterface:XmlClient.EncryptionKey =
Server.HostInterface:XmlClient.EncryptionKeyFile =
Server.HostInterface:XmlClient.EncryptionKeyHexEncoded = true
Server.HostInterface:XmlClient.EncryptionKeyStrength = 128
Server.HostInterface:XmlClient.HandshakeEncryptionAlgorithm = RSA
Server.HostInterface:XmlClient.HandshakeEncryptionKey =
Server.HostInterface:XmlClient.HandshakeEncryptionKeyFile = rsa_
public_encryption_key
Server.HostInterface:XmlClient.HandshakeEncryptionKeyHexEncoded = true
```

```
Server.HostInterface:XmlClient.XmlDebug = false
Server.HostInterface:XmlClient.UseTransferFields =
Server.HostInterface:XmlClient.MinimumNumberHandlers = 0
Server.HostInterface:XmlClient.MaximumNumberHandlers = 50
```

In the above example, the IP names / addresses of the primary (first) and secondary gateways are listed in order separated by commas. The 'sequential' selection method indicates that the list is tried in first to last sequence. You may also specify 'random' where any IP in the list is tried until one is successful.

5.7 Diagnostics and Tracing

5.7.1 Diagnostics and Tracing

This section provides information on the various mechanisms available to perform technical diagnostics on the system and trace internal processes. This may be necessary to help identify and resolve problems usually under instruction from Kony support.

5.7.2 Key Topics

MEAP Server Tracing
SAP Internal Tracing and Diagnostics
Secure Container Tracing

5.7.3 MEAP Server Tracing

5.7.3.1 About

These are the options available to trace communications and internal processing on the SAP host. All trace files are written to the `VTI` working directory as configured in the system defaults. If this is not configured (null), then tracing is ignored.

> ***Important:*** Some traces generate a large amount of output, so caution is necessary when using these options. Also remember to turn them off.

5.7.3.2 Key Topics

The MEAP Server Trace Facility
Data Object Trace
Local Database Trace
RFC Call Trace
MEAP Performance Trace
Session Manager Call Trace
The Heartbeat Trace

5.7.3.3 The MEAP Server Trace Facility

As this option suggests, it enables to optionally configure the following traces for a specific Server ID:

- MEAP session manager calls
- LDB traces
- DOB traces
- RFC calls
- Performance trace
- Heartbeat trace

To configure server trace entries, select the option from the workbench "**Utilities** > **Host Requests** > **Traces** > **Server trace utility**".

? Unknown Attachment

As per all Kony for SAP maintenance screens, you position your cursor and select the toolbar option or just double-click a line to change. The server group orders the display.

? Unknown Attachment

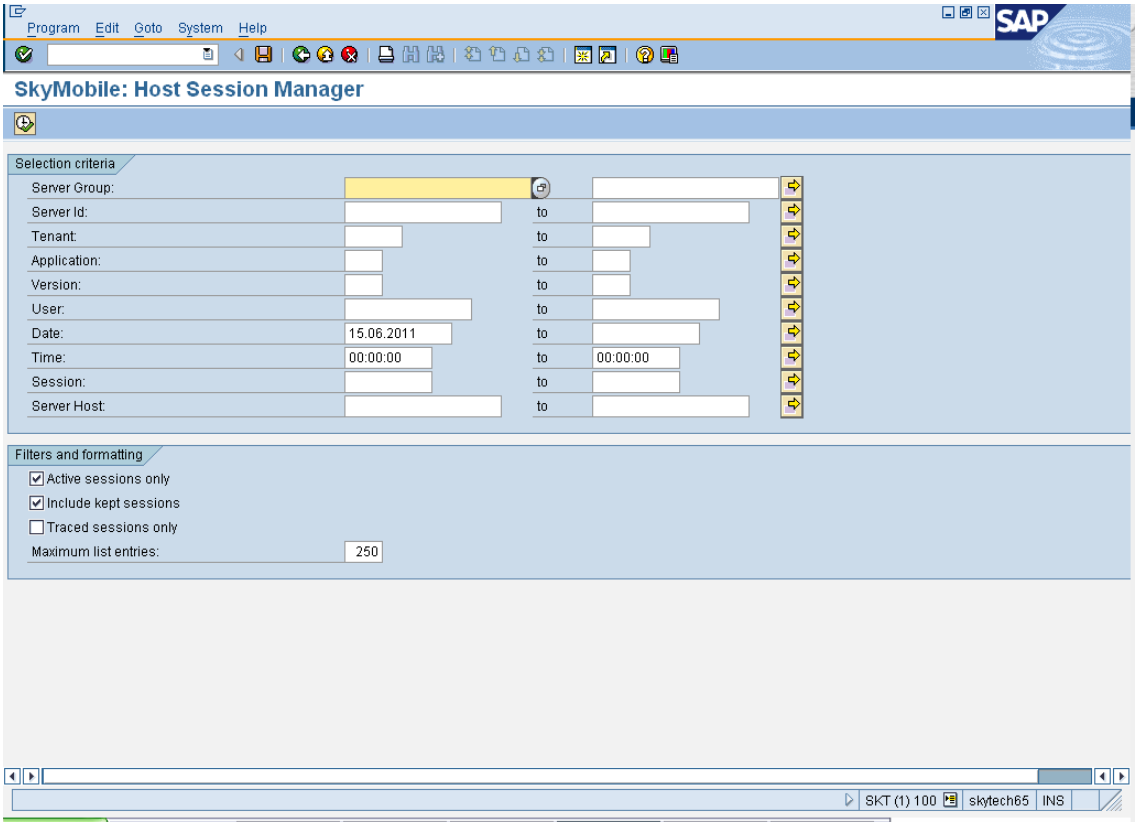
Each trace entry has an expiry date; the trace is automatically deactivated on that date. The retention governs how long the generated trace entries are kept within the context of the server trace. Select the traces required using the check boxes. To deactivate tracing, simply delete the definition or change the settings; that is expiry date or trace selection. The following sections describe each of the trace options in more detail.

> ***Note:*** The explicit trace settings override these trace settings at the system or definition level. You may branch to the relevant trace area by positioning your cursor and clicking the Sessions, DOB trace, LDB trace, RFC call trace or Performance trace buttons in the application toolbar.

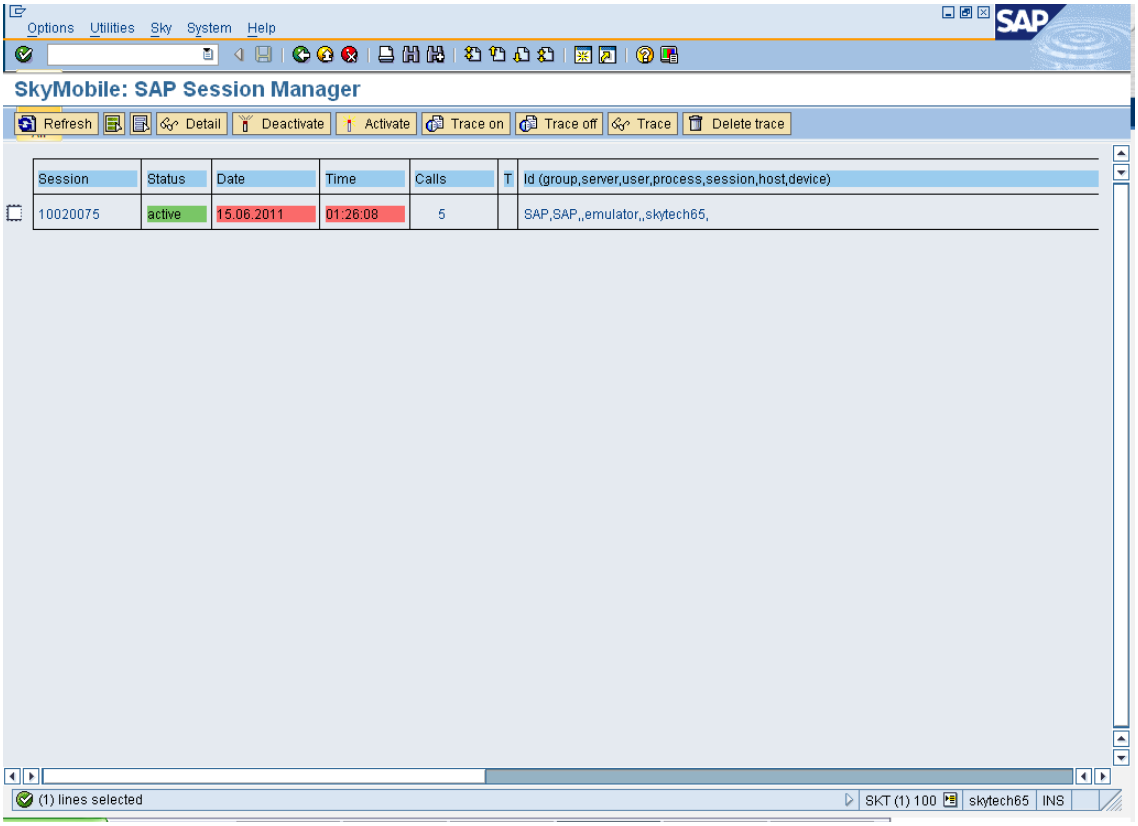
5.7.3.4 Session Manager Call Trace

This trace details the technical call information; data structures input and output to the MEAP host session manager. It also provides a flow trace of processing that breaks down the performance and time taken to process the various aspects of the call. The session trace is configured either using the server trace facility (above) or the MEAP host session manager facility. The host session manager is invoked using the **Workbench**-> **Utilities**-> **Host requests** ->**Session Manager** menu option or the transaction YVTS. A selection screen is used to restrict the display.

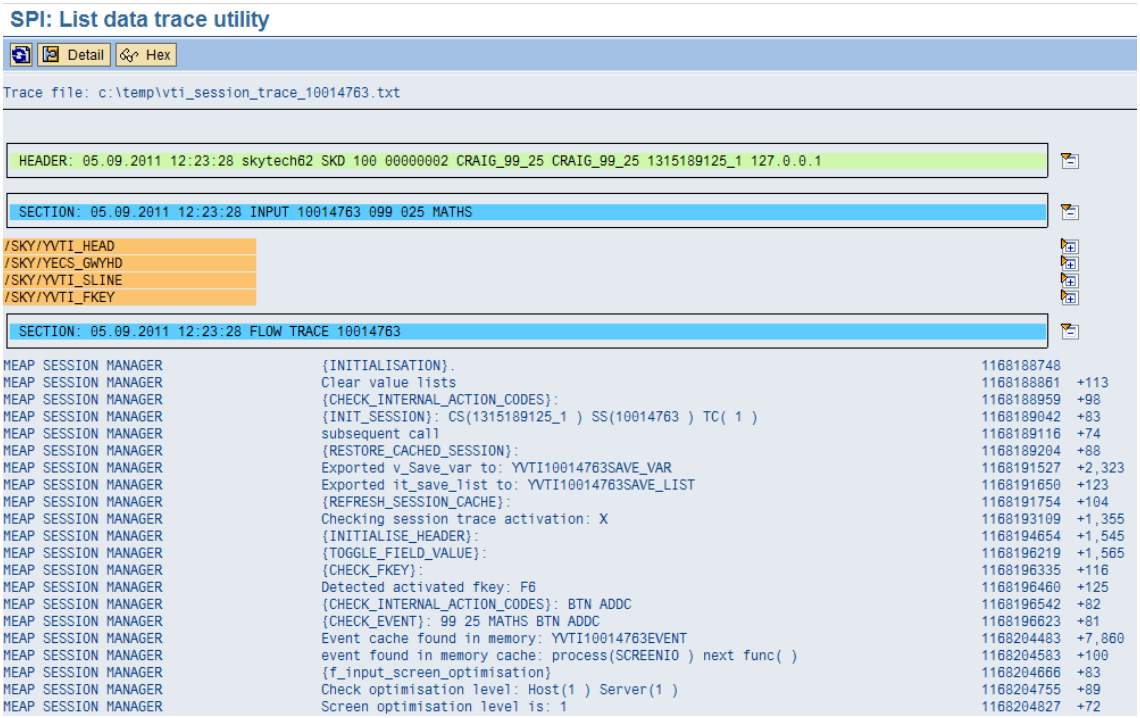
> ***Note:*** An alternative and more generic way to configure the session manager trace is through the server trace facility, using the Container (server) group and ID.



A list of sessions appear. The trace is activated by positioning your cursor or selecting the check box and clicking the **Trace on** button in the toolbar. The trace shows it is activated with a highlighted 'T' and that trace data exists by 'glasses' symbol on the right-hand side.



Click the glasses symbol to display the detailed trace list. The list is broken into sections that show the input data, flow trace and output data. Any custom procedure flow trace entries are also included. You may view the detailed data structures by double-clicking any line. Multiple calls to the MEAP sessions manager are listed separately. You can expand / collapse each one using the icon on the right-hand side.

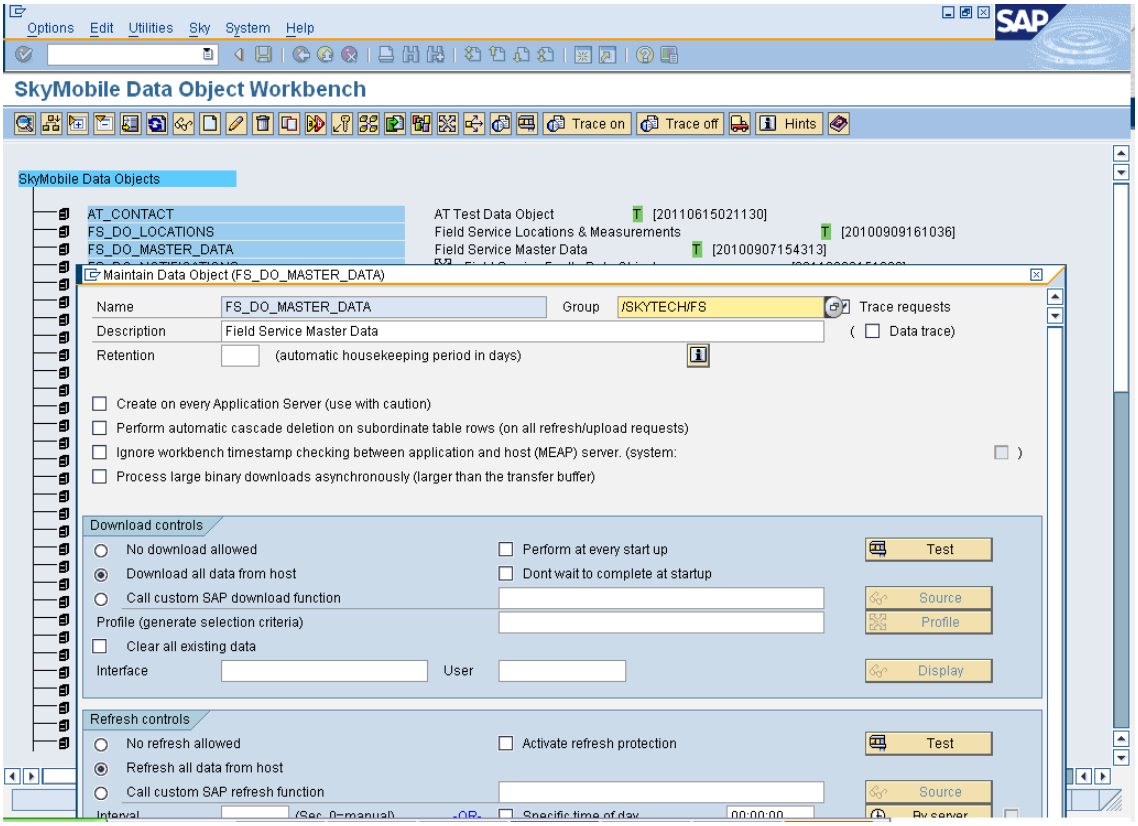


The +nnn readings on the right-hand side of the flow trace are the processing steps in milliseconds (millionths of a second) from the previous step, that is, the time taken to execute the previous step. A total run time is given at the end of the flow trace.

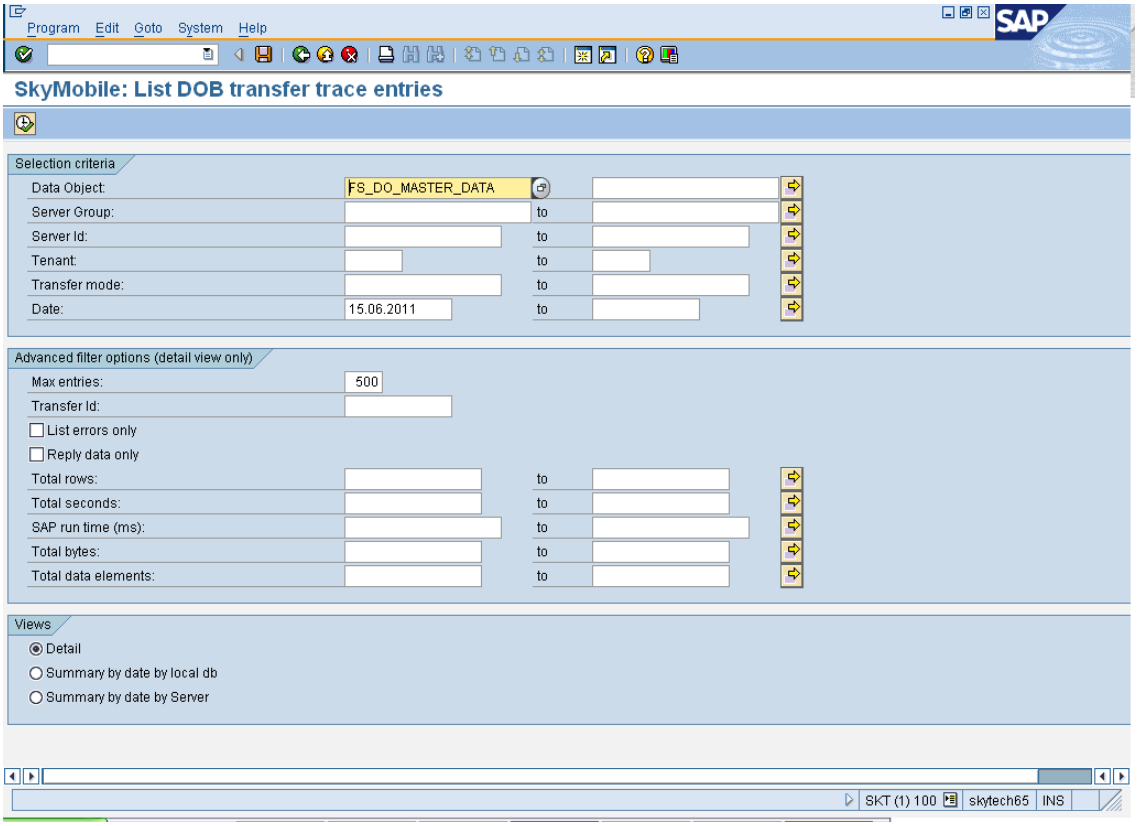
5.7.3.5 DOB Trace

The Data Object (DOB) trace is an effective way to see the frequencies and volumes of DOB requests. The trace is activated and deactivated for each DOB using the SAP data object manager that you may start directly using transaction YVTD or through the workbench (YVTW) **Utilities** menu and application toolbar options. There are two levels of trace, the high level statistical trace entry and a detailed data trace. When activated, a highlighted green 'T' appears next to the DOB name if just the high level trace is activated or a 'D' if the detailed data level trace is activated. The T / D indicators are also 'hotspots' to automatically launch the trace section screen. Traces are written out as files to the Add-in working directory and are automatically maintained by the Add-in housekeeping routine depending on the system and / or server trace retention periods.

> ***Note:*** You may also dynamically set the LDB trace using the server trace facility DOB trace settings are also applicable to the underlying database tables (LDBs), that is, if tracing is specified at the data object level; it is applied to all of its table references. This is so that all the technical information is available for diagnosis. You can drill-down from the DOB detailed trace entry to display all the related LDB calls.



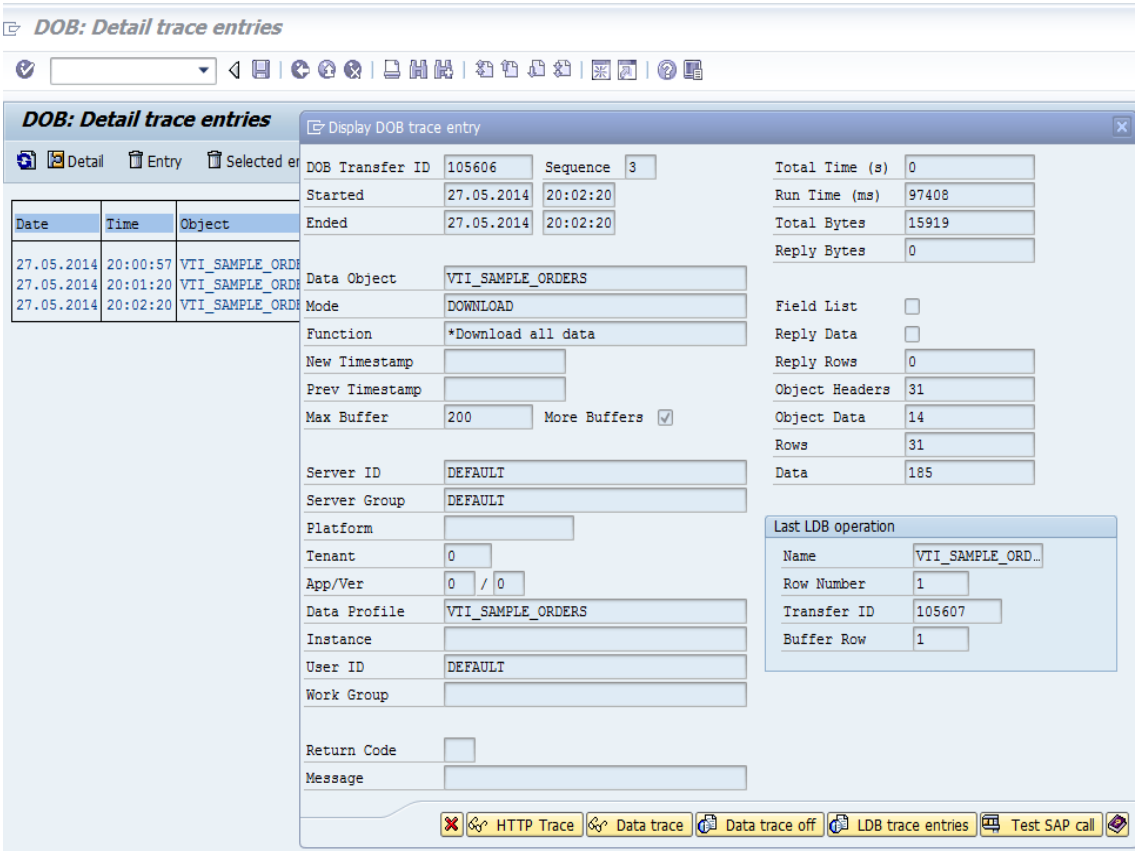
To list and manage trace entries, you invoke the trace utility from either the T / D hotspots adjacent to the data object or through the **Utilities** menu. A selection screen allows you to filter the listed entries and select the type of report you want.



Detailed Report

The trace detail report provides information on all DOB download, refresh and upload requests. It shows the total bytes transferred and the time taken.

> ***Note:*** A separate entry is written for every buffered request. You may double-click a line to display the details.



Field	Description
DOB Transfer ID	Internal ID for the DOB trace entry
Sequence	Denotes the sequence number of the current trace entry if the data is buffered
Started	Start date and time of the DOB Operation
Ended	End date and time of the DOB Operation
Data Object	Data Object (DOB) whose trace captured by the current trace entry
Mode	Operation on the DOB that was traced (DOWNLOAD/UPLOAD/REFRESH)

Field	Description
Function	Short description of the DOB Mode
New Timestamp	Timestamp value generated for the current DOB operation for the Server Id
Prev Timestamp	Timestamp value for the previous DOB operation for the Server Id
Max Buffer	Maximum buffer size of the data transferred
More Buffers	Indicates if more buffers are available to read
Server ID	Server Id of the device invoking the DOB operation
Server Group	Group name of the device invoking the DOB operation
Platform	Logical platform of the calling device
Tenant	Tenant for the User ID
App/Ver	Application and version for which the DOB operation occurred
Data Profile	Data Profile associated with the DOB
Instance	Instance of the Data Profile
User ID	User name with which the DOB operation is invoked
Work Group	Work Group for the User ID

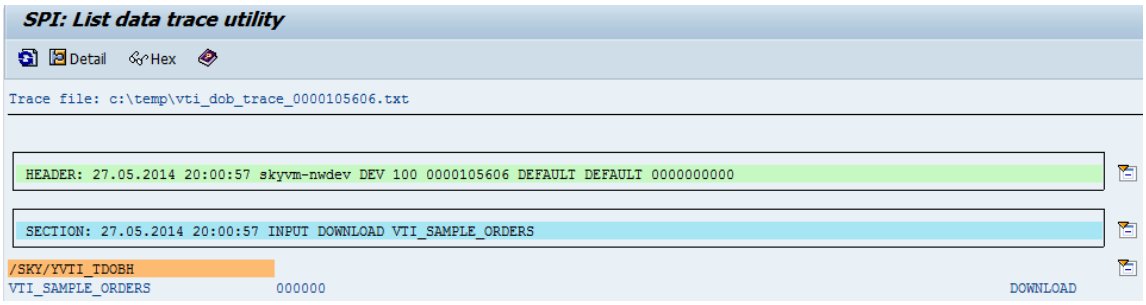
Field	Description
Return Code	Return Code of the DOB operation
Message	Message pertaining to the Return Code
Total Time (s)	Total time in seconds for the complete round trip for the DOB operation
Run Time (ms)	Execution time in milliseconds for the actual DOB operation processing
Total Bytes	Size of the data transferred during the DOB operation invocation
Object Headers	Number of rows of Object headers transferred as part of the DOB operation. Used to transfer data fields > 40 characters.
Object Data	Number of rows of Object Data transferred as part of the DOB operation. This is accessed using offsets present in the Object Header table
Rows	Number of Logical data records transferred as part of the DOB operation
Data	Total number of table data entries (Rows * number of fields in a record) transferred as part of the DOB operation

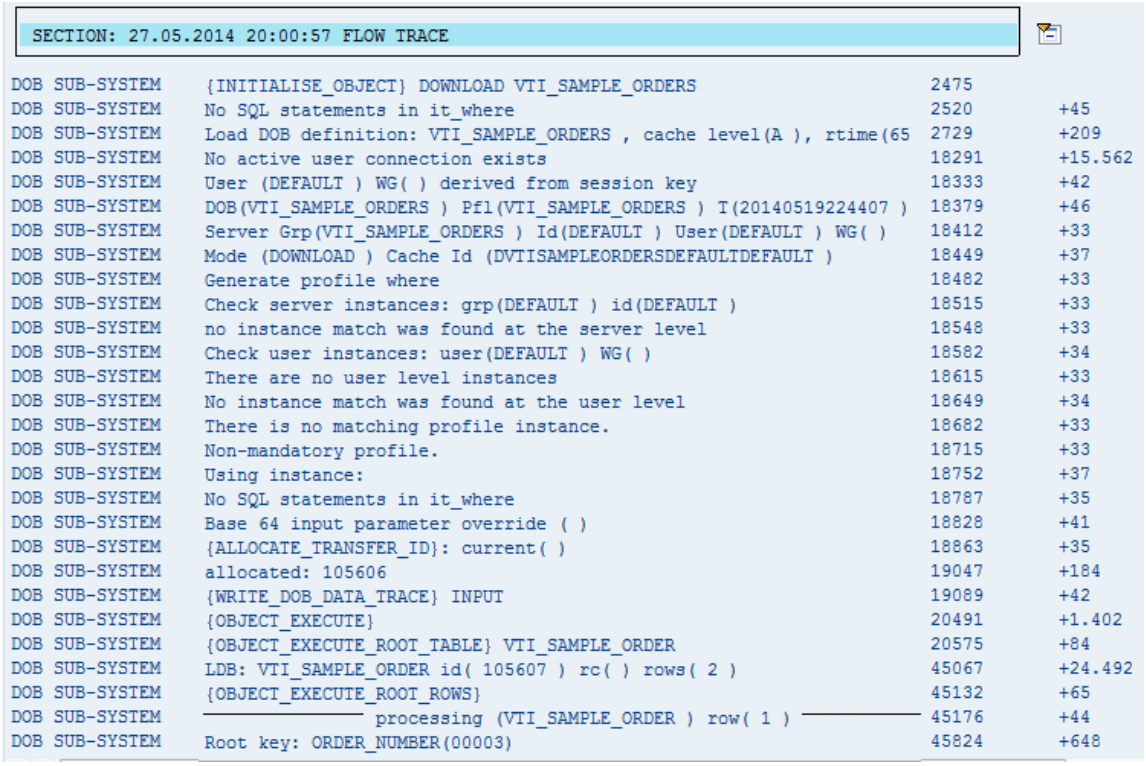
The Last LDB operation section has the following fields:

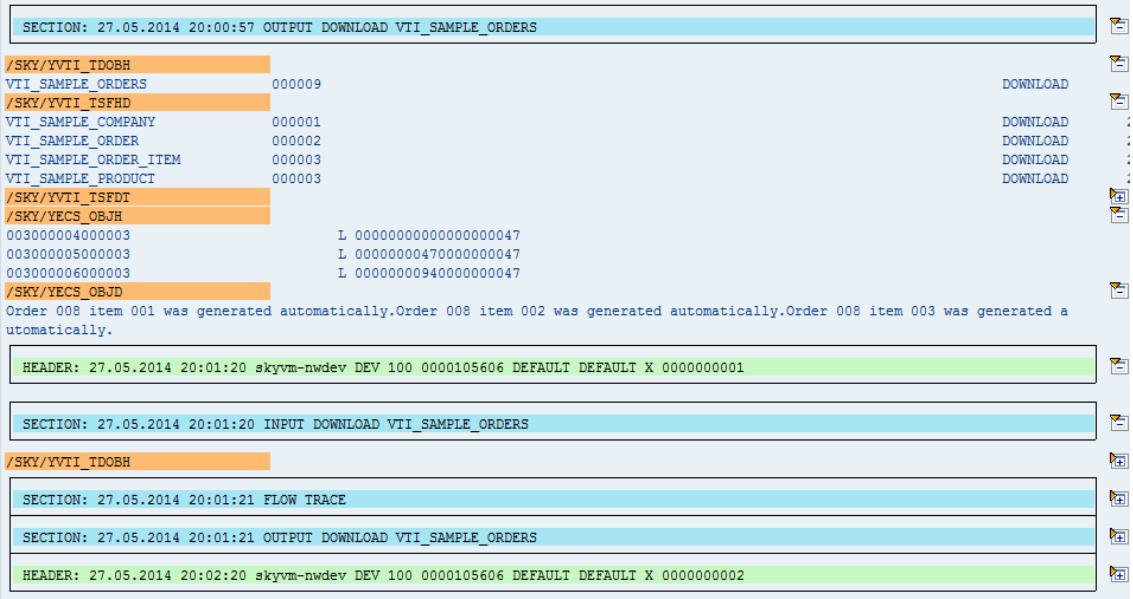
Field	Description
Name	In case of additional buffers to be read, this denotes the LDB that is last saved in the current buffer.
Transfer ID	Internal Id for the last LDB trace entry in the current buffer

Field	Description
Buffer Row	Number of logical rows of the LDB in the current buffer

a. If a data level trace exists, then you may click **Hex** to display the details. You can also view the related LDB trace entries by clicking the **LDB trace entries** button.







The data trace listing displays the technical structures that are passed between the Java Server and the host. The header (/sky/yvti_tdobh) contains control attributes about the call, followed by the LDB tables and data elements / object information. Double-click any line for a structured display of the data.

SPI: List structure components (/SKY/YVTI_TDOBH)

OBJECT_NAME	C 32	VTI_SAMPLE_ORDERS
TOTAL_ROWS	N 6	000000
RC	C 3	
MSG	C 79	
TRNSFR_MOD	C 15	DOWNLOAD
LASTTMSTMP	C 14	
VTI_SERVER	C 15	DEFAULT
MAXBUFFER	N 10	0000000200
DOBTRANSFERID	N 10	0000105606
MOREBUFFER	C 1	
MOREBUFFERSEQ	N 10	0000000000
APPID	N 3	000
VERID	N 3	000
SERVER_GROUP	C 32	DEFAULT
SAPTRACE	C 1	X
SAPDATATRACE	C 1	X
SAPRUNTIME	N 15	000000000000000
WBTIMESTAMP	C 14	
TOTAL_BYTES	N 10	0000000000
LASTLDB	C 32	
LASTLDBROW	N 6	000000
LASTLDBTID	N 10	0000000000
LASTLDBBUF	N 6	000000

b. If any operation is performed on a DOB as part of a HTTP service invocation, you can view the trace entries of this service call by clicking the corresponding HTTP Trace **Hex** button. This redirects to the HTTP Services Trace Report where you may view the trace entries for the HTTP Service call.

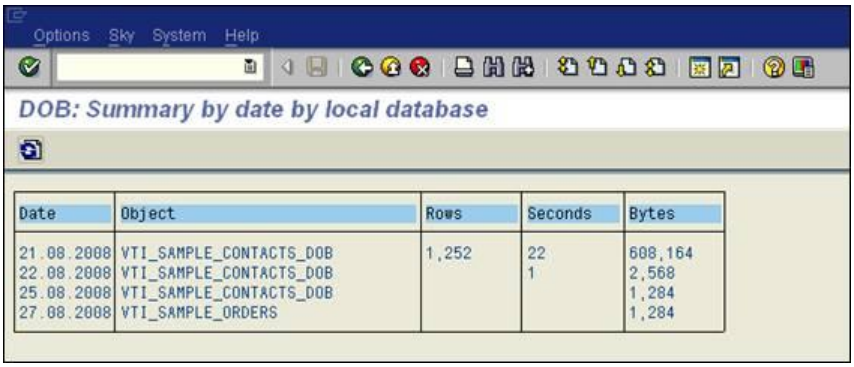
HTTP: Detail trace entries

Detail | Entry | Selected entries

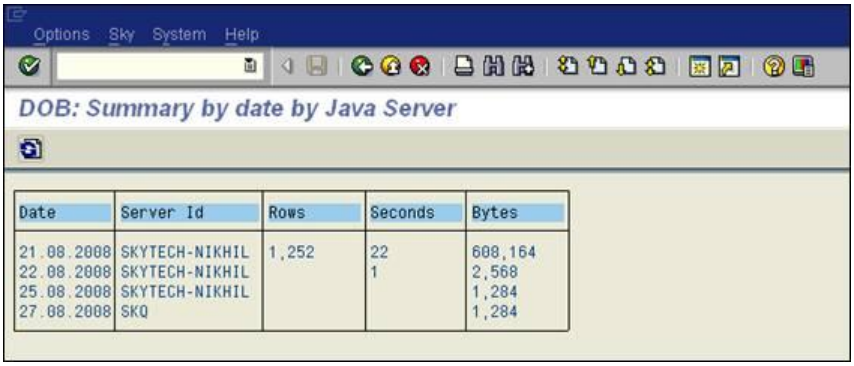
Date	Time	Transfer ID	Service	Method	Server ID	Seconds	SAP Runtime	Total Bytes	RC
27.05.2014	20:00:57	0000105605	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	212	1906	200

Summary Report

The summary reports provide condensed statistics, for example, by date by local database. These reports are useful to analyze the DOB traffic, that is, volumes and frequencies. In these scenarios, you should configure tracing on all DOB definitions for a day or more and then summarize the details to see how frequently the SAP host is requested.



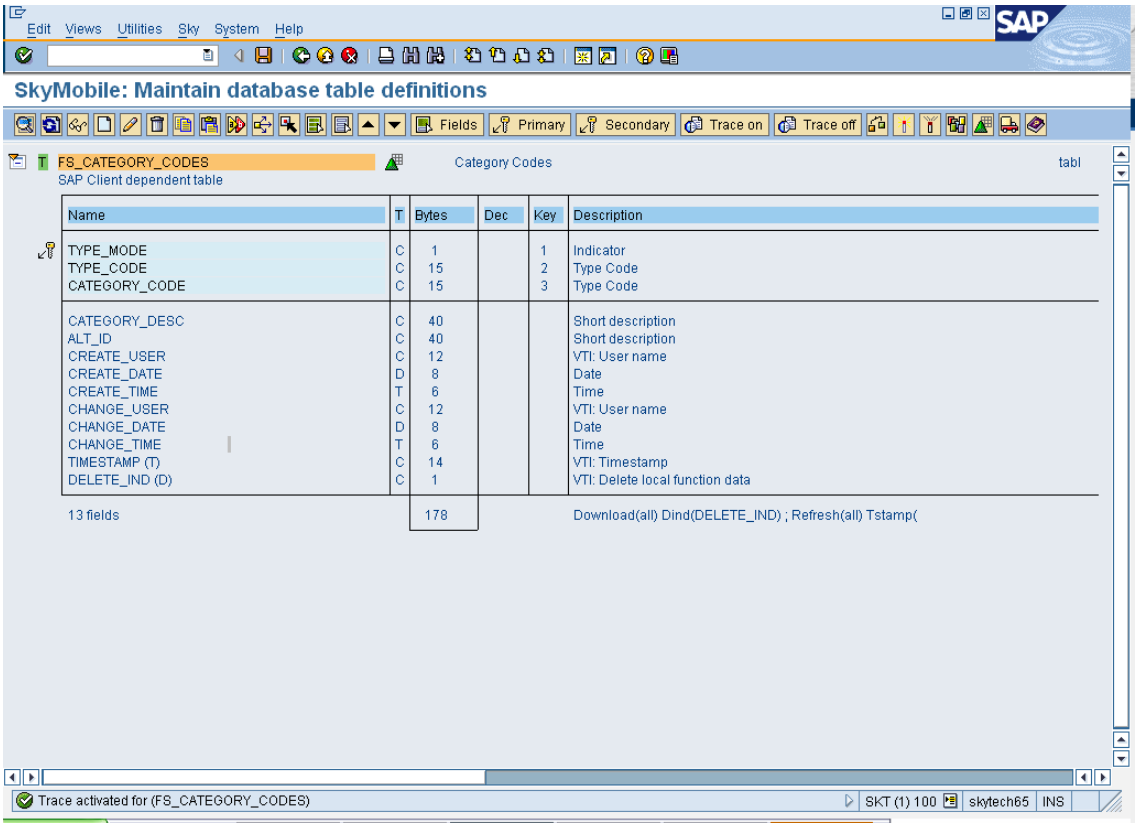
Or by date by VTI Java server ID:



5.7.3.6 LDB Trace

The Local Database (LDB) trace is an effective way to see the frequencies and volumes of LDB requests. The trace is activated and deactivated for each LDB using the SAP local database manager that you may start directly using transaction YVTL or through the workbench (YVTW) **Utilities** menu and application toolbar options. There are two levels of trace, the high level statistical trace entry and a detailed data trace. When activated, a highlighted green 'T' appears next to the LDB name if just the high level trace is activated or a 'D if the detailed data level trace is activated. The T / D indicators are also 'hotspots' to automatically launch the trace section screen. Traces are written out as files to the SkyMobile working directory and are automatically maintained by the Sky Mobile housekeeping routine depending on the system and / or server trace retention periods.

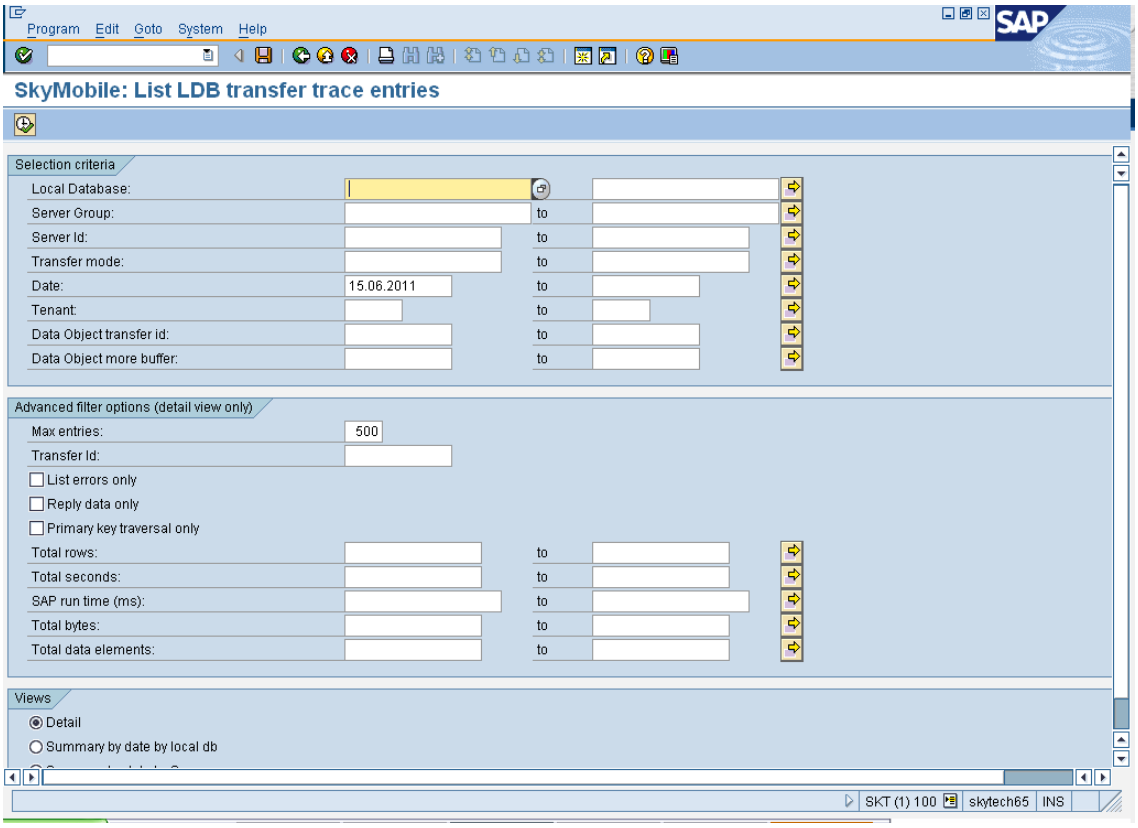
> ***Note:*** You may also dynamically set the LDB trace using the server trace facility.



You may use the trace attributes against the LDB header definition, or trace on and trace off application toolbar options, to toggle whether the trace is activated or deactivated. To manage and view the LDB trace, use the **Utilities** menu -> **Local database trace** option. The following options are available:

Option	Description
Activate trace all	Activates the trace for all local databases
Deactivate trace	Deactivates the trace for all local databases
Clear trace entries	Selectively clears the trace depending on criteria
List trace entries	Invokes a facility to selectively view LDB trace entries

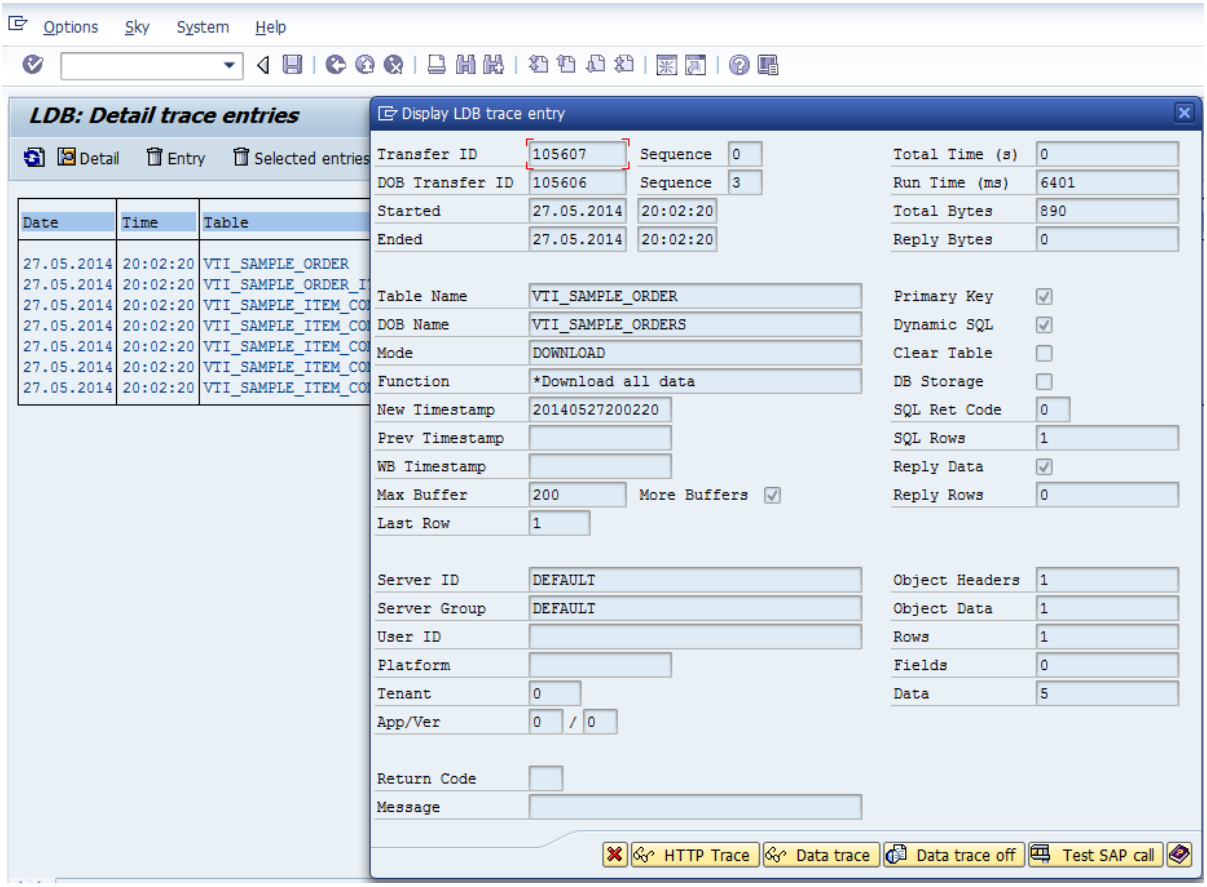
When listing trace entries, a selection screen allows you to restrict the entries and the type of report:



Detailed Report

The trace detail report provides information on all LDB download, refresh and upload requests. It shows the total bytes transferred and the time taken.

> ***Note:*** A separate entry is written for every buffered request. You may double-click a line to display the details.



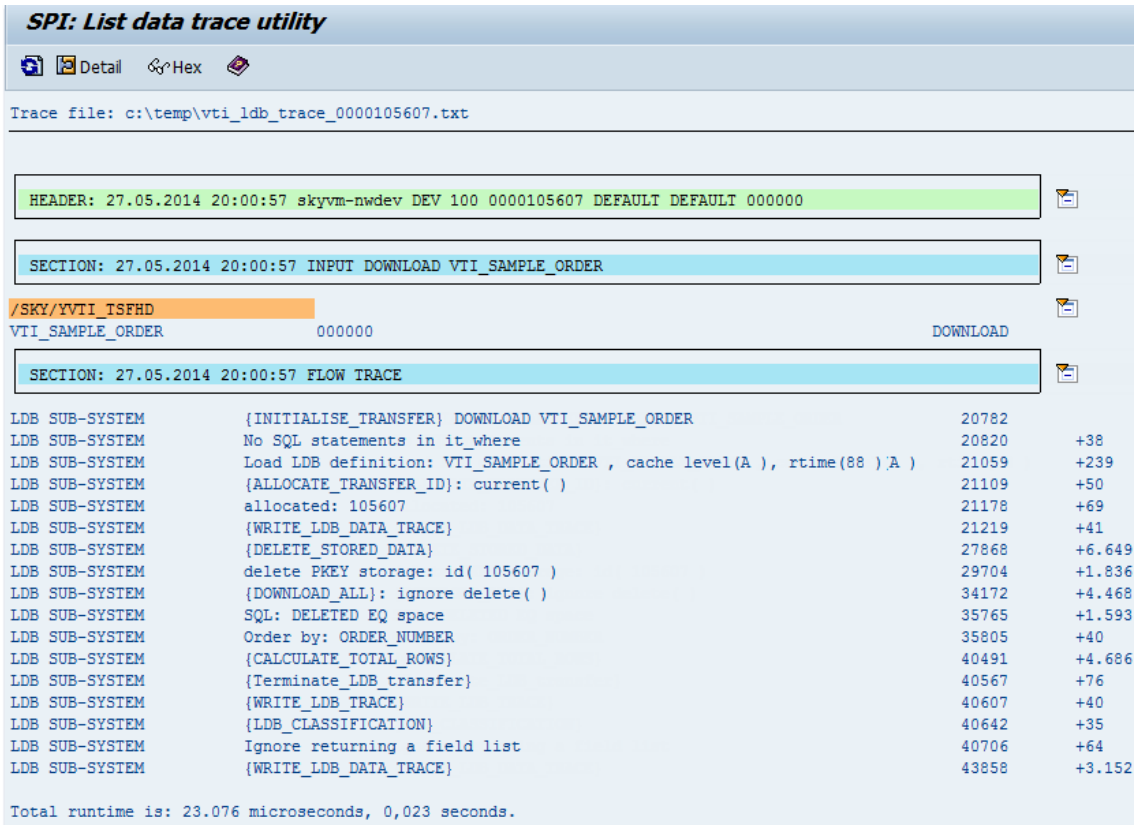
Field	Description
Transfer ID	Internal ID for the LDB trace entry.
Sequence	Denotes the sequence number of the current LDB trace entry if the data is buffered

Field	Description
DOB Transfer ID	Internal ID for the associated DOB trace entry
Sequence	Denotes the sequence number of the associated DOB trace entry if the data is buffered
Started	Start date and time of the DOB Operation
Ended	End date and time of the DOB Operation
Table Name	LDB Table Name whose trace is captured by the current trace entry
DOB Name	Associated data object for the LDB
Mode	Operation on the LDB that was traced (DOWNLOAD/UPLOAD/REFRESH)
Function	Short description of the LDB Mode
New Timestamp	Timestamp value generated for the current LDB operation for the Server Id
Prev Timestamp	Timestamp value generated for the previous LDB operation for the Server Id
WB Timestamp	Workbench timestamp. Refers to the timestamp of the most recent change of the LDB definition
Max Buffer	Maximum buffer size of the data transferred
More Buffers	Indicates if more buffers are available to read.

Field	Description
Last Row	Last logical record number of the data in the current buffer
Server ID	Server Id of the device invoking the LDB operation
Server Group	Group name of the device invoking the LDB operation
Platform	Logical platform of the calling device
Tenant	Tenant for the User ID
App/Ver	Application and version for which the LDB operation occurred
Return Code	Return code of the LDB operation
Message	Message pertaining to the Return Code
Total Time (s)	Total time in seconds for the complete round trip for the LDB operation
Run Time (ms)	Execution time in milliseconds for the actual LDB operation processing
Total Bytes	Size of the data transferred during the LDB operation invocation
Primary Key	Indicates if primary key traversal is used for the LDB operation
Object Headers	Number of rows of object headers transferred as part of the LDB operation. Used to transfer data fields > 40 characters.

Field	Description
Object Data	Number of rows of object data transferred as part of the LDB operation. This is accessed using offsets present in the Object Header table
Rows	Number of logical data records transferred as part of the DOB operation
Data	Total number of table data entries (Rows * number of fields in a record) transferred as part of the LDB operation

a. The data trace listing displays the technical structures that are passed between the Java Server and the host. The header (/sky/yvti_tsfhd) contains control attributes about the call, followed by the data elements and field information. Double-click any line for a structured display of the data.



SPI: List structure components (/SKY/YVTI_TSFHD)

LOCAL_DB	C 32	VTI_SAMPLE_ORDER
TOTAL_ROWS	N 6	000000
RC	C 3	
MSG	C 79	
TRNSFR_MOD	C 15	DOWNLOAD
LASTTMSTMP	C 14	
INTERNAL1	N 10	0000000000
CLRCNTENTS	C 1	
VTI_SERVER	C 15	DEFAULT
FLDROW_ST	N 6	000001
FLDROW_EN	N 6	000008
DATAROW_ST	N 6	000000
DATAROW_EN	N 6	000000
REPLY_DATA	C 1	X
MAXBUFFER	N 10	0000000200
LASTBUFROW	N 6	000000
TRANSFERID	N 10	0000105607
MOREBUFFER	C 1	
APPID	N 3	000
VERID	N 3	000
SERVER_GROUP	C 32	DEFAULT
SAPTRACE	C 1	X
SAPDATATRACE	C 1	X
SAPRUNTIME	N 15	000000000000000

b. If any operation is performed on a LDB as part of a HTTP service invocation, you can view the trace entries of this service call by clicking the corresponding HTTP Trace **Hex** button.
The **HTTP: Detail trace entries** window appears where you may view the trace entries for the HTTP Service call.

HTTP: Detail trace entries

Detail | Entry | Selected entries

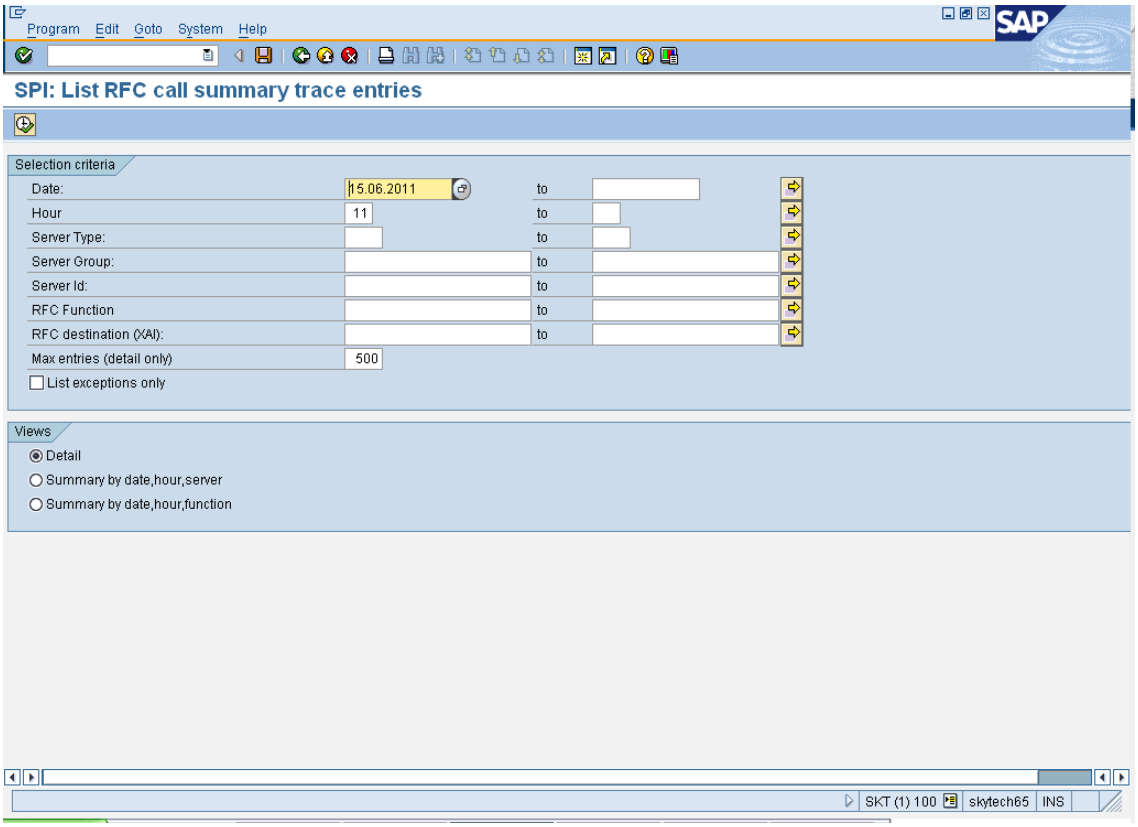
Date	Time	Transfer ID	Service	Method	Server ID	Seconds	SAP Runtime	Total Bytes	RC
27.05.2014	20:00:57	0000105605	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	212	1906	200

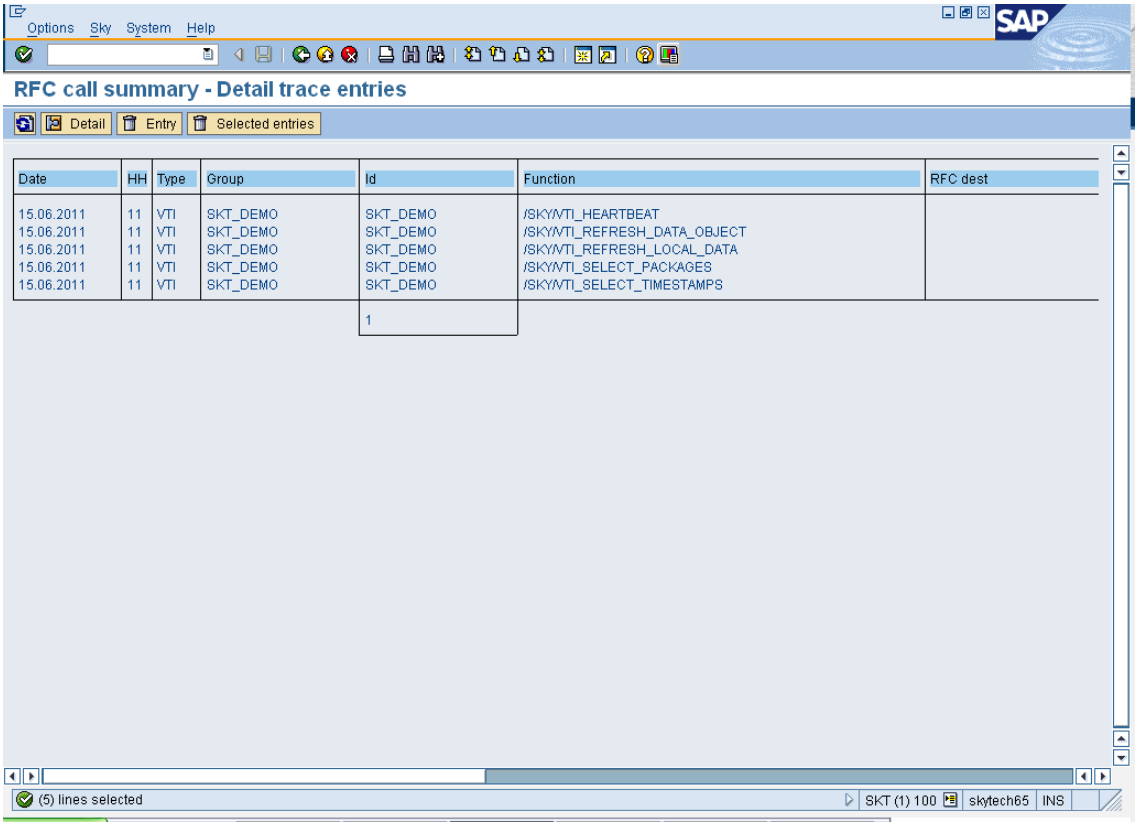
Summary Reports

The summary reports provide condensed statistics, for example, by date by local database. These reports are useful to analyze the LDB traffic, that is, volumes and frequencies. In these scenarios, you should set tracing on all LDB definitions for a day or more and then summarise the details to see how frequently the SAP host is requested.

5.7.3.7 RFC Call Trace

This trace summarizes all the RFC calls made to the SAP host. It is used primarily to identify frequencies and performance bottlenecks and tune appropriately. The RFC call trace is set either using the server trace facility (above), at the system level (setup configuration) or the **Workbench** > **Utilities** > **SAP Host** > **RFC call trace** menu option. A selection screen is used to restrict the display and specify different reporting options, for example, detailed, or summaries of the collected data.





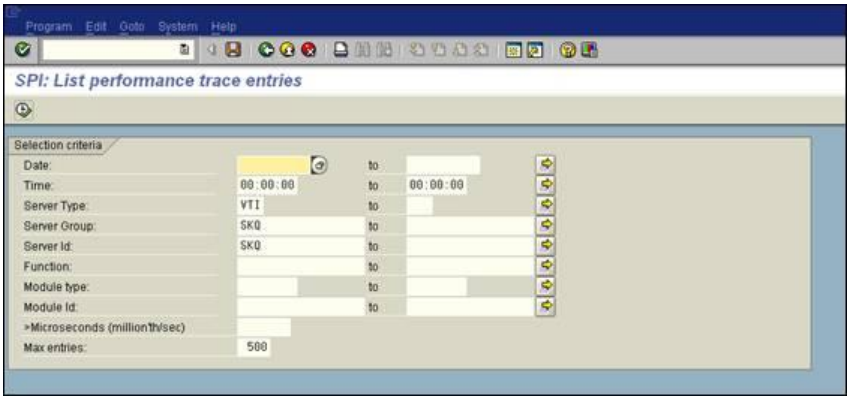
By default, the detailed list summarizes all RFC calls to the SAP Host by date and hour. The run time is in milliseconds (a thousandth of a second).

5.7.3.8 MEAP Performance Trace

This trace monitors all Kony functions, custom functions and programs called by a Secure Container. It is used to:

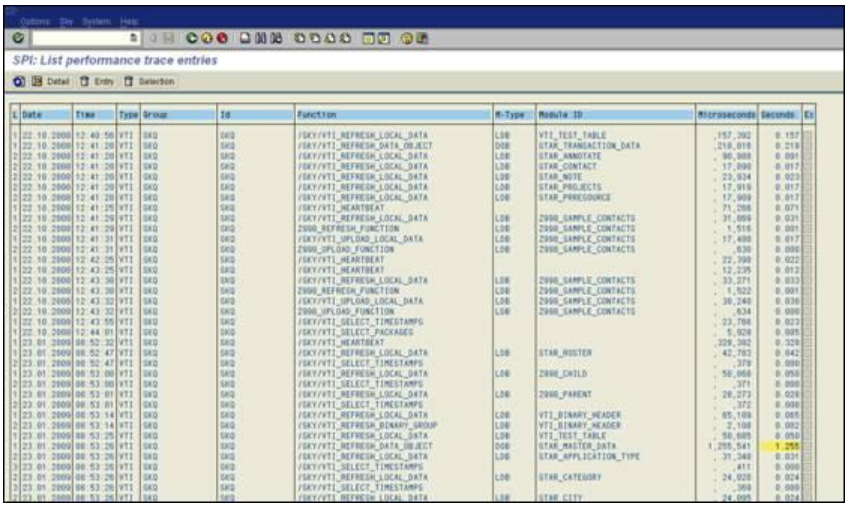
- Highlight areas that perform badly and need to tune. These are usually custom programs / function modules, data object (DOB) or local database (LDB) calls.
- Monitor the frequency of calls to the SAP host, for example, heartbeat, data synchronization, session manager.

The performance trace is configured either using the server trace facility (above) or the **Workbench** > **Utilities** > **SAP Host** > **Performance trace** menu option. A selection screen is used to restrict the display.



A list is generated showing each call. Nested calls are identified by the level (L) column on the left hand side and the time taken for each call is shown in both microseconds (millionths) and seconds.

> ***Note:*** The top level (for example,1) timings are inclusive of the sub-levels (nested calls).

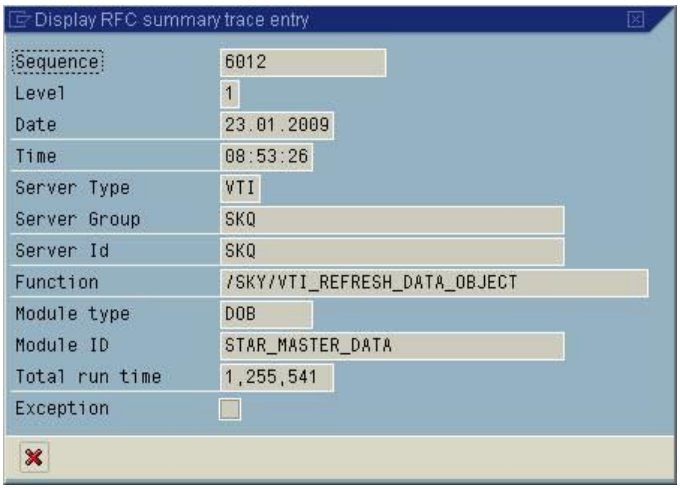


The module type (M-Type) is used to identify LDB (local database), DOB (data object) and EXIT (ABAP program) type calls and the module id is used to identify the specific name.

Colors are used to help highlight excessive call times using the seconds:

- > 0.5Light blue
- >1.0Yellow
- > 5Red

You may double-click any line to display the recorded details:



5.7.3.9 The heartbeat Trace

You may switch on the MEAP heartbeat trace facility globally through the Container management console configuration, or you may switch on for specific Containers through the MEAP Server trace facility. The later is recommended because of the large amount of data generated in a busy system. When activated, a highlighted "T" appears against the Secure Container in the management console list. You can click it, or invoke the trace list through the **Utilities** menu. The trace works in exactly the same way as the LDB and Data Object traces in that a basic list appears from which you double-click to display the detail and the actual input / output data and processing flow trace (if required and specified). The heartbeat trace is normally used to diagnose Container / User association problems, that is, what is passed up to the MEAP server, the processing assumptions and what is returned back.

5.7.4 SAP Internal Tracing and Diagnostics

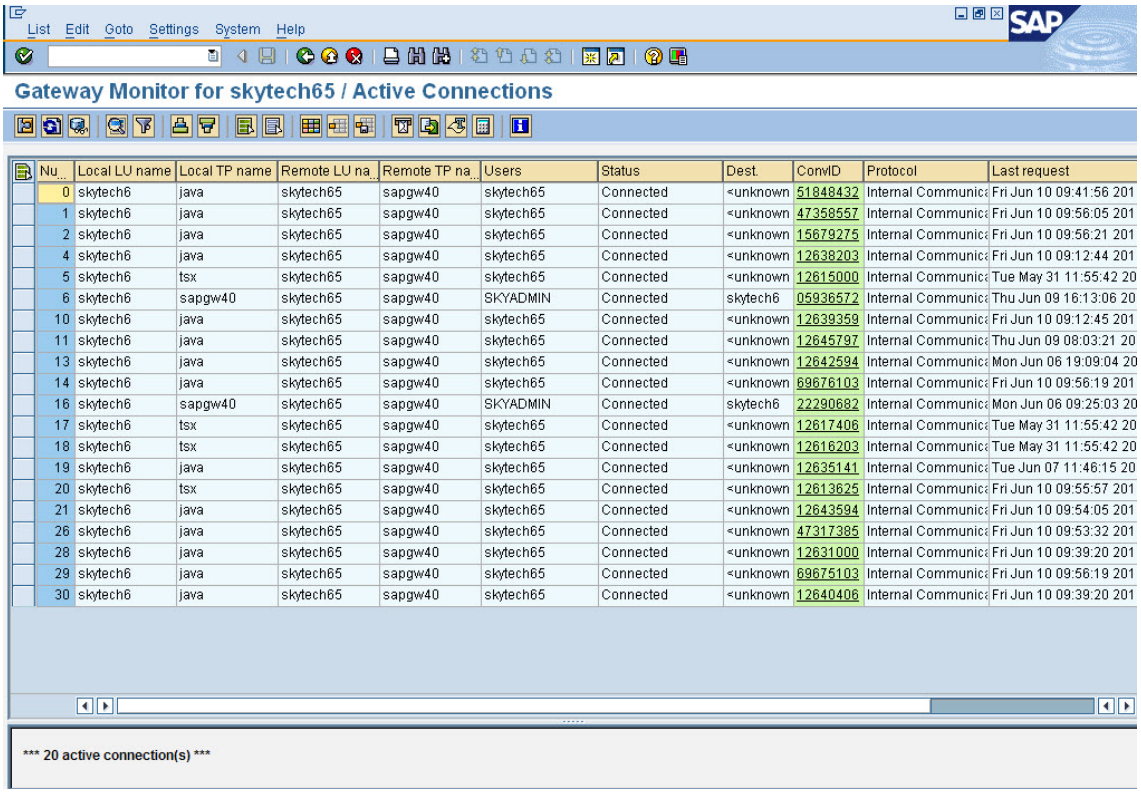
As well as the standard tools, there are several SAP tools that you may use to help diagnose more complex issues with RFC communication and connections. This section outlines some of the tools and techniques available. For more comprehensive documentation, refer to SAP documentation.

Monitoring the SAP Gateway

Viewing and Monitoring Gateway and Server Connections in SAP

5.7.4.1 Monitoring the SAP Gateway

SAP transaction SMGW displays active RFC client and registered connections for a specific SAP gateway. In a multi-instance SAP environment, you must log onto the specific SAP host where the gateway executes. Execute SMGW and the following screen appears:

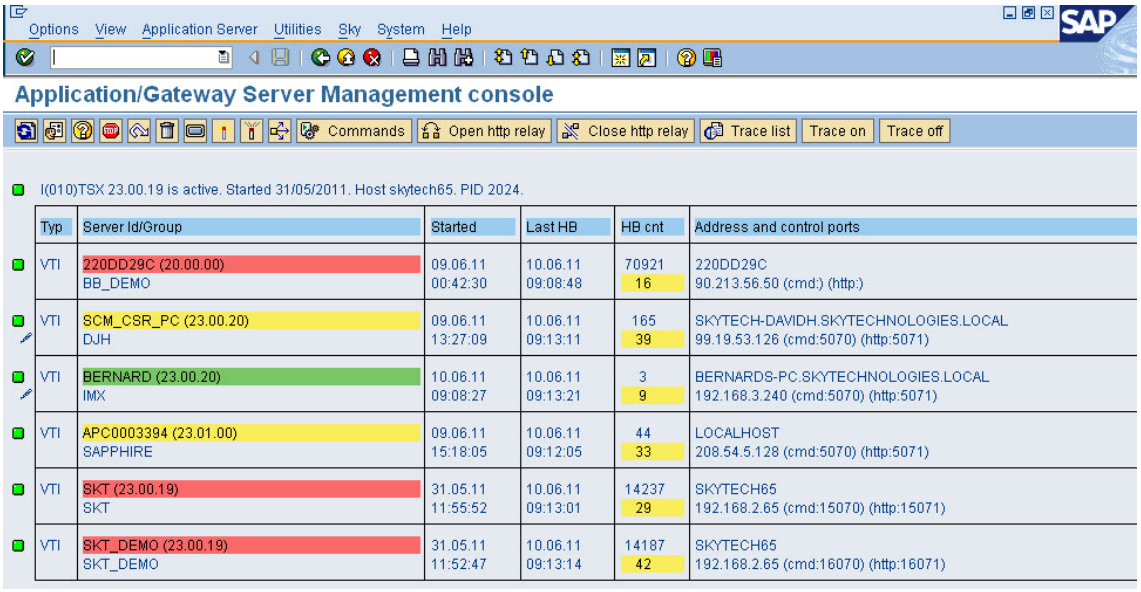


Nu...	Local LU name	Local TP name	Remote LU na...	Remote TP na...	Users	Status	Dest.	ConvID	Protocol	Last request
0	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	51848432	Internal Communic	Fri Jun 10 09:41:56 201
1	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	47358557	Internal Communic	Fri Jun 10 09:56:05 201
2	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	15679275	Internal Communic	Fri Jun 10 09:56:21 201
4	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12638203	Internal Communic	Fri Jun 10 09:12:44 201
5	skytech6	tsx	skytech65	sapgw40	skytech65	Connected	<unknown	12615000	Internal Communic	Tue May 31 11:55:42 20
6	skytech6	sapgw40	skytech65	sapgw40	SKYADMIN	Connected	skytech6	05936572	Internal Communic	Thu Jun 09 16:13:06 20
10	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12639359	Internal Communic	Fri Jun 10 09:12:45 201
11	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12645797	Internal Communic	Thu Jun 09 08:03:21 20
13	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12642594	Internal Communic	Mon Jun 06 19:09:04 20
14	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	69676103	Internal Communic	Fri Jun 10 09:56:19 201
16	skytech6	sapgw40	skytech65	sapgw40	SKYADMIN	Connected	skytech6	22290682	Internal Communic	Mon Jun 06 09:25:03 20
17	skytech6	tsx	skytech65	sapgw40	skytech65	Connected	<unknown	12617406	Internal Communic	Tue May 31 11:55:42 20
18	skytech6	tsx	skytech65	sapgw40	skytech65	Connected	<unknown	12616203	Internal Communic	Tue May 31 11:55:42 20
19	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12635141	Internal Communic	Tue Jun 07 11:46:15 20
20	skytech6	tsx	skytech65	sapgw40	skytech65	Connected	<unknown	12613625	Internal Communic	Fri Jun 10 09:55:57 201
21	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12643594	Internal Communic	Fri Jun 10 09:54:05 201
26	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	47317385	Internal Communic	Fri Jun 10 09:53:32 201
28	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12631000	Internal Communic	Fri Jun 10 09:39:20 201
29	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	69675103	Internal Communic	Fri Jun 10 09:56:19 201
30	skytech6	java	skytech65	sapgw40	skytech65	Connected	<unknown	12640406	Internal Communic	Fri Jun 10 09:39:20 201

This screen displays all current client connections. Those marked with a TP name of 'Java' are those initiated usually by a Gateway or directly from Application Servers.

5.7.4.2 Viewing and Monitoring Gateway and Server Connections in SAP

Every Access Gateway and Application Server that has an SAP connection publishes a 'heartbeat' with the designated SAP system. This heartbeat request logs the server group / ID, IP address and other attributes. You may view these connections using the /SKY/YECJ or YECJ transaction. This can be extremely useful to centrally manage remote servers directly from within SAP. For more detailed information on remote device management and this option, refer to Systems Management.

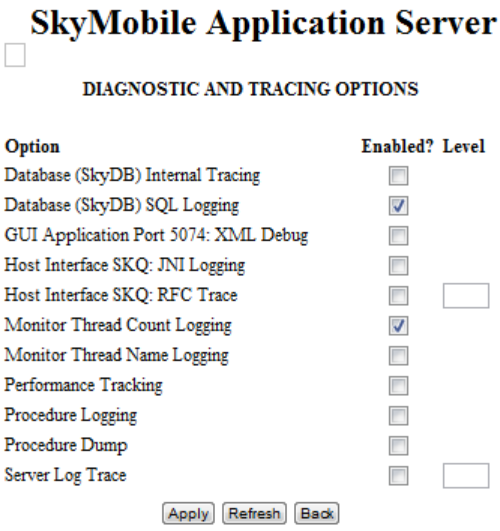


In addition to manually monitoring connections, you may implement a scheduled task to automatically monitor Gateway and Server connections to a particular SAP system. This runs as a ECS Schedule that monitors the heartbeats and reports any errors, that is, those connections that have missed two or more consecutive heartbeats.

5.7.5 Secure Container Tracing

There are several technical traces available to help support diagnose issues at the Application Server and presentation client end. As you can appreciate, subtly different traces are implemented depending on the platform, for example, Windows Mobile, BlackBerry, and IOS.

You can enable traces either through editing the configuration, or through the web status page. The web status page allows the diagnostics to turn on and off whilst the application server runs. This page is available upon clicking the **Diagnostics** button on the main web status page.



> ***Important:*** Some traces generate a large amount of output, so caution is necessary when you use these options. Also remember to turn them off.

5.7.5.1 Key Topics

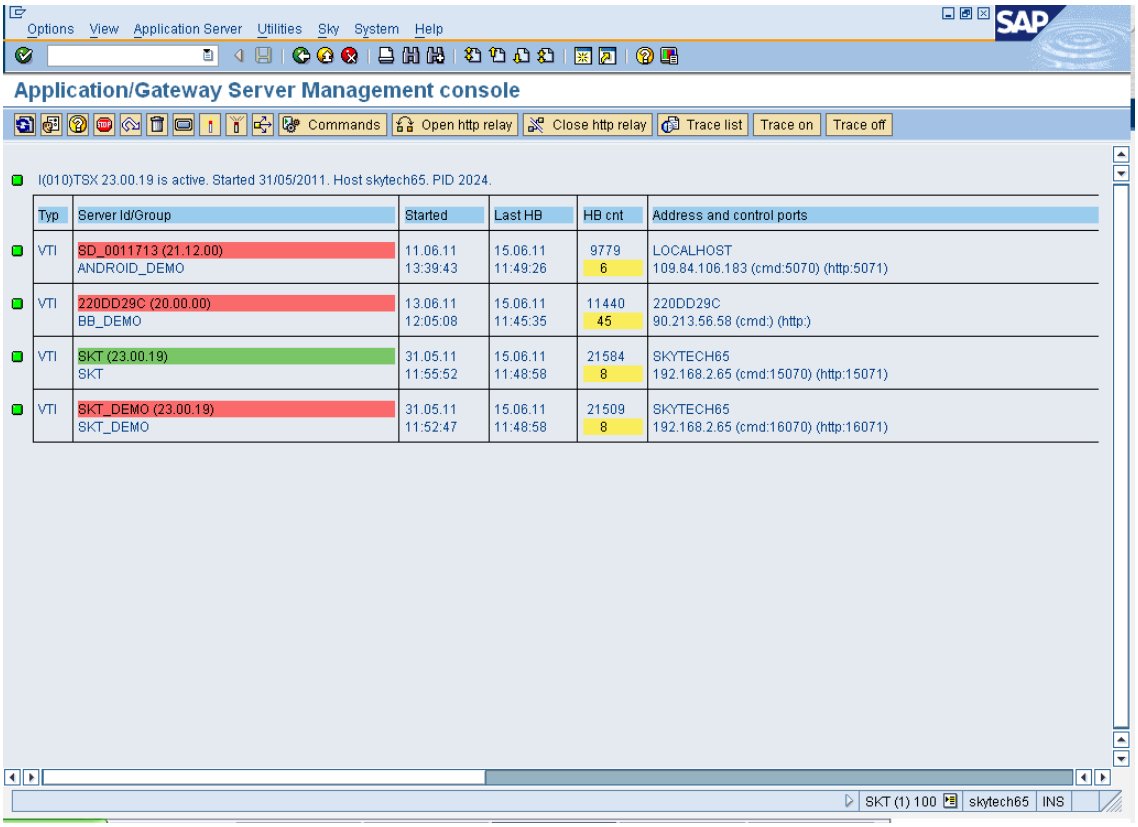
System Log
Performance Trace
SAP RFC Trace
JNI Trace
Gateway XML Trace

Presentation Client XML Trace
Custom Java Exit Trace
Device about Screen
Device Self Test

5.7.5.2 System Log

Each application server writes all its information, warning, error and abort messages to a central system log. The log file name, location and retention characteristics are controlled by the log configuration options specified in the application server configuration file. The log is the first place to go if an application server or application is experiencing problems. The log is a text file that you may view by any text editor, for example, Notepad, WordPad, and VI (Unix). The Web status page also has a facility to view the log that is useful to view the log of remote servers. The download log option is also extremely useful to retrieve log information from a remote device. When an application server is first started, it writes its configuration to the log file before starting up its services.

A summary error and warning count of system log entries is passed up to the MEAP server with every heartbeat command. This information appears in the Application Server Management Console to help highlight potential problems. You may reset these counters using the main web status page.



System Log

```
07/11/2007 14:23:39 INFO: [VtiLdbUpdater10] Finished refresh of local
database table VTI_BINARY_HEADER for host interface SkyPrdRfcClient.
07/11/2007 14:23:40 INFO: [VtiLdbUpdater10] Skipping upload of local
database table VTI_SAMPLE_CONTACTS for host interface SkyPrdRfcClient
(no data to send).
07/11/2007 14:23:41 INFO: [VtiLdbUpdater10] Skipping upload of local
database table VTI_TEST_TABLE for host interface SkyPrdRfcClient (no
data to send).
07/11/2007 14:24:10 INFO: [VtiLdbUpdater10] Skipping upload of local
database table VTI_SAMPLE_CONTACTS for host interface SkyPrdRfcClient
(no data to send).
07/11/2007 14:24:11 INFO: [VtiLdbUpdater10] Skipping upload of local
database table VTI_TEST_TABLE for host interface SkyPrdRfcClient (no
data to send).
07/11/2007 14:24:14 INFO: [VtiLdbUpdater10] Started refresh of local
database table VTI_SAMPLE_CONTACTS for host interface SkyPrdRfcClient.
07/11/2007 14:24:14 INFO: [VtiLdbUpdater10] Finished refresh of local
database table VTI_SAMPLE_CONTACTS for host interface SkyPrdRfcClient.
07/11/2007 14:24:16 INFO: [VtiLdbUpdater10] Started refresh of local
database table VTI_TEST_TABLE for host interface SkyPrdRfcClient.
07/11/2007 14:24:16 INFO: [VtiLdbUpdater10] Finished refresh of local
database table VTI_TEST_TABLE for host interface SkyPrdRfcClient.
```

Each log entry has the following format:
{date} {time} {severity} {thread/service} {text}

> ***Note:*** You can control the system messages that are written to the log, using configuration options, that is, suppress information (INFO) messages, and garbage collection messages.

5.7.5.3 Application Server Performance Trace

You can set on a performance level trace in both the presentation client configuration (for example, `wpc.cfg`) or the Java Server configuration (`skymobile.cfg`). This causes low level timings of the internal operations to be written to the log files. You can use this data to highlight where most of the time is spent, for example, LDB requests, SAP host calls, and custom Java exits. Some examples of a WPC performance trace and a Secure Container performance trace follow:

Windows Presentation Client

```
01/10/2009 15:57:34 PERF: [New Timer: REQUEST] Starting screen request
(current function name: ADDITEMSTOPRINT, requested fkey: ENTER).
Time to prepare and send request to the server
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 19 ms, Delta: 19
ms] Finished XML construction.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 34 ms, Delta: 15
ms] Starting request compression.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 54 ms, Delta: 20
ms] Finished request compression.
Received a message from the server only. Still waiting for screen
response.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 68 ms, Delta: 14
ms] Request sent.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 508 ms, Delta: 440
ms] Received server response.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 548 ms, Delta: 40
ms] Processing screen response.
01/10/2009 15:57:34 PERF: [Timer: REQUEST, Elapsed: 563 ms, Delta: 15
ms] Received message only. Waiting for response.
Time to receive and process (decompression, XML parsing, etc ) of
screen response.
01/10/2009 15:57:36 PERF: [Timer: REQUEST, Elapsed: 2130 ms, Delta:
1567 ms] Received server response.
01/10/2009 15:57:36 PERF: [Timer: REQUEST, Elapsed: 2144 ms, Delta: 14
```

```
ms] Processing screen response.
01/10/2009 15:57:36 PERF: [Timer: REQUEST, Elapsed: 2211 ms, Delta: 67
ms] Finished screen response processing.
Time to layout the screen.
01/10/2009 15:57:36 PERF: [New Timer: RENDERING] Started rendering the
screen.
01/10/2009 15:57:36 PERF: [New Timer: LAYOUT] Started laying out the
screen.
01/10/2009 15:57:36 PERF: [Timer: LAYOUT, Elapsed: 25 ms, Delta: 25
ms] Finished laying out the screen.
01/10/2009 15:57:36 PERF: [Timer: RENDERING, Elapsed: 157 ms, Delta:
157 ms] Finished rendering the screen.
```

Application Server

```
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: New Tracker:
SESSION1254376308_1] Started processing message received from client.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 8ms Delta: 8ms] Message from client has
now been fully received.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 15ms Delta: 7ms] Started decompressing
message received from client.
Determined that input is a screen request.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 23ms Delta: 8ms] Finished decompressing
message received from client.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 33ms Delta: 10ms] Processing message
received from client.
Started VTI session manager processing.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 42ms Delta: 9ms] Receiving screen
request.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
```

```
SESSION1254376308_1 Elapsed: 51ms Delta: 9ms] Started Java session
manager processing.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 63ms Delta: 12ms] Started processing for
selection of possible (drop-down/list) values.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 81ms Delta: 18ms] Finished processing for
selection of possible (drop-down/list) values.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 92ms Delta: 11ms] Started host
transaction processing.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 102ms Delta: 10ms] Bundling header
information for transmission to host.
Transaction will require processing against host. Prepare parameters
sent to the host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 118ms Delta: 16ms] Bundling function keys
for transmission to host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 129ms Delta: 11ms] Bundling screen tables
for transmission to host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 140ms Delta: 11ms] Bundling screen lines
for transmission to host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 216ms Delta: 76ms] Bundling LDB transfer
data for transmission to host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 294ms Delta: 78ms] Bundling object data
for transmission to host.
01/10/2009 15:57:29 PERF: [ConnectionProcessingThread14: Tracker:
SESSION1254376308_1 Elapsed: 303ms Delta: 9ms] Bundling RFC parameters
received from the host.
```

> ***Note:*** You can use the SAP host performance trace in conjunction to break down where the most processing time is spent in SAP.

5.7.5.4 SAP RFC Trace

This is the standard SAP RFC tracing facility and is only relevant for direct RFC host connections to SAP. Specify a value from 1-5 for increasing detail. The trace lists the data areas and is written to the current working directory, for example, `VTI\ devrfc001.trc`. For a detailed description of the trace facility, read the SAP RFC documentation.

Application Server Configuration

HostInterface:SAP

```
SERVER.HOSTINTERFACE:SKDRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SKDRFCCLIENT.JNILOGGING = false
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MINIMUMNUMBERHANDLERS = 3
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPHOST = demo.skytechnologies.com
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBGROUP =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBHOST =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPPASSWORD = FRED123
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPSYSTEM = 99
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPTRACELEVEL = 2
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPUSELB = 0
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPUSER = CPIC_VTI
```

You can configure the Application Server to automatically housekeep the SAP RFC trace files based on the following configuration options:

Monitor

```
SERVER.MONITOR.CLEARRFCTRACEFILES = true
SERVER.MONITOR.GARBAGECOLLECT = true
SERVER.MONITOR.LOGMEMORYUSAGE = true
SERVER.MONITOR.LOGTHREADCOUNT = true
SERVER.MONITOR.LOGTHREADNAMES = false
SERVER.MONITOR.MONITORINTERVAL = 60
SERVER.MONITOR.RFCTRACEEXPIRYDAYS = 3
SERVER.MONITOR.USEMONITOR = true
```

Trace File

```
**** Trace file opened at , SAP-REL 710,0,0 RFC-VER nU 3 880476 MT-SL
>>> RfcCallReceiveEx [4] ...
>>> RfcCallEx [4] ...
*> RfcCallEx
FUNCTION /SKY/VTI_REFRESH_LOCAL_DATA
handle = 4
parameter[0].name = I_HEADER
parameter[0].nlen = 8
parameter[0].addr = 0BBA0160
parameter[0].leng = 265
parameter[0].type = RFCTYPE_CHAR

tables[0].name = IT_DATA
tables[0].nlen = 7
tables[0].leng = 57
tables[0].type = RFCTYPE_CHAR
tables[0].newitab = 0
tables[0].itmode = RFC_ITMODE_BYREFERENCE
ItFill(tables[0].ithandle) = 0
ItLeng(tables[0].ithandle) = 57
```

5.7.5.5 JNI Trace

This activates the Java Native Interface trace. The JNI layer is responsible for all calls to the SAP RFC layer and other system functions. It provides a comprehensive listing of all parameters output and input. The trace is written to the Java Server system log; use the normal log display / download facilities to view the trace details.

JNI Logging Configuration

```
SERVER.HOSTINTERFACE:SKDRFCCLIENT.HOSTINTERFACETYPE = RfcClient
SERVER.HOSTINTERFACE:SKDRFCCLIENT.JNILOGGING = true
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MAINTAINHOSTCONNECTIVITY = true
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MAXIMUMNUMBERHANDLERS = 10
SERVER.HOSTINTERFACE:SKDRFCCLIENT.MINIMUMNUMBERHANDLERS = 3
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPCLIENT = 100
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPCONNECTIONTIMEOUT = 60
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPDESTINATION = VTI
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPHOST = demo.skytechnologies.com
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLANGUAGE = E
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBGROUP =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBHOST =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPLBSYSTEM =
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPPASSWORD = FRED123
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPSYSTEM = 99
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPTRACELEVEL = 0
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPUSELB = 0
SERVER.HOSTINTERFACE:SKDRFCCLIENT.SAPUSER = CPIC_VTI
```

The following displays sample log trace entries.

skytechnologies

SkyMobile Application Server

Show All | Go To Page | Next -> | Refresh | Back

LOG FILE MESSAGE LISTING (PAGE 1 OF 213)

Date	Time	Type	Thread	Thread
15/06/2011	00:00:29	INFO	VtiMonitor	Before garbage collect: total memory = 64950272, free memory = 37254464.
15/06/2011	00:00:29	INFO	VtiMonitor	After garbage collect: total memory = 64950272, free memory = 53188008.
15/06/2011	00:00:29	INFO	VtiMonitor	Number of active threads: 27.
15/06/2011	00:01:29	INFO	VtiMonitor	Before garbage collect: total memory = 64950272, free memory = 46512512.
15/06/2011	00:01:29	INFO	VtiMonitor	After garbage collect: total memory = 64950272, free memory = 57808920.
15/06/2011	00:01:29	INFO	VtiMonitor	Number of active threads: 27.
15/06/2011	00:01:50	INFO	VtiLdbUpdater6	Started refresh of data object STAR_TRANSACTION_DATA for host interface PRD.
15/06/2011	00:01:50	INFO	VtiLdbUpdater6	Finished refresh of data object STAR_TRANSACTION_DATA for host interface PRD.
15/06/2011	00:02:29	INFO	VtiMonitor	Before garbage collect: total memory = 64950272, free memory = 47968112.
15/06/2011	00:02:29	INFO	VtiMonitor	After garbage collect: total memory = 64950272, free memory = 58826160.
15/06/2011	00:02:29	INFO	VtiMonitor	Number of active threads: 29.
15/06/2011	00:02:45	INFO	VtiLdbUpdater6	Started refresh of data object STAR_CERT_TEST_DATA for host interface PRD.
15/06/2011	00:02:45	INFO	VtiLdbUpdater6	Finished refresh of data object STAR_CERT_TEST_DATA for host interface PRD.
15/06/2011	00:03:29	INFO	VtiMonitor	Before garbage collect: total memory = 64950272, free memory = 41235296.
15/06/2011	00:03:29	INFO	VtiMonitor	After garbage collect: total memory = 64950272, free memory = 53549744.

Show All | Go To Page | Next -> | Refresh | Back

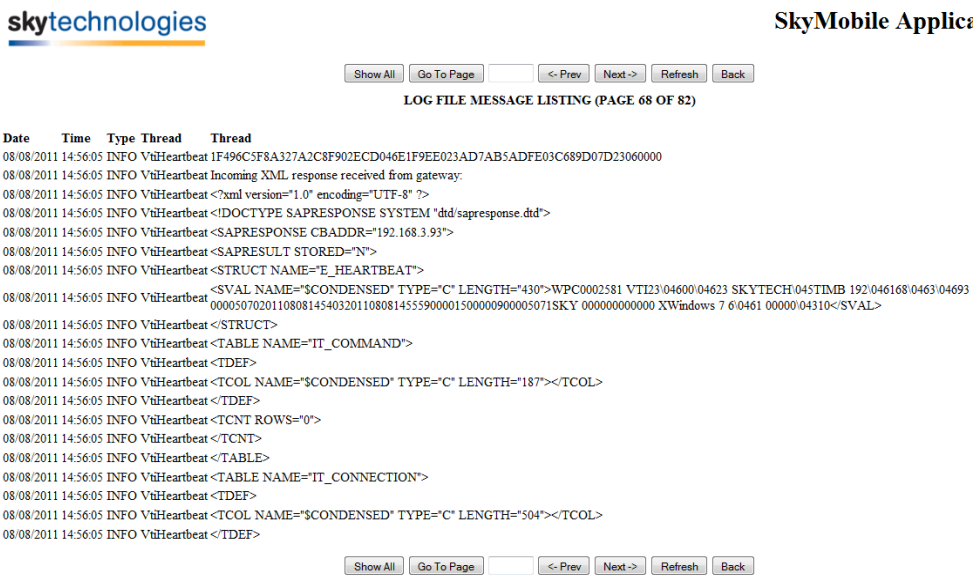
5.7.5.6 Gateway XML Trace

The gateway trace provides a comprehensive list of all the generated XML calls to and from the specified Gateway. The trace is written to the application server system log; use the normal log display / download facilities to view the trace details.

Application Server Configuration

```
SERVER.HOSTINTERFACE:SKYCLOUD.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:SKYCLOUD.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:SKYCLOUD.MAXIMUMNUMBERHANDLERS = 50
SERVER.HOSTINTERFACE:SKYCLOUD.MINIMUMNUMBERHANDLERS = 0
SERVER.HOSTINTERFACE:SKYCLOUD.USECOMPRESSION = true
SERVER.HOSTINTERFACE:SKYCLOUD.USEENCRYPTION = false
SERVER.HOSTINTERFACE:SKYCLOUD.XMLDEBUG = true
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYHOSTS =
demo.skytechnologies.com
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYHOSTSELECTION = Sequential
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYPORT = 40000
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYRESPONSETIMEOUT = 120
SERVER.HOSTINTERFACE:SKYCLOUD.XMLGATEWAYSTALLTIMEOUT = 120
```

The following displays sample log trace entries.



skytechnologies SkyMobile Applica

Show All | Go To Page | <- Prev | Next -> | Refresh | Back

LOG FILE MESSAGE LISTING (PAGE 68 OF 82)

Date	Time	Type	Thread	Thread
08/08/2011	14:56:05	INFO	VtiHeartbeat	1F496C5F8A327A2C8F902ECD046E1F9EE023AD7AB5ADFE03C689D07D23060000
08/08/2011	14:56:05	INFO	VtiHeartbeat	Incoming XML response received from gateway:
08/08/2011	14:56:05	INFO	VtiHeartbeat	<?xml version="1.0" encoding="UTF-8" ?>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<!DOCTYPE SAPRESPONSE SYSTEM "dtd/sapresponse.dtd">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<SAPRESPONSE CBADDR="192.168.3.93">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<SAPRESULT STORED="N">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<STRUCT NAME="E_HEARTBEAT">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<SVAL NAME="$CONDENSED" TYPE="C" LENGTH="430">WPC0002581 VTI23\04600\04623 SKYTECH\045TIMB 192\046168\0463\04693 0000507020110808145403201108081455590000150000090000 5071SKY 000000000000 XWindows 7 6\0461 00000\04310</SVAL>
08/08/2011	14:56:05	INFO	VtiHeartbeat	</STRUCT>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TABLE NAME="IT_COMMAND">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TDEF>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TCOL NAME="$CONDENSED" TYPE="C" LENGTH="187"></TCOL>
08/08/2011	14:56:05	INFO	VtiHeartbeat	</TDEF>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TCNT ROWS="0">
08/08/2011	14:56:05	INFO	VtiHeartbeat	</TCNT>
08/08/2011	14:56:05	INFO	VtiHeartbeat	</TABLE>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TABLE NAME="IT_CONNECTION">
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TDEF>
08/08/2011	14:56:05	INFO	VtiHeartbeat	<TCOL NAME="$CONDENSED" TYPE="C" LENGTH="504"></TCOL>
08/08/2011	14:56:05	INFO	VtiHeartbeat	</TDEF>

Show All | Go To Page | <- Prev | Next -> | Refresh | Back

5.7.5.7 Presentation Client XML Trace

The presentation clients have XML trace options that allow the calls between the Client and the application server to be output to either one or both of:

- the client screen through a pop-up
- the client log

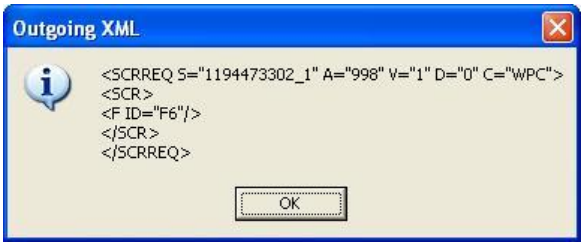
These options are controlled through the application server configuration file.

Configuration File

```
CLIENT.SERVER.COMPRESSIONALGORITHM = GZIP
CLIENT.SERVER.LAUNCHSERVER = True
CLIENT.SERVER.SERVERCOMMANDPORT = 5070
CLIENT.SERVER.SERVERHOSTNAME = localhost
CLIENT.SERVER.SERVERPORT = 5074
CLIENT.SERVER.SERVERRESPONSETIMEOUT = 30
CLIENT.SERVER.USEENCRYPTION = false
CLIENT.SERVER.XMLDEBUG = false
CLIENT.SERVER.XMLDEBUGTOLOG = false
```

> ***Important:*** The XMLDEBUGTOLOG option is not supported on BlackBerry.

Pop-up Display

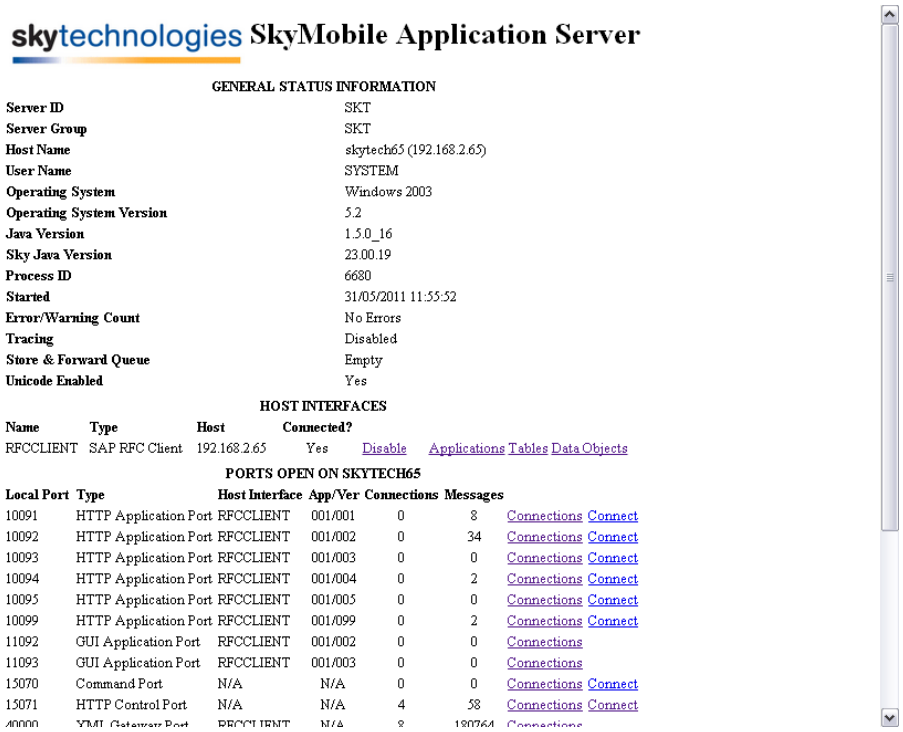


Example log

```
08/11/2007 09:08:23 INFO:  Outgoing XML request:
<SCRREQ C="WPC">
</SCRREQ>
08/11/2007 09:08:25 INFO:  Incoming XML response:
<SCRRSP>
<MSG T="Waiting for host\046\046\046"/>
</SCRRSP>
08/11/2007 09:08:25 INFO:  Incoming XML response:
<SCRRSP>
<MSG T="Loading function INIT\046\046\046"/>
</SCRRSP>
08/11/2007 09:08:26 INFO:  Incoming XML response:
<SCRRSP>
<MSG T="Loading function MAINTESTMENU\046\046\046"/>
</SCRRSP>
```

5.7.5.8 Custom Java Exit Tracing

You use the trace+ and trace- buttons on the main web status page to increment and decrement the Java Server tracing level. This effect is immediate and you should take care as to how much information is written to the Java Server system log. This facility is designed for tracing custom Java exit processing, that is, programs can have embedded trace statements that you can dynamically activate / deactivate using this mechanism. As well as this, this level triggers internal diagnostic tracing that helps give some context to the processing. As with all technical tracing, you should take care to avoid excessive output.



skytechnologies SkyMobile Application Server

GENERAL STATUS INFORMATION

Server ID	SKT
Server Group	SKT
Host Name	skytech65 (192.168.2.65)
User Name	SYSTEM
Operating System	Windows 2003
Operating System Version	5.2
Java Version	1.5.0_16
Sky Java Version	23.00.19
Process ID	6680
Started	31/05/2011 11:55:52
Error/Warning Count	No Errors
Tracing	Disabled
Store & Forward Queue	Empty
Unicode Enabled	Yes

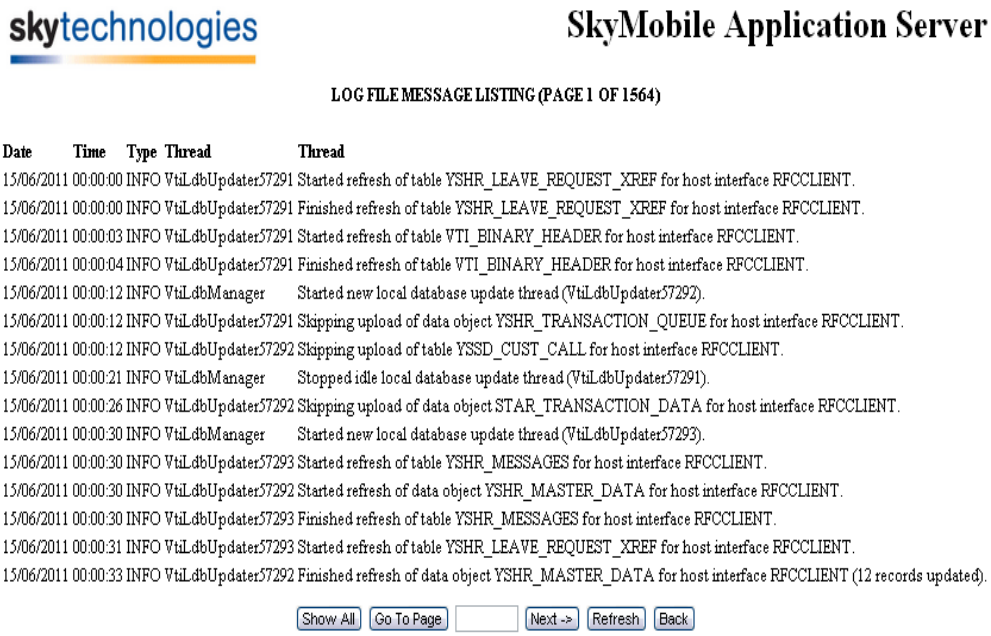
HOST INTERFACES

Name	Type	Host	Connected?			
RFCCLIENT	SAP RFC Client	192.168.2.65	Yes	Disable	Applications	Tables Data Objects

PORTS OPEN ON SKYTECH65

Local Port	Type	Host Interface	App/Ver	Connections	Messages	
10091	HTTP Application Port	RFCCLIENT	001/001	0	8	Connections Connect
10092	HTTP Application Port	RFCCLIENT	001/002	0	34	Connections Connect
10093	HTTP Application Port	RFCCLIENT	001/003	0	0	Connections Connect
10094	HTTP Application Port	RFCCLIENT	001/004	0	2	Connections Connect
10095	HTTP Application Port	RFCCLIENT	001/005	0	0	Connections Connect
10099	HTTP Application Port	RFCCLIENT	001/099	0	2	Connections Connect
11092	GUI Application Port	RFCCLIENT	001/002	0	0	Connections
11093	GUI Application Port	RFCCLIENT	001/003	0	0	Connections
15070	Command Port	N/A	N/A	0	0	Connections Connect
15071	HTTP Control Port	N/A	N/A	4	58	Connections Connect
40000	XML Gateway Port	RFCCLIENT	N/A	8	180764	Connections

All trace information is written to the Java Server system log.



skytechnologies

SkyMobile Application Server

LOG FILE MESSAGE LISTING (PAGE 1 OF 1564)

Date	Time	Type	Thread	Thread
15/06/2011	00:00:00	INFO	VtiLdbUpdater57291	Started refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
15/06/2011	00:00:00	INFO	VtiLdbUpdater57291	Finished refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
15/06/2011	00:00:03	INFO	VtiLdbUpdater57291	Started refresh of table VTI_BINARY_HEADER for host interface RFCCLIENT.
15/06/2011	00:00:04	INFO	VtiLdbUpdater57291	Finished refresh of table VTI_BINARY_HEADER for host interface RFCCLIENT.
15/06/2011	00:00:12	INFO	VtiLdbManager	Started new local database update thread (VtiLdbUpdater57292).
15/06/2011	00:00:12	INFO	VtiLdbUpdater57291	Skipping upload of data object YSHR_TRANSACTION_QUEUE for host interface RFCCLIENT.
15/06/2011	00:00:12	INFO	VtiLdbUpdater57292	Skipping upload of table YSSD_CUST_CALL for host interface RFCCLIENT.
15/06/2011	00:00:21	INFO	VtiLdbManager	Stopped idle local database update thread (VtiLdbUpdater57291).
15/06/2011	00:00:26	INFO	VtiLdbUpdater57292	Skipping upload of data object STAR_TRANSACTION_DATA for host interface RFCCLIENT.
15/06/2011	00:00:30	INFO	VtiLdbManager	Started new local database update thread (VtiLdbUpdater57293).
15/06/2011	00:00:30	INFO	VtiLdbUpdater57293	Started refresh of table YSHR_MESSAGES for host interface RFCCLIENT.
15/06/2011	00:00:30	INFO	VtiLdbUpdater57292	Started refresh of data object YSHR_MASTER_DATA for host interface RFCCLIENT.
15/06/2011	00:00:30	INFO	VtiLdbUpdater57293	Finished refresh of table YSHR_MESSAGES for host interface RFCCLIENT.
15/06/2011	00:00:31	INFO	VtiLdbUpdater57293	Started refresh of table YSHR_LEAVE_REQUEST_XREF for host interface RFCCLIENT.
15/06/2011	00:00:33	INFO	VtiLdbUpdater57292	Finished refresh of data object YSHR_MASTER_DATA for host interface RFCCLIENT (12 records updated).

Show All | Go To Page | Next -> | Refresh | Back

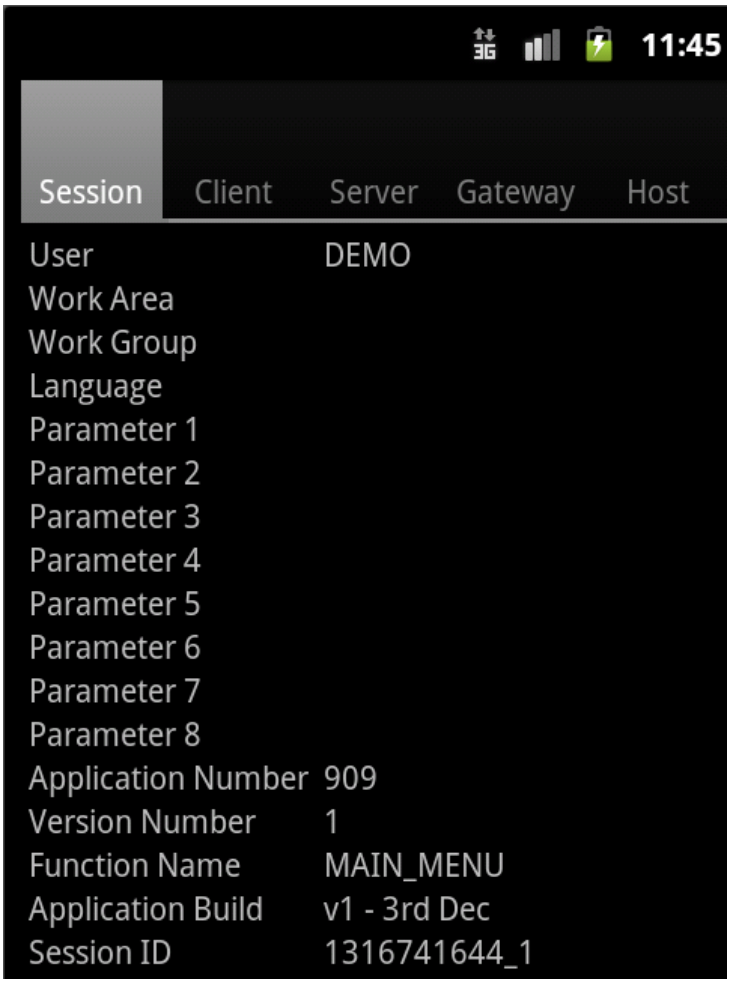
5.7.5.9 Device about Screen

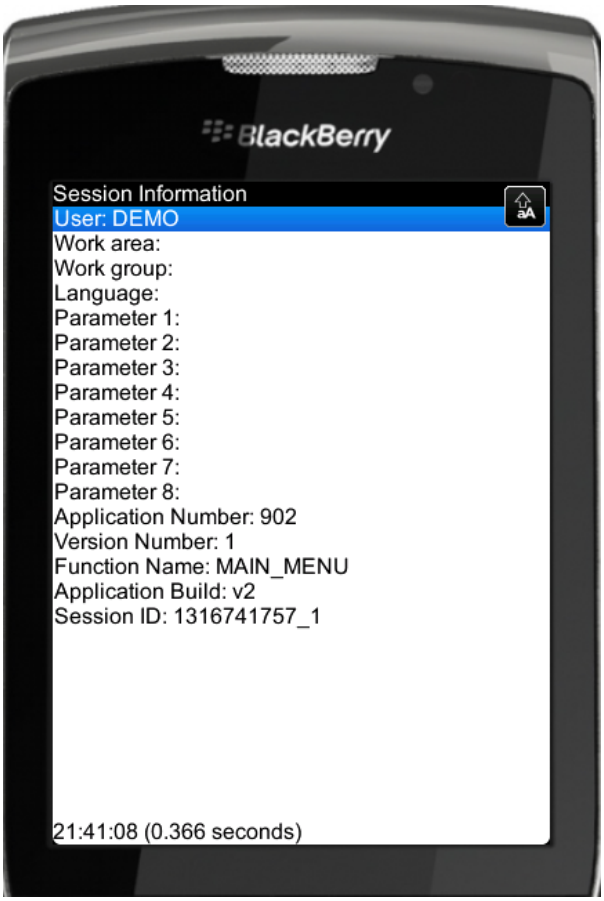
An "About" screen is available on each client (except VT200 and HTTP) that provides a range of information the include:

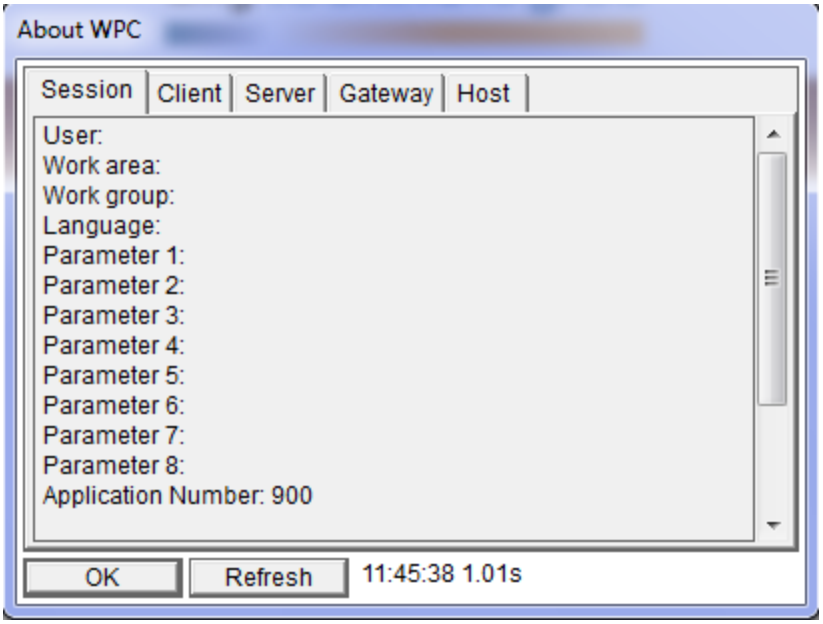
- User session information
- Client software information
- Secure container software information
- Gateway information (if connecting through a gateway)
- MEAP information.

> ***Important:*** If the device does not have a network connection, then gateway and host information is not available and there may be a slight delay in the other screens updating with the details.

Each area of information is presented on a different tab or screen (depending on device).







Session Tab

The Session tab displays information about the user current session on the device that include:

- Logged on User ID
- User work area and group
- User current language
- Parameter 1 to 8 values
- Current accessed Application ID, Version ID and function name
- Currently accessed application build info
- Session ID

Client Tab

The tab displays information about the client running on the device that include:

- Client version
- Device operating system version
- Unique Device ID
- Device hostname / IP address

Server Tab

The tab displays information about the secure container running on the device that include:

- Server ID
- Server Group
- Hostname
- IP address

- OS version
- JRE version
- SkyMobile version
- JVM process ID
- Whether the server is operating in unicode mode

Gateway Tab

The tab displays information about the gateway the device is connected to include:

- Server ID
- Server Group
- Hostname
- IP address
- Gateway port number
- OS version
- JRE version
- SkyMobile version
- JVM process ID
- Whether the server is operating in unicode mode

Host Tab

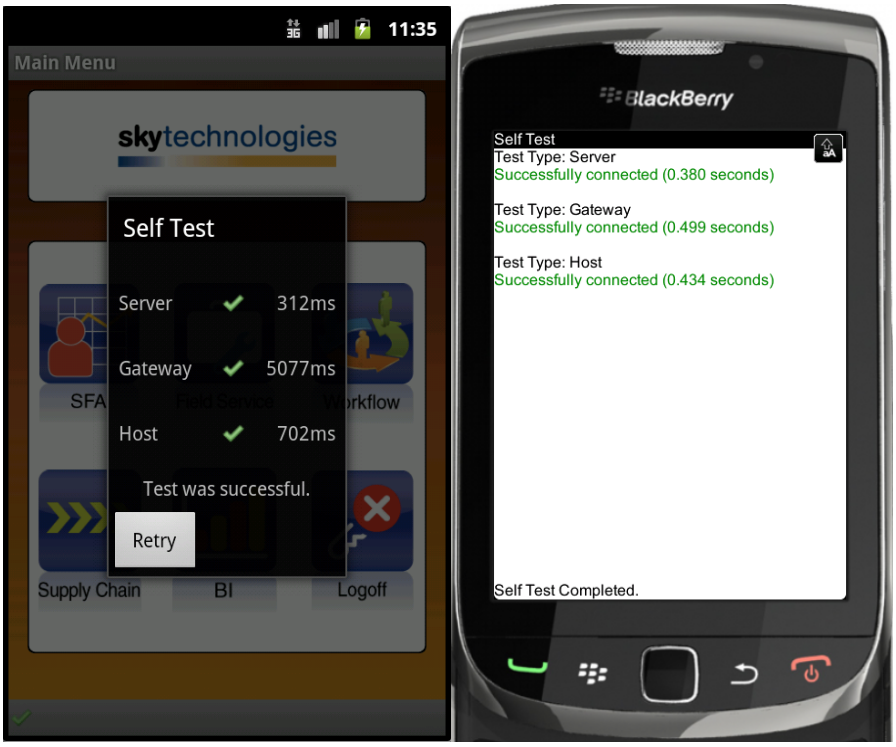
The tab displays information about the MEAP the device connected that include:

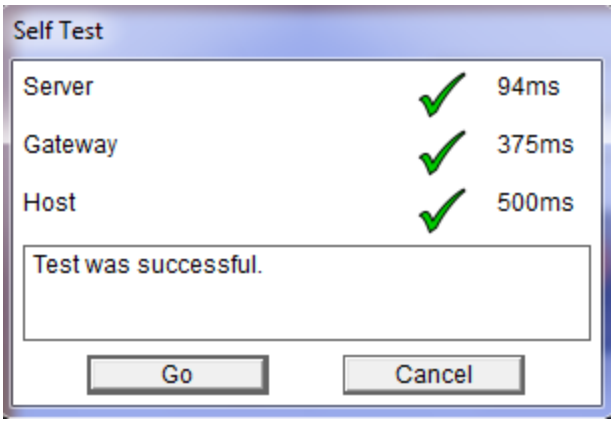
- Hostname
- OS version

- SAP Installation ID
- SAP SID
- SAP Client

5.7.5.10 Device Self Test

The "Self Test" function is available on all clients except for VT220 and HTTP). It is accessed from the system menu at any time. A basic pop-up appears indicating Server, Gateway and SAP Host tests. You click **go** and this invokes the connection test sequence. A green success message or tick means all is well. If there is a problem, then a cross appears with the relevant problem area description.





The purpose of the self test function is to enable remote users to perform their own basic network diagnostic testing to help identify where possible problems are, for example, to identify at which point communications fail. This can help support quickly escalate problems to the correct area in an organization or more importantly any third party service provided involved in the infrastructure, for example, Telecommunications provider.

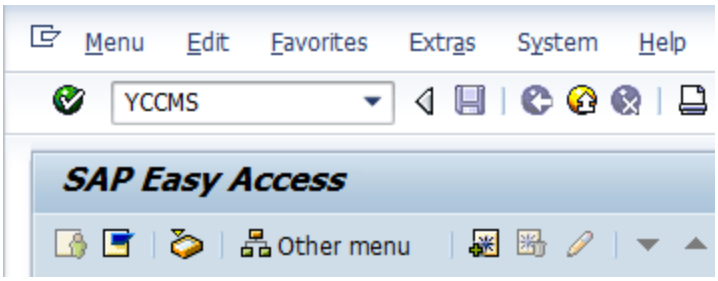
5.8 Computer Center Management System Monitor Utility

The Computer Center Management System (CCMS) Monitor utility allows you to maintain monitoring rules that control the alerts from Kony for SAP that are published to SAP Computer Center Management System.

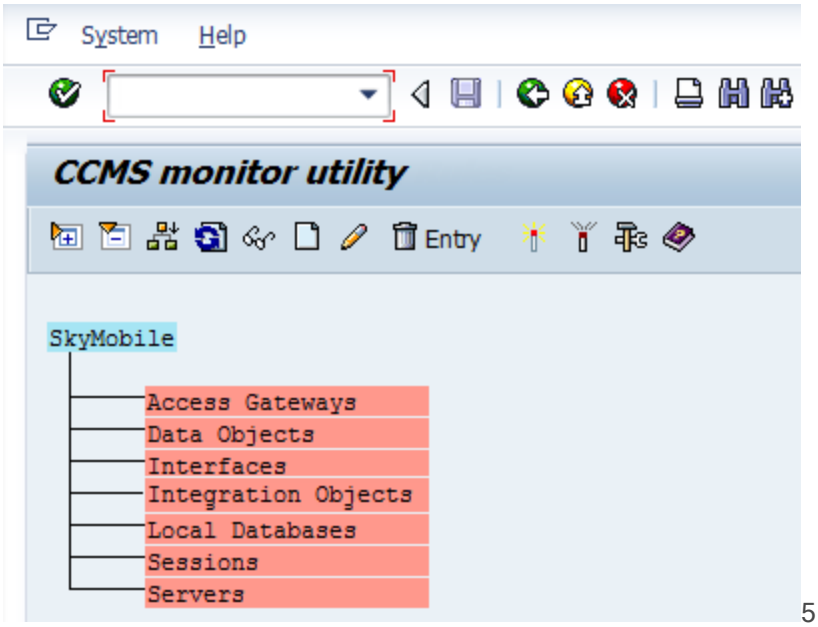
5.8.1 Launching the CCMS Monitor Utility Workbench

To launch the CCMS Monitoring utility workbench, follow these steps:

1. Run the transaction **/SKY/YCCMS (YCCMS)** in the SAP system.



   The **CCMS monitor utility** window appears with all the configured components as node types.

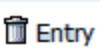


5

Note: On launch of this transaction, **/SKY/YCCMS (YCCMS)** for the first time, monitoring is deactivated for all node types. Based on your requirement, you can activate the node types.

The following is the list of application tool bar buttons and its usages:

Button	Button Name	Use
	Expand all levels	Expands all levels of tree structure
	Collapse all levels	Collapses all levels of tree structure
	Expand one level	Expands one level of tree structure
	Refresh	Refreshes the output list

Button	Button Name	Use
	Display	Displays the details of the selected rule
	Create	Creates a new rule
	Change	Changes the selected rule
	Delete selected entry	Deletes the selected rule entry
	Activate Monitoring	Activates the selected node type or rule
	Deactivate Monitoring	Deactivates the selected node type or rule
	Simulate an alert	Simulates an alert for selected values
	Online help	Displays online help for YCCMS transaction

You can perform the following operations using the utility for maintaining monitoring rules:

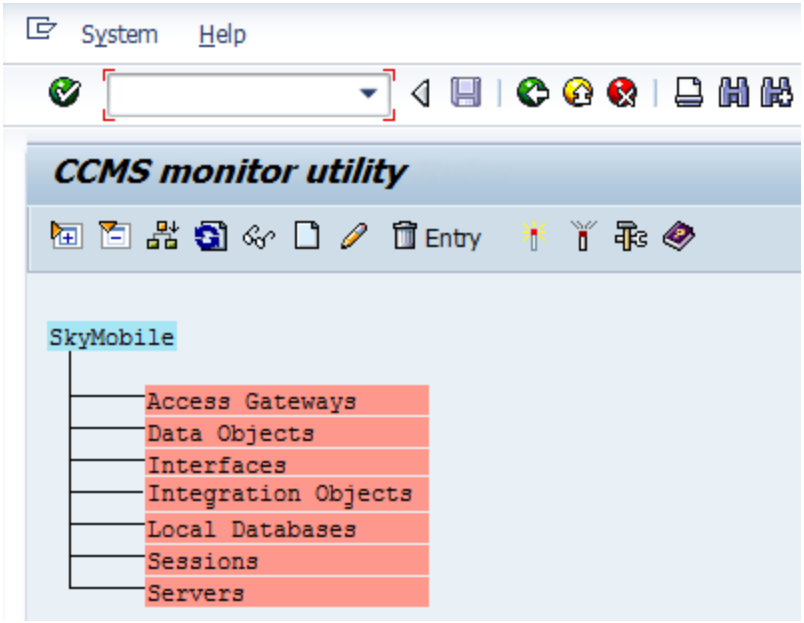
- Creating a Monitoring Rule
- Changing a Monitoring Rule
- Deleting a Monitoring Rule
- Activating a Monitoring Rule
- Deactivating a Monitoring Rule
- Simulating an Alert

5.8.2 Scenarios of Monitoring Rules

You can activate monitoring at the root node level or at the rule level. The possible scenarios are:

- *Root Node Inactive*
- *Root Node Active with No Child Rule*
- *Root Node Active with Inactive Child Rules*
- *Root Node Active with Active Child Nodes*

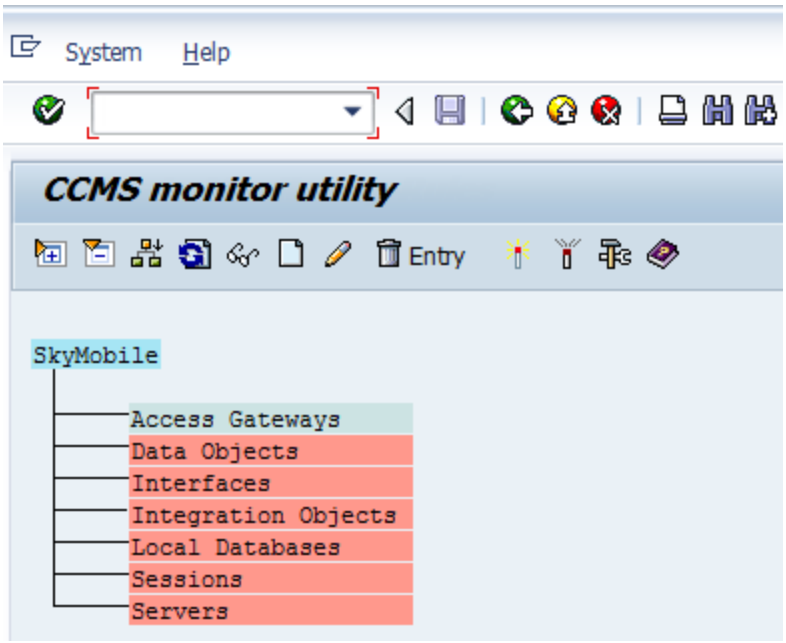
Root Node Inactive



In the above example, all the root nodes are deactivated. That means monitoring is not allowed for all the node types.

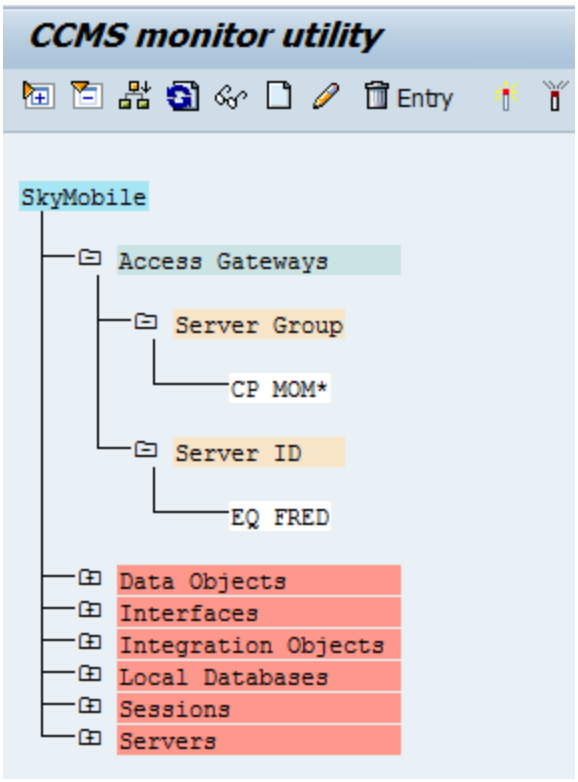
> ***Note:*** If you deactivate a root node, then all child rules under that node are also deactivated.

Root Node Active with No Child Rule



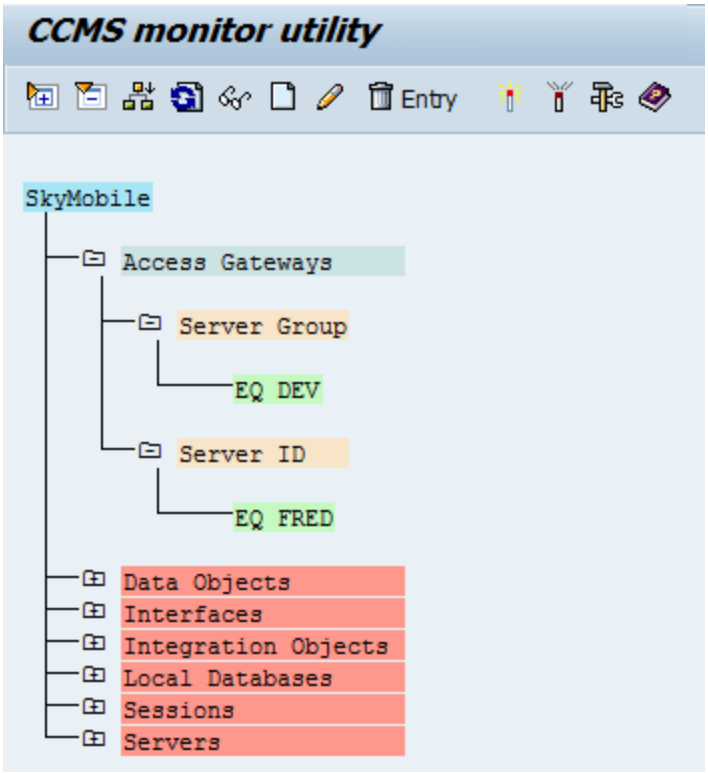
In the above example, the only active node is **Access Gateways**. In this case, any alerts raised for the remaining nodes are ignored. Any alerts for Access Gateways are published to SAP CCMS.

Root Node Active with Inactive Child Rules



In the above example, the only active node is **Access Gateways**. In addition, two rules are created under the **Access Gateways** node, but the rules are not activated. In this case, any alerts raised for the remaining nodes are ignored. Any alerts for Access Gateways are published to SAP CCMS.

Root Node Active with Active Child Nodes



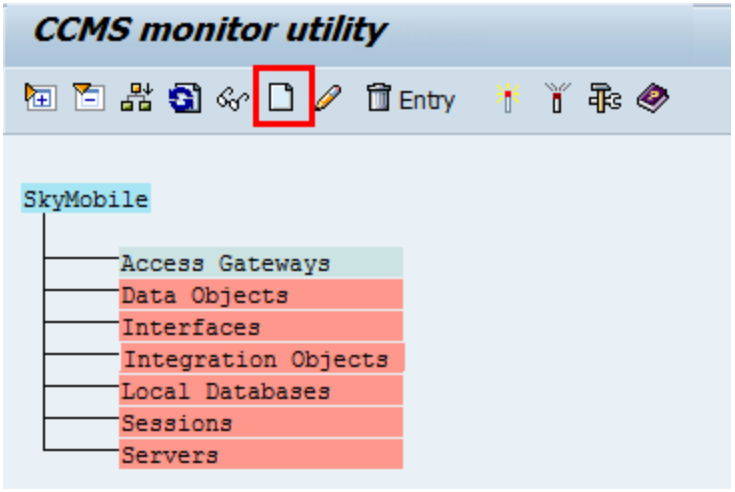
In the above example, the only active node is **Access Gateways**. In addition, two rules are created under the **Access Gateways** node and are also activated. If an alert for an Access Gateway is raised, it must match one of the rules to publish to SAP CCMS. In this case, either the server group needs to be “DEV” or the Server ID needs to be “FRED”. If the alert does not match either rule, then the alert is ignored.

5.8.3 Creating a Monitoring Rule

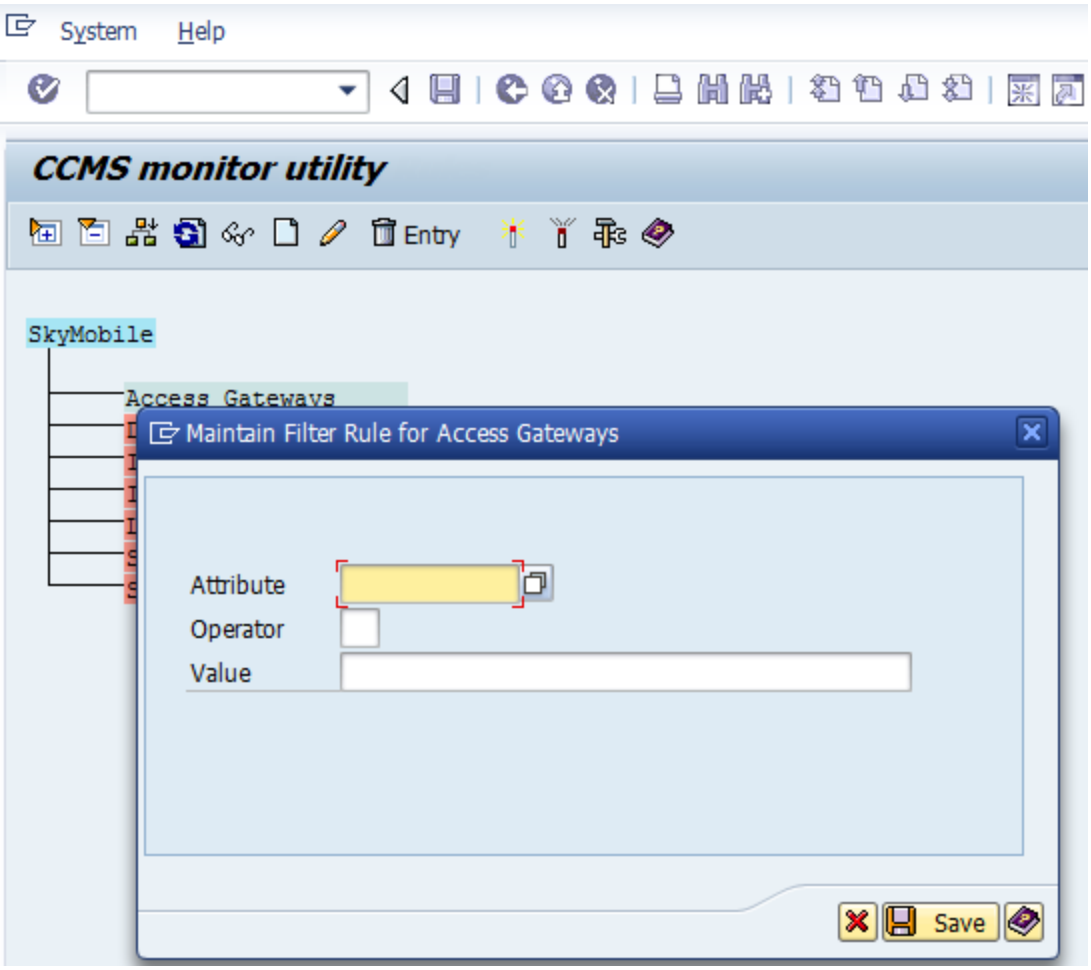
You can create monitoring rules for the selected node type at its attribute level.

To create the monitoring rules, follow these steps:

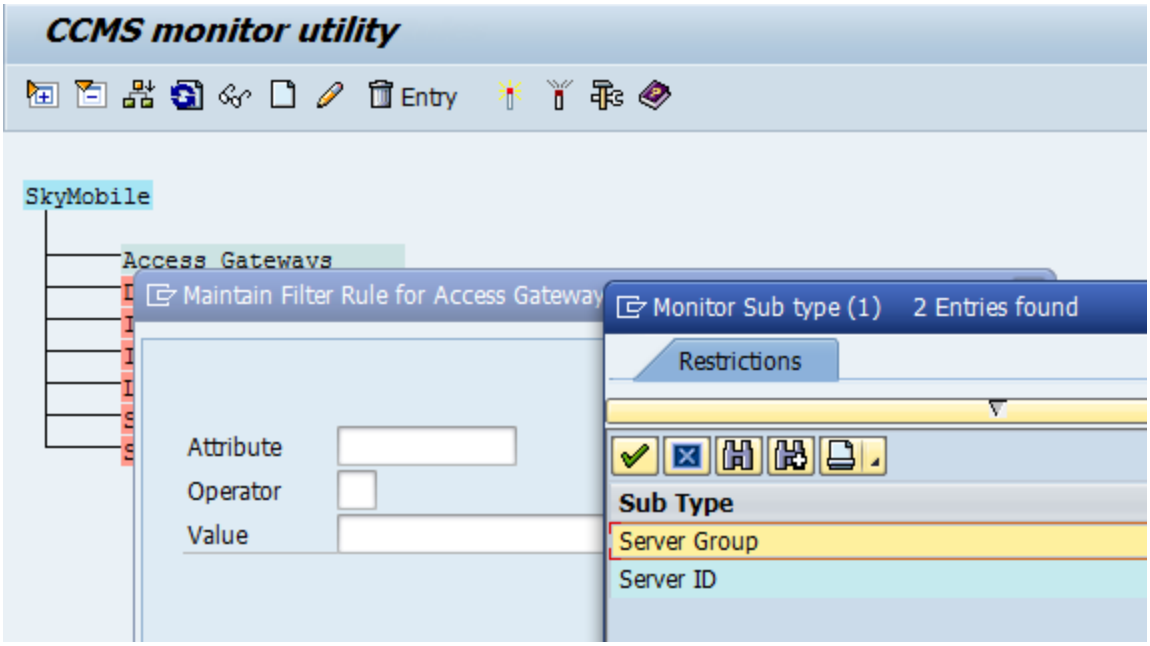
1. Select the node type and click the **Create** icon from the application tool bar.



The **Maintain Filter Rule for <Node name>** dialog appears.



2. You can choose the attribute type for which you want to create the monitoring rule.



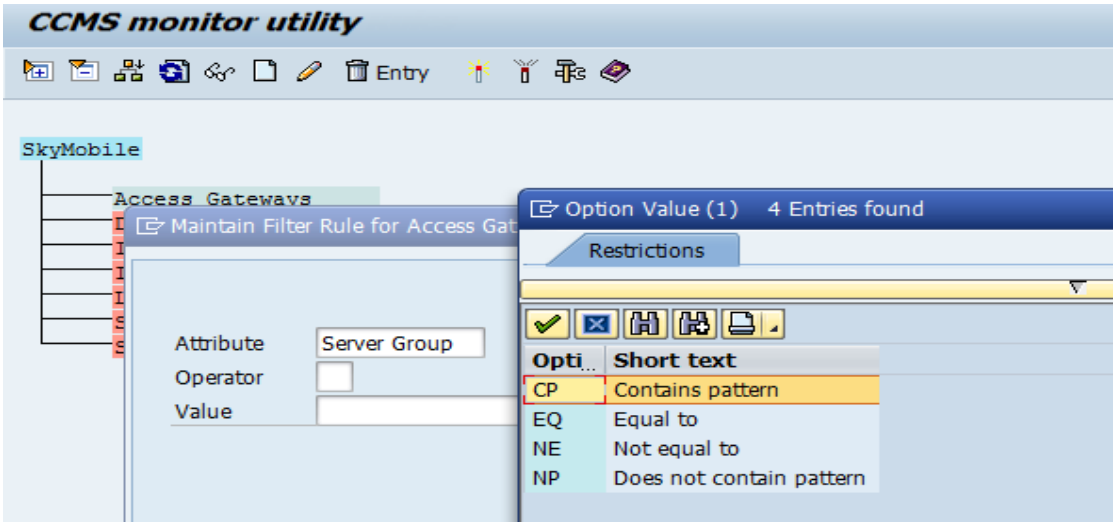
   In this example, you can choose the **Server Group** or **Server ID**.

> ***Note:*** Attribute values are dependent on the type of node.

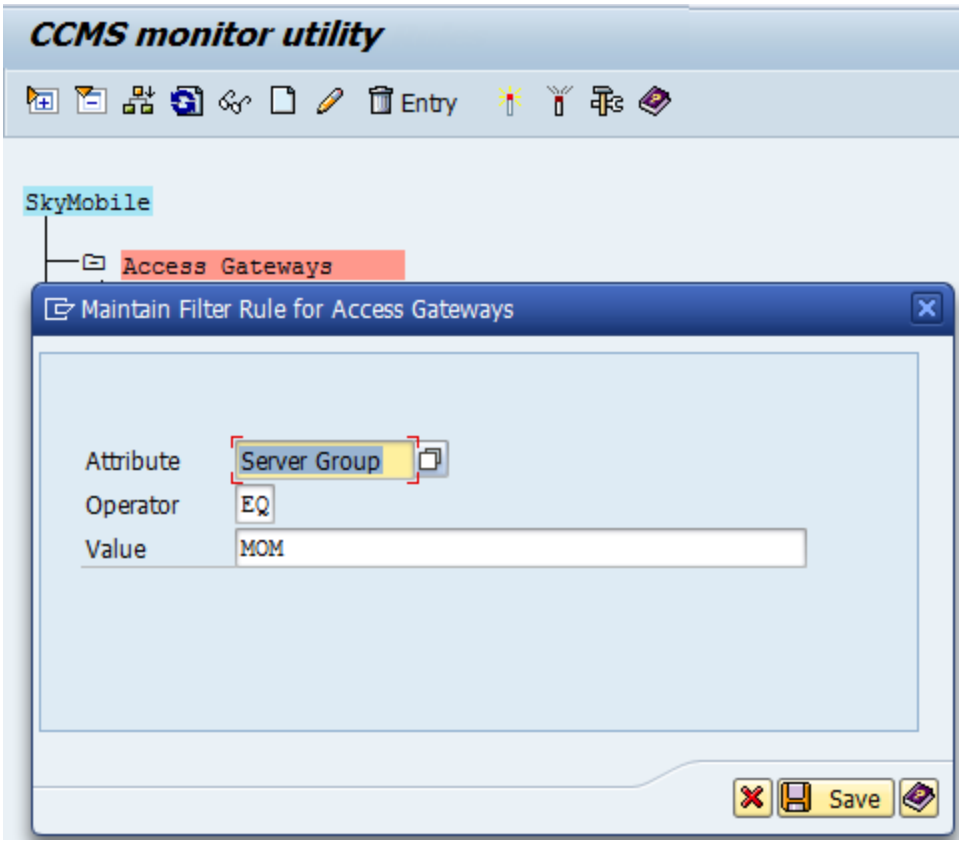
3. Click in the text box next to the **Operator**.
   An icon appears.

4. Click the icon.
   The **Option Value** dialog appears.

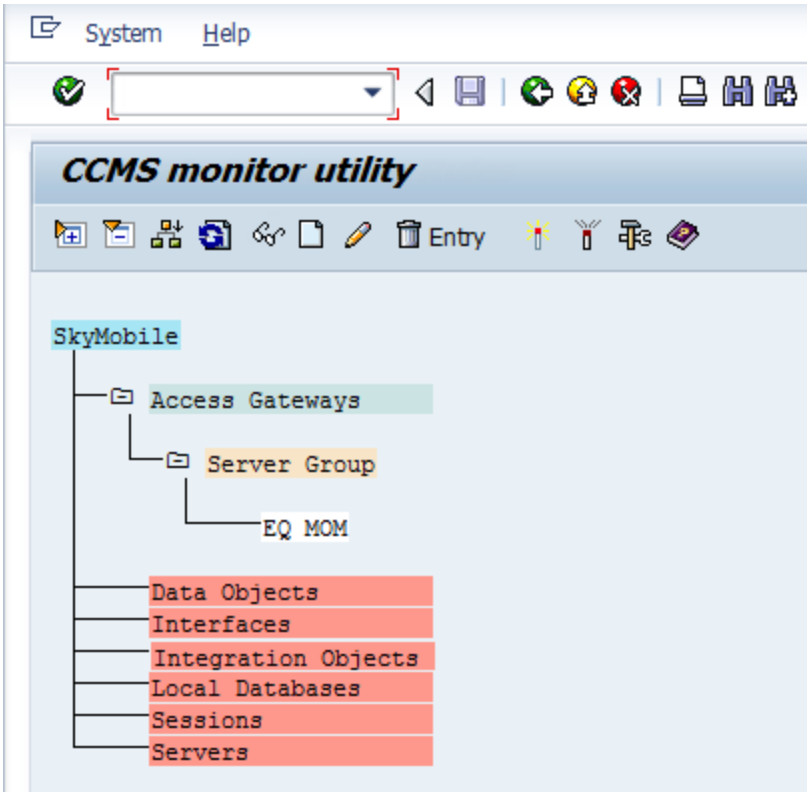
5. Choose the value of the operator for which you want to create the monitoring rule. You may choose (**Equal to** / **Not equal to** / **Contains pattern** / **Does not contain pattern**).



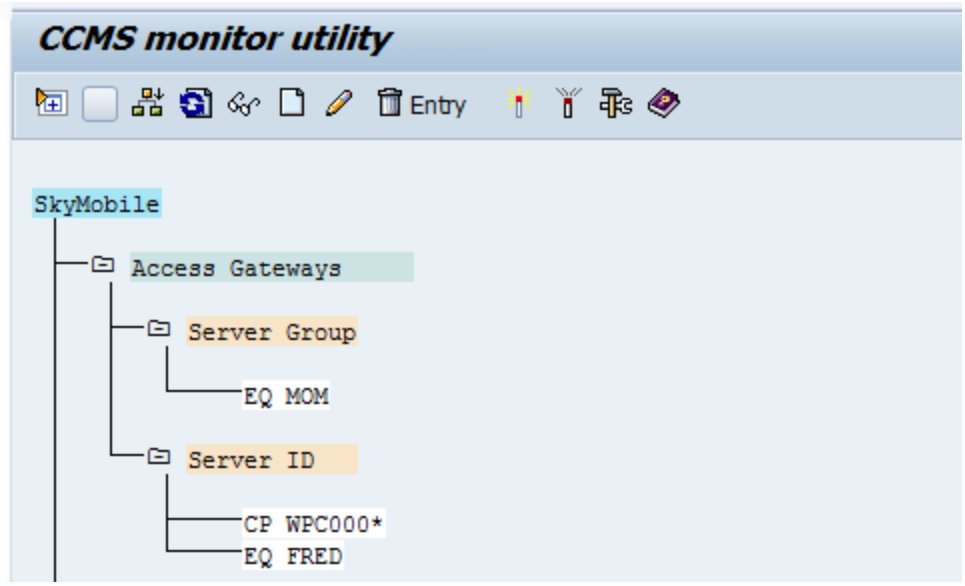
6. You may enter the attribute value of the Gateway for Value.



7. After you enter the values for all fields, click **Save**.



The monitoring filter rule is created under the selected node type with the Server Group attribute type. You may create any number of rules for all the other attributes.



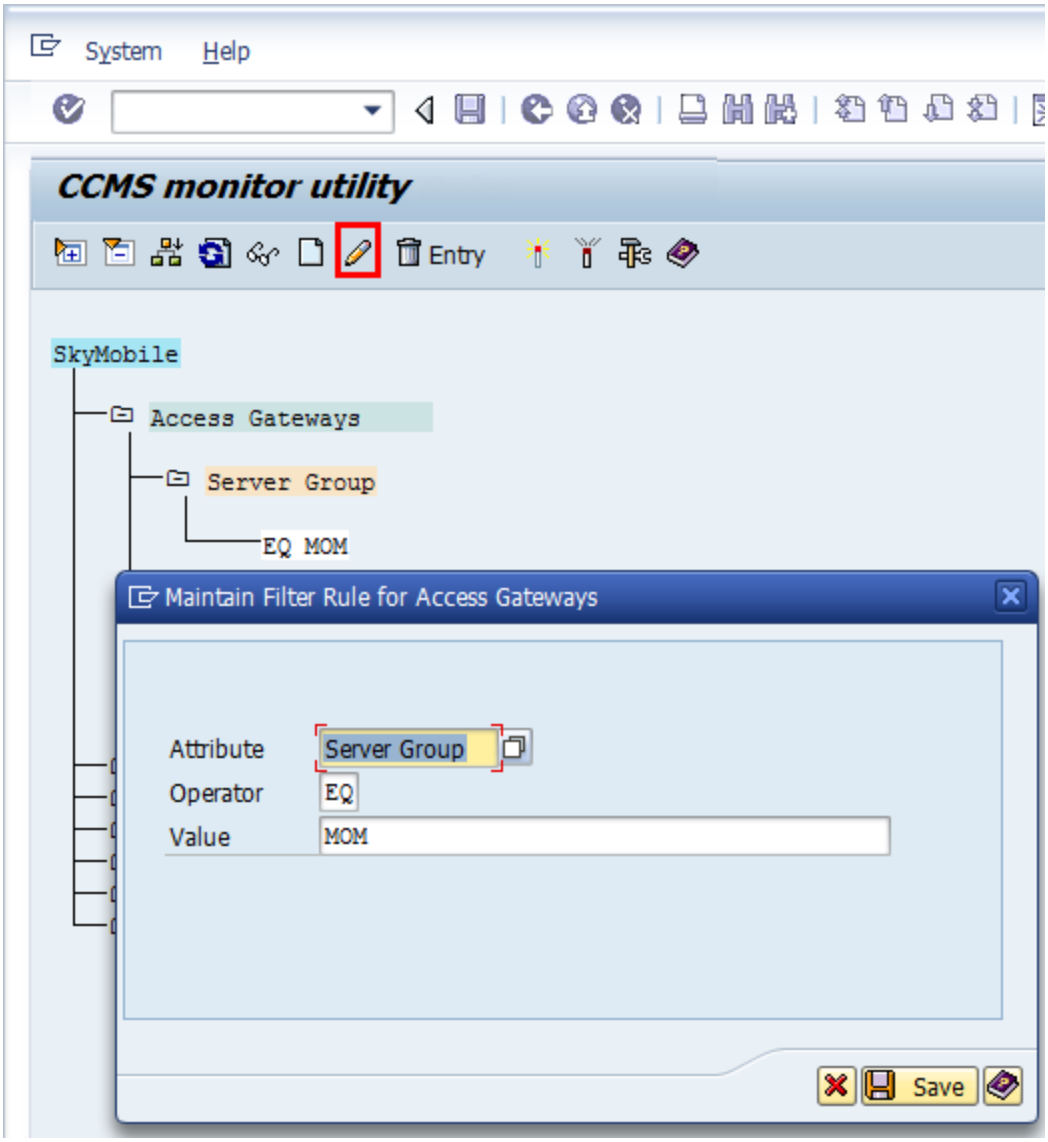
> ***Note:*** By default the newly created rules are deactivated. For the filters to work, choose the rule and activate the rule by clicking the **Activate** icon on the application tool bar.

5.8.4 Changing a Monitoring Rule

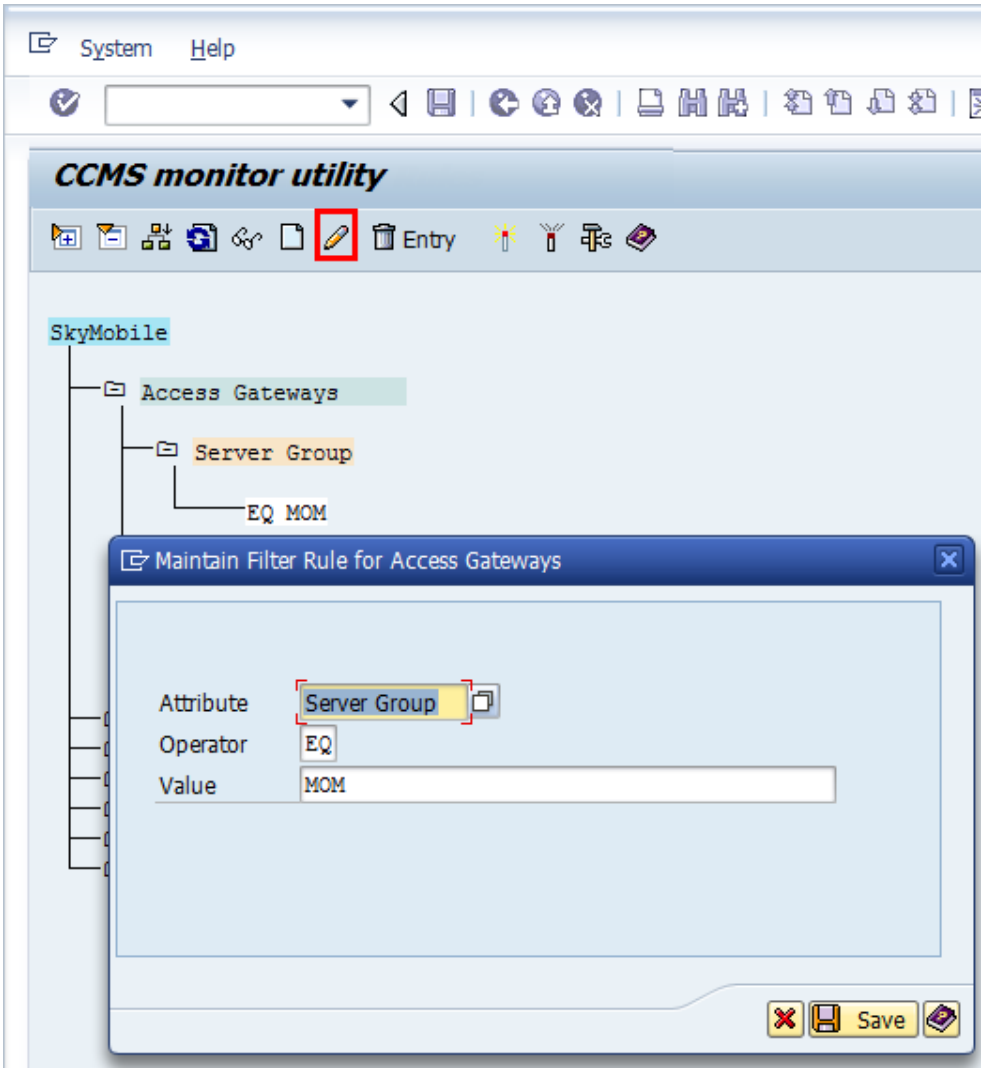
To change the monitoring rule, follow these steps:

1. Select the deactivated monitoring rule.
2. Double-click the rule or click the **Change** icon from the application tool bar.

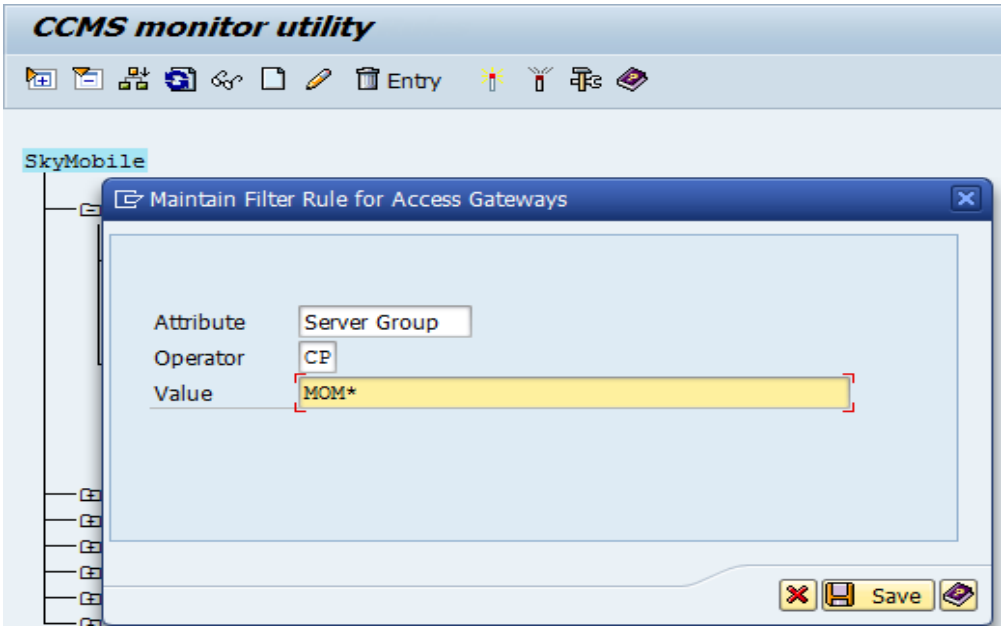
The **Maintain Filter Rule for <Node name>** dialog appears with selected values.



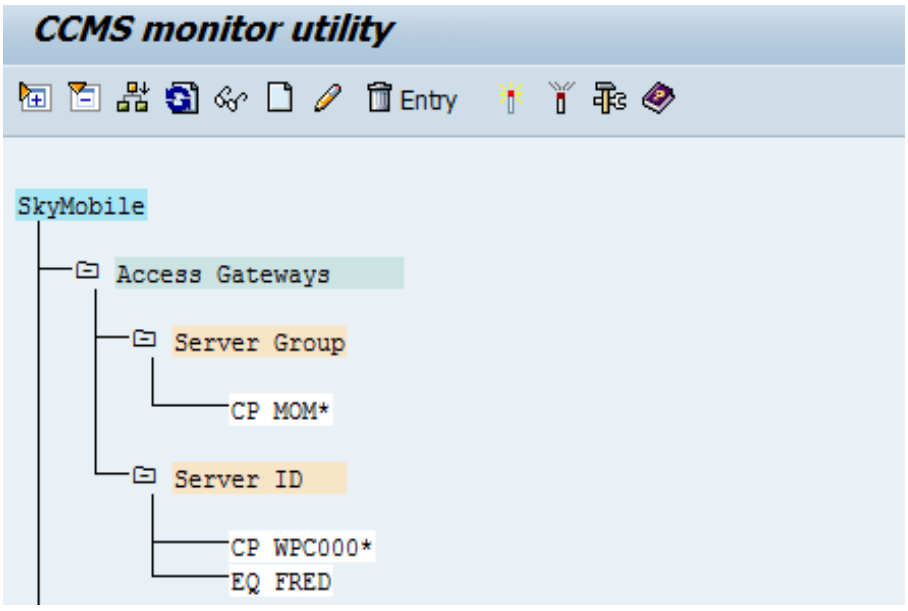
3. The **Maintain Filter Rule for Access Gateways** dialog appears with selected values.



4. Change the required values and click **Save**.



The changed rule appears.

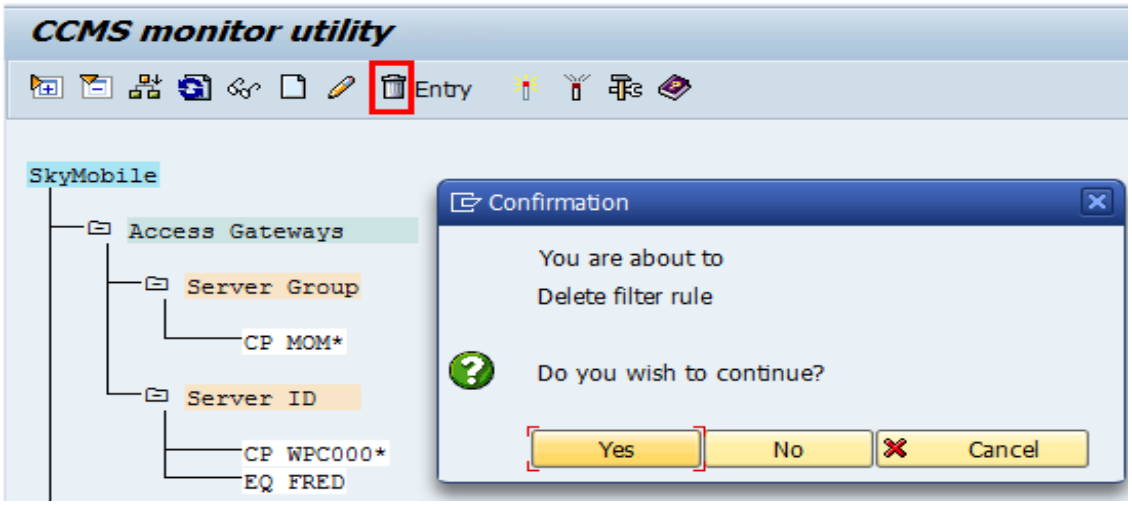


> ***Note:*** You must deactivate an existing rule to change it.

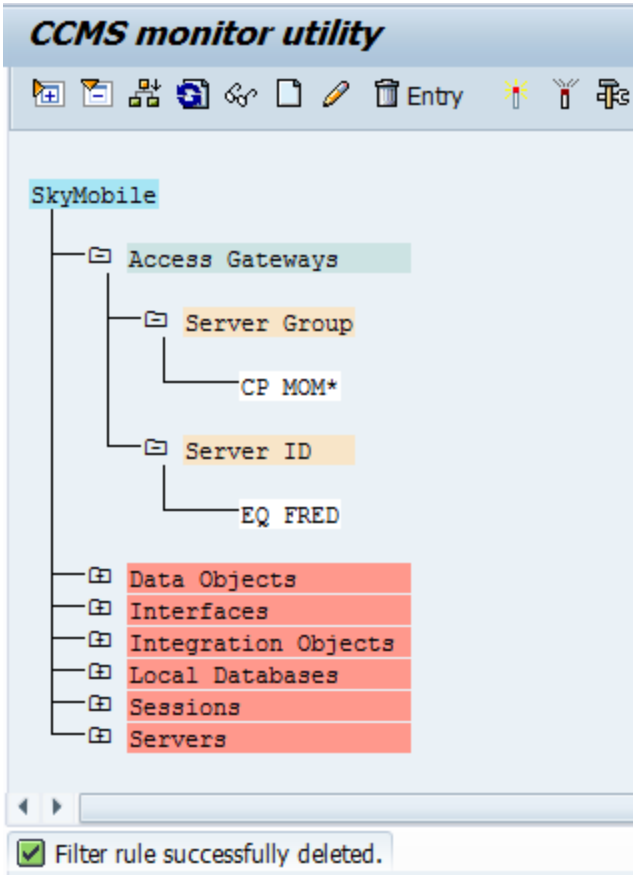
5.8.5 Deleting a Monitoring Rule

To delete the monitoring rule, follow these steps:

1. Select the deactivated monitoring rule, and click the **Delete** icon from the application tool bar. The **Confirmation** dialog appears.



2. Click **Yes** to delete the rule.



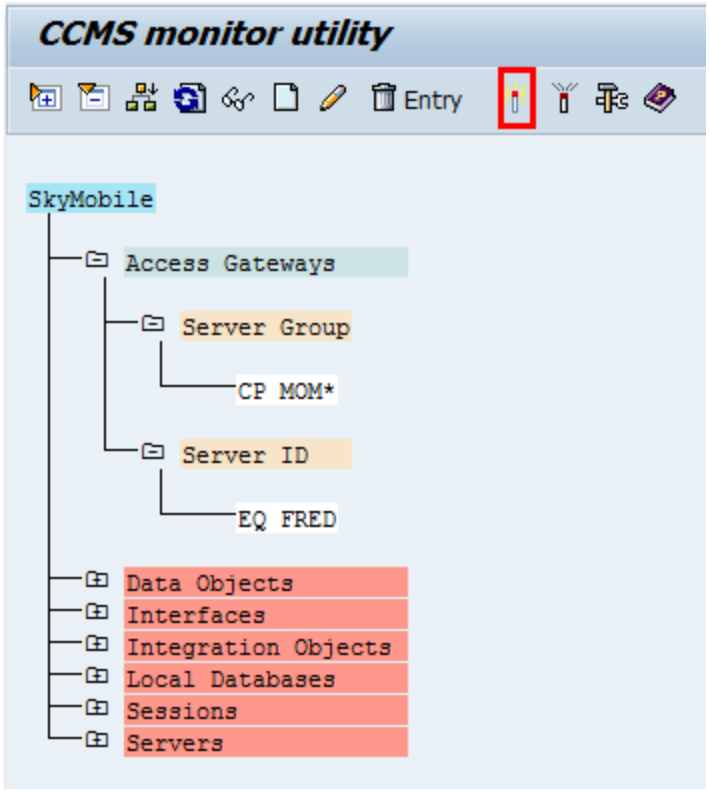
The deleted rule is removed from the list.

> ***Note:*** You must deactivate an existing rule to delete it.

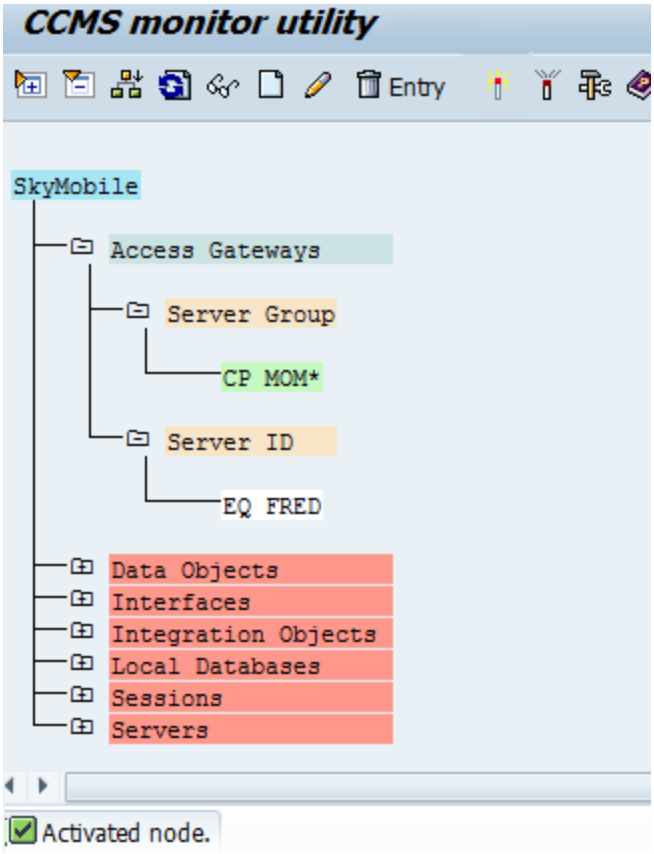
5.8.6 Activating a Monitoring Rule

To activate a monitoring rule, follow these steps:

1. Select the deactivated monitoring rule, and click the **Activate** icon from the application tool bar.

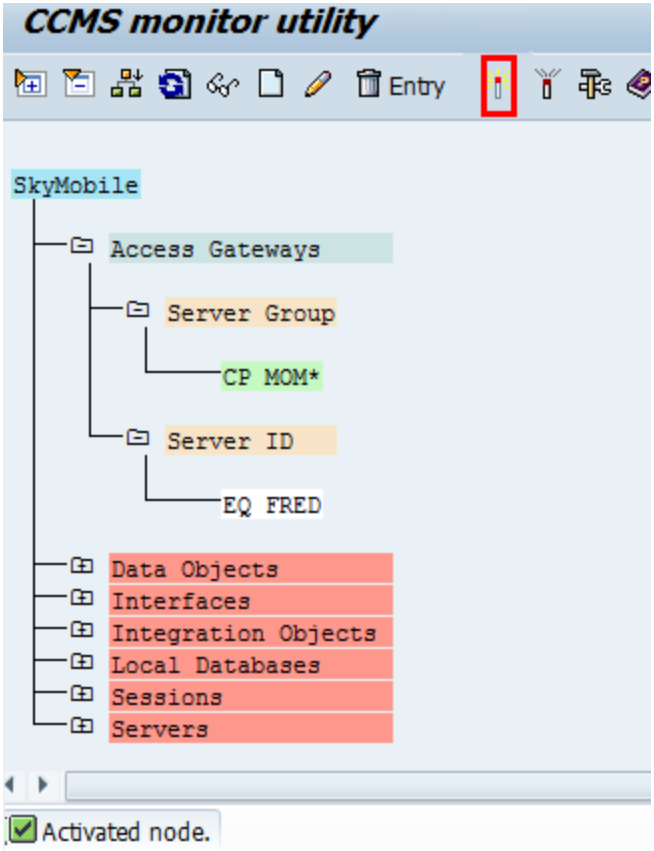


The rule is activated as follows:

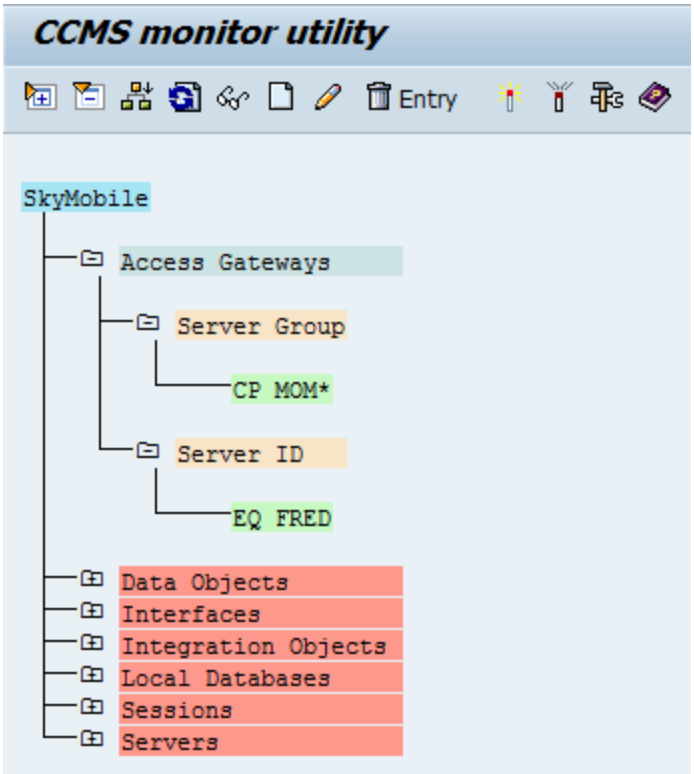


In the above example, entries of Access Gateways that have the Server Group containing the MOM* pattern are only allowed for CCMS Monitoring.

2. Activate the rule.

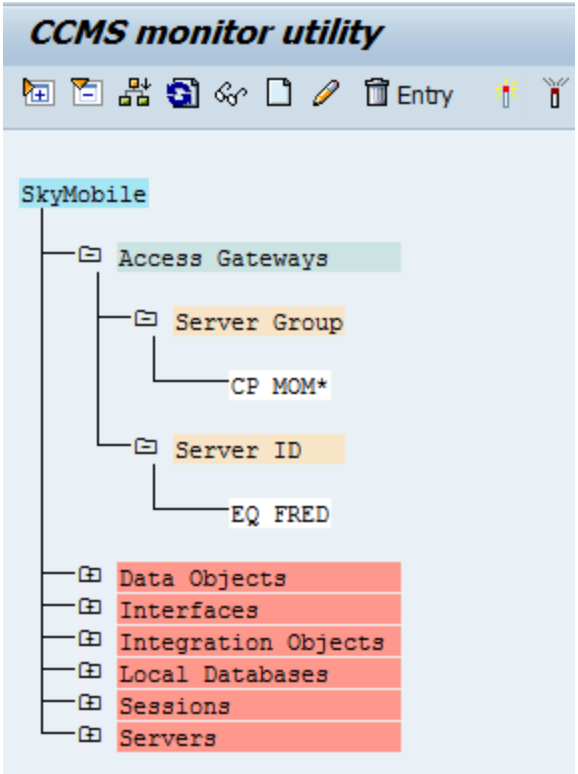


The rule is activated as follows:



In the above example, entries of Access Gateways that satisfy the criteria Server Group containing the MOM* pattern or Server ID equal to FRED are only allowed for CCMS Monitoring.

3. Activate only root Node type. Select the deactivated node type (Access gateways) and click the **Activate** icon.



In the above example, only the node type (Access Gateways) is activated whereas the rules under it are deactivated. This allows CCMS monitoring for all entries of type Access Gateways.

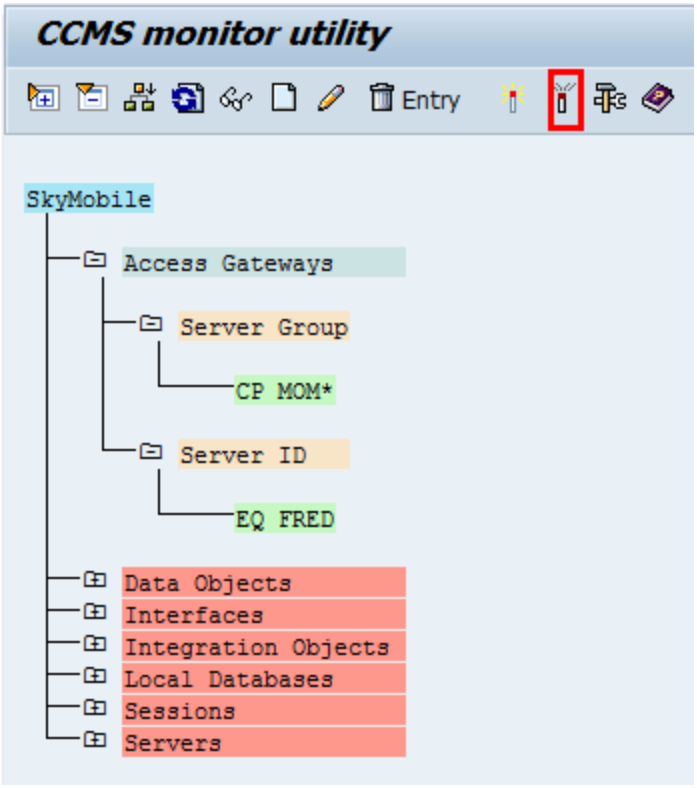
Note:

1. You can activate monitoring at node level or you can active monitoring at rule level.
2. Only activated node types and monitoring rules are considered for the CCMS Monitoring.

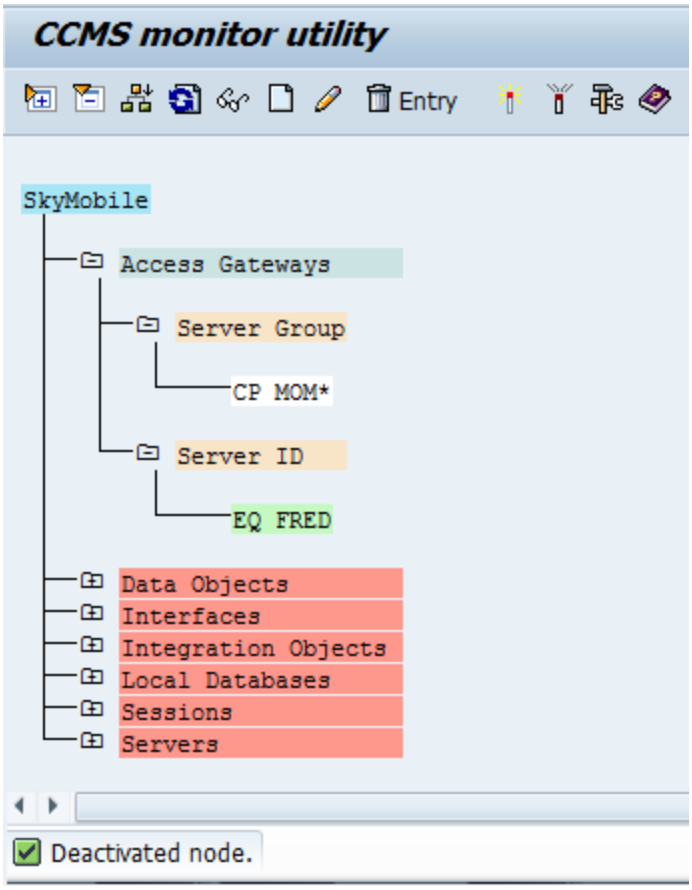
5.8.7 Deactivating a Monitoring Rule

To deactivate a monitoring rule, follow these steps:

1. Select any active monitoring rule and click the **Deactivate** icon from the application toolbar.

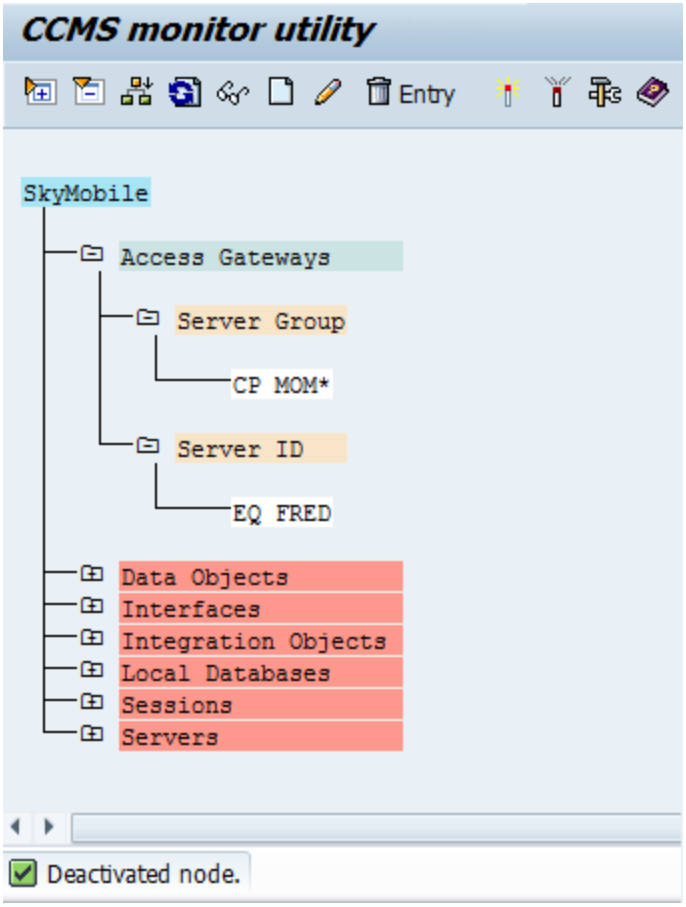


The rule is deactivated as follows:



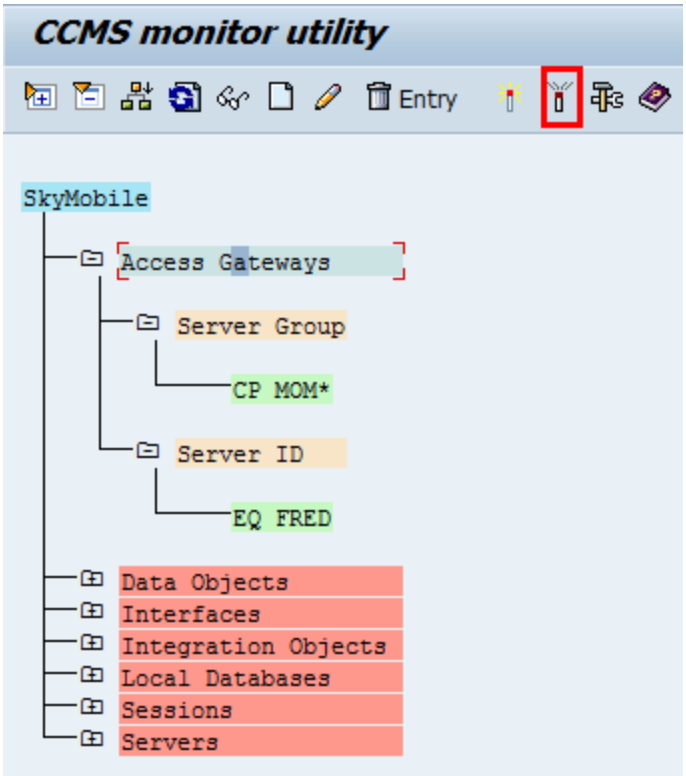
In the above example, by deactivating the Server Group rule, only entries of Access Gateways that satisfy the criteria, Server ID EQ FRED are allowed for monitoring.

2. Deactivate the rule.

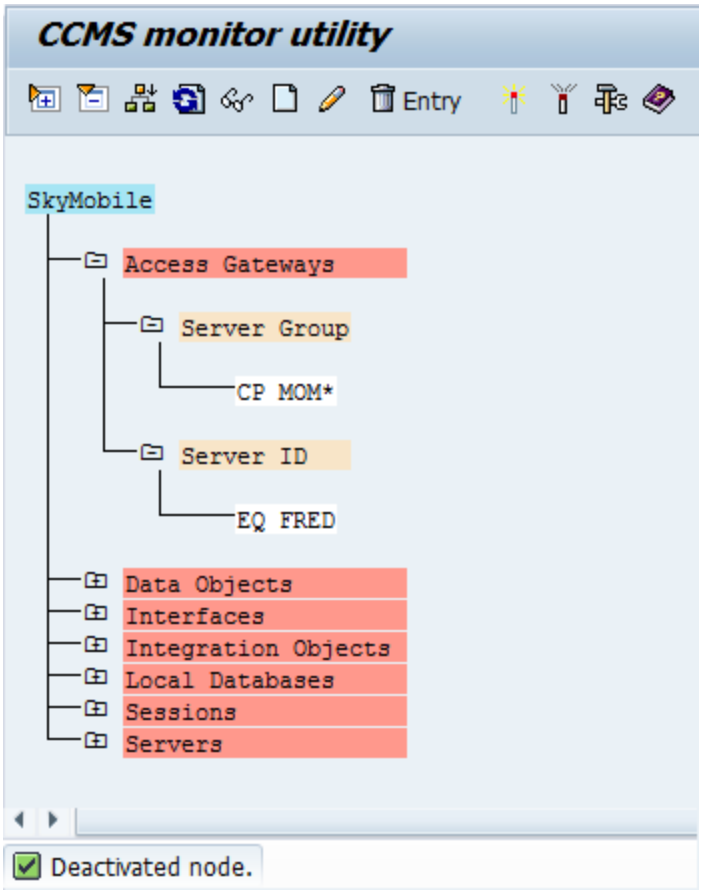


In the above example, by deactivating the Server ID rule, only the root node type Access Gateways is activated, and all other rules under it are deactivated. That means all the entries of Access Gateways are allowed for CCMS monitoring.

3. To deactivate the root node type, select the node type and click the **Deactivate** icon.



The node is deactivated as follows:

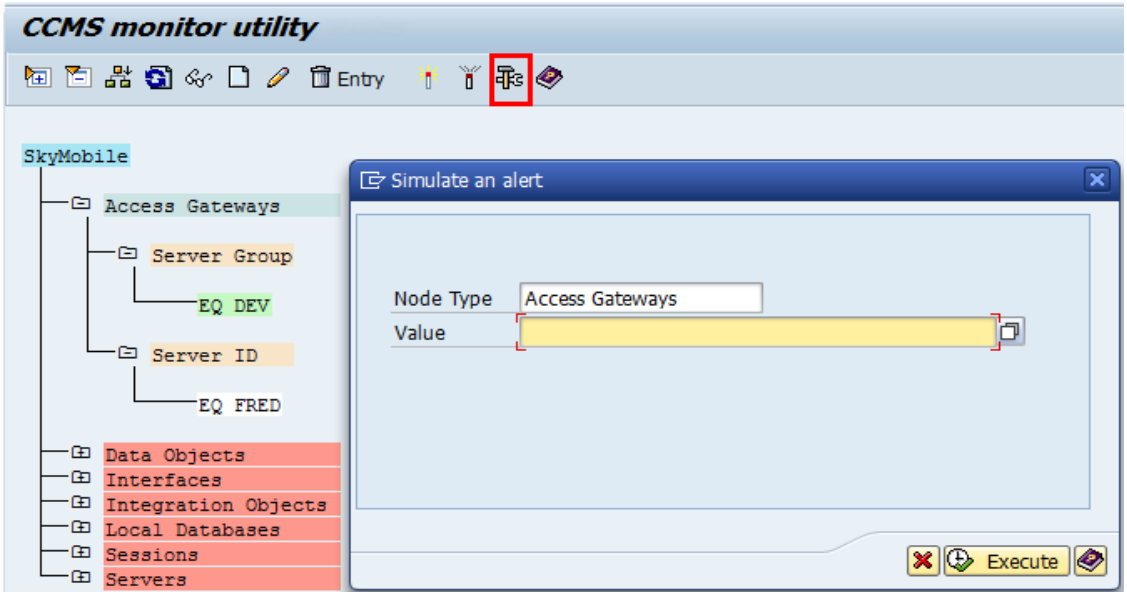


In the above example, when the root node is deactivated, all the rules under that node are also deactivated. That means no Access Gateways entry is allowed for CCMS monitoring.

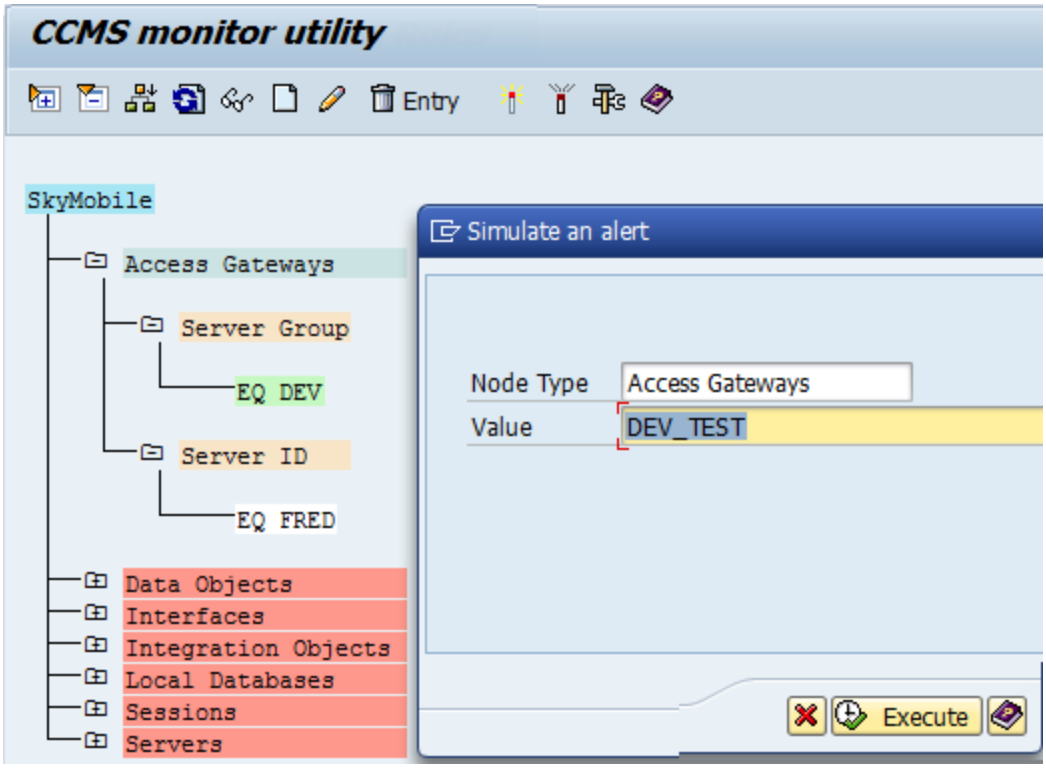
5.8.8 Simulating an Alert

To simulate an alert, follow these steps:

1. Select any of the root node or its attribute type, and click the **Simulate an alert** icon from the application tool bar.
   A dialog appears with the filled in node type (Access Gateways).

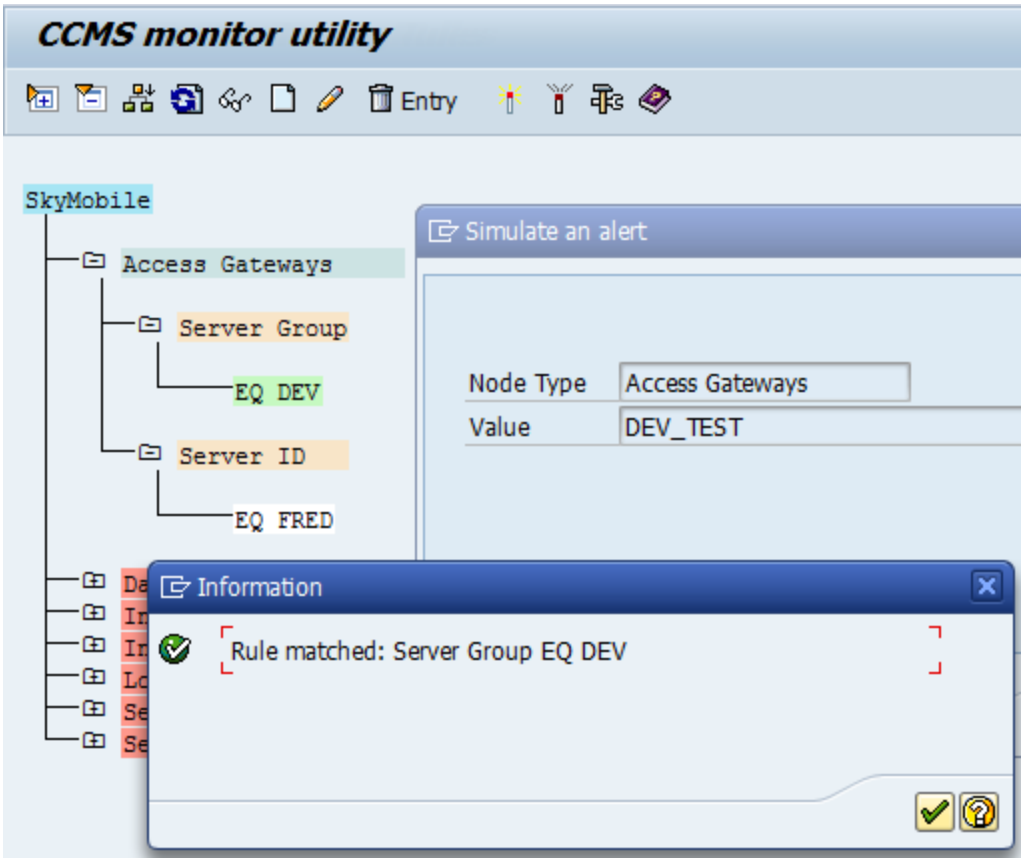


2. Fill or choose the **Value** for which you need to simulate an alert.

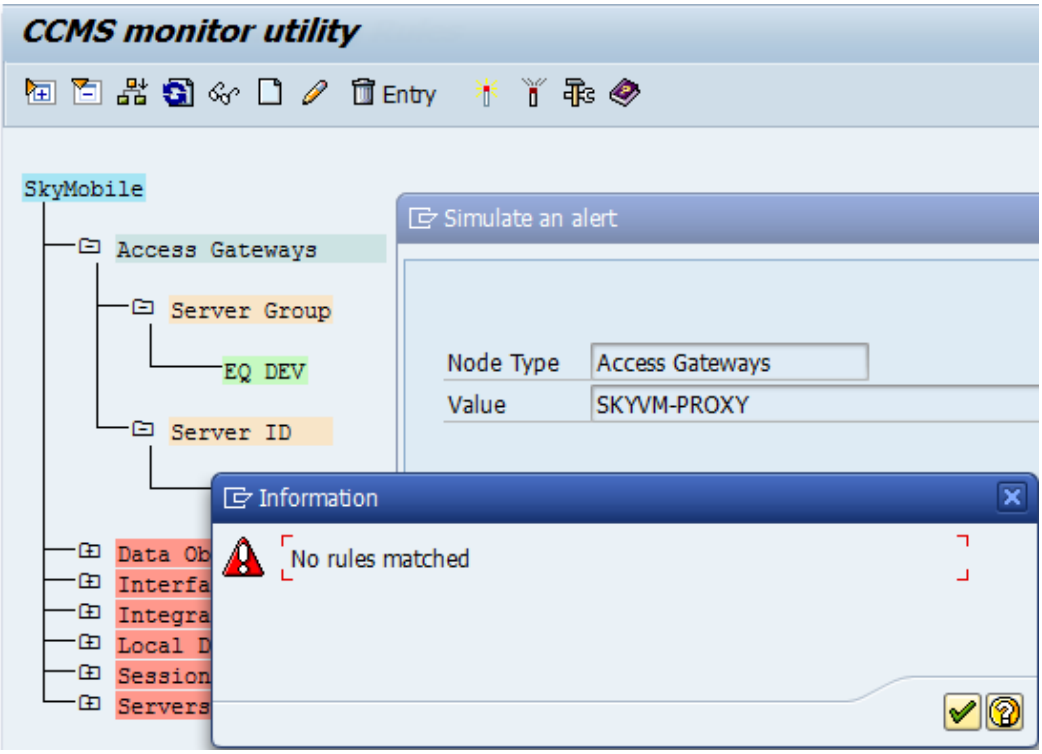


3. Click **Execute**.

   a. If monitoring is allowed for the entered value, confirmation message that reads "Rule matched" appears, as shown below.



   b. If monitoring is not allowed for the entered value, an error message appears as shown below.



You may check the monitoring alerts using the **Simulate an alert** option. Based on outcome, you may adjust the monitoring rules accordingly.

5.8.9 Monitoring from Solution Manager

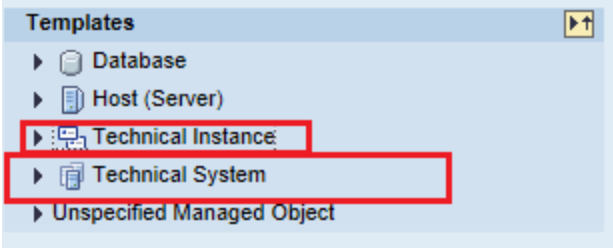
The alerts that Kony for SAP publishes to SAP CCMS can in turn be published to Solution Manager. To do this, you need to define some custom metrics and alerts in Solution Manager. The following sections describe the configuration process.

5.8.9.1 Prerequisites

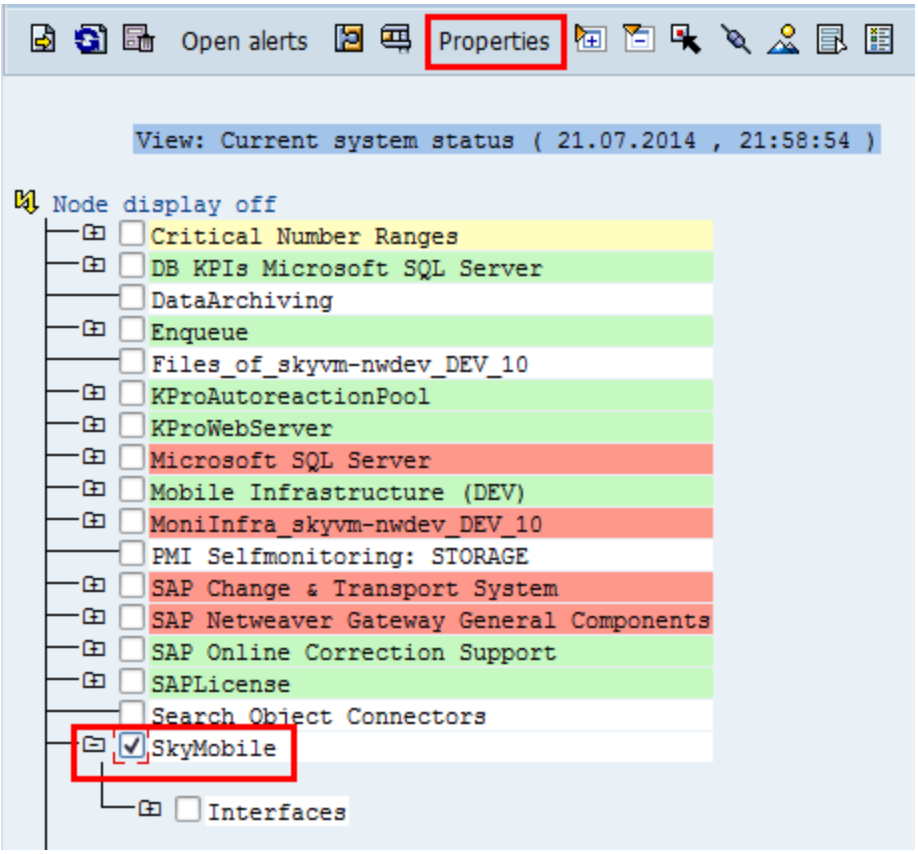
This section describes the necessary requirements before you define custom metrics and alerts.

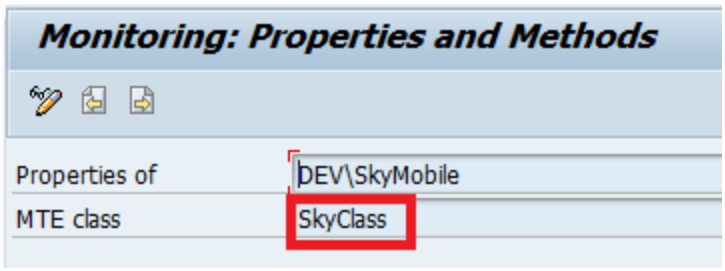
- You can create custom metrics and alerts only for Solution Manager Version 7.1, and you need to use SOLMAN_ADMIN user to create Metric/Alert.

- To find out if you must configure a CCMS MTE metric in a technical instance template or a technical system template, check the MTE in the CCMS (RZ20 Tcode). If the MTE is placed under a sub-tree that has the same name as the instance, then MTE belongs to a technical instance template.



- The parameter value is the full name of the CCMS MTE (for example, `SID \ srv_SID_00 \ Object_name \ Attribute_name`).





- The parameter value is a managed object name - for example, an instance name.

> ***Note:*** The names of the managed object and the names of the CCMS object are completely different.

- The correct object name is the second last part of the full name of the CCMS MTEs.

In custom metrics and alerts, you can perform the following operations:

1. Creating a Custom Metric
2. Creating a Custom Alert and Assigning it to a Custom Metric
3. Troubleshooting

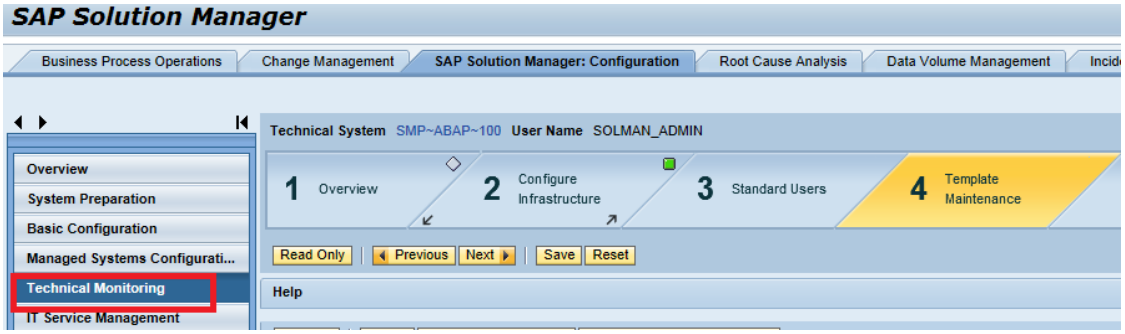
You can create custom metrics and alerts only in a custom template. The first step is to copy the SAP standard template for the system type you want to create the metric into a custom template.

You may note that metrics can be on an instance or on the system level. You must create a metric in a template for a Technical Instance on an instance level. You must create a metric on system level in a template for a Technical System.

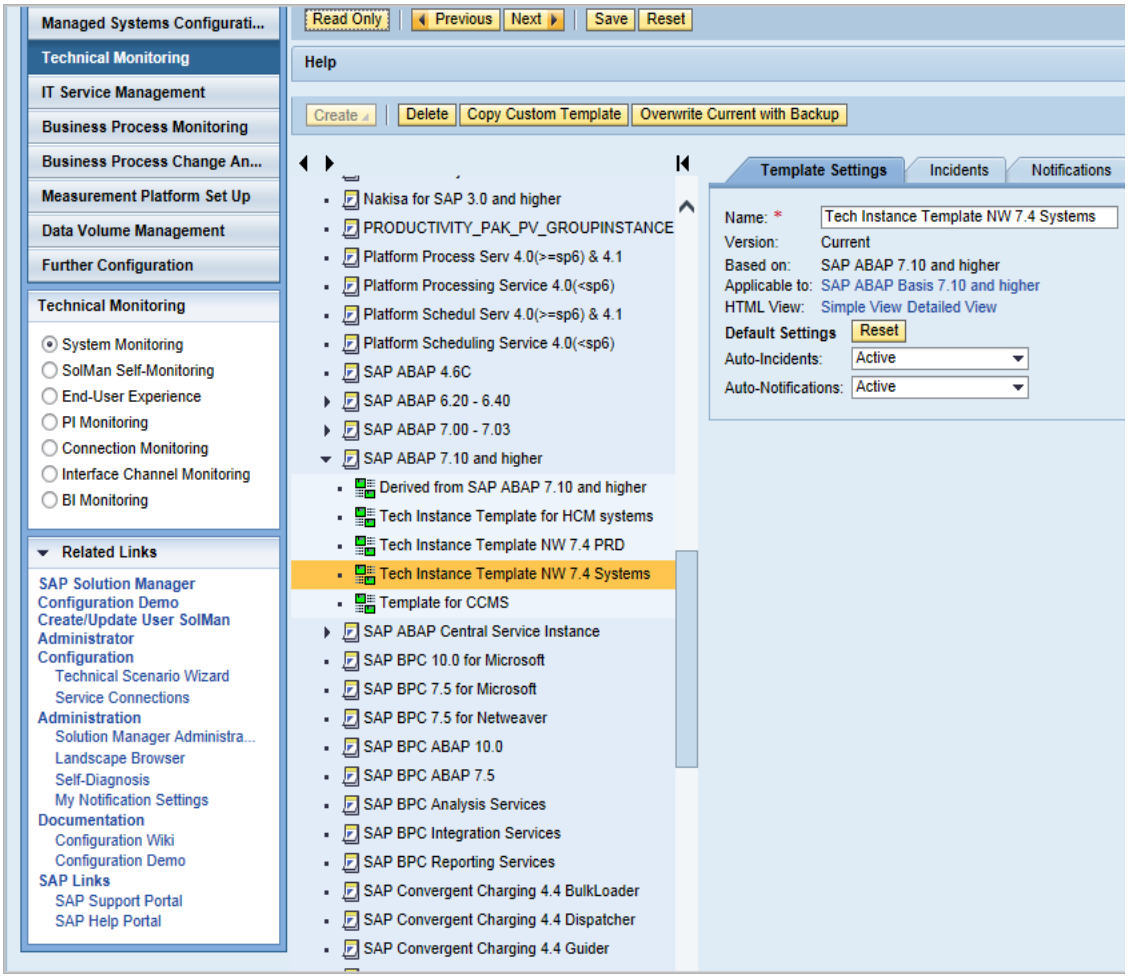
5.8.9.2 Creating a Custom Metric

To create a custom metric, follow these steps:

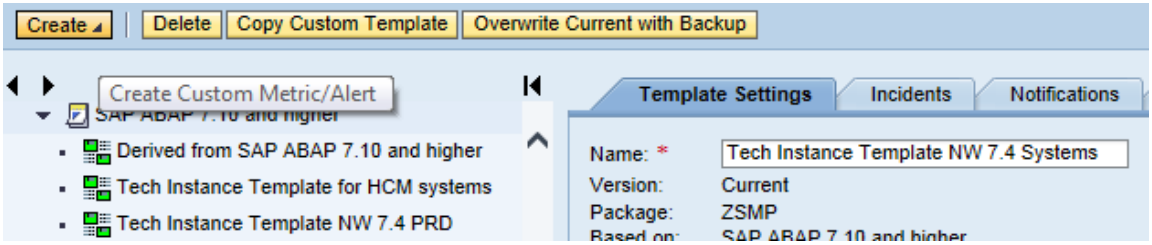
1. Run the transaction, **SOLMAN_SETUP** in the SAP system and switch to **Technical Monitoring** > **Step 4 Template Maintenance**.



2. Select the correct SAP standard template for the system and switch to change mode: If a custom template is already used, you can use it.

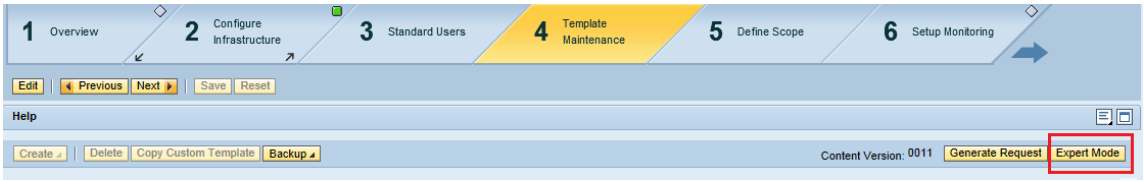


3. Click **Create**.

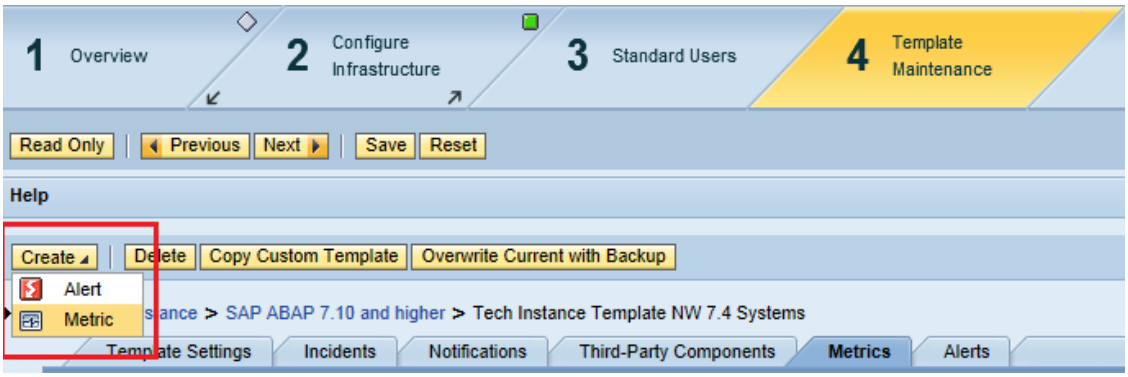


4. Enter a name for your template in **Name** and save it.

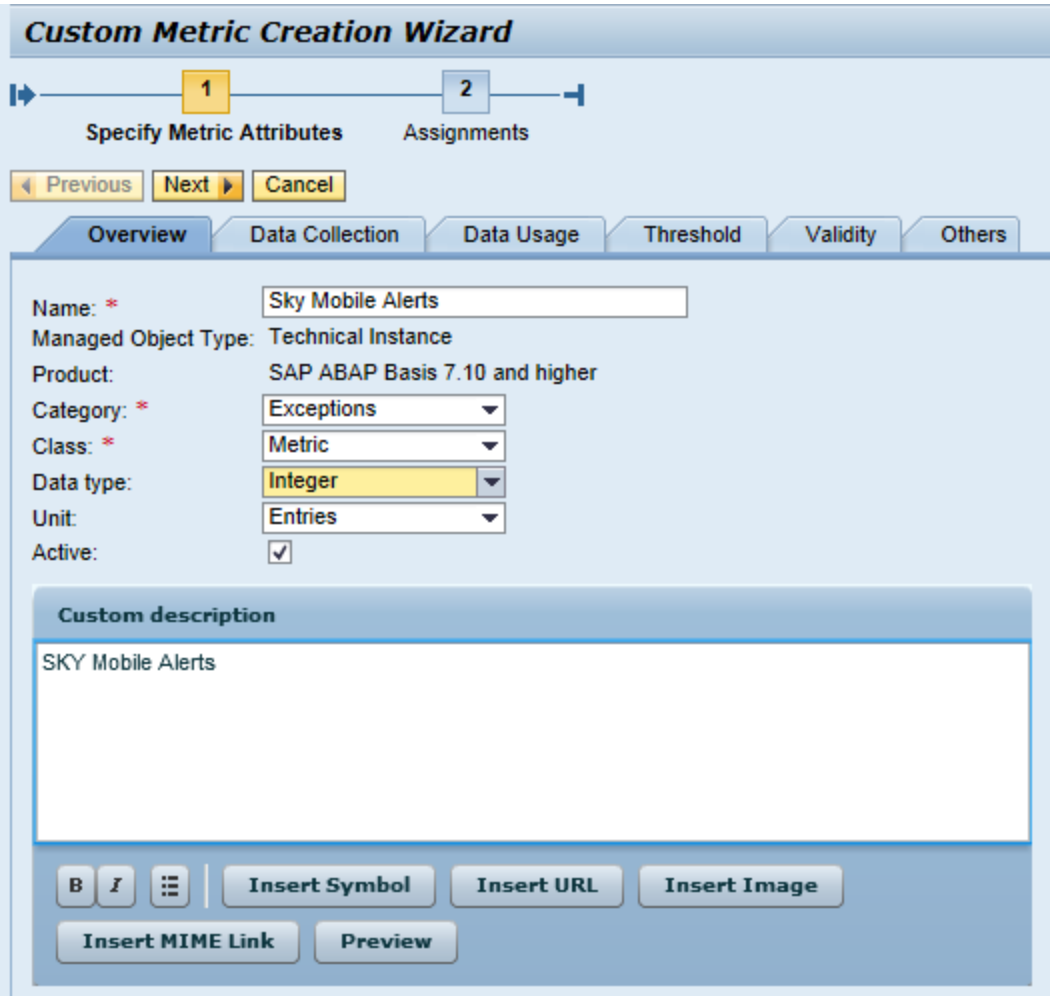
5. You can only create custom metrics and alerts in expert mode. Switch to **Expert Mode**.



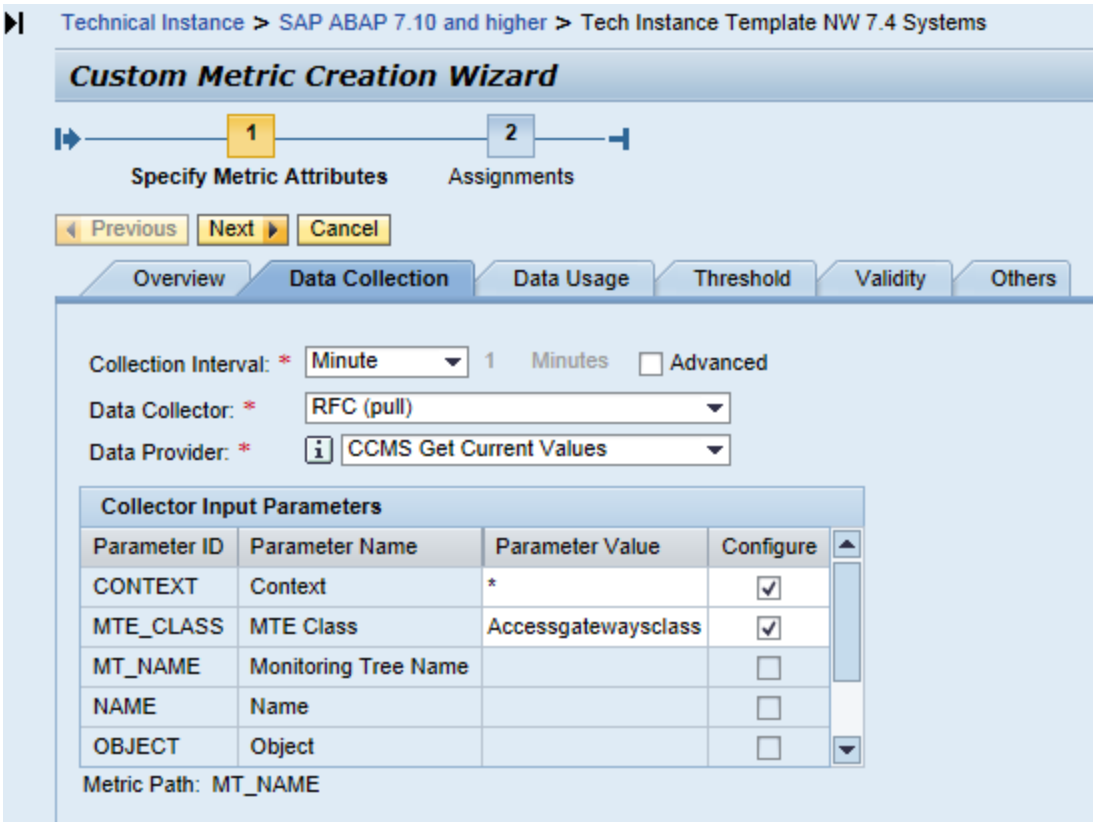
6. Click **Create** > select **Metric**.



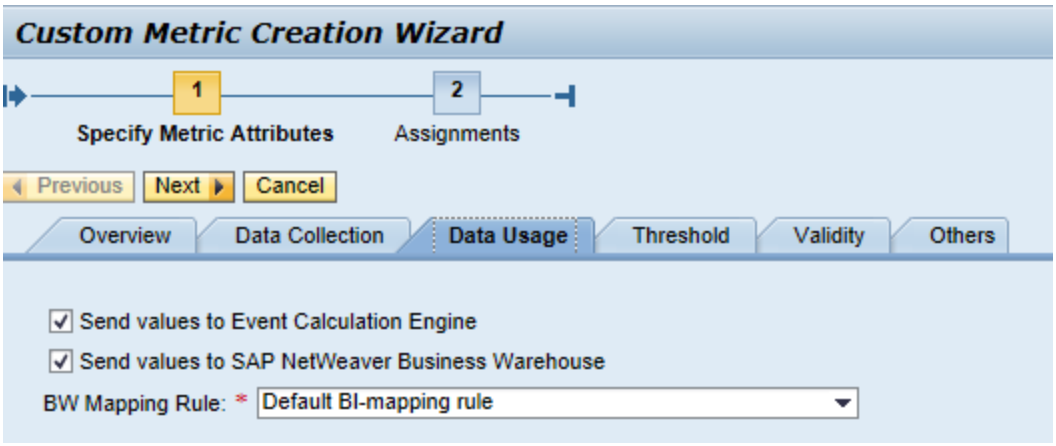
The **Custom Metric Creation Wizard** appears.



7. Under the **Data Collection** tab, you maintain the data collection. Make sure that you select the right data provider. For example, it has to be the CCMS MTE data provider that uses the data provider coding `/SDF/E2E_CCMS_MTE`. All CCMS data providers belong to the data collector type **Pull RFC Collectortype** as they are called through an RFC destination to the managed system.



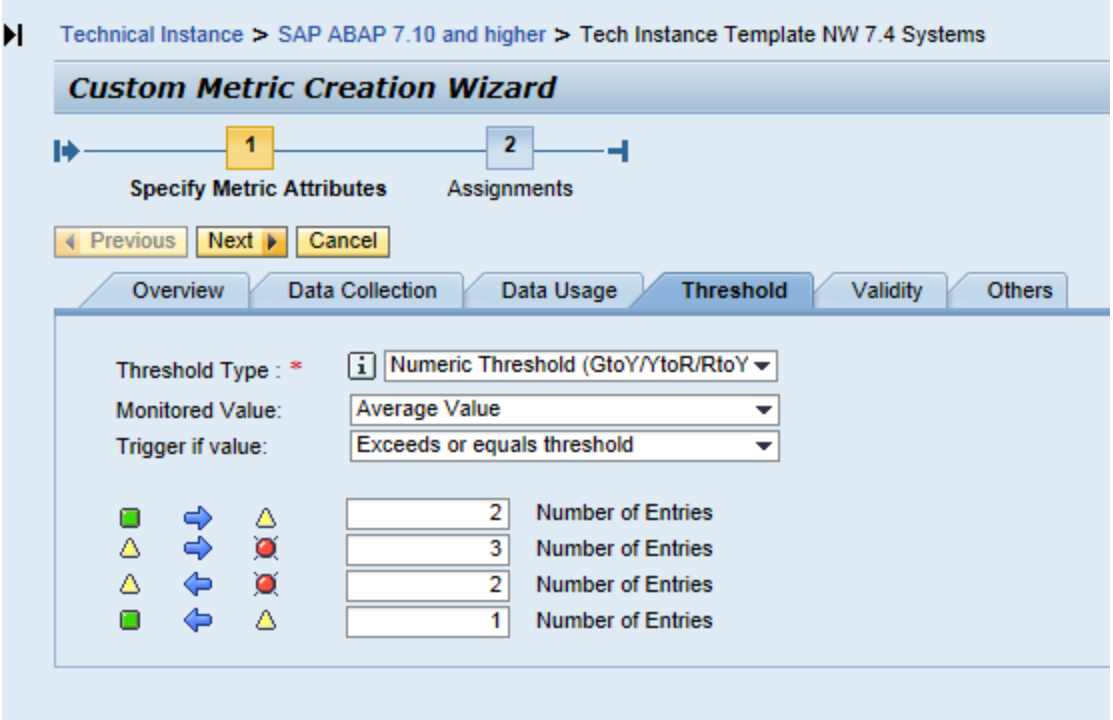
8. On the **Data Usage** tab, you can define if the metric need to be used for monitoring (Alert Store) or for reporting (Metric Store) only. You can also change the Business Warehouse (BW) mapping rule.



a. To generate the metric, select the **Send value to Event Calculation Engine** check box.

b. To generate the alert, select the **Send values to SAP NetWeaver Business Warehouse** check box.

9. On the **Threshold** tab, you have to define the thresholds for the metric. This is necessary for metrics collected by "CCMS MTE", as the return value is a numerical value. For this metric, all numeric thresholds work.

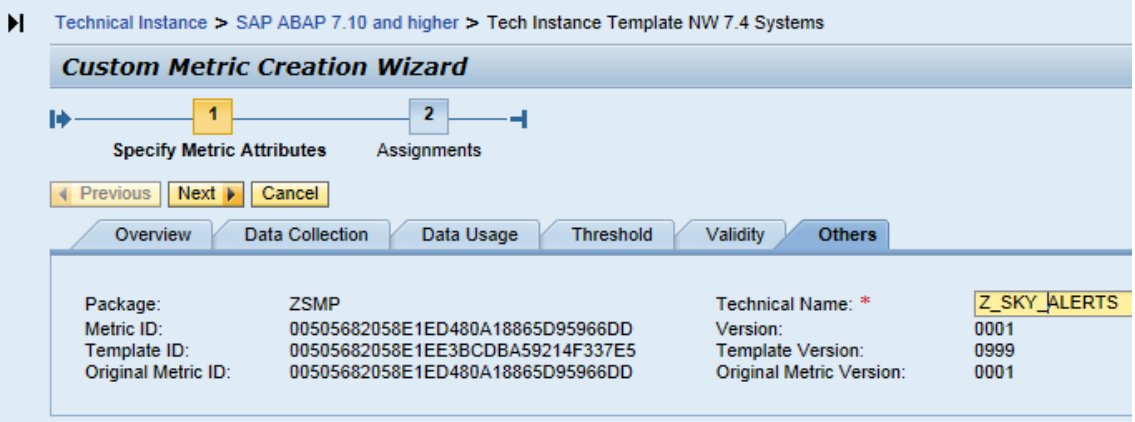


Based on requirement, you need to define the metric value. For example, in Custom Metric Creation Wizard, the metric value change from

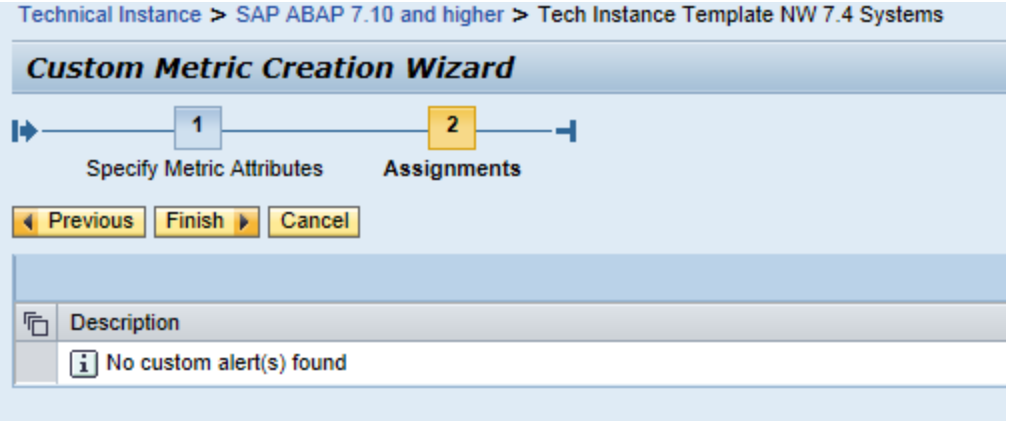
- Green to Yellow is 2
- Yellow to Red is 3
- Red to Yellow is 2
- Yellow to Green is 1

10. Under the **Validity** tab, you can define if the metric has a special validity, for example, is only valid if collected for a virtual host. Do not change any details on this tab, unless you know what to do.

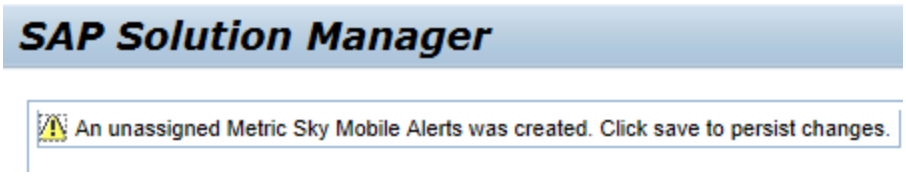
11. Under the **Others** tab, define a technical name for the metric. Make sure that the name is in customer namespace, starting with "Z" character.



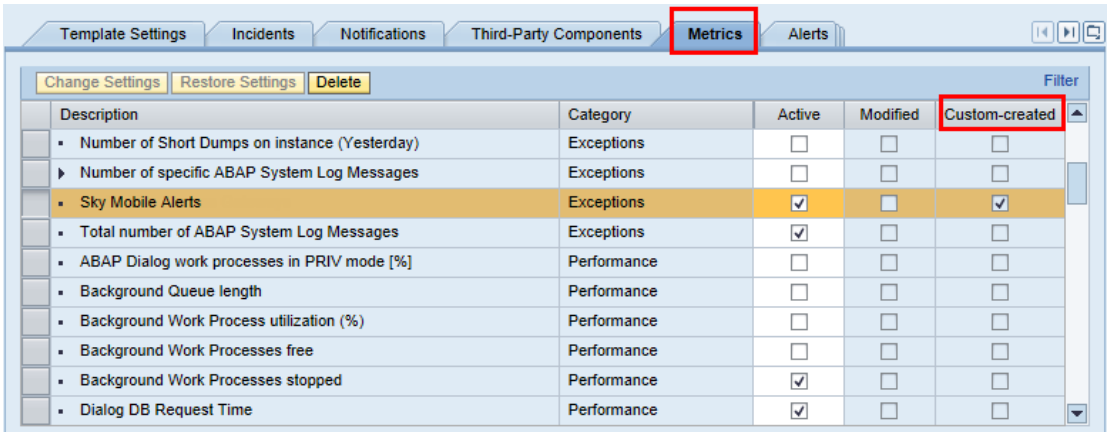
12. In Step **2**, assign the metric to a custom alert in the same category. You do not have a created alert yet, so no alert is available. Click **Finish**.



The **SAP Solution Manager** dialog appears.



You can see your metric under the **Metrics** tab of the template. It is flagged as **Custom-created**.



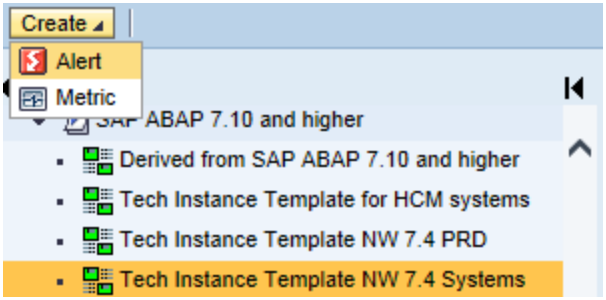
5.8.9.3 Creating a Custom Alert and Assigning it to a Custom Metric

The custom metric alone is not helpful. You need a custom alert to use it. If you have other metrics that fit in the same context (for example, other metrics that describe the custom metrics on instance level), then you can create these metrics first. You can later on, assign all of them to the same alert to group them.

You only have this metric for now. Create a custom alert for your metric.

To create a custom alert, follow these steps:

1. Click **Create** > select **Alert**.

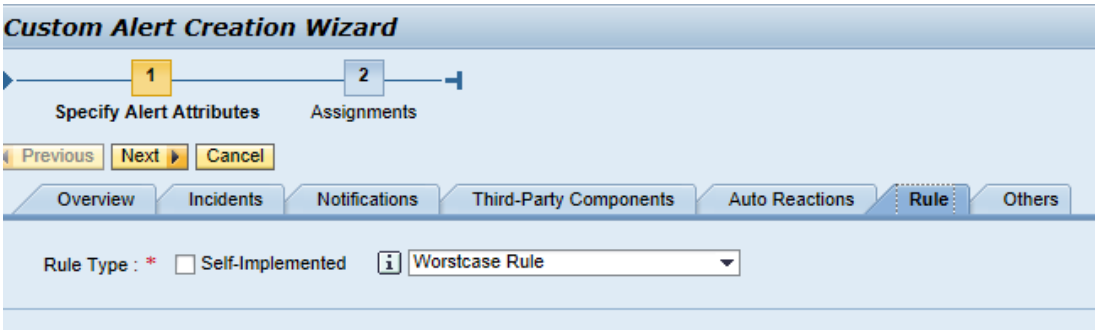


2. Alerts do not need much maintenance. On the **Overview** tab, you maintain the alert name, the category, the severity and the custom description. Make sure that category is the same as the category of your metrics; otherwise you may not be able to assign the metrics to the alert.

3. You can use the input help, **Insert Symbol** tab to insert variables for your managed system context to the custom description. This variable is replaced with the description of the actual instance at run-time.

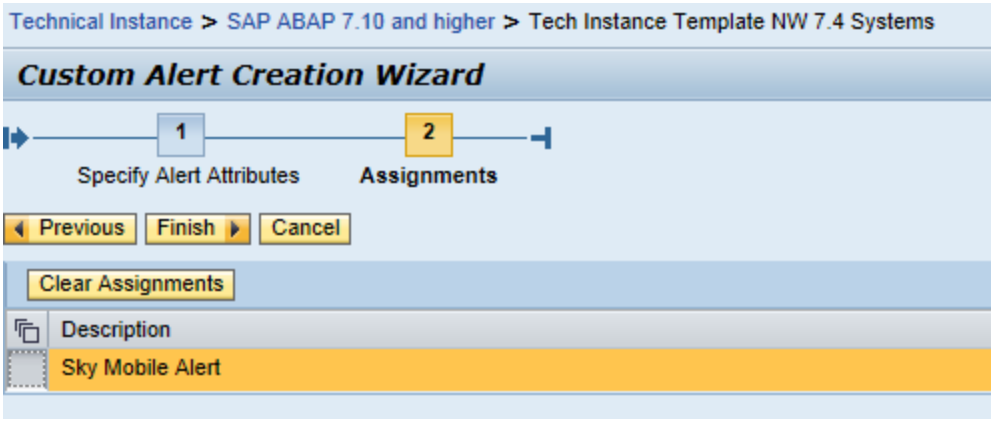
You can maintain the **Incidents**, **Notifications**, **Auto Reactions** and **Third-Party Components** tabs. You can also maintain the tabs later on.

The rule is per default, a worst case rule. This means an alert is raised, if one metric belonging to this alert has a red or yellow rating. This is applicable for most of the alerts. Per default, you can also have a best case and an average case rule. You can also implement an own alert rule, if the provided ones are not sufficient.



4. Under the **Others** tab, you again need to assign a technical name to your alert. Make sure it is in customer namespace starting with the "Z" character.

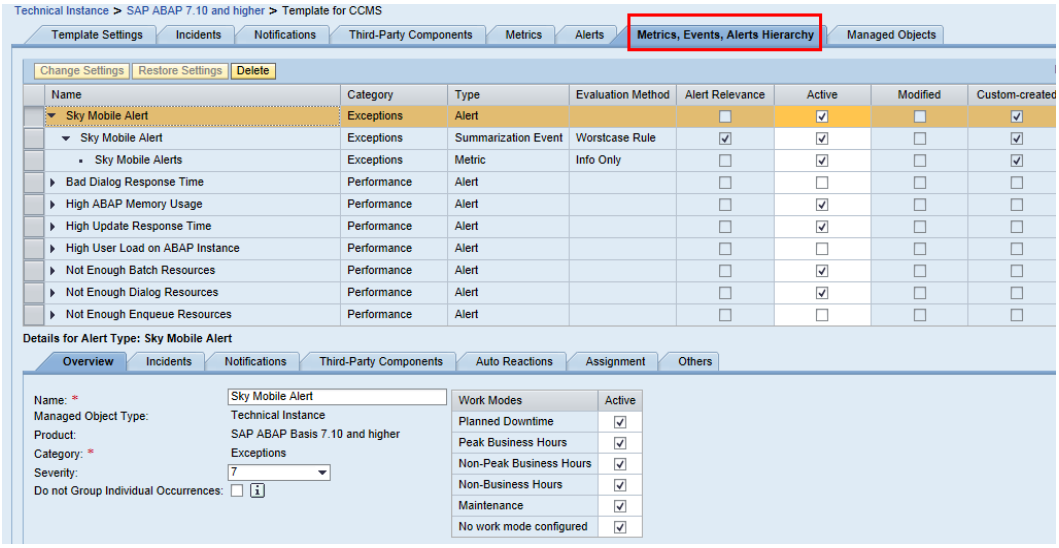
5. Click **Next**.
   In the next step, you now see all unassigned metric of the category the alert is created. Select the metrics you want to assign to this alert from the list (press **CTRL** to select multiple alerts).



6. Click **Finish**.
   The following alert appears.

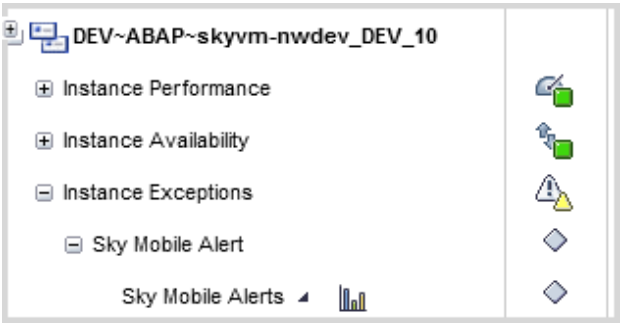
Make sure that you save the template to save your changes.

You can now see your alert with the assigned metric on the **Metrics**, **Events**, **Alerts Hierarchy** tab.



After you apply your alert and activate your template to a managed object, you can find your metric in the system monitoring application.

After all the extractors have run, the values are reported in the **Solution Manager**.



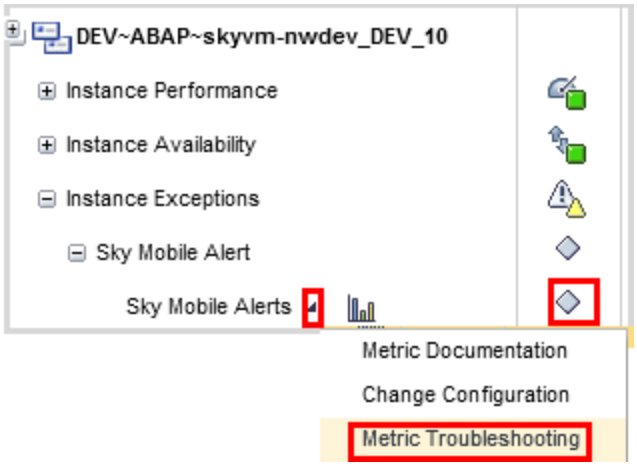
5.8.9.4 Troubleshooting

If no values for your self-provided metrics are delivered, check the parameters of your data collection implementation.

To check the parameters of your data collection implementation, follow these steps:

1. Call the directory browser (Transaction code, **SOLMAN_SETUP** and select Technical Monitoring) , and select the corresponding managed object.

2. In Technical Monitoring, choose the **Metrics** tab > select the metric that is grey in color as marked in the below image. Click the metric. From the drop-down list that appears, select **Metric Troubleshooting**.



   For that metric, choose the **Data Collection** tab in the **Details for Metric** area.

3. Check the values of the **Collection Parameter**. The values need to be same as the corresponding values for the CCMS metric.

4. If **OBJECT_NAME** is key of one of the **Collection Parameter**, check for the following errors that already occurred in real-life:

- The parameter value is empty (space).
- The parameter value is the full name of the CCMS MTE (for example, `SID \ srv_SID_00 \ Object_name \ Attribute_name`).

The parameter value is a managed object name, for example, an instance name.

> ***Note:*** The managed object names and CCMS object names are completely different.

If one of the above mentioned errors occur, correct the value of OBJECT_NAME in the implementation of the data collector analog to the implementation mentioned in the Prerequisites section.

5.9 High Availability and DRP Mechanisms

Kony for SAP provides options to automatically 'switch over' to one or more alternate secondary connections if the primary connection is not available. It also provides automatic backup and recovery mechanisms to make sure data is not lost and applications are in a stable state after device failure. The following high availability and DRP (Disaster Recovery) options are available to provide a fully robust infrastructure in case of failure. Contact Sky for more information.

5.9.1 Automatic Network Re-connection

In the case of a stand alone application, Kony for SAP automatically detects that the network connection and / or host system is not available. A distinct tick or cross at the bottom left corner of the screen indicates this status to the user. The system automatically attempts to reconnect lost network connections in the background (configurable) and re-authenticates using credentials in its secure local cache. In this way, network connectivity is transparent to the user. The user may also "self test" network connectivity to help identify which component of the infrastructure is at fault. Refer to Device about screen for more details.

5.9.2 Gateway Switchover

You may configure multiple gateways to enable automatic switchover if one or the other is not available. This means that remote devices in the field may still connect to, and synchronize data with, the backend SAP system. You may also deploy multiple gateways to help load balance a high number of connections, that is, you may configure each gateway to handle a maximum number of connections and if this is exceeded, the client switches to use another. Refer to Gateway switchover for more details.

5.9.3 Stand Alone and Store and Forward Modes

Kony for SAP fully supports "stand alone" applications, that is, can run totally independent of the SAP host. Data synchronization is performed automatically in the background when a host connection is re-established. In addition to "stand alone", a store and forward mode is also supported where transactions are performed real-time against the host and if the connection is lost, are stored and processed asynchronously when a connection is next available. Refer to SAP Host Store and forward manager for more information.

5.9.4 Local Backup or Recovery Mechanism (Windows only)

The Java server on Windows has a built-in data backup and recovery mechanism that can take full and/or delta backups of the local database on the device while the application runs. These backups are written to removable disk, for example, a SD card. This mechanism provides 'point in time' recovery of software and data, should the device be damaged or corrupted in some way thus minimizing the amount of down time experienced by the user. Refer to Java Server Backup and Recovery (Windows only) section for more information.

5.9.5 Host Backups

The SAP add-in resides inside SAP and all its data and objects are stored in SAP tables. The SAP add-in inherits the same backup / recovery strategy as the host SAP system. Therefore, no specialized procedures are necessary. If you are running a central Java Server or Gateway on the SAP host, then ensure that it is included in the overall OS backup / recovery schema.

5.9.6 Stand Alone Dedicated Servers

If you deploy the Gateway on its own server, or a shared utility server, you should incorporate it into the overall backup / recovery server schema, no special procedures are necessary. The gateway itself holds no data and thus you can easily re-install and configure from scratch if required.

5.10 Logs and Reports

Kony for SAP has a number of logs and reports to provide comprehensive information about various system, server and gateway attributes.

5.10.1 Key Topics

Server Log Utility

Support Report

Users and Sessions Report

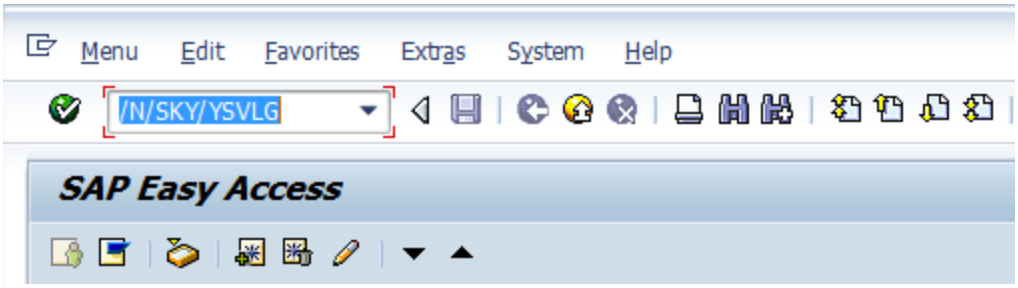
5.10.2 Server Log Utility

The server log utility provides a standard ABAP report to view the server log entries based on server group, server id and date.

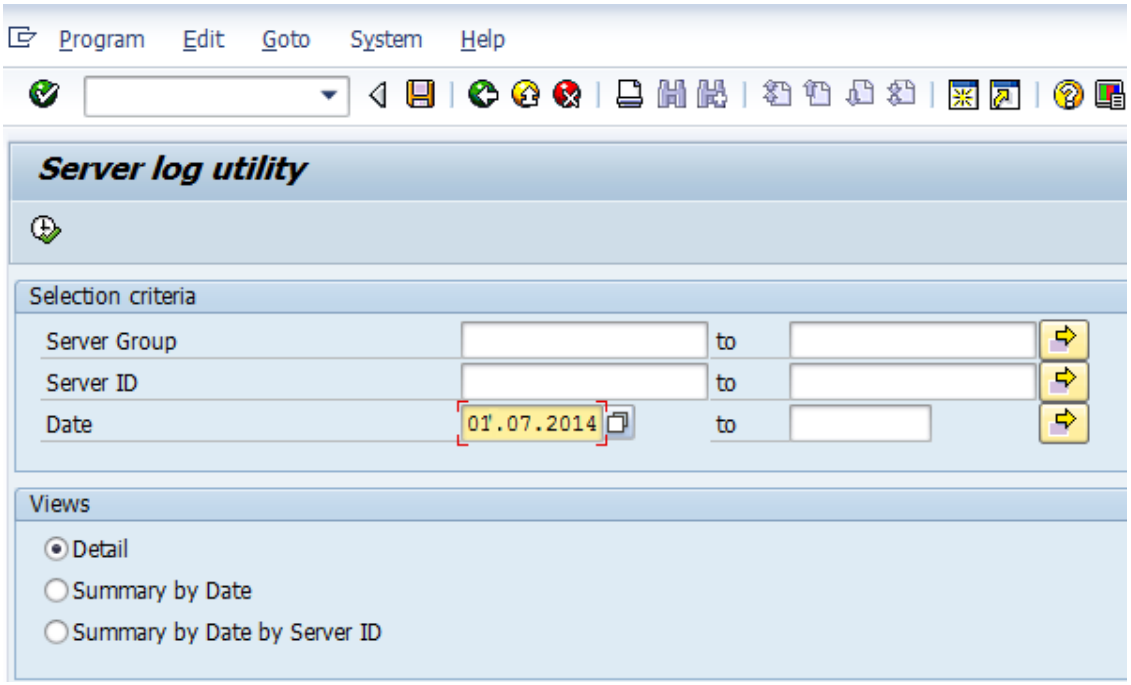
These logs contain a list of error, warning, and abort messages generated by application servers during the period they are connected. A heartbeat command from the server sends these messages to SAP periodically.

To generate the server log view report, follow these steps:

1. Run the transaction **/SKY/YSVLG (YSVLG)** in the SAP system.



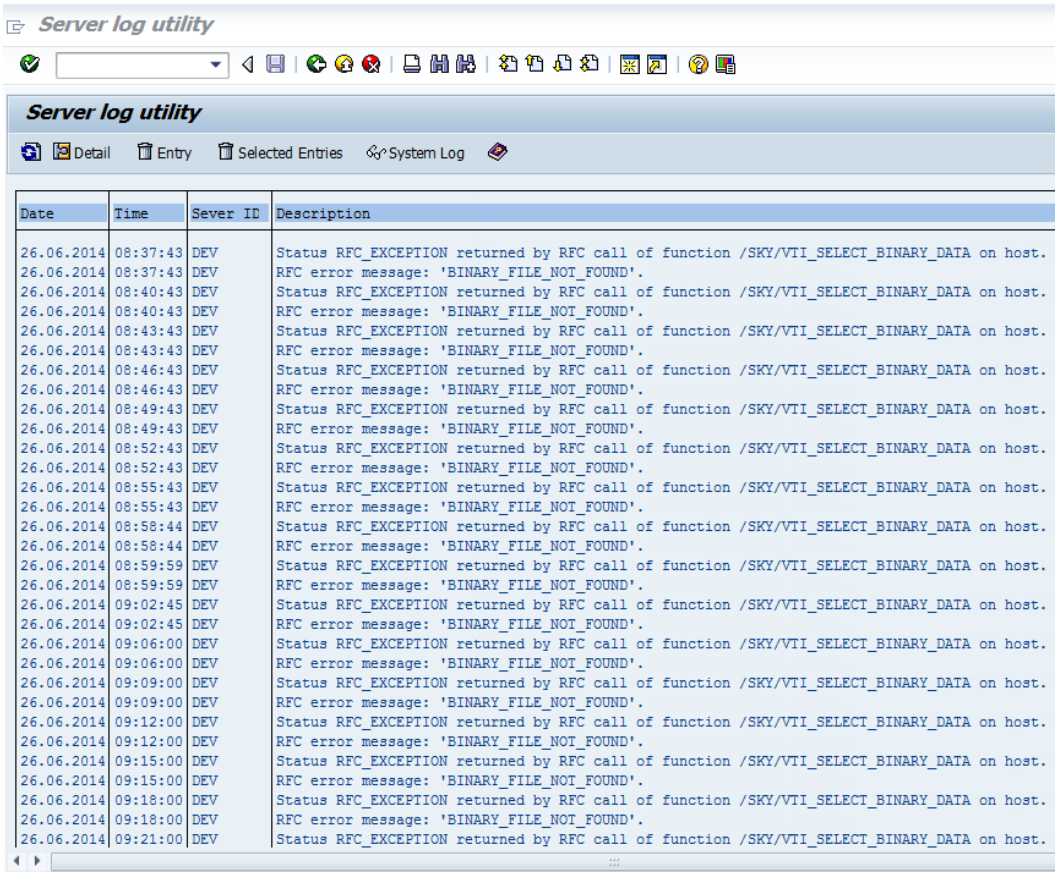
The **Server log utility** selection window appears with the default system date.



2. The server logs are displayed in three views:
   - Detail: Provides in-depth information about the server log.
   - Summary by Date: Provides a summary of the server log entries by date.
   - Summary by Date by Server ID: Provides a summary of the server log entries by date and Server ID.

5.10.2.1 Detail View

a. On the **Server log utility** window, under the **Views** section, select the **Detail** option and click **Execute**.
   The **Server log utility** window appears with details.



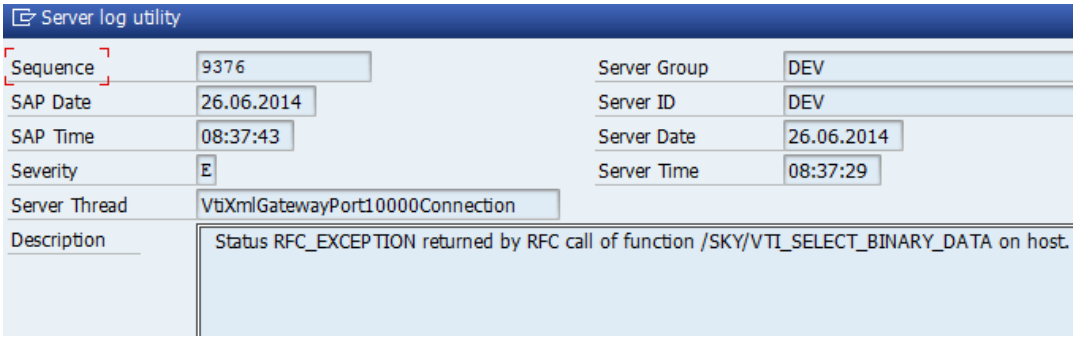
Date	Time	Sever ID	Description
26.06.2014	08:37:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:37:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:40:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:40:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:43:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:43:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:46:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:46:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:49:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:49:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:52:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:52:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:55:43	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:55:43	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:58:44	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:58:44	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	08:59:59	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	08:59:59	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:02:45	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:02:45	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:06:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:06:00	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:09:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:09:00	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:12:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:12:00	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:15:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:15:00	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:18:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
26.06.2014	09:18:00	DEV	RFC error message: 'BINARY_FILE_NOT_FOUND'.
26.06.2014	09:21:00	DEV	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.

The following table displays the list of buttons available in the application tool bar of the detail view display.

Button	Use
Refresh	Refreshes the displayed data
Detail	Gets the detailed view of the selected record

Button	Use
Entry	Deletes the record based on cursor position
Selected Entries	Deletes all the selected records based on selection screen filtering
System Log	Navigates system log utility by defaulting the server group, server ID, and date of selected log
On line help	Displays the online help about report functionality

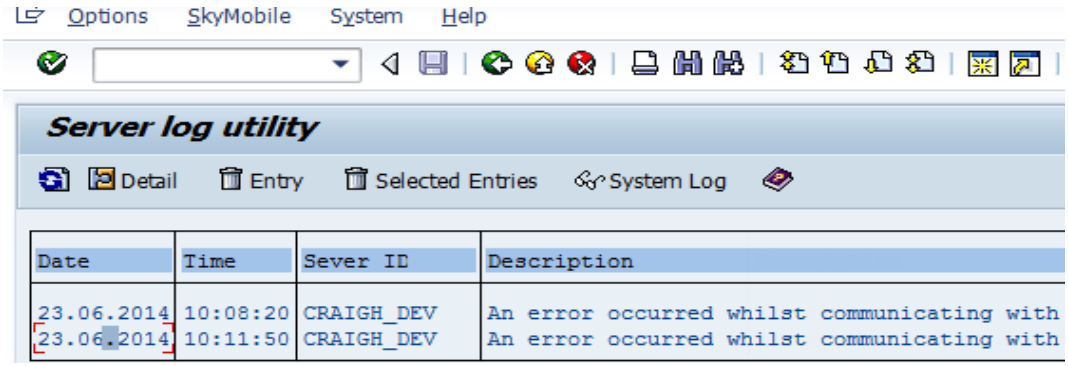
b. Double-click on any row of the detail view.
The **Server log utility** window appears with a detailed view of the selected record as shown below.



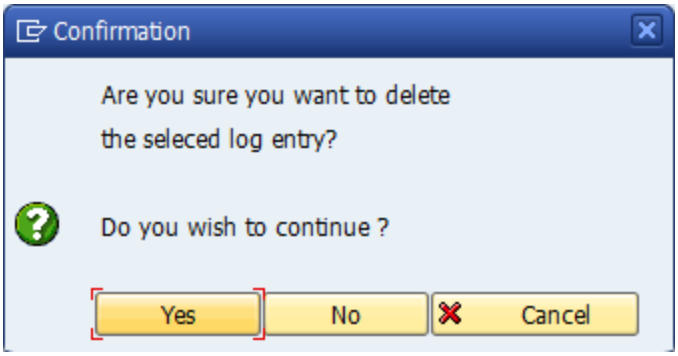
Field	Description
Sequence	Sequence number of the server log
Server Group	Group name of the device

Field	Description
SAP Date	SAP system date when the message is logged
Server ID	Server ID of the device
SAP Time	SAP system time when the message is logged
Server Date	Application server date on which the message is logged
Severity	Severity of the log (error, warning or abort)
Server Time	Application server time when the message is logged
Server Thread	Thread to which the server belongs
Description	Description of the error

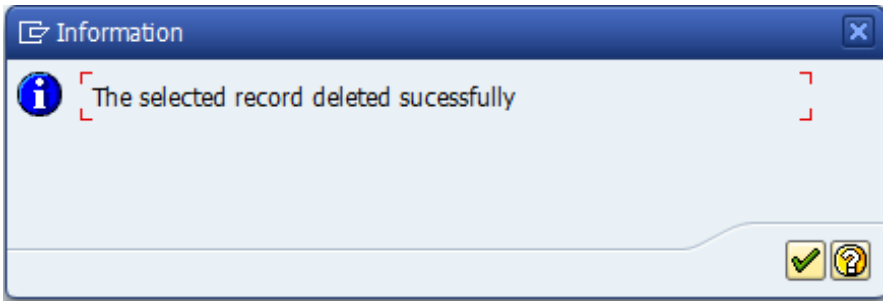
c. To delete the selected record from the displayed set of records, place the cursor on any of the record. Click the **Entry** button on the application tool bar as shown below.



A **Confirmation** dialog appears.

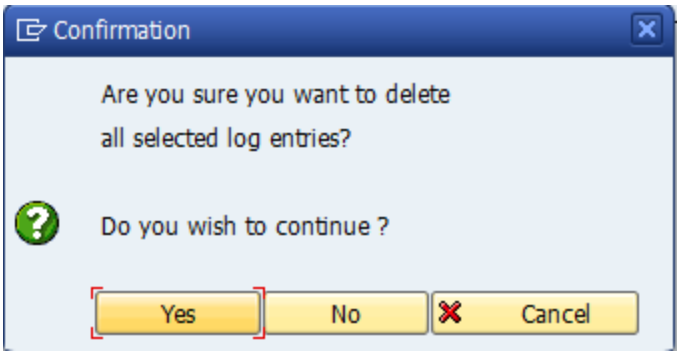


i. Click **Yes** to continue.
The **Information** dialog appears after when the deletion is complete.

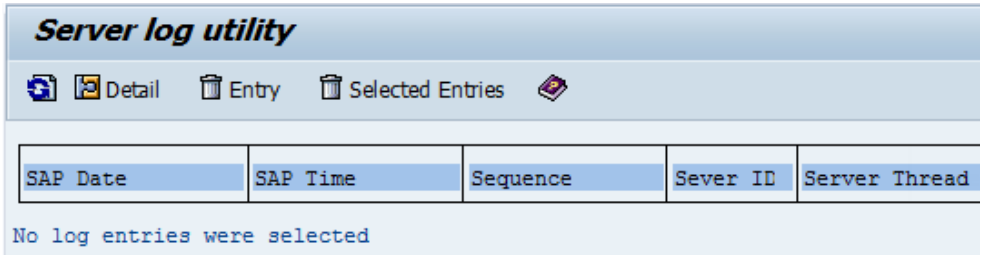


ii. Click **No** or **Cancel** to exit.

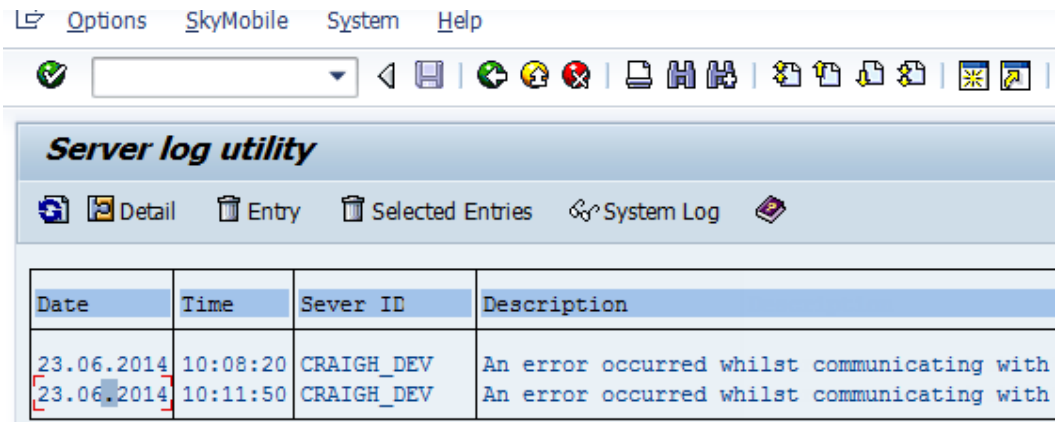
d. To delete the selected range of entries, click the **Selected Entries** button on the application tool bar.
A **Confirmation** dialog appears.



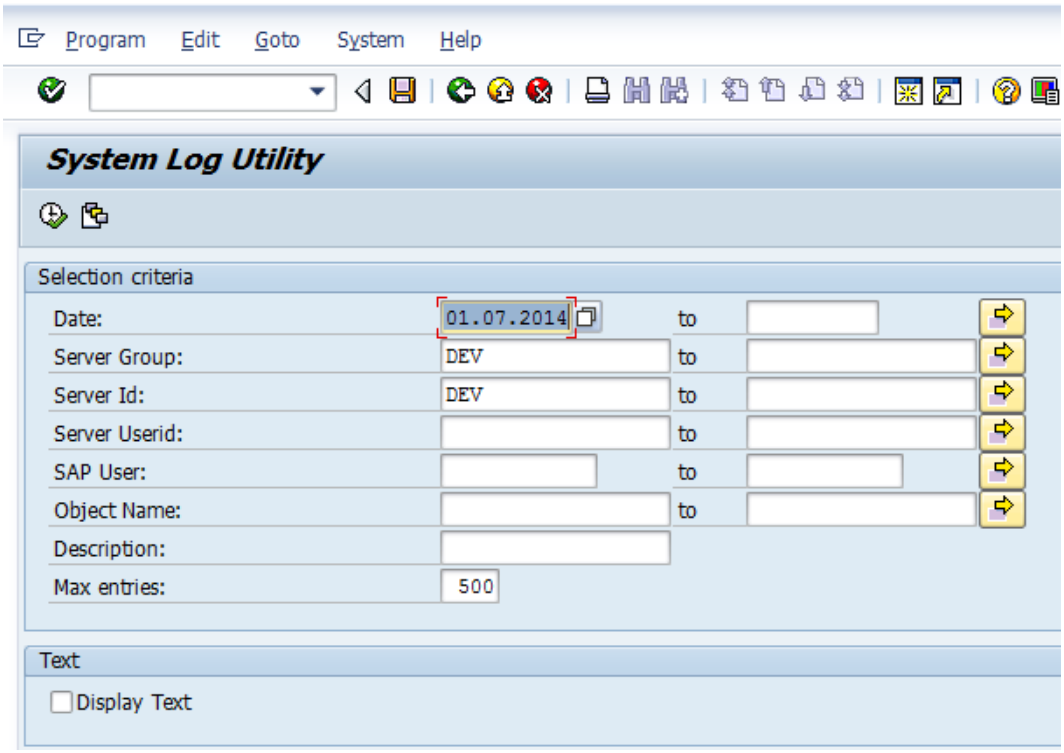
i. Click **Yes** to continue.
All the displayed entries get deleted and the output list is refreshed as below.



e. To use the system log utility, place the cursor on any of the record, and click the **System Log** button on the application tool bar as below.

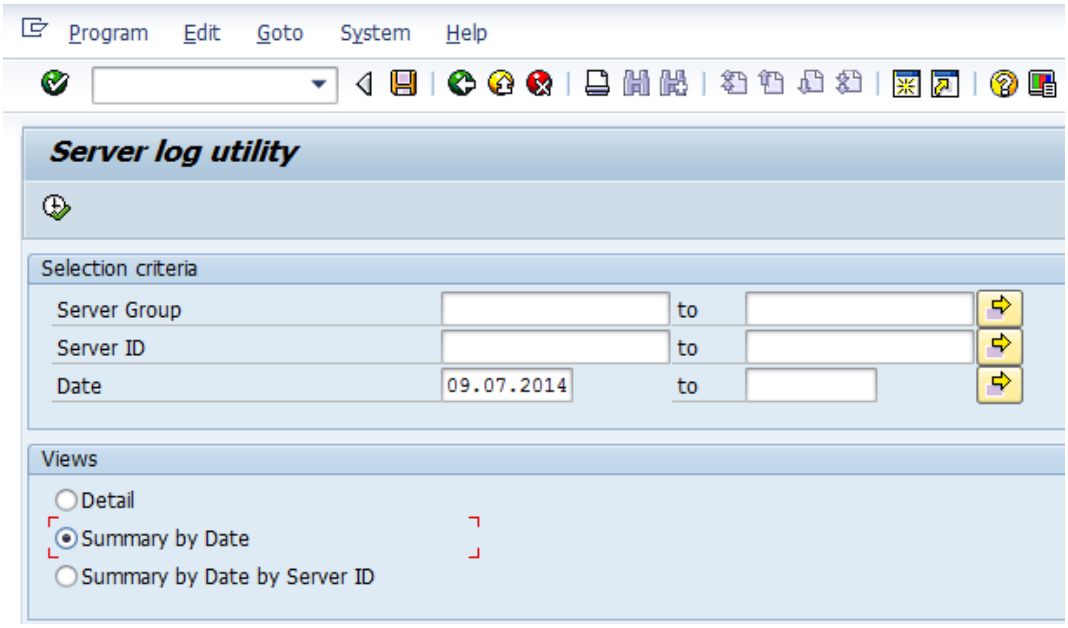


The **System Log Utility** selection window appears with the server group, server ID and date of the selected record.

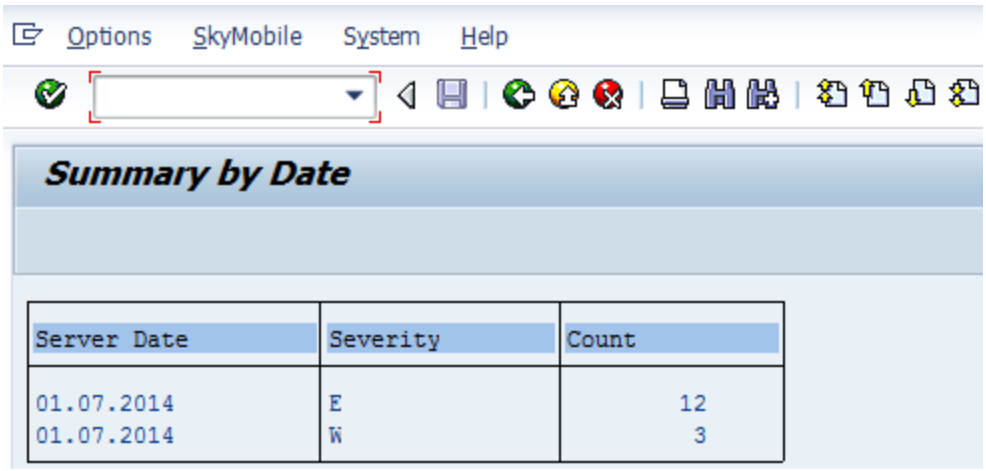


5.10.2.2 Summary by Date View

a. On the **Server log utility** window, under the **Views** section, select the **Summary by Date** option.

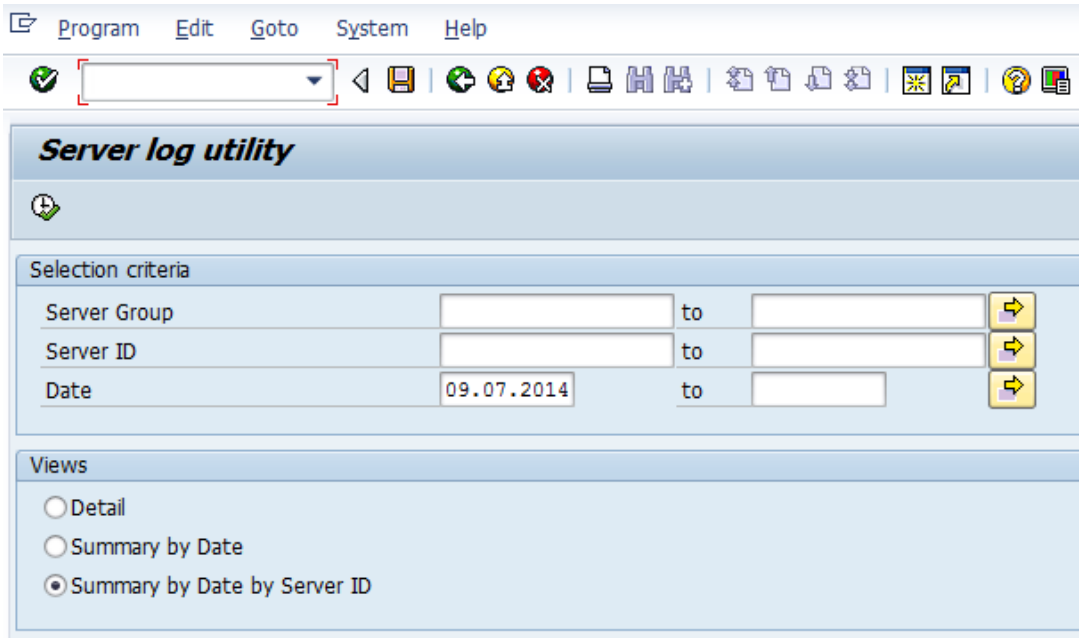


The **Summary by Date** display appears with the summary as shown below.

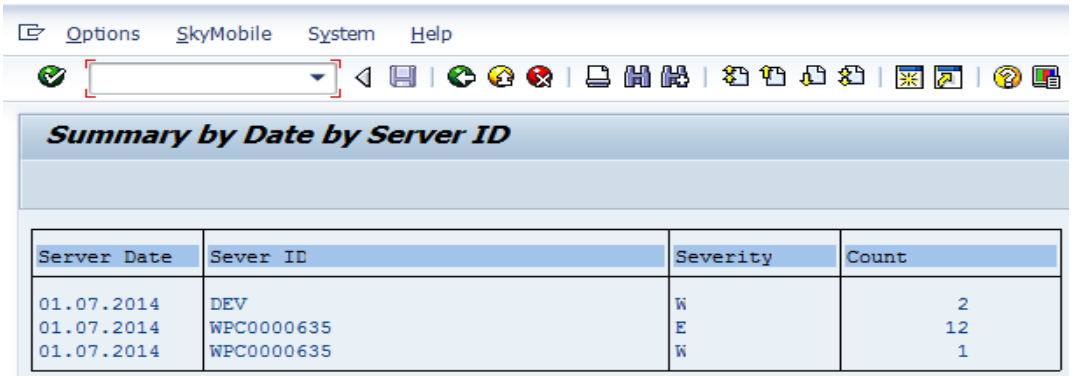


5.10.2.3 Summary by Date by Server ID View

a. On the **Server log utility** window, under the **Views** section, select the **Summary by Date by Server ID** option.



The **Summary by date by Server ID** display appears with the summary as below.

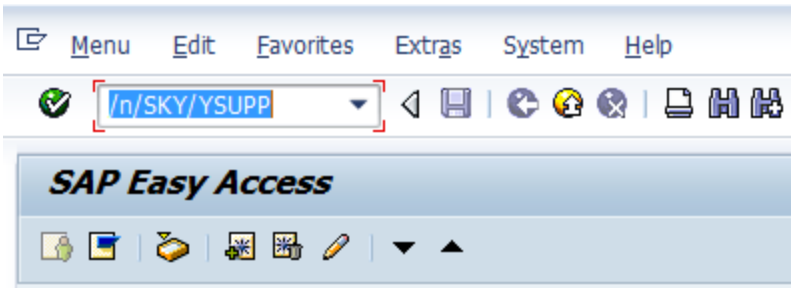


5.10.3 Support Report

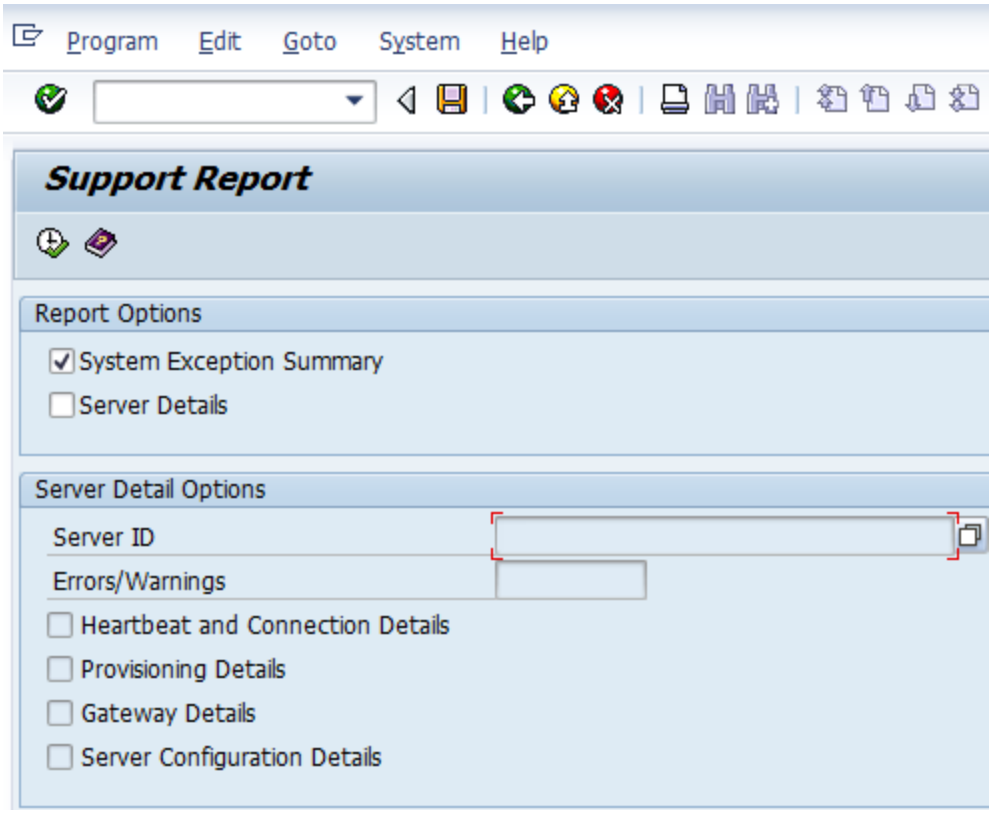
The support report provides a comprehensive information about the system that helps the Kony SAP support team to diagnose problems faster. Customers also find the report useful in diagnosing issues. All the environment and specified server information is collected centrally in SAP and then consolidated into a standard support report.

To generate the support report, follow these steps:

1. Run the transaction **/SKY/YSUPP (YSUPP)** in the SAP system.



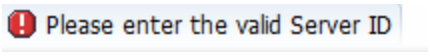
   The **Support Report** dialog appears as shown below.



The following table represents a list of selection screen elements and its usage details:

Selection Screen Element	Use
System Exception Summary	The **System Exception Summary** check box is selected by default when the selection screen appears. This check box displays the section that summarizes the overall system health and highlights areas where major problems occur. If the support report is run by selecting this check box, the following process is performed: All the logs in the system for the last seven days are retrieved, and a concise summary of all errors by date of each category is provided. The following logs are included in the report display: • Add-in Details • Application Server • Access Gateways • Provisioning log • Data object trace (errors) • LDB trace (errors) • ECS interfaces • Session manager • Short dumps (associated with Kony for SAP) • SAP system log messages (associated with Kony for SAP)

Selection Screen Element	Use
Server Details	The **Server Details** check box displays the server details of the input, and the **Server ID** (entered on the selection screen field **Server ID**). If you do not select the **Server Details** check box, only the **Report Options** section appears in the report output. If you select the **Server Details** check box, the following selection screen elements are enabled and the corresponding server specific sections appear in report output • Server ID • Errors/warnings • Heartbeat and connection details • Provisioning details • Gateway details • Server configuration details

Selection Screen Element	Use
Server ID	The **Server ID** parameter is disabled by default when the **Support Report** window appears. This parameter is enabled when you select the **Server Details** check box. If you run the report with the Server ID blank, an error message appears as shown below  If you execute the report with an invalid Server ID, an information message appears as shown below. The report output appears with the **Report Options** summary:  If you run the report with a valid server ID, the output appears with both the **Report Options** and **Server Detail Options** summary sections.

Selection Screen Element	Use
Errors/Warnings Since	The **Errors/Warnings Since** parameter is enabled when you select the **Server Details** check box. The date defaults to the System date (minus two days). If you run the report with the **Errors/Warnings Since** date parameter blank, the following sections appear in the report output: • Gateway Server Error Logs in **Gateway Details** • Errors and Warnings If you run the report with a valid date in **Errors/Warnings Since** parameter, the following sections appear in the report output: • A list of gateway server error logs appears since the specified date. This contains a maximum of 100 entries. This action correlates any server errors with errors occurring on the gateway, for example, timeouts or Identity issues. • **Errors and Warnings** section appears with all the known errors and warnings for the server since the specified date. Log entries from the following sources appear: ◦ System Log (maximum 100 entries) ◦ Server Log (maximum 100 entries) ◦ DOB Logs (maximum 100 entries) ◦ LDB Logs (maximum 100 entries)

Selection Screen Element	Use
Heartbeat and Connection Details	The **Heartbeat and Connection Details** check box is enabled and selected when you select the **Server Details** check box. If you execute the report by selecting this check box, the server registration and connection details of entered Server ID are retrieved and displayed in the **Heartbeat and Connection Information** section.
Provisioning Details	The **Provisioning Details** check box is enabled and selected when you select the **Server Details** check box. If you run a report by selecting the **Provisioning Details** check box, all server provisioning details of Server ID are retrieved and displayed in the **Provisioning details** section. The application summary also appears.
Gateway Details	The **Gateway Details** check box is enabled and selected when you select the **Server Details** check box. If you run the report by selecting this check box, the system gets the gateway server registration details of the Server ID and lists the details in **Gateway Details** section. The gateway server error logs since the **Errors/Warnings** date also appear.
Server Configuration Details	The **Server Configuration Details** check box is enabled and selected when you select the **Server Details** check box. If you execute the report by selecting the **Gateway Details** check box, the server configuration details of the Server ID are retrieved and displayed in the **Server Configuration Details** section.

The following table represents application toolbar buttons on the selection screen:

Button	Use
 Online help	Displays online help for the transaction **/SKY/YSUPP**.

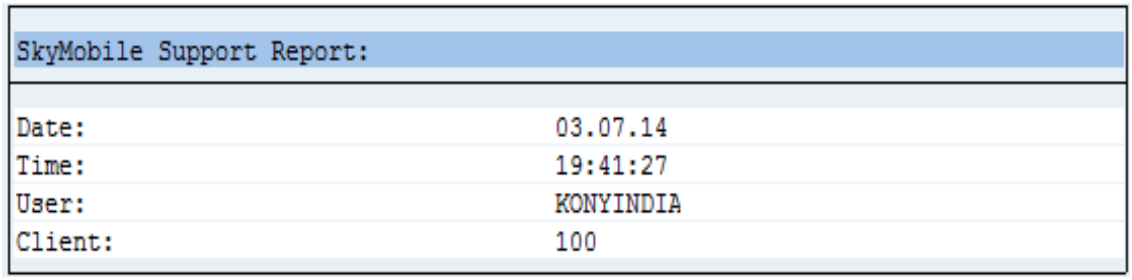
5.10.3.1 Report Output

Running the report with the proper selection screen data displays the report output with the following summary sections:

- Report Options
  - Add-in Support Report
  - SAP System
  - Add-in System
  - Add-in Access Gateways
  - Add-in Server Release Summary
  - Hearbeat Connection and Session Summary
  - System Exception Summary
- Server Detail Options
  - Heartbeat and Connection Information
  - Provisioning Details
  - Gateway Details
  - Errors and Warnings
  - Server Configuration Details

Support Report

The Support Report section appears with the generated report details.



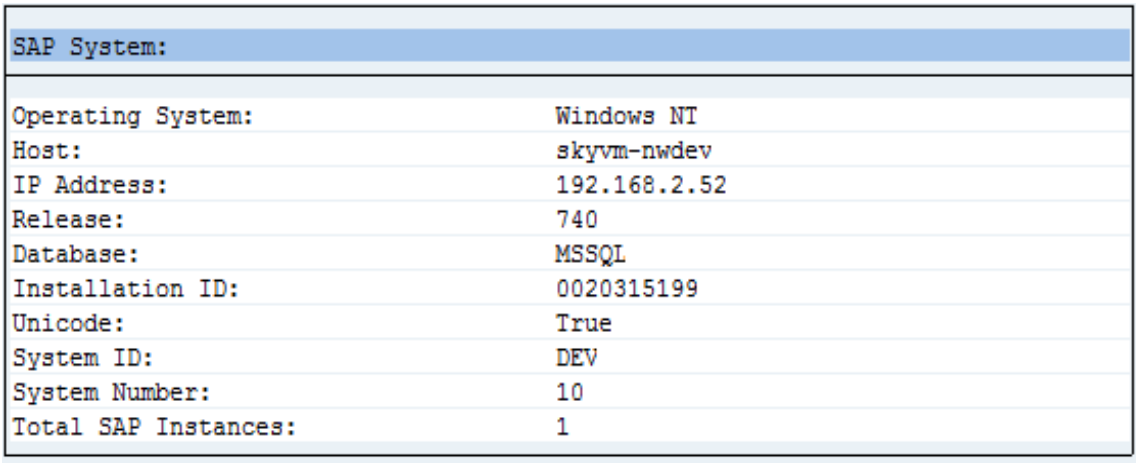
SkyMobile Support Report:	
Date:	03.07.14
Time:	19:41:27
User:	KONYINDIA
Client:	100

The following table represents the fields and field description:

Field	Description
Date	Date on which the report is generated
Time	Time when the report is generated
User	User who generated the report
Client	SAP client number on which the report is generated

SAP System

The SAP System attributes section contains the following SAP system details:



SAP System:	
Operating System:	Windows NT
Host:	skyvm-nwdev
IP Address:	192.168.2.52
Release:	740
Database:	MSSQL
Installation ID:	0020315199
Unicode:	True
System ID:	DEV
System Number:	10
Total SAP Instances:	1

Field	Description
Operating System	Server operating system on which SAP is installed
Host	SAP host name of server on which the report is generated
IP Address	SAP server IP address on which the report is generated
Release	Installed SAP release number of the server on which the report is run
Database	The type of the database that is used by the SAP server
Installation ID	SAP installation ID
Unicode	SAP system is Unicode, check enabled or not (displays *True* or *False*)
System ID	SAP system ID on which the report is run
System Number	SAP system number on which the report is run
Total SAP Instances	The number of instances on the SAP server

System

The System section contains the following details:

- Add-in System details
- Licensing information
- General configuration
- ECS configuration
- Add-in configuration
- Application Server configuration

SkyMobile System:	
Current Release:	24.04.01
Registered (Central):	29.05.2014 07:13:47 SKYTECH
Registered (Client):	04.07.2013 14:39:55 SKYADMIN
Last Checked:	29.05.2014 10:23:47 SKYTECH
Previous Release:	24.04.00
Previous Registration:	23.04.2014 03:45:40 SKYTECH
Licensing Information:	
Status:	Licensed
Expires	31.12.99
Installation ID:	0020315199
Maximum Daily Connections:	999999
Maximum Monthly Runs:	999999999
Total Active Connections:	
Total Runs This Month:	
Total Sessions This Month:	5
Total Users This Month:	2
Total Devices This Month:	2
General Configuration:	
Authorization Exit:	
Suppress Sky Menu Options:	False

ECS Configuration:	
Working Directory:	C:\temp
Directory Delimiter:	\
Temporary File Prefix:	YECS
File Copy Command:	
(e.g. YECS COPY):	True
Default Job Prefix:	Y_ECS
Default BDC Prefix:	Y_ECS_
Authorization Exit:	
Turn Off Authorization Check Cache:	False
ECS Shutdown Wait Period:	300
Monitor Job Class:	
Default Print Format:	
Commit Wait At Start:	False
Backgroud Server Attributes:	
Max Concurrent:	1
Fixed Priority 1:	
Wait Interval (Seconds):	5
Max Wait Cycles:	60
Queue Snatch Size:	5
Server timeout (Minutes):	
Don't Failover:	False
Ignore SAP Locks:	True
Commit Work Wait:	False

Server Job Controls:	
Job Class:	
Job Group:	
User ID:	
RFC Group:	
RFC Destination:	
Snatched Queue Entries In Parallel:	False
SkyMobile Configuration:	
Housekeeping:	
Retain Inactive Sessions For:	7
Cancel Idle Sessions After:	8
Retain Inactive Kept Sessions:	90
Retain Completed ISF Runs For:	7
Retain Classification Data For:	30
Retain GPS Container Data For:	1
Retain GPS Host Data For:	7
General:	
Make The Workbench Display Only:	False
Default LDB/DOB Transfer Buffer:	500000
Don't Do Runtime ABAP Syntax Checks:	False
Suppress SAP Screen Optimization:	False
Force Session Manager Commit&Wait:	False
Enable XAI Cross System Support:	True
Ignore DOB/LDB Timestamp Check:	False
White List Only Provisioning:	False
Extended DOB/LDB Upload Check:	False
Activate SAP Session Key Checking:	False
In-Active Time Out:	
Update SkyMobile SAP user:	True
Dialog Only:	False

```
Working Directory (Optional):
Directory:                          c:\temp
Delimiter:                          \
ABAP System Exits (Optional):
Authorization:
Switchboard Manager:
Store and Forward:
Traces:
Retain DOB/LDB Transfer Trace For:  7
Retain App/Ver Refresh Trace For:   7
Retain RFC Call Trace For:          7
Retain Gateway Service Trace:       7
Collect SAP RFC Call Statistics:    False
UTC Timezone Controls:
Host UTC Offset:
Update Automatically:               False
Numeric Formatting (SAP):
Decimal Symbol:                     .
Grouping Symbol:                    ,
Online Help:
URL:    http://developer.kony.com/twiki/pub/Portal/Docs/SAP/KonyforSAP/V2404/De
        fault.htm#SkyMobileTopics/
Local:                              False
```

```
Application Server Configuration:

Heartbeat Command Retention:
Retain Commands For (Any Status):     7
Keep Completed Commands For(Hours):
Keep Pending Commands For(Hours):
Keep Inprogress Commands For(Hours):
Heartbeat Trace:
Directory:                            c:\temp
Delimiter:                            \
Activate Global Trace:                False
With Data Trace:                      False
Registration/Cross Reference Retention:
Unique Registration By Type/Id Only:  True
Keep In-Active Registrations For:     60
Rationalise Cross Reference:          True
Keep In-Active References For:        60
General:
Suppress Heartbeat Server Enqueueing: False
System Log Retention:                 30
Server Log Retention:                 30
Server Group/ID Exit:
TSX Command Interface (specialised):
Disable Suppress Command Interface:   True
Destination:                          SPI_JAVA_CONSOLE
```

The below table represents the fields on the selection screen:

Field	Description
Add-in Details	
Current Release	Release number of the installed Kony for SAP
Registered (Central)	Concatenation of registration date, registration time, and registered user
Registered (Client)	Concatenation of client registration date, registration time, and registered user
Last Checked	Concatenation of last checked date, last checked time, and last checked user

Field	Description
Previous Release	Release number of the previous Kony for SAP release
Previous Registration	Concatenation of previous release registration date, registration time, and registered user
Licensing Information	
Status	Status of Kony for SAP release (for example, Temporary version, Free Version, Licensed, and so on)
Expires	Expiry date of the Kony for SAP. It is applied depending on the service level and maintenance agreement with Kony
Installation ID	The installation number of Kony for SAP
Maximum Daily Connections	The limit for maximum number of daily connections. It is applied depending on service level and maintenance agreement with SKY
Maximum Monthly Runs	Maximum number of monthly runs limit. It is applied depending on service level and maintenance agreement with SKY
Total Active Connections	Total active connections established to Server
Total Runs This Month	Total number of runs by Server in this month
Total Sessions This Month	Total number of sessions opened by Server in this month
Total Users This Month	Total number of unique users connected to Server in this month

Field	Description
Total Devices This Month	Total number of unique devices connected to Server in this month
General Configuration	
Authorization Exit	This is a custom ABAP program that may be installed to further receive authorization requests
Suppress Sky Menu Options	If *True*: All the standard Kony for SAP menu options are suppressed.
ECS Configuration	
Working Directory	Working ECS directory for external file management
Directory Delimiter	Directory delimiter (for example, \...) is used for SAP external file management
Temporary File Prefix	Prefix for ECS generated temporary file names
File Copy Command	SAP external command
(YECS COPY)	*True*: Uses YECS COPY *False*: Uses ECS internal file copy instead
Default Job Prefix	Default job prefix used for all jobs generated by ECS
Default BDC Prefix	Default BDC prefix used for BDC sessions created by ECS

Field	Description
Authorization Exit	This is a custom ABAP program that may be installed to further receive authorization requests
Turn Off Authorization Check Cache	*True*: Turns off the authorization check cache
ECS Shutdown Wait Period	Grace period before forcing shutdown
Monitor Job Class	Blank - defaults same as job, else A/B/C
Default Print Format	Used for ABAP spool output
Commit Wait At Start	*True* : Ensures that the database updates are committed
Max Concurrent	Value *0* : Disables server functionality
Fixed Priority 1	Servers permanently dedicated to priority 1 processing
Wait Interval (Seconds)	Seconds to wait in between while checking the server queue
Max Wait Cycles	Wait cycles to perform before shutting down
Queue Snatch Size	Maximum entries to select for processing each cycle

Field	Description
Server timeout (Minutes)	Maximum minutes spent in-progress, before auto-cancellation
Don't Failover	*True*: Do not auto-switch to job mode, if server is not available
Ignore SAP Locks	*True*: Ignore low level SAP lock/enqueue checking
Commit Work Wait	*True*: Force DB commit work and wait on queue processing
Job Class	Valid entries are A/B/C. The default is *C*
Job Group	SAP backgroup based on SM61
User ID	The SAP user ID to run under. By default, it is appears with user name whoever initiates the start
RFC Group	RFC log-on group as defined using RZ12
RFC Destination	Blank value: Starts the server using a local RFC destination, otherwise uses the specified destination
Snatched Queue Entries In Parallel	*True*: All the snatched queue entries that are executed parallely
Add-in Configuration	
Retain Inactive Sessions For	How long (days) in-active sessions can remain in the system before the housekeeping deletes them automatically
Cancel Idle Sessions After	Automatically cancels active sessions that are not used for a displayed number of hours

Field	Description
Retain Inactive Kept Sessions	How long (days) in-active stored sessions can remain in the system before the housekeeping deletes them automatically
Retain Completed ISF Runs For	How long (days) successfully completed store and forward requests are kept for reference, before the housekeeping deletes them automatically
Retain Classification Data For	How long (days) the system should keep classification historical data
Retain GPS Container Data For	How long (days) the system should keep GPS container historical data
Retain GPS Host Data For	How long (days) the system should keep GPS host historical data
Make The Workbench Display Only	*True*: You may only import changes using the transport facility along with change management procedures
Default LDB/DOB Transfer Buffer	Default transfer buffer size in bytes. If an LDB or a DOB download or refresh synchronization request is received in SAP with no transfer buffer, this value is automatically defaulted
Don't Do Runtime ABAP Syntax Checks	*True*: Suppresses all syntax checking. It is recommended for production environments
Suppress SAP Screen Optimization	*True*: Effectively turns off screen optimization

Field	Description
Force Session Manager Commit & Wait	*True*: Forces the SAP session manager (host screen processing) to wait for all its commit works to confirm before continuing with processing
Enable XAI Cross System Support	*True*: Enable to transparently invoke Kony for SAP functionality across multiple back end SAP systems - for example, ERP, CRM, and BW
White List Only Provisioning	*True*: Enforces that all Application Servers and Gateways must have a provisioning white list entry.
Extended DOB/LDB Upload Check	*True*: Perform extended DOB/LDB upload checking.
Activate SAP Session Key Checking	*True*: Activate SAP Session Key Checking.
In-Active Time Out	How many minutes the system can stay inactive. *0*: disable
Update SkyMobile SAP user	*True*: Update Kony for SAP SAP user.
Dialog Only	Only update SAP dialog (logged on) user IDs. Ignore for CPIC and service user type connections.
Directory	If data is written to host files during Kony for SAP processing, host directory and delimiter displayed here are used for external file processing.
Delimiter	
Authorization	Authorization exit may be installed.

Field	Description
Switchboard Manager	Legacy switchboard ABAP exit.
Store and Forward	Legacy store and forward ABAP exit.
Retain DOB/LDB Transfer Trace For	How long (days) the system should keep DOB/LDB transfer trace historical data.
Retain App/Ver Refresh Trace For	How long (days) the system should keep application version. Refreshes historical data.
Retain RFC Call Trace For	How long (days) the system keeps RFC call trace historical data.
Retain Gateway Service Trace	How long (days) the system keeps gateway service trace historical data.
Collect SAP RFC Call Statistics	*True*: Collect all SAP host RFC call statistics. ***Note:*** This can generate high volumes.
Host UTC Offset	UTC time zone for the MEAP Server (Time offset from GMT).
Update Automatically	*True*: Kony for SAP configures automatically.
Decimal Symbol	Numeric formatting decimal symbol (SAP) (for example: . or ,).

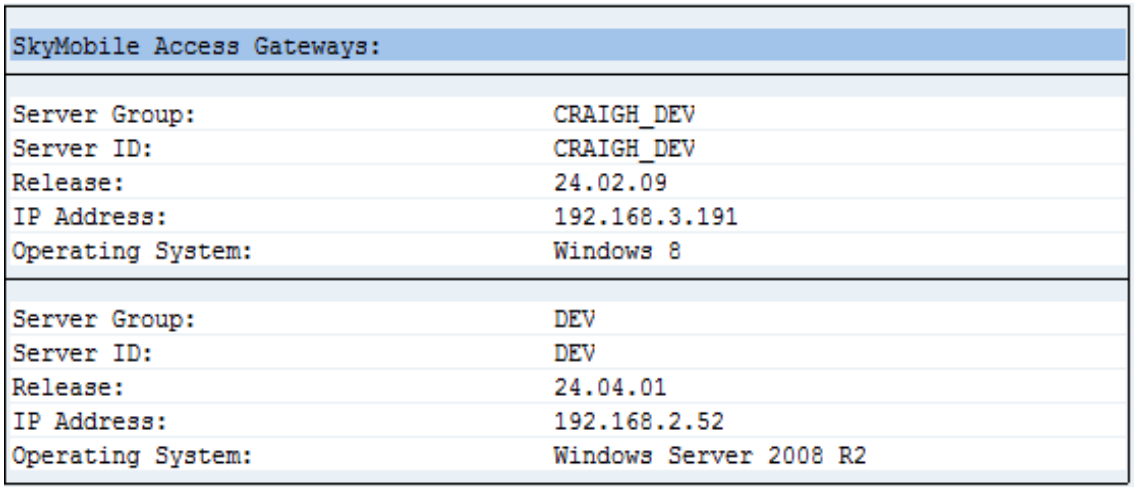
Field	Description
Grouping Symbol	Numeric formatting grouping symbol (SAP) (for example: . or ,).
URL	Help URL.
Local	*True*: Alternative help URL, that is, non-Sky standard.
Application Server Configuration	
Retain Commands For (Any Status)	For how long (days) the system keeps heartbeat (any status) commands.
Keep Completed Commands For (Hours)	For how long (hours) the system keeps heartbeat completed commands.
Keep Pending Commands For (Hours)	For how long (hours) the system keeps heartbeat pending commands.
Keep In-progress Commands For (Hours)	For how long (hours) the system keeps heartbeat in-progress commands.
Directory	Heartbeat trace directory.
Delimiter	Delimiter used for heartbeat trace file management.
Activate Global Trace	*True*: Activates the global heartbeat trace.

Field	Description
With Data Trace	*True*: Activates the global heartbeat trace with data trace.
Unique Registration By Type/Id Only	*True*: Registration and reference by type and ID only.
Keep In-Active Registrations For	For how long (days) the system keeps inactive registrations.
Rationalise Cross Reference	Rationalize cross-reference.
Keep In-Active References For	For how long (days) the system keeps inactive references.
Suppress Heartbeat Server Enqueueing	*True*: Supresses the heartbeat server enqueue process.
System Log Retention	For how long (days) the system keeps system logs.
Server Log Retention	For how long (days) the system keeps server logs
Server Group/ID Exit	Server Group/ID Exit program.

Field	Description
Disable Suppress Command Interface	*True*: Suppress TSX command interface.
Destination	Destination.

Access Gateways

The Access Gateways contain the list of all the active Access Gateways connected to the SAP environment. For each connected gateway, the following details appear.



```
SkyMobile Access Gateways:

Server Group:                   CRAIGH_DEV
Server ID:                      CRAIGH_DEV
Release:                        24.02.09
IP Address:                     192.168.3.191
Operating System:               Windows 8

Server Group:                   DEV
Server ID:                      DEV
Release:                        24.04.01
IP Address:                     192.168.2.52
Operating System:               Windows Server 2008 R2
```

Field	Description
Server Group	The Server Group of the Gateway Server.
Server ID	The Server ID of the Gateway Server.
Release	Kony for SAP Release.
IP Address	IP address of Gateway Server.

Field	Description
Operating System	Operating system of Gateway Server.

Server Release Summary

The Server Release Summary contains details of the consolidated summary list of the following table fields for Server and Gateway Server.

SkyMobile Server Release Summary:

Client	Release	Operating System	Last Used	Count
APC	v24.00.00	ANDROID 2.3.6	03.06.2014	1
BPC	24.04.00	BLACKBERRY OS 6.0.0.141	04.06.2014	1
IPC	24.03.00	IPHONE OS 6.1.3	04.06.2014	1
SSM		WINDOWS NT 6.3	06.06.2014	1
SSX		QNX 8.0.0	05.06.2014	1
WPC	24.02.21	WINDOWS 7/SERVER 2008 R2 6.1	30.06.2014	1
WPC	24.03.16	POCKETPC 5.2	05.06.2014	1
WPC	24.04.01	WINDOWS 7/SERVER 2008 R2 6.1	01.07.2014	7
WPC	24.04.02	WINDOWS 8 6.2	03.07.2014	1
GWY	24.04.01	Windows Server 2008 R2 6.1	03.07.2014	1

Field	Description
Client	Client type (appears as GWY for gateway server details)
Release	The release number of Kony for SAP
Operating System	Concatenation of operating system and operating system version
Last Used	Last connected date of server
Count	Total number of server connections grouped by the Client, Release, Operating System, and Operating System Version

Heartbeat Connection and Session Summary

The **Heartbeat Connection and Session Summary** section recaps the size and use of the system. The section displays the consolidated count summary of the last three months.

Heartbeat Connection and Session Summary:

Year	Month	Unique Devices	Unique Users	Total Sessions
2014	04	9	2	3.959
2014	05	25	5	813
2014	06	11	1	296
2014	07	2	2	5

Field	Description
Year	Year for which Heartbeat Connection and Session Summary is displayed, in (YYYY format)
Month	Month for which Heartbeat Connection and Session Summary is displayed in (MM format)
Unique Devices	Unique number of devices connected
Unique Users	Unique number of users connected
Total Sessions	Unique number of sessions connected

System Exception Summary

The **System Exception Summary** appears only when you select the **System Exception Summary** check box on the selection screen. The system goes through all the logs for the last seven days and provides a summary of all errors by date for each category. This action informs a user about the overall health of the system and highlights areas where major problems occur. The following is a list of logs that are included:

- SkyMobile System
- SkyMobile Server
- SkyMobile Access Gateways
- Provisioning log
- Data object trace (errors)
- LDB trace (errors)
- ECS Interfaces
- Session Manager
- Short Dumps (associated with SkyMobile)
- SAP System Log Messages (associated with SkyMobile)

The following fields appear in the section output.

System Exception Summary:

Date	Component	Total
26.06.2014	SkyMobile System	299
	SkyMobile Access Gateways	286
	LDB's	21
27.06.2014	SkyMobile System	369
	SkyMobile Access Gateways	344
	LDB's	27
	ECS Interfaces	1
28.06.2014	SkyMobile System	1
	ECS Interfaces	1
29.06.2014	ECS Interfaces	1
30.06.2014	SkyMobile System	91
	SkyMobile Access Gateways	75
	LDB's	15
	ECS Interfaces	1
01.07.2014	SkyMobile System	24
	SkyMobile Servers	12
	Data Objects	15
	LDB's	5
	ECS Interfaces	1
02.07.2014	ECS Interfaces	1
	Short Dumps	1
	SAP System Log Messages	1

Field	Description
Date	Log date (DD.MM.YYYY)
Component	Log Area listed above (for example, SkyMobile System, SkyMobile Server, and so on)
Total	Count of errors for the specified period

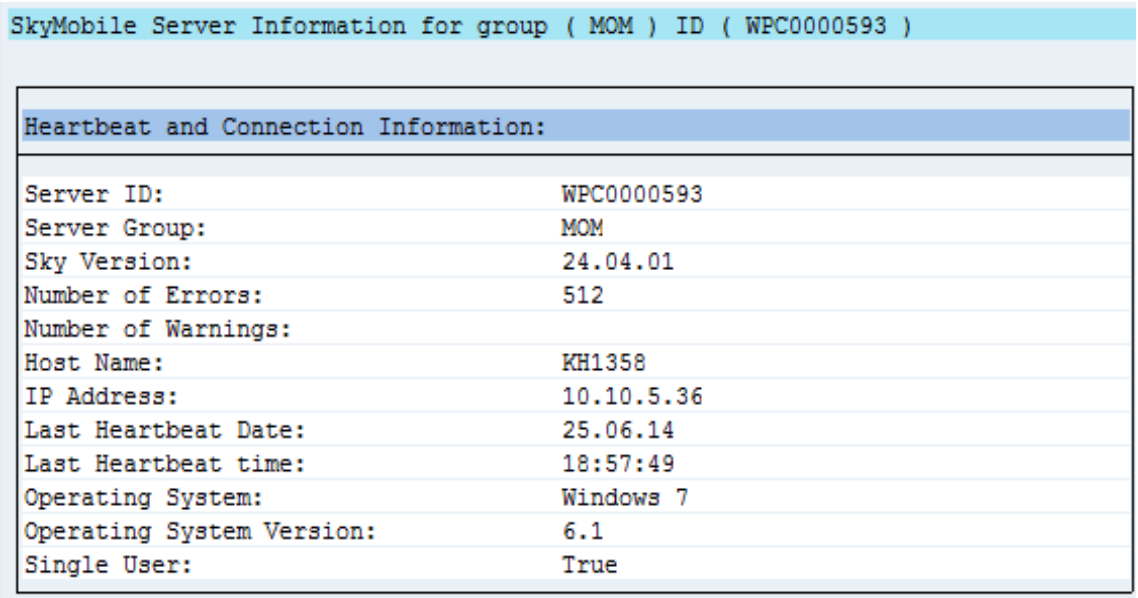
Heartbeat and Connection Information

The **Heartbeat and Connection Information** section appears only when you select the **Server Details** check box and the **Heartbeat and Connection Details** check box and enter a valid server ID on the selection screen.

The section lists all the heartbeat registration details and the associated connection details of the server ID that you enter as shown below. The **Connections** subsection is repeated for every available client connection.

- Registration Details
- Connections

The following fields appear in the section output.



SkyMobile Server Information for group (MOM) ID (WPC0000593)

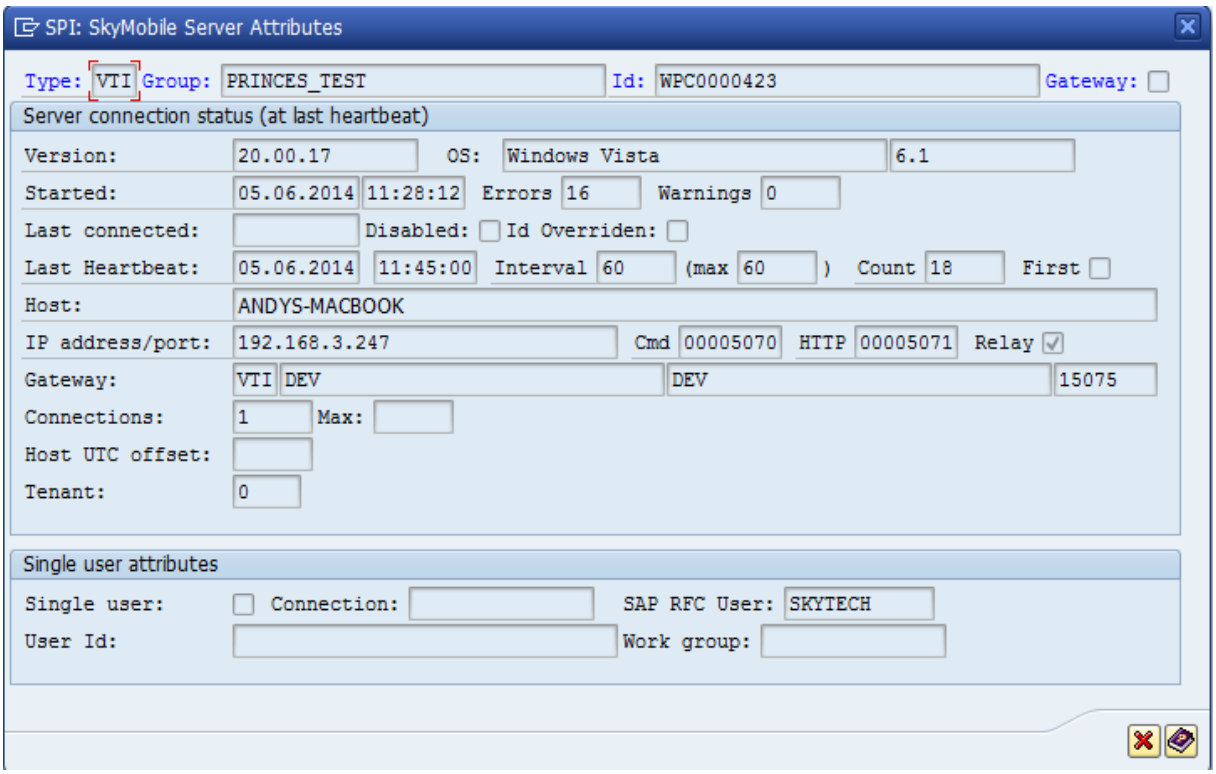
Heartbeat and Connection Information:

Server ID:	WPC0000593
Server Group:	MOM
Sky Version:	24.04.01
Number of Errors:	512
Number of Warnings:	
Host Name:	KH1358
IP Address:	10.10.5.36
Last Heartbeat Date:	25.06.14
Last Heartbeat time:	18:57:49
Operating System:	Windows 7
Operating System Version:	6.1
Single User:	True

Field	Description
Registration Details	
Server ID	Server ID connected

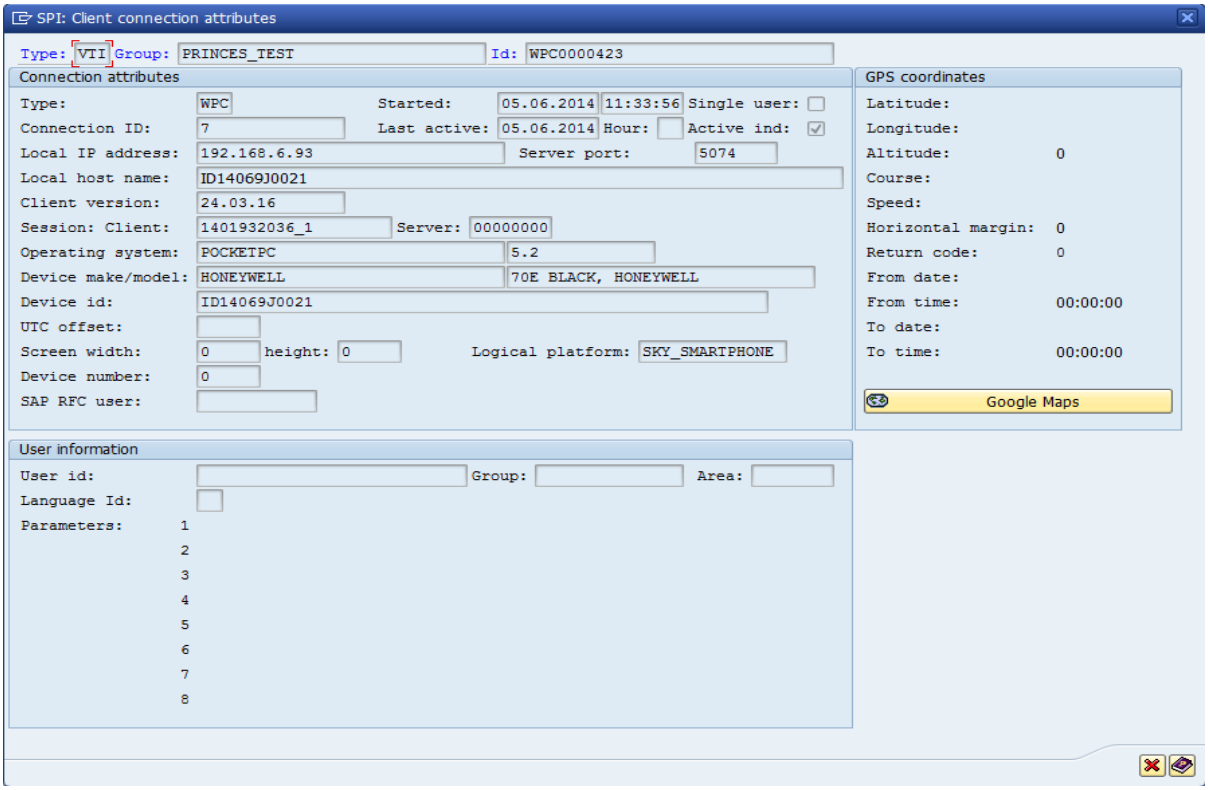
Field	Description
Server Group	Server group
Sky Version	Installed Sky release version
Number of Errors	Errors count
Number of Warnings	Warnings count
Host Name	Server host name
IP Address	Server IP address
Last Heartbeat Date	Last heartbeat date (DD.MM.YY)
Last Heartbeat time	Last heartbeat time (HH:MM:SS)
Operating System	Operating system of server
Operating System Version	Operating system version
Single User	*False*: Multiple users
Connections (This section repeats for every established client connection)	
Client Type	Type of the client (for example, WPC, and so on)
Device Make	Make of the connected device
Device Model	Model of the connected device
Device ID	The unique ID of the connected device

If you double-click any line of the registrations block, a new dialog appears with all registration technical attributes. You can use the details to diagnose the heartbeat details as shown below:



- Click the **Online help function** button on the **SPI: SkyMobile Server Attributes** dialog to open the online documentation page.
- Click the **Cancel** button to navigate back to Support report output display.

If you double-click any line of the connections block, a new dialog appears with client connection technical attributes details as shown below:



- Click the **Online help function** button on the **Client Connection Attributes** dialog to open the online documentation page.
- Click the **Google Maps** button on the **Client Connection Attributes** dialog.
  The Google map appears with the details in the **GPS coordinates** section.
- Click the **Cancel** button to navigate back to Support report output display.

Provisioning Details

The **Provisioning Details** section appears only when you select the **Server Details** check box and **Provisioning Details** check box and enter a valid server ID on the selection screen. The section lists all the server provisioning attributes and the associated application summary details.

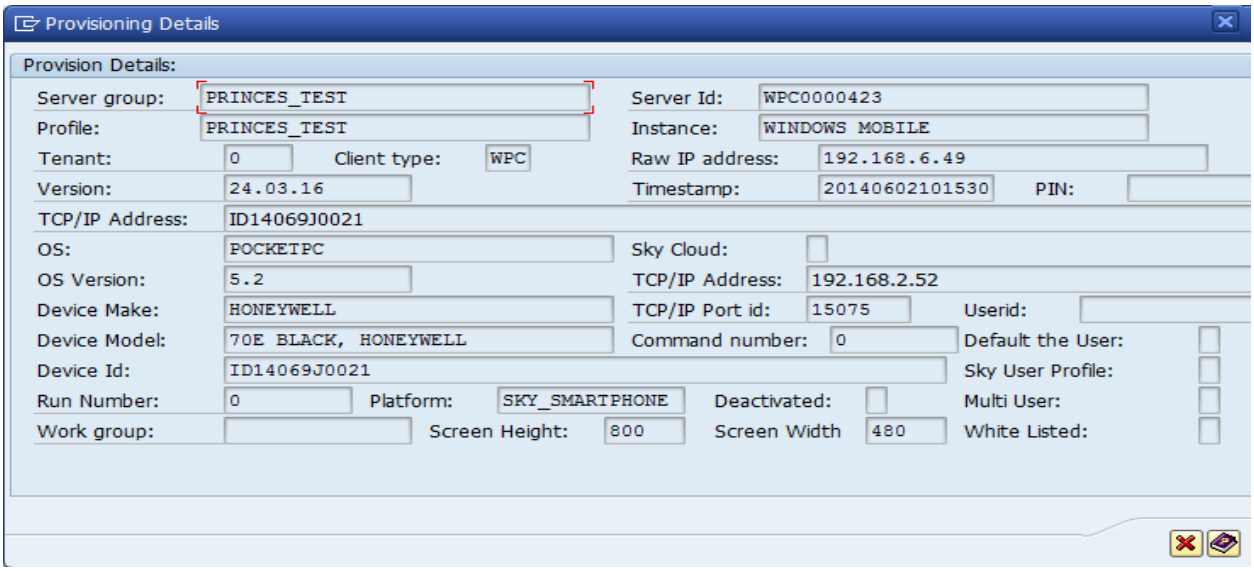
The following fields appear in the section output.

Provisioning Details:	
Server ID	WPC0000593
Server Group:	MOM
Profile ID	MOM
Client Version:	24.04.01
Tenant:	00000
IP Address:	10.10.5.27
Gateway Address:	192.168.2.52
Application Summary:	
Application Definition:	905 001
Database tables:	MOM_CUSTOMER
	MOM_EVENT
	MOM_INSTANCE
	MOM_ORDER_HEADER
	MOM_ORDER_ITEM
	MOM_PRODUCTS
	MOM_TEST
	MOM_TEST_LOG
	MOM_TEST_RUN
	MOM_THREAD
	MOM_USER
Data Objects:	MOM_MASTER_DATA
	MOM_TESTS
	MOM_TEST_RUNS
	MOM_TRANSACTIONS
Binary Objects:	No Binary Objects Provisioned

Field	Description
Provisioning Details	
Server ID	Server ID connected
Server Group	Server group
Profile ID	The server profile assigned to the last server provision
Client Version	Client Kony for SAP release
Tenant	The unique tenant associated with the last server provision

Field	Description
IP Address	Client server IP address
Gateway Address	Gateway server IP address
Application Summary: (Repeated for every application/version)	
Application Definition	Concatenation of application and version
Database tables	List of LDBs involved in application version
Data Objects	List of DOBs involved in application version
Binary Objects	List of binary objects involved in application version

If you double-click any line of the Provisioning Details block, a new dialog appears with provisioning technical attributes. Details are shown below.



- Click the **Online help function** button on the **Provisioning Details** dialog to open the online documentation page.

- Click the **Cancel** button on the **Provisioning Details** dialog to navigate back to the Support report output display.

Gateway Details

The **Gateway Details** section appears only when you select the **Server Details** check box and the **Gateway Details** check box and enter a valid server ID on the selection screen.

This section lists the details of the gateway that the server is connected to and also lists the errors of the gateway server since the **Errors/Warnings Since** date of the selection screen. No errors appear if the **Errors/Warnings Since** field is blank. You can use this section to correlate any server errors with errors that occur on the gateway - for example, timeouts or Identity issues and so on.

The following fields appear in the section output.

Gateway Details:

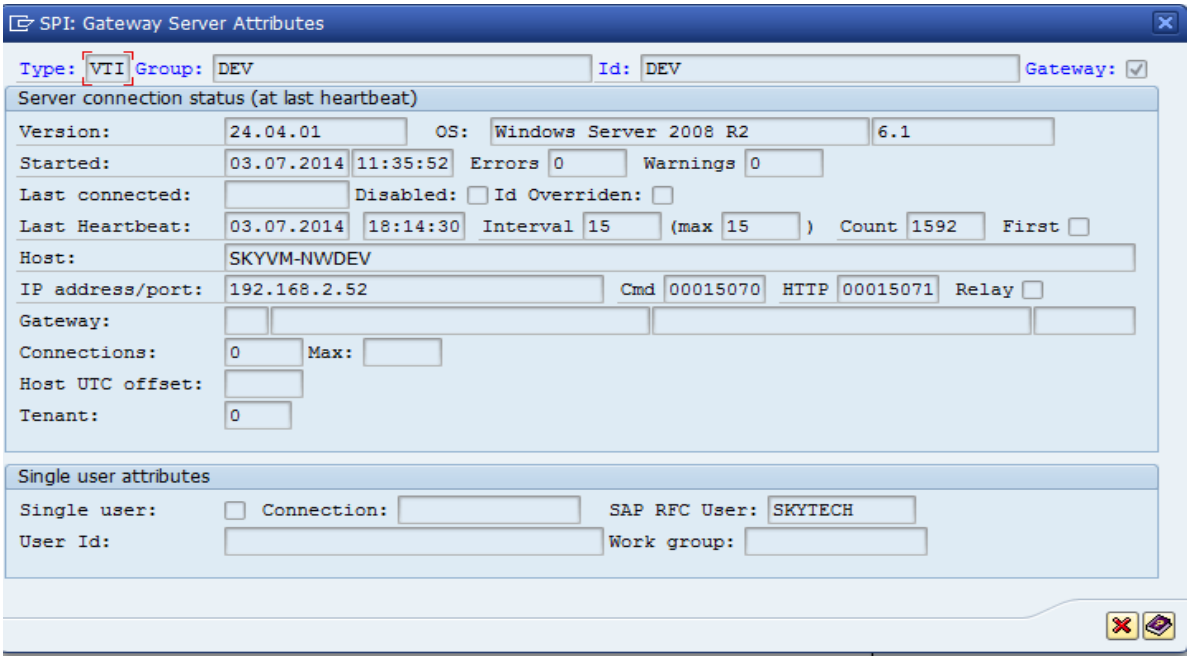
Server ID:	DEV
Server Group:	DEV
Sky Version:	24.04.01
Number of Errors:	
Number of Warnings:	
Host Name:	SKYVM-NWDEV
IP Address:	192.168.2.52
Last Heartbeat Date:	03.07.14
Last Heartbeat time:	20:02:05
Operating System:	Windows Server 2008 R2
Operating System Version:	6.1
Single User:	False

Date	Time	Error Message
20.06.14	14:46:20	Invalid request.
20.06.14	15:34:46	Invalid request.
20.06.14	15:46:44	An HTTP error response (400) was received from the portal.
20.06.14	16:06:59	An HTTP error response (400) was received from the portal.
24.06.14	15:58:23	Invalid request.
24.06.14	15:58:53	Invalid request.
24.06.14	16:04:08	Invalid request.
24.06.14	16:04:38	Invalid request.
24.06.14	16:06:08	Invalid request.
24.06.14	21:46:11	An error occurred whilst communicating with the portal.
24.06.14	21:46:11	An error occurred whilst communicating with the porta
25.06.14	13:09:02	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
25.06.14	13:09:02	RFC error message: 'BINARY_FILE_NOT_FOUND'.
25.06.14	13:12:02	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
25.06.14	13:12:02	RFC error message: 'BINARY_FILE_NOT_FOUND'.
25.06.14	13:15:02	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
25.06.14	13:15:02	RFC error message: 'BINARY_FILE_NOT_FOUND'.
25.06.14	13:18:02	Status RFC_EXCEPTION returned by RFC call of function /SKY/VTI_SELECT_BINARY_DATA on host.
25.06.14	13:18:02	RFC error message: 'BINARY_FILE_NOT_FOUND'.

*** Max of 100 entries listed ***

Field	Description
Gateway Details	
Server ID	The server ID of Gateway
Server Group	The server group of Gateway
Sky Version	The version number of the Kony for SAP
Number of Errors	The number of errors
Number of Warnings	The number of warnings
Host Name	The host name of the Gateway
IP Address	The server IP address of Gateway
Last Heartbeat Date	Date of the last heartbeat (DD.MM.YY)
Last Heartbeat time	The last heartbeat time (HH:MM:SS)
Operating System	The operating system of the Gateway Server
Operating System Version	Version of the operating system
Single User	*False*: Multiple user
The following fields appear for error logs:	
Date	Date (DD.MM.YY) on which the Gateway Server error is logged
Time	Time (HH:MM:SS) on which the Gateway Server is logged
Error Message	Error text message

If you double-click any line of the gateway details block, a new dialog appears with all gateway technical attributes.



- Click the **Online help function** button on the **SPI: Gateway Server Attributes** dialog to open the on-line documentation page.
- Click the **Cancel** button on the **SPI: Gateway Server Attributes** dialog to navigate back to the Support report output display.

Errors and Warnings

The **Errors and Warnings** section appears only when you select the **Server Details** check box and when the **Errors/warnings Since** date is not blank.

The section summarizes all the known errors and warnings for the server since the date specified in **Errors/Warnings Since**. There are many sources for this with different attributes, and so the list content may vary. The entries from the following sources of the server appear:

- SkyMobile System Log (maximum 100 entries)
- Server Log (maximum 100 entries)

- DOB trace (maximum 100 entries)
- LDB trace (maximum 100 entries)

For each of the above sources, the following fields appear in the section output:

```
SkyMobile System Log (Max 100 Entries):

Date      Time      Type Error Message

01.07.14  14:41:28  E    3 log errors detected
01.07.14  14:42:07  E    Data Object REFRESH failed
01.07.14  14:42:20  E    Data Object REFRESH failed
01.07.14  14:42:22  E    Data Object REFRESH failed
01.07.14  14:42:38  E    9 log errors detected

Server Log (Max 100 Entries):

Date      Time      Type Error Message

01.07.14  14:42:38  E    Accompanying message: Host/local DOB workbench timest
                       amp is different
01.07.14  14:42:38  E    Failure during refresh of data object MOM_MASTER_DATA
                        for host interface SKYCLOUD.
01.07.14  14:42:38  E    Return code 025 received during refresh of data objec
                       t MOM_MASTER_DATA for host interface SKYCLOUD.
01.07.14  14:42:38  E    Accompanying message: Host/local DOB workbench timest
                       amp is different
01.07.14  14:42:38  E    Failure during refresh of data object MOM_MASTER_DATA
                        for host interface SKYCLOUD.
01.07.14  14:42:38  W    Host interface SKYCLOUD has been disabled.
```

Fields on DOB Logs and LDB Logs:

DOB Logs (Max 100 Entries):

Date	Time	Type	Error Message
01.07.14	14:26:41	E	Host/local DOB workbench timestamp is different
01.07.14	14:41:14	E	Host/local DOB workbench timestamp is different
01.07.14	14:42:07	E	Host/local DOB workbench timestamp is different
01.07.14	14:42:20	E	Host/local DOB workbench timestamp is different
01.07.14	14:42:22	E	Host/local DOB workbench timestamp is different

LDB Logs (Max 100 Entries):

Date	Time	Type	Error Message
01.07.14	14:32:21	E	Host/local LDB workbench timestamp is different
01.07.14	14:32:46	E	Host/local LDB workbench timestamp is different
01.07.14	14:33:21	E	Host/local LDB workbench timestamp is different
01.07.14	14:33:50	E	Host/local LDB workbench timestamp is different
01.07.14	14:34:38	E	Host/local LDB workbench timestamp is different

Field	Description
Date	Log date (DD.MM.YYYY)
Time	Log time (HH:MM:SS)
Type	Message type (for example; E, W, and so on)
Error Message	Message text

Server Configuration Details

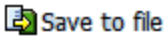
The **Server Configuration Details** section appears only when you select the **Server Details** check box and the **Server Configuration Details** check box and enter a valid server ID on selection screen.

The **Server Configuration Details** section lists the configuration of the server entered on selection screen as below:

```
Server Configuration Details:

CLIENT.GENERAL.ALLOWMULTIPLEINSTANCES = false
CLIENT.GENERAL.DISABLELOGOFFMESSAGE = false
CLIENT.GENERAL.DISABLESOUND = false
CLIENT.GENERAL.UNICODEENABLED = true
CLIENT.GENERAL.KEYBOARDONLY = false
CLIENT.GENERAL.HOSTSTATUSMESSAGES = All
CLIENT.LOG.LOGACTIVE = true
CLIENT.LOG.LOGFILEDIRECTORY = ./log
CLIENT.LOG.SHOWMILLISECONDS = false
CLIENT.LOG.PERFORMANCETRACKING = false
CLIENT.LOG.STRATEGY = Daily
CLIENT.LOG.MAXFILES = 7
CLIENT.PRESENTATION.FIXEDSIZELINEHEIGHT = 25
CLIENT.PRESENTATION.FONTNAME = Arial
CLIENT.PRESENTATION.FONTSIZE = 9
CLIENT.PRESENTATION.CHARPADDINGHEIGHT = 2
CLIENT.PRESENTATION.CHARPADDINGWIDTH = 1
CLIENT.PRESENTATION.DISABLEABOUTMENU = false
CLIENT.PRESENTATION.DISABLEMINIMIZEMENU = true
CLIENT.PRESENTATION.DISABLESELFTESTMENU = false
CLIENT.PRESENTATION.DISABLESHUTDOWNMENU = false
CLIENT.PRESENTATION.HIGHRESOLUTIONGRAPHICS = true
CLIENT.PRESENTATION.MAXIMISEONSTARTUP = false
CLIENT.PRESENTATION.WINDOWHORIZONTALALIGNMENT = Centre
CLIENT.PRESENTATION.WINDOWVERTICALALIGNMENT = Centre
CLIENT.PRESENTATION.SHOWMENU = true
CLIENT.PRESENTATION.SHOWFUNCTIONKEYSINMENU = false
CLIENT.PRESENTATION.SHOWWINDOWCONTROLS = true
CLIENT.PRESENTATION.SHOWHORIZONTALSCROLLBAR = true
```

5.10.3.2 Application Toolbar Buttons

You can also convert the report output to a PDF file from the Support report output display screen, export to the local system, and send mail to several recipients with a subject and body text. Application toolbar buttons and their uses on the support report output display are listed below.

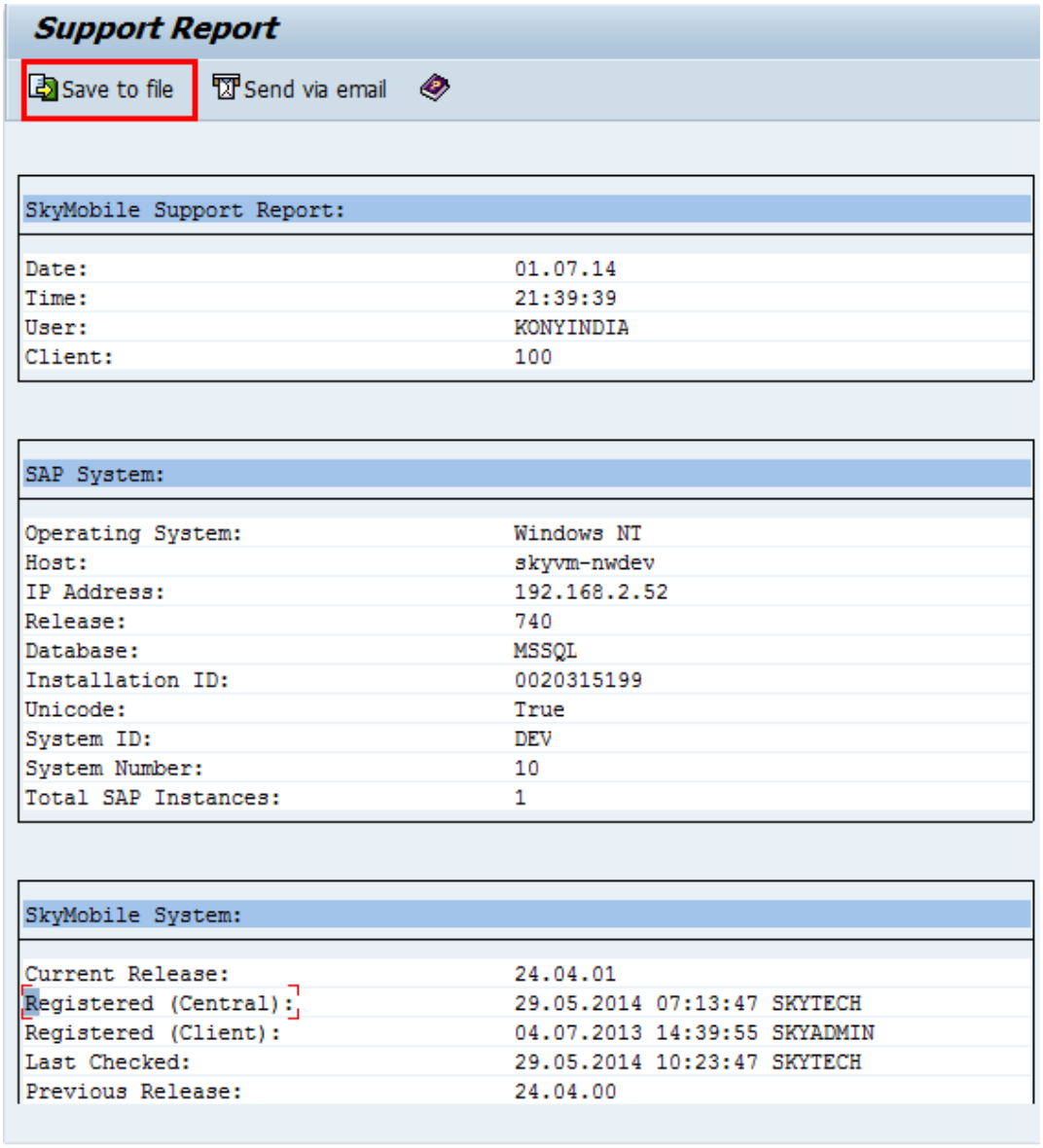
Button	Description
 Save to file	Convert the support report output to a PDF file and save to local system

Button	Description
Send via email	Convert the support report output to a PDF file and send mail to several recipients
	To display online help for the **/SKY/YSUPP** transaction

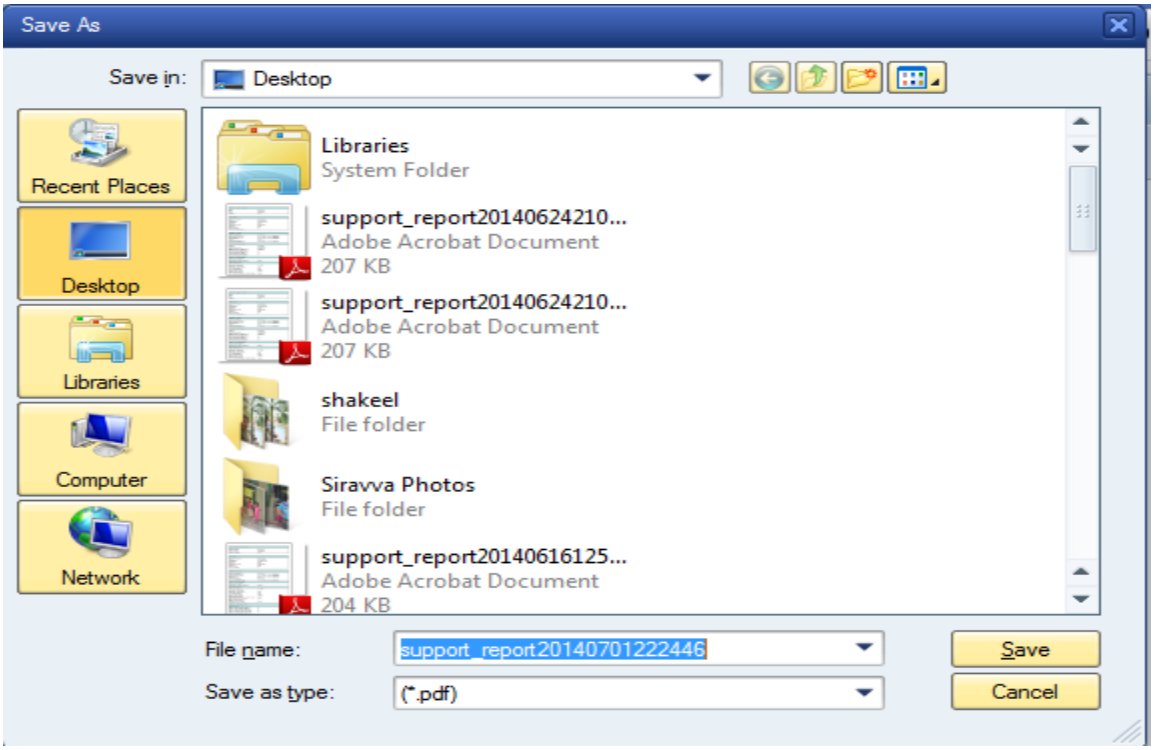
5.10.3.3 Generate PDF File

You can generate a PDF file from the output display by clicking the **Save to File** button from the application toolbar.

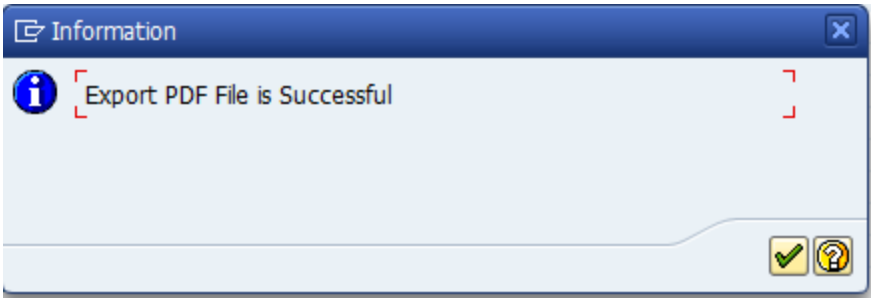
Click the **Save to file** button as shown below:



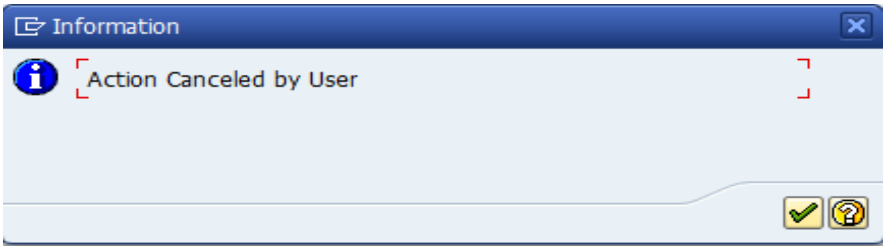
The **Save As** dialog appears by defaulting the file name as **support_reportYYYYDDMMHHMMSS**.



Once you save the file, a PDF file is generated. The following message appears.

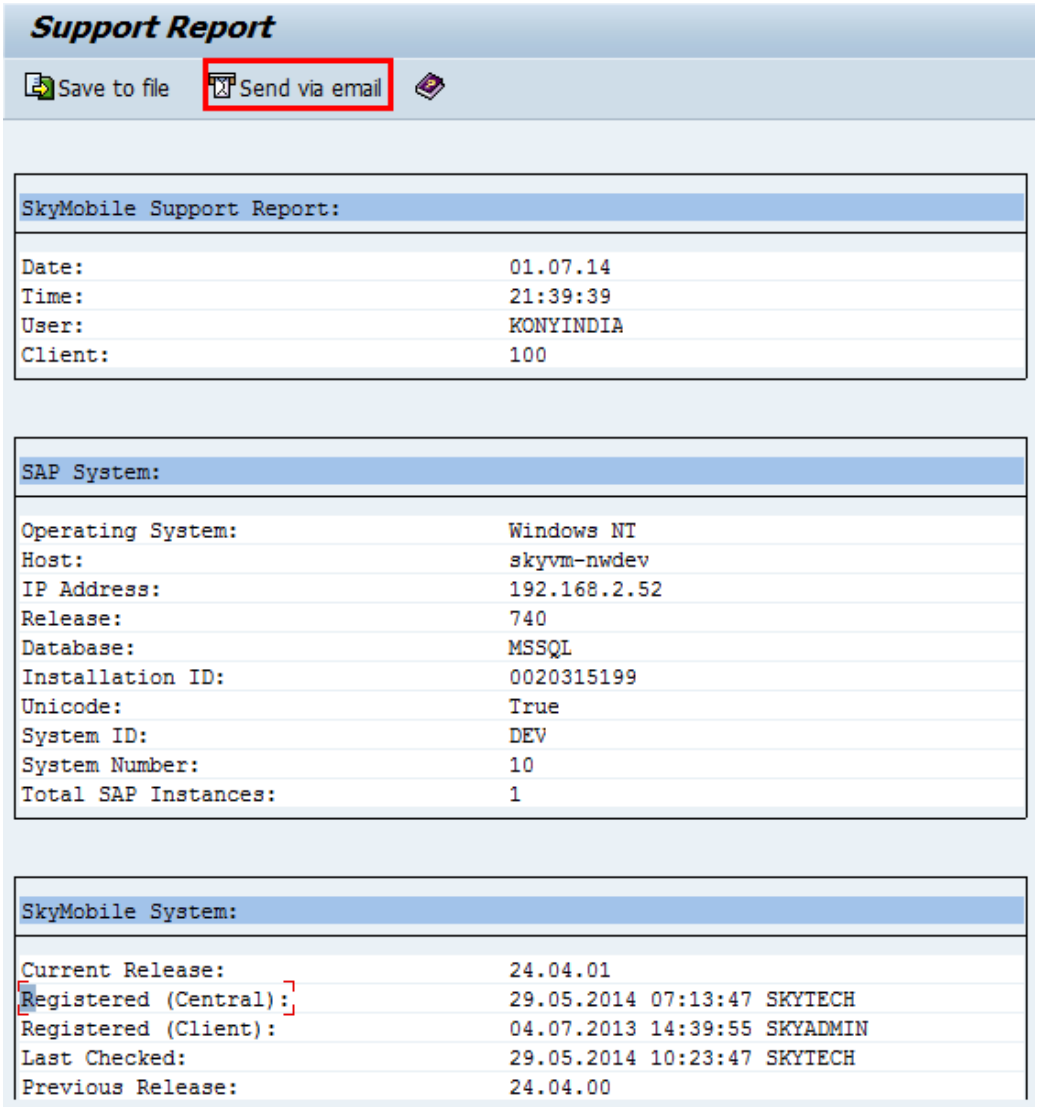


To cancel the PDF file generation, click the **Cancel** button. The following message appears.

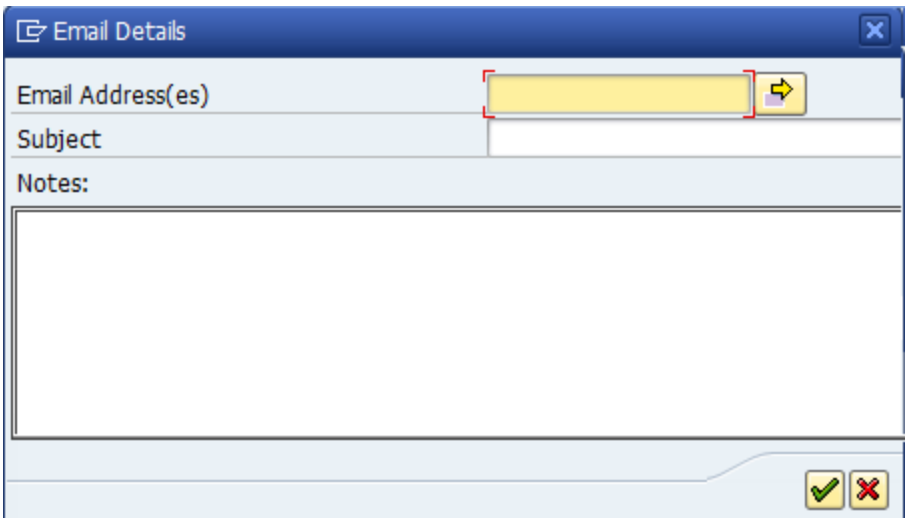


5.10.3.4 Sending an Email

You can send an email to several IDs by clicking the **Send via email** button on the application toolbar of the **Support Report** as shown below:



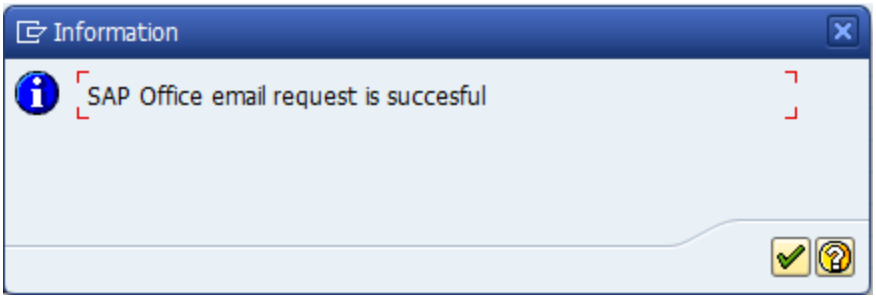
The **Email Details** dialog appears as shown below:



Field	Description
Email Address (es)	Recipient email ID. You can enter multiple email IDs by clicking the right arrow button.
Subject	Subject of the email.
Notes	Mail content of the email.

If you press **Enter** by filling the mail information on the **Email Details** dialog, the following occurs:

- A PDF file is generated for output display content.
- An email is sent with the subject, mail content entered on screen, and PDF file as an attachment to the list of recipients entered.
- The following message appears:



If you click the **Cancel** button, the email is not sent.

5.10.4 Users and Sessions Report

5.10.4.1 About

You may get information about the users and devices that are logged into the system for a specified period of time through the Users and Sessions report. You get collated data on monthly basis.

5.10.4.2 Working with Users and Sessions Report

Generating a Report of Users and Sessions

Displaying Reports

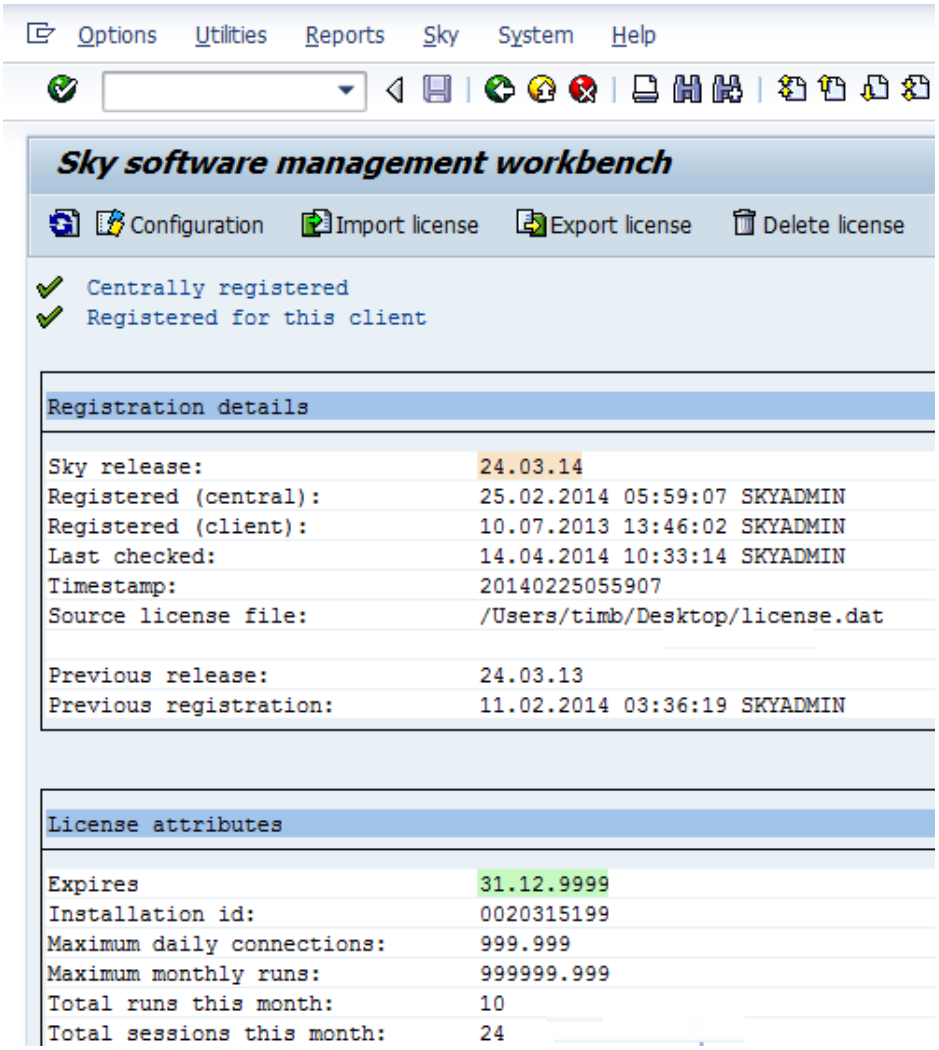
- Summary View
- Detail View

5.10.4.3 Generating a Report of Users and Sessions

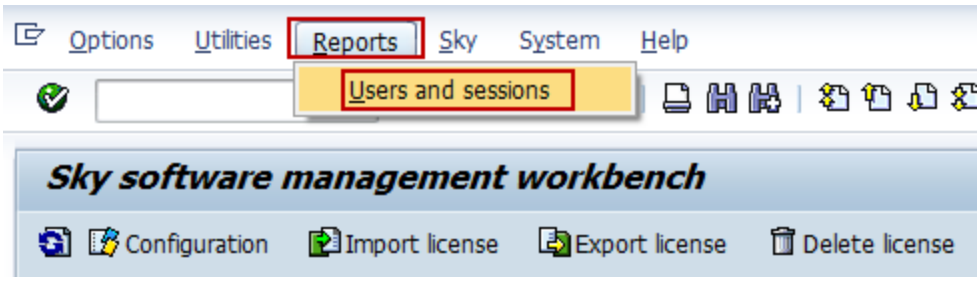
To generate a report of users and sessions, follow these steps:

1. Execute the transaction,**YECI** in SAP system.

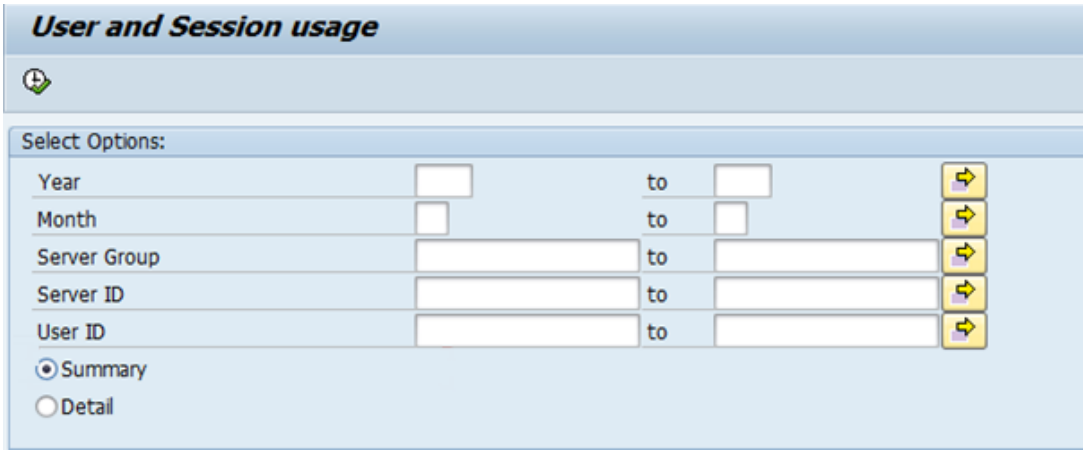
   The **Sky Software Management Workbench** window appears.



2. Go to **Reports** on the Application Workbench > **Users and sessions**.



The **User and Session Usage** dialog appears with various filters available for the report as below:



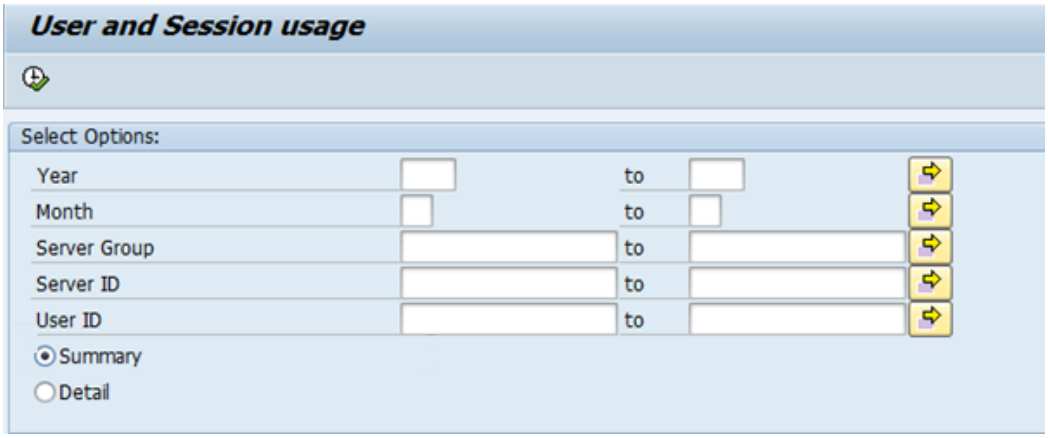
> ***Note:*** By default, the data for the last six months appears in the report.

3. You can display the report in two different ways.

5.10.4.4 Displaying Reports

To display the types of reports, follow these steps:

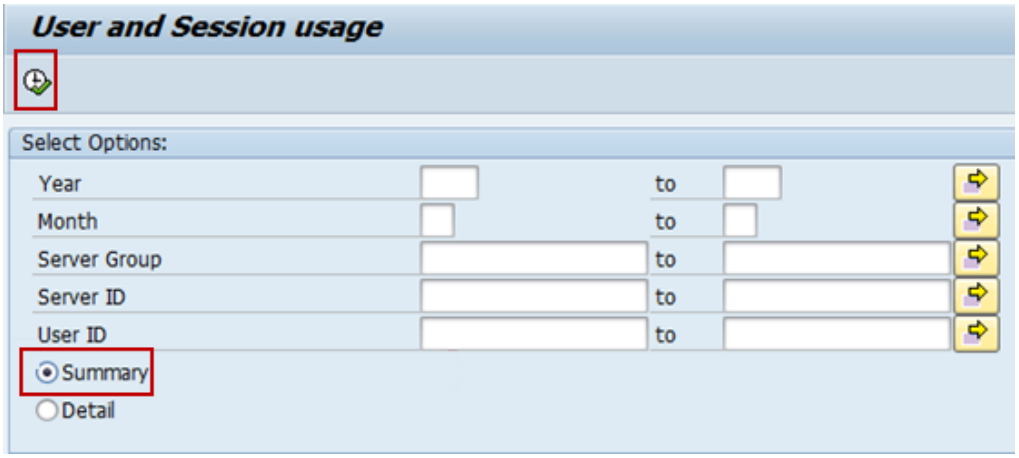
1. Log on to the SAP system and generate a report of users and sessions.
2. While generating the report, you may choose to generate in the following modes of views
   a. Summary
   b. Detail



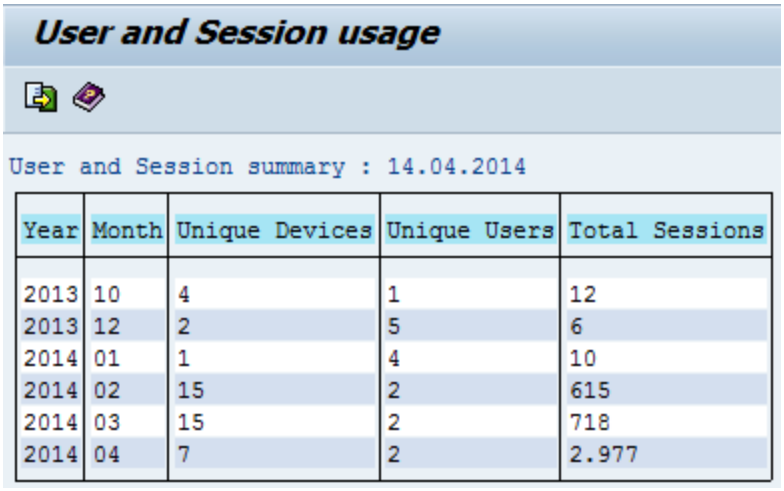
Summary View

To display the report in Summary view, follow these steps:

1. In the **User and Session usage** window, select the **Summary** option and click **Execute**.



The **User and Session Summary** view appears with the number of devices and users along with the total number of sessions held in each month that you enter in the **Select Options** area.

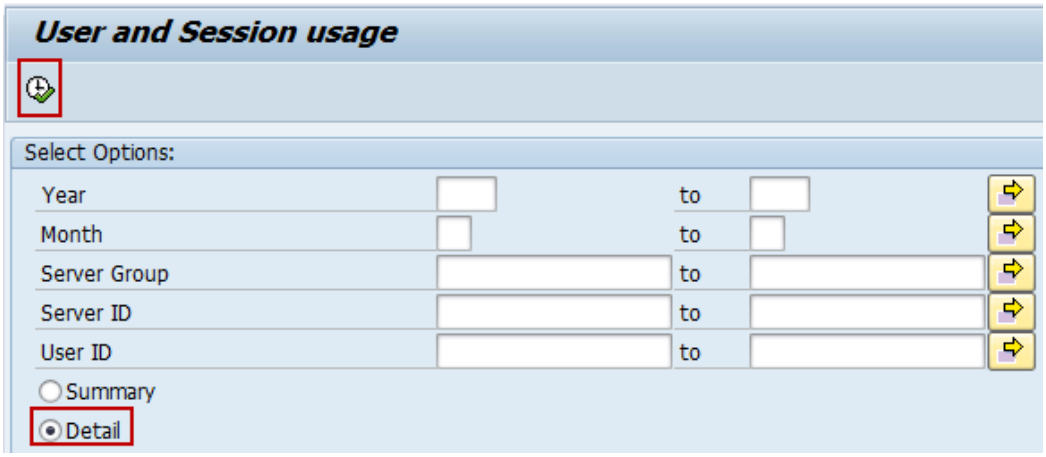


Year	Month	Unique Devices	Unique Users	Total Sessions
2013	10	4	1	12
2013	12	2	5	6
2014	01	1	4	10
2014	02	15	2	615
2014	03	15	2	718
2014	04	7	2	2.977

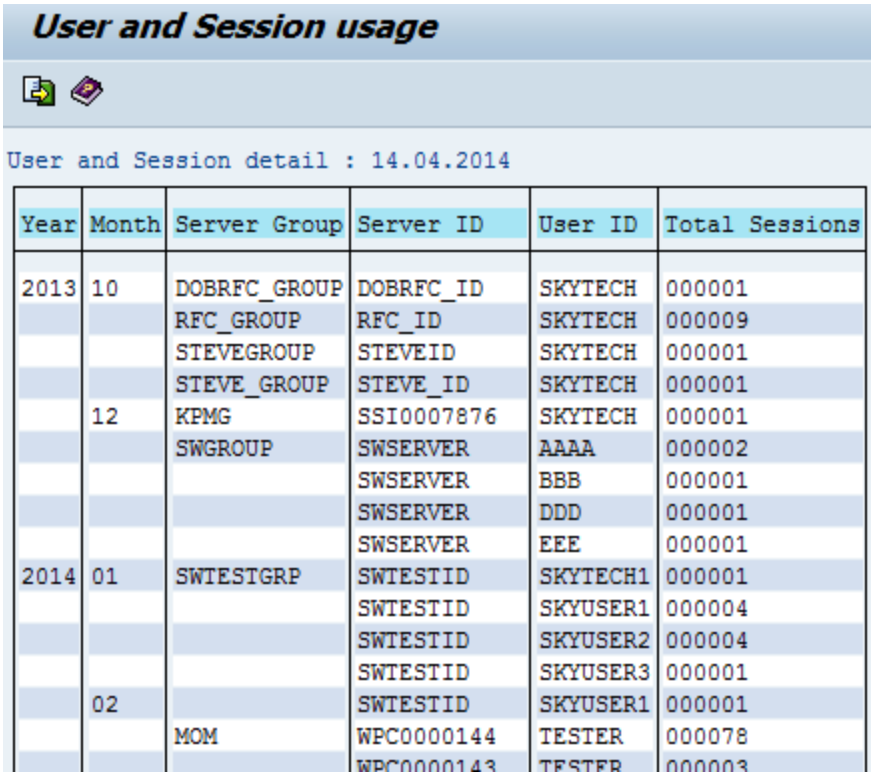
Detail View

To display the report in Detail view, follow these steps:

1. In the **User and Session Usage** dialog, select the **Detail** option and click **Execute**.



The **User and Session Usage** view appears with the details about the individual Server Groups, Server IDs and User IDs along with the number of sessions.



Year	Month	Server Group	Server ID	User ID	Total Sessions
2013	10	DOBRFC_GROUP	DOBRFC_ID	SKYTECH	000001
		RFC_GROUP	RFC_ID	SKYTECH	000009
		STEVEGROUP	STEVEID	SKYTECH	000001
		STEVE_GROUP	STEVE_ID	SKYTECH	000001
	12	KPMG	SSI0007876	SKYTECH	000001
		SWGROUP	SWSERVER	AAAA	000002
			SWSERVER	BBB	000001
			SWSERVER	DDD	000001
			SWSERVER	EEE	000001
2014	01	SWTESTGRP	SWTESTID	SKYTECH1	000001
			SWTESTID	SKYUSER1	000004
			SWTESTID	SKYUSER2	000004
			SWTESTID	SKYUSER3	000001
	02		SWTESTID	SKYUSER1	000001
		MOM	WPC0000144	TESTER	000078
			WPC0000143	TESTER	000003

5.10.5 Performance Tuning and Optimization

This section describes various mechanisms that you may use to diagnose and tune the system that is optimize performance. Kony for SAP recommends that you read the key sections shown opposite.

5.10.5.1 Application Performance Trace

Many of the trace utilities discussed here overlap with Diagnostics and tracing that is described in more detail. The most important facility to identify initial areas of concern and bottlenecks is the Client/Server performance trace and SAP performance trace that lists all the processing performed and the time taken. Using the performance trace, all Kony for SAP internal functions, DOB/LDB calls and custom program / function module calls are itemized with timings. From this list, you should be able to determine excessive:

- Processing times that need tuning (often custom code and database calls)
- Call frequencies that you need to address, for example, configuration timings

It is Kony for SAP experience that most performance problems arise from in-efficient or large volume database processing, and so initial performance tuning should focus on data extracts, transfers and selections, that is, minimize the frequency and amounts of data processed on both the SAP host on device. The LDB and Data Object traces are also useful to ascertain how often data is selected and the volumes are retrieved.

> ***Note:*** Kony for SAP recommends that you use the performance trace in all unit and system testing scenarios, prior to production, so that you can quickly identify and rectify any potential problems. In cases where production volumes are a factor, you may invoke the trace at any time for specific components without impacting the system.

5.10.5.2 Key Topics

High Volume Performance Strategies
RFC Call Recommendations
Database Call Recommendations
Access Gateway Statistics

5.10.5.3 Other Considerations

The following table lists some other performance related considerations.

Topic	Description
Screen optimization	To help minimize network traffic and processing overhead, the Container and SAP session mangers optimize the screen data transferred for real-time and post local SAP requests, that is, only the required screen fields (containing values) are transferred between them. This screen optimization is on by default, but you can turn off either globally in SAP using the YVTI system configuration or the MaxOptimiseLevel option in the host connection section in the Container configuration (`skymobile.cfg`). Before optimization can take place, a handshake is performed to determine what either side is capable of and then the lowest capability is agreed. In this way, processing is backward compatible with older components of the software.
Turning off SAP syntax checking	By default, the SAP session manager performs syntax checks on all custom ABAP programs and function modules. The results are cached, but even so, you can achieve a performance improvement by switching this off in a stable production system. You do this through the systems management console (YVTI) configuration.

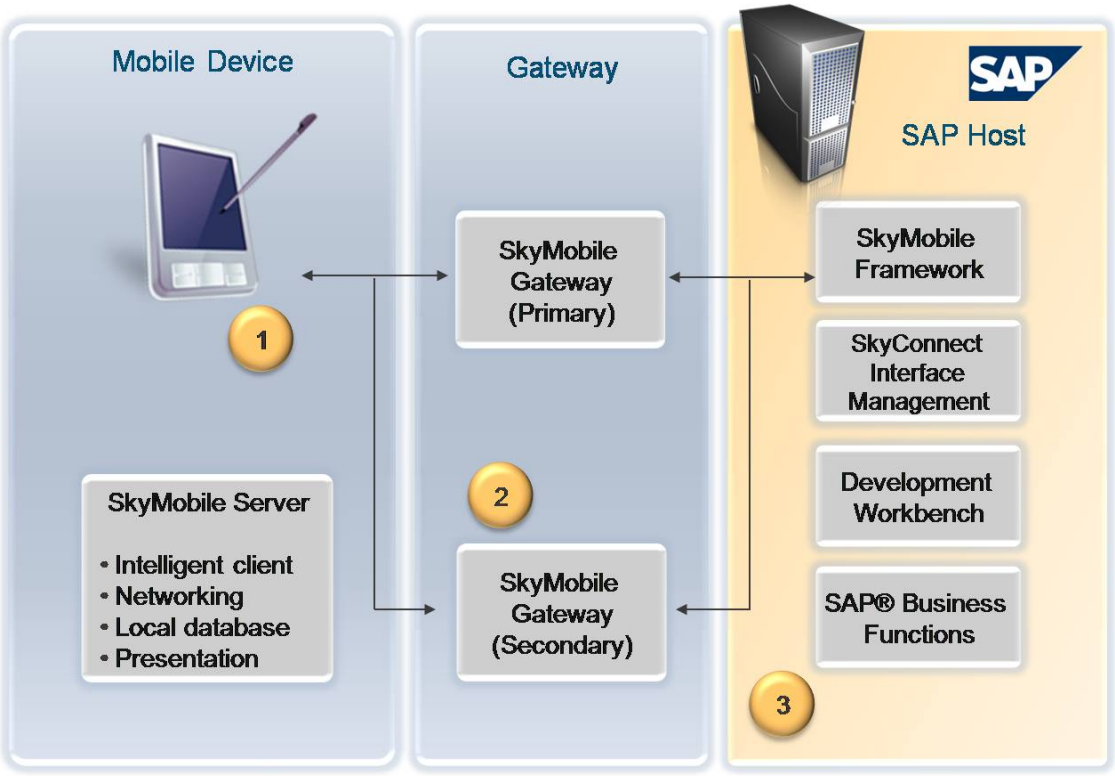
5.10.5.4 High Volume Performance Strategies

The purpose of this section is to outline how you can configure (scale) Kony for SAP framework and SAP can to handle high volume connections and transactions in-order to limit the impact of the mobile application on the back end SAP system. Kony for SAP framework is developed specifically for SAP and thus makes full use of the excellent SAP Client / Server architecture. While some detractors claim that Kony 'inside SAP' architecture can affect the performance of the Customers SAP system, quite the opposite is true. Because Kony for SAP software has control 'inside SAP', it can schedule, manage and monitor SAP resources and integrate much more effectively than external middle-ware platforms. You must realize that while Kony for SAP has many built-in features to help control and manage a large volume of mobile devices; the actual SAP resources required for the extraction and posting of SAP data and transactions can vary considerably from site to site. Thus, there is no magic formula to take all the different SAP configurations and capacities into account. It is therefore vital to have a thorough understanding of what SAP data extractions and transaction postings are required by the mobile application, the average timings of each and the current capacity of the SAP system. The objectives of this section are to understand:

The Typical Kony for SAP Architecture

In the following diagram, there are three discrete areas to the framework:

1. Secure Container running on the remote mobile device.
2. Access Gateway, acting as a central secure entry point into the LAN / WAN.
3. The SAP add-in component, providing centralized system and interface management services.



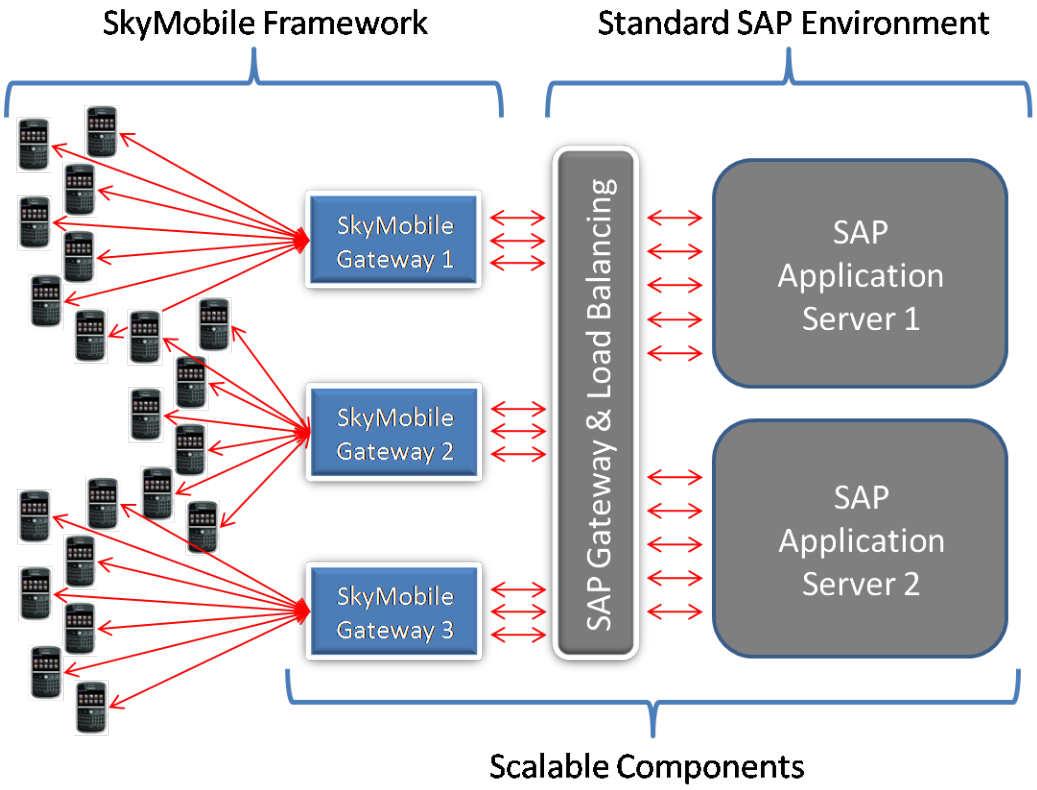
When considering a high volume scenario consisting of 1000s of remote devices that all need to synchronize data and post transactions on the back end SAP system; the immediate questions are:

- How you can scale the architecture to handle the mass volume of connections, data synchronization requests and transaction postings.
- What controls are in place to manage the impact to the overall SAP system?

Kony for SAP framework has a unique advantage in this respect in that it is specifically architected and optimized for the proven SAP client / server model. The 'inside SAP' approach enables far more control in terms of event driven processing, monitoring and the overall management of the internal SAP sub-systems.

SAP Connections

Each remote Secure Container that runs on the mobile device connects to a Gateway using TCP / IP Sockets or Http. You may use one or more Gateways for automatic fail over and load balancing (scaling). The Gateway maintains a 'pool' of connections with the back end SAP system using SAP RFC (Remote Function Call) mechanism. It fully supports SAP logon load balancing and multi-instance scenarios, that is, multiple SAP application servers. In this way, you may effectively manage 1000s of device connections by scaling the number of Access Gateways and 'back end' SAP application servers.



Heartbeat Requests

When a mobile connection to SAP is initiated, Kony for SAP Java Server first registers itself, that is, provides identification and system information to the centralized System Management 'inside SAP'. After this, it periodically checks the connection status with SAP to ensure there are no problems and to

receive notifications from the SAP system (for example, 'push' data requests). This process is known as a 'Heartbeat Request'. The time in-between each heartbeat request is configurable, usually every 60 seconds. Each heartbeat request takes an average of 1,000µs microseconds (millionths of a second) that is, 0.01 seconds to execute in SAP.

Data Synchronization Mechanism

Kony for SAP has a built-in data synchronization engine that controls all facets of the process and is optimized for SAP. Data objects are automatically transferred directly between the Secure Container and SAP, that is, no specialized middle-ware broker and staging area is required. Kony for SAP Secure Container has a built-in database that is used to store all 'local' information. The overhead of Kony for SAP internal synchronization processing inside SAP is trivial in comparison to the actual SAP extraction / posting process; averaging 50,000µs microseconds (millionths of a second), that is, 0.05 seconds per request. This means that the low end benchmark SAP application instance (documented in the appendix) can handle 20 concurrent data object requests per processing second. In addition, you can schedule data object uploads to execute asynchronously in the background and you can use Kony for SAP interface management functionality to effectively schedule and load balance transactions thus avoiding 'flooding' the system.

Kony for SAP framework fully supports the concept of 'Data Objects'. A Data Object definition consists of all the inter-related tables and binary objects (files) that constitute a transaction. The primary benefits that all related data is synchronized together in an all or nothing scenario; guaranteeing delivery and avoiding incessant problems with synchronizing individual tables and files. This functionality significantly improves performance by reducing the number of calls and is a vital consideration for high volume applications.

The role of the Sky Data profiler is to efficiently distribute data based on configured rules, that is, which user / device gets what data? It effectively automates the 'push' data functionality (discussed in the next section). Because the 'profiler' executes inside of SAP, it can control synchronization in a single phase, rather than use temporary device staging areas and / or a central data store controlled by a middle-ware broker.

Utilizing a Push Synchronization Model

Kony for SAP fully supports both 'push' and 'pull' mechanisms to synchronize data objects. A 'pull' model has the Secure Container constantly polling the SAP environment at regular intervals to see if there is any data to download. The problem with this approach is that SAP gets spammed with requests that unnecessarily clog up the system (most of which don't return anything). A better approach is to use a hybrid 'push' model, that is, notify the remote mobile device only when it has data objects to download from SAP, that is, event driven. This eliminates the device having to constantly poll and the need for complex interim device data stores.

For example:

- A work order is created in SAP and an event is raised (through a change document, user exit, and configuration).
- The data profiler is invoked that determines which devices / users you should notify through a preconfigured set of rules, for example, a specific work centre / operation is related to a specific work gang or crew in the field.
- Notification requests are created and are transmitted to the Java Server on the remote device the next time it checks its connectivity status with SAP.
- The remote Java Server then issues a download request for the data object(s) and updates its own local database.

In this way, 1000s of remote devices in the field only need to synchronize data with SAP when required. In addition, Kony for SAP Data Object technology means that all the related tables and binary objects are synchronized in a single call.

Scalable Components

As mentioned in the previous architecture section, you can scale the Gateway and SAP Application Server components to service the volume of mobile connections and data transactions and also provide redundancy in the process. On the SAP side, a lot does depend on the data extraction and transaction posting requirements that are variable depending on the type of mobile application and the customer configuration.

Access Gateway

You may deploy these on stand alone servers or on the SAP server instances and support both Windows and UNIX operating systems. You can configure the maximum connections that a gateway can handle and the escalation chain, that is, tries Gateway1 then Gateway2. Each Gateway does require memory to decompress, de-crypt data and so there is a limit of the number of concurrent connections that you can process. The gateway is multi-threaded and so the amount of memory and processing power of its host server is the limiting factor. For high volume scenarios, Kony recommends the following minimum Gateway configuration on either a stand alone UNIX or Windows 64 bit server (or virtual machine):

Item	Value	Unit
Max connections	500	-
Min memory	3	GB
Min CPU	2	GHZ
Min Disk	150	MB
Max SAP RFC connections	25	-

Note:

- The Access Gateway requires a JVM (Java Virtual Machine) installed and configured to utilise the memory.
- Virtual machine environments (for example, VMWARE ESX) are fully supported.
- 500 connections are conservative, the above configuration can support double that, but there is risk if large amounts of memory are required concurrently to de-compress and / or de-crypt data.
- The SAP RFC connections are connected 'handles' to the SAP RFC gateway and do not represent SAP 'dialog' processes.

- You should configure a DRP backup Access Gateway to handle large connection volumes on a temporary basis.
    - You must have the operating system (OS), a server edition, for example, Windows 2008 Server that can handle large amounts of TCP / IP connections and traffic.
    - The Access Gateway fully supports SAP load balancing groups that you can use to segregate which SAP Application Servers the mobile system can utilise.

Example:

Using the above recommendation, a 3000 mobile device implementation requires six gateway servers (500 connections each) and a single DRP gateway server for a unlimited number of connections (also catering for any overflow).

SAP RFC Gateway

All SkyMobile Gateway RFC client / server connections with SAP are directed through a SAP Gateway service that handles all CPI-C services within the SAP world. It is important to note that SAP supports multiple of these gateways (one per instance as a default) and has configuration that controls the maximum number of connections, and timeouts. When considering a potentially large number of SkyMobile Gateway RFC connections, it is a good idea to review the existing SAP set-up and consider implementing multiple SAP RFC gateways to segregate / control the mobile connections and improve performance. This functionality and recommended settings is fully documented in the SAP help, and OSS notes. It is important to realize that the SkyMobile Access Gateway manages a connection pool with SAP that is used to process concurrent requests and therefore even though hundreds / thousands of devices are connected to the gateway, a much smaller number of concurrent requests with SAP is processed.

Sky recommends that you consult your SAP documentation and OSS on the limitations of the SAP RFC gateway and what parameters are used to tune it. This information is not listed here as it varies between SAP releases and is much better documented in the SAP documentation.

SAP Application Servers

To cater for the SAP processing load to extract data and post transactions, you may add additional dedicated SAP Application Server Instances to the SAP production environment and assign them to a specific mobile load balancing group. This effectively segregates these Application Servers away from the other SAP users for the exclusive use of the SkyMobile Gateways. This also means that the SAP 'Dialog' and 'Background' processes allocated to these SAP servers are used exclusively to process the mobile RFC and transaction load. You should view this as a last resort and should become evident after performing a capacity planning exercise. In addition to this, you may control advanced transaction management and SAP resource allocation 'inside SAP' using the SkyConnect interface management, scheduling and high performance functionality (included as a standard component of SkyMobile). See the following discussion on Interface Management for more details.

Potential advantages of implementing additional SAP Application Servers is to:

- Avoid any impact on the existing SAP system.
- Fully utilize the excellent SAP Client / Server architecture and low hardware costs to balance the load across as many computing resources as possible.
- Provide a higher level of availability and redundancy.
- Reuse the existing SAP system management, security and DRP processes.
- Utilize existing SAP technical and support skills.

Note:

- It may also be practical or cost effective to implement one or more 'high end' servers or even use a virtual machine environment such as VMWARE.
- It is rare that you need to increase the capacity of the SAP Central Database Instance. But in the exceptional case, you can add more update tasks and database connections.

Modelling:

Since the processing time of the SAP data extracts and transaction posts is entirely variable depending on the mobile application and SAP configuration, it is important to model and prototype the resources required and apply them to the expected frequency and volume. For example, identify the:

- Interface types:
    - Data Object extracts and synchronization requests
    - Transaction postings
    - Interface frequencies (especially the predicted peak periods)
    - User expectation on turn around.

Taking the above variables, it is then a reasonably straight forward process to calculate the SAP resources required to effectively service the demands of the mobile application and calculate how many application servers are required. See the example capacity model later in this section.

Interface Management

The SkyMobile framework has an interface management component called SkyConnect that runs 'inside SAP' to effectively and efficiently schedule, manage and monitor all data extracts and transaction postings. Since SAP interfacing is probably the most critical component of the architecture that has the most bearing on performance and the impact on the SAP environment; it is vital to control and manage interfaces 'inside SAP'. In this way, you can properly utilize SAP resources to prevent transaction flooding and negative impact on non-mobile users and processes. SkyConnect is a SAP certified interface management and scheduling tool that provides the following key functionality:

- Queue management and prioritization
- 'Processing zone' management and load balancing
- Dependency control
- Centralized monitoring
- Technical workflow of transactions

- Data translation and SAP integration
- High volume transaction processing
- Cross SAP integration, for example, CRM and ERP
- Safe restart / recovery mechanisms

Since SkyConnect is event driven and optimized, the impact of its internal processing on the SAP environment is negligible, that is, it takes 0.03 of a second to start an interface. In a high volume scenario, SkyConnect can considerably improve the processing of asynchronous transactions through its 'background server' functionality. This option stores all the interface details in a queue that is subsequently analyzed and processed by a scalable number of dedicated background tasks (SAP parallel processing is fully supported). In this way, you can place a 'throttle' on the SAP transactions to fully control the load volume and frequency. Short / sharp transactions, such as postings, benefit most from this technology that can improve throughput by over 25% (Customer Benchmark). If you do not have SkyConnect functionality 'inside SAP' and all transaction postings are submitted using standard SAP mechanisms, you have to ask the questions: 'Just what is going to schedule and manage the work?' and 'How do I minimize the impact on the SAP environment?'.

Networking Considerations

All SkyMobile network communications use standard TCP / IP sockets or HTTP and thus all SkyMobile Gateways require a dedicated IP address and unique port. In a high volume scenario, a large number of device connections generating IP traffic can impact the network and consideration and you must give to the available bandwidth, and IP configuration. SkyMobile already optimizes data packets by sending only what is required and compressing the data as much as possible. However, 1000s of devices generating packets can cause congestion.

Sky recommends the following considerations:

- Use one or more dedicated APNs (Access Point Name) with a high speed link from the telecommunications provider to the Customers LAN (on its own sub-net).
- Use intelligent switches and VLAN to segregate and prioritize the mobile traffic within the LAN.

- Use multiple SkyMobile gateways to spread the network port connections and load across multiple subnets, TCP / IP stacks, and network cards.
- Use multiple SAP Gateways, Application Server Instances and a dedicated SAP load balancing group to effectively segregate and spread the RFC traffic across the SAP environment.
- TCP / IP tuning , that is, optimal system settings, configuration and hardware. It is worth investing in specialist consultants and software to perform the capacity planning and benchmarking required. You can do this best using a prototype mobile application and scaling it up to measure the network traffic.

Performance Analysis and Capacity Planning

It goes without saying that applications need performance testing to make sure that they meet user and system expectation. Typically, you need to analyze and tune to ensure that processing outside of SkyMobile control such as database and logic processing don't cause excessive waits and high resource consumption. You find that a lot of the time, these problems only manifest themselves in production when using larger volumes of data. It is therefore important to 'benchmark' high volume performance as part of the QA (quality assurance) and UAT (user acceptance) phases of the project; and also simulate load scenarios, for example, 100 concurrent requests to measure the impact on the SAP system. A well designed data model utilizing SkyMobile Data Objects and Profiling, a 'push' data strategy, and event driven SAP integration using SkyConnect, are all essential design considerations to avoid performance problems in high volume scenarios.

SkyMobile provides the following tools to help analyze performance:

- Application performance traces in SAP and the Secure Container that highlight where the elapsed and processing time is spent.
- An RFC call trace in SAP that summarizes the type and volume of requests issued against the SAP system.
- Data Object and LDB traces in SAP that can highlight the frequency and cost of data transfers.
- SkyMobile Gateway statistics. These provide a summary of total gateway throughput and can help highlight insufficient SAP connections, inefficient requests, and network latency.

- A performance test tool (PTT) that you can use to record and playback screen dialogs, that is, simulate load.

By using these tools in conjunction with network monitoring and standard SAP performance tools, you can document the actual impact of the mobile application; and you can put in place a effective capacity planning model as part of the 'go live' process.

Example capacity model

The following tables show an example of how to compile a resource model of the primary mobile application interfaces. These give a very rough guide only, but help to demonstrate the overhead of the interfaces and what SAP processing capacity is required. As a rule of thumb, you should also apply a 25% contingency. You should also plan against the worst peak scenario and take into account the additional overheads (waits) incurred with concurrent processing, that is, locks, and memory paging.

Interface overhead:

Interface	Component	Elapsed (s)	Run time (ms)
Extract Work Order	SAP change document trigger and SkyMobile notification.	0.5	700
	'Push' heartbeat command notification	10 queue	-
	Data Object refresh (download)	0.5 network	900
	Total	11	1,600 (1.6 sec)
Post Work Order	Data object upload to SAP	0.5 network	800
	Asynchronous BAPI call	2	1,300
Total		2.5	2,100 (2.1 sec)

Peak SAP impact:

Interface	Peak Period	Volume	Run time (seconds)	Avg Rtime per second
Extract Work Order	7am - 8am (1 hour)	1000 x 1.6 seconds	1,600 (per hour)	0.4 (400 ms)
Post Work Order	Trickle fed during day Max 20 per minute	20 x 2.1 seconds	42 (per minute)	0.7 (700 ms)
Total				1.1
	+25% contingency			1.4

The current peak total run time per second for the whole SAP environment without this extra load is 4,700 ms per second. Therefore the percentage increase in the overall capacity required, that is, CPU, Memory, Disk I / O is approximately 30% (1,400 / 4,700). So if the current SAP system is peaking at 85% CPU, then obviously more CPU is required. You need to apply the same to memory and the disk I / O throughput.

5.10.5.5 RFC Call Recommendations

Sky recommends that you activate the RFC call trace at the system level for a period of time to capture the frequency and overhead of all the RFC calls to the SAP Host. These statistics are captured at the date, hour, server group, server id, function level and you may view using the SAP Host RFC call facility. Based on the results, you can tweak the Secure Container configuration to help minimize the impact of processing against the SAP host. Some of the more common functions and their associated configuration option is listed below:

Function	Configuration
/SKY/VTI_ HEARTBEAT	The frequency of the heartbeat is controlled by the following gateway and server configuration (in seconds): SERVER.HEARTBEAT.HEARTBEATINTERVAL= 15 (default) SERVER.HEARTBEAT.HEARTBEATCONNECTEDINTERVAL = Central gateways on the LAN do not generally need a frequent heartbeat and you can set this to HeartbeatInterval = 300, that is, (5 minutes). On devices, you can make use of the "lazy heartbeat" feature to space out the frequency of heartbeats once one is established, for example, SERVER.HEARTBEAT.HEARTBEATINTERVAL= 60 SERVER.HEARTBEAT.HEARTBEATCONNECTEDINTERVAL = 180 What happens here is that the lazy frequency is used until a connection exception is raised, whereupon the server resorts back to the original heartbeat interval until a connection is established again.

Function	Configuration
/SKY/VTI_ DOWNLOAD_ OBJECT_ DATA /SKY/VTI_ REFRESH_ OBJECT_ DATA /SKY/VTI_ UPLOAD_ OBJECT_ DATA	Data Object (DOB) data transfer operations. The tuning of these frequencies is covered in Database call recommendations.
/SKY/VTI_ DOWNLOAD_ LOCAL_DATA /SKY/VTI_ REFRESH_ LOCAL_DATA /SKY/VTI_ UPLOAD_ LOCAL_DATA	Local Database (LDB) data transfer operations. The tuning of these frequencies is covered in Database call recommendations.
/SKY/VTI_ SELECT_ TIMESTAMPS	In a production / QA environment with minimal application change, you should consider the following application refresh check configuration: SERVER.APPLICATIONCACHE.APPLICATIONREFRESHINTERVAL = 30 (default) You can also scale back to something like 900 (that is, 15 minutes).

Function	Configuration
/SKY/VTI_ SELECT_ PACKAGES	This is how often the secure container should check the host for "Packaged software updates". Sky recommends that you can make this fairly infrequent for QA and production SAP environments, for example, every 1 hour. SERVER.BINARYFILES.PACKAGEREFRESHINTERVAL = 3600
/SKY/VTI_ SESSION_ MANAGER	These are real-time or "post local" calls to the SAP session manager. If these are excessive, you may want to review the design of the application with a view to changing as much processing as possible to be local, that is, processed directly by the Java Server.

5.10.5.6 Database Call Recommendations

This section describes how you can tune Data Object (DOB) and Local Database (LDB) operations to help minimize the impact on the SAP host. Also refer to the Data Object, Local Database and MEAP performance trace sections for more details on how to drill down the calls and diagnose what is happening.

> ***Note:*** You are also highly recommended the following options and techniques, and you should use wherever possible:
>
> Use keyed (indexed) retrievals.
>
> Use LDB primary key traversals in conjunction with transfer buffering wherever possible to minimize the selected and stored data set sizes.
>
> Use the "Push" command technique to avoid excessive server polling of the SAP host.

Key Topics

- Buffering Data
- Scheduling Data Transfers
- Changing Table Definitions

Buffering Data

For large data transfers, you may overcome the network congestion and memory issues by implementing a maximum buffer size. This has the effect of breaking the overall data transfer into more manageable 'chunks'. The Secure Container then receives and processes a 'chunk' of data at a time, and repeatedly calls SAP for more until the total transfer is complete. The 'TransferBufferSize' option

determines the maximum number of bytes that you may transfer from SAP to the Container at any one time. You specify this parameter in the local database section of the Container configuration file. A value of '0' (default) indicates no imposed limit. The buffering also applies to upload processing. By default, SAP imposes a transfer buffer of 500K.

> ***Note:*** Data objects return the entire 'child' dataset for each top level parent (root) table row irrespective of the max buffer size. You must undertake the careful design consideration of root table (top level parent) and child table dependencies if buffering is required.

Go back to the top.

Scheduling Data Transfers

Secure Container Pull Requests

In the case of 'pull' requests, the Secure Container refreshes and uploads functions always against the SAP host. You may schedule these to occur on an interval, for example, every five minutes, or at a scheduled time of day, for example, 19:00:00. You configue these scheduling options against each DOB / LDB definition in SAP using the respective workbenches. You may override scheduled times for specific servers or groups using the **Utilities** > **Application Server** > **Schedule LDB / DOB transfers** menu options in the Workbench. You can also configure Downloads of data to always occur at Secure Container start-up. Kony recommends that you use the "push" technique wherever possible based on events in the SAP back end. This reduces any unnecessary polling.

> ***Note:*** You can use the data transfer and RFC call tracing to determine how often and how expensive data transfer operations are. From the results, you can alter the intervals and scheduled times to optimize processing.

Push commands

In addition to the normal 'pull' processing using intervals and scheduled times, you may also implement 'push' commands to secure containers using the heartbeat command queue. This can considerably

reduce, or eliminate completely, the number of routine calls to the host. This technique effectively implements an event driven model where processing on the host determines when data synchronization should be triggered. The ABAP programming guide has information on how to use the heartbeat command API to do this.

Go back to the top.

Changing Table Definitions

The following table documents how SkyMobile processes changes to an existing database table definition. All the functions are supported for the SkyMobile built-in database system, however you may need to check the compliance level if you are using a JDBC driver. Refer to the General Sky Database installation and configuration guide for more details.

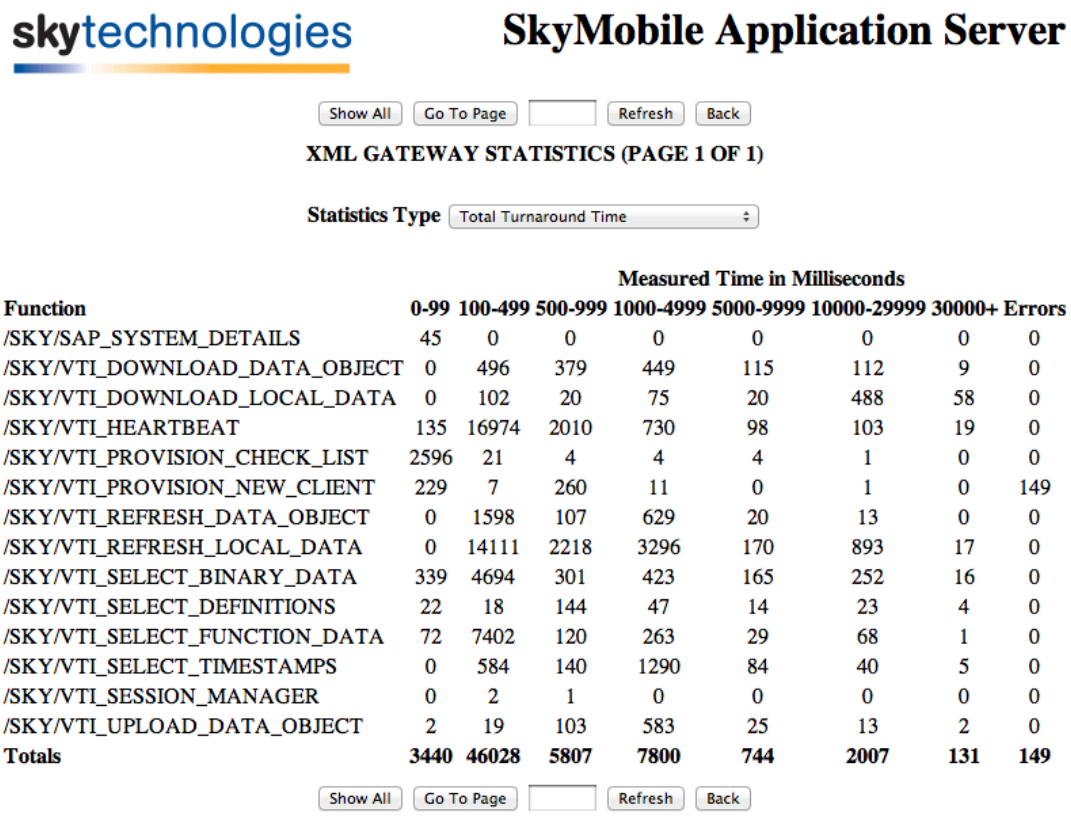
Action	Result
Insert a new field	The new column is created in the table with a null value.
Delete a existing field	If the database system supports it, the field is removed from the database table
Change a field type or length	If the database system supports it, the field is changed and the data is converted.

Action	Result
Change a primary key	***Important:*** You should take extreme care when altering the primary key of a table That is, the field(s) that make each row unique. If you are simply adding fields to the end of the existing key, this is usually fine. However if you remove fields or change the data type of an existing field, then you should carefully consider the impact. ***Important:*** Sky recommends that you implement an export data, delete/change the definition, import data strategy.

Go back to the top.

5.10.5.7 Access Gateway Statistics

The SkyMobile Access Gateway server keeps a list of throughput statistics that are useful to ascertain potential bottlenecks. The statistics button is available on the gateways web status page. The following screen appears:



skytechnologies

SkyMobile Application Server

Show All | Go To Page | Refresh | Back

XML GATEWAY STATISTICS (PAGE 1 OF 1)

Statistics Type Total Turnaround Time

	Measured Time in Milliseconds							
Function	**0-99**	**100-499**	**500-999**	**1000-4999**	**5000-9999**	**10000-29999**	**30000+**	**Errors**
/SKY/SAP_SYSTEM_DETAILS	45	0	0	0	0	0	0	0
/SKY/VTI_DOWNLOAD_DATA_OBJECT	0	496	379	449	115	112	9	0
/SKY/VTI_DOWNLOAD_LOCAL_DATA	0	102	20	75	20	488	58	0
/SKY/VTI_HEARTBEAT	135	16974	2010	730	98	103	19	0
/SKY/VTI_PROVISION_CHECK_LIST	2596	21	4	4	4	1	0	0
/SKY/VTI_PROVISION_NEW_CLIENT	229	7	260	11	0	1	0	149
/SKY/VTI_REFRESH_DATA_OBJECT	0	1598	107	629	20	13	0	0
/SKY/VTI_REFRESH_LOCAL_DATA	0	14111	2218	3296	170	893	17	0
/SKY/VTI_SELECT_BINARY_DATA	339	4694	301	423	165	252	16	0
/SKY/VTI_SELECT_DEFINITIONS	22	18	144	47	14	23	4	0
/SKY/VTI_SELECT_FUNCTION_DATA	72	7402	120	263	29	68	1	0
/SKY/VTI_SELECT_TIMESTAMPS	0	584	140	1290	84	40	5	0
/SKY/VTI_SESSION_MANAGER	0	2	1	0	0	0	0	0
/SKY/VTI_UPLOAD_DATA_OBJECT	2	19	103	583	25	13	2	0
Totals	**3440**	**46028**	**5807**	**7800**	**744**	**2007**	**131**	**149**

Show All | Go To Page | Refresh | Back

The statistics type drop-down determines:

- Total turn around time of the request (inclusive of host processing time)
- Host processing time (just the host overhead) to process the request
- Gateway latency (handler wait) time. The amount of time spent waiting for a SAP host connection to become available from the connection pool.

Statistics are listed by the function calls and are categorized in groups of milliseconds that is in thousandths of a second. The number of system errors encountered is also listed. These statistics are only held in memory and are only indicative for the current executions of the server, that is, they start again when the gateway restarts.

5.10.5.8 Security

About

The security of applications and data is important to most organizations when deploying mobile devices that store data locally on the device and transmit data across wireless networks.

This section describes the various mechanisms available to secure the SkyMobile infrastructure and mobile applications in general.

Key Topics

The Secure Container Concept

This section describes the concept of a secure container and why it is required in a multi-application, multi-device world.

Overview

As mobility becomes the next big thing and traditional enterprise and consumer business processes merge; CIOs face with a number of key challenges:

- Managing different device types and operating systems
- Securing applications and data on both corporate and personal devices
- Identity management and “role” based provisioning of applications and data
- Providing dynamic context and location aware solutions
- Sharing devices between multiple users
- Managing multiple integrated B2B, B2E and B2C mobile applications.
- Globalization, that is, supporting multiple languages and time zones.

Deploying device specific code and browser solutions quickly leads to a fragmented change management nightmare. In addition, how do all these applications:

- Share a consolidated integrated database?
- Properly identify who the real user is and what role they perform?
- Secure the business process and company data?
- Seamlessly interact with each other?

Many vendors provide complex and costly middleware and device management solutions, mainly to address the need to manage fragmented solutions. Isn't the logical solution to run all of the applications and store all of the data in a kind of secure "virtual" environment on the device? And, deploy applications that can run everywhere without having to write complex code for every different platform or browser?

The answer is to deploy a complete mobility framework that supports a single consistent secure method to provision applications and synchronize data across all the major device types. Such a solution exists and it is called SkyMobile.

The key component to Kony for SAP success is the concept of a "Secure Container" that completely abstracts applications and data away from the specifics of the device and operating system. This paper outlines the benefits of the secure container and how it lowers the cost of development, deployment and operation when compared to other fragmented, code driven applications.

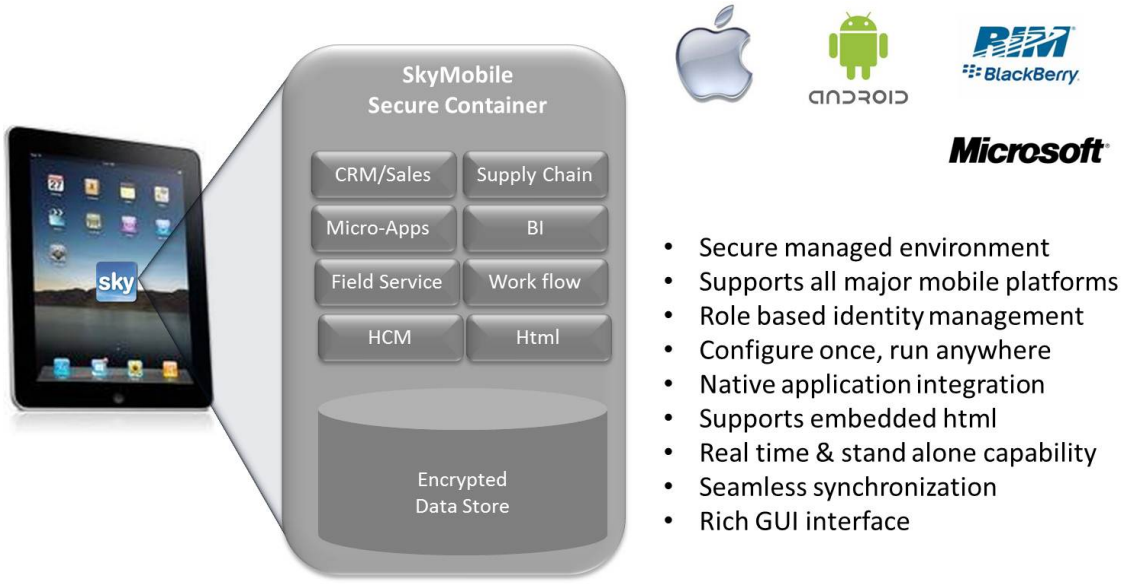
What is a Secure Container?

As mentioned in the overview, companies are faced with the problem of managing multiple mobile applications and data across a diverse range of devices. As well as requiring expensive specialists and technology, there are the added complexities of: different programming skill sets, application change management and security. To most CIO who successfully implemented back end ERP systems and effectively downsized their IT capability as a result; the thought of re-skilling in Objective C, Java, C++, and HTML is both a daunting and expensive one. As well as fragmentation, there are also the issues of identity management, role based application provisioning and data security. At this stage, some may argue that a browser running HTML applications solve these problems; but in reality, browsers have their limitations and fall well short of the overall requirement. Particularly in the areas of security, offline access, rich user interface, role based application provisioning and performance.

The logical solution is to implement a "secure virtual area" (container) that:

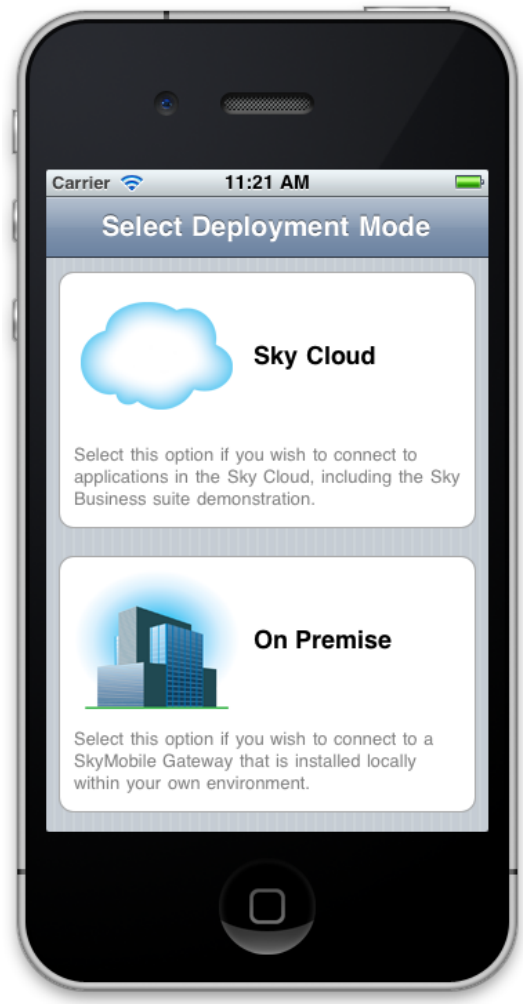
- Abstracts the application definitions away from the specifics of the operating system
- Provides a central secure data storage area
- Performs proper identity management, that is, user authentication and role authorization
- Automatically synchronizes application and data definitions
- Enables both native and “shared” application integration (including HTML)
- Supports global applications, that is, language and time zones
- Utilizes the full capability of the native user interface
- Supports multiple users on a single device.

The Secure Container meets all these requirements, providing a safe area on the device within which to run all applications and store all data.



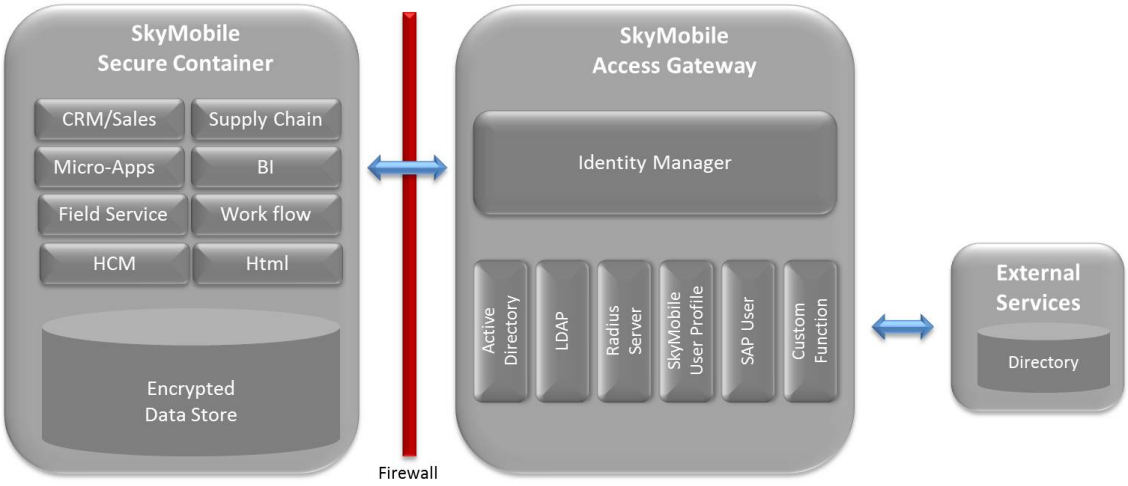
Provisioning the Container

The Secure container is a native executable available for all the major mobile platforms. You can download it and implement from either a public application store, a private enterprise store using device management software or through a vendor work bench. When the container starts for the first time, the user is prompted to connect to either the Kony for SAP cloud or to an “on premise” MEAP server, where its configuration automatically downloads. You also have the option to totally automate the process for a specific device using a “white list”, that is essentially a trusted list of unique device IDs.



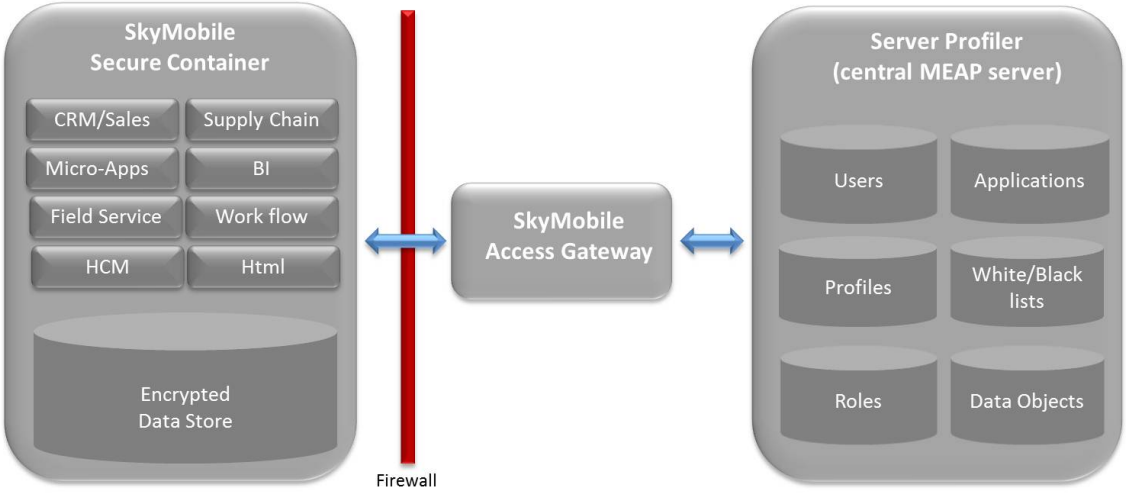
Identity management:

Once the container is successfully provisioned, the next step is to identify the user/device so that the correct role can be assigned. This process is called Identity Management and is usually performed against a “back end” directory service, by either prompting the user for information, for example, a user id/password, or silently deriving the associated user through a single sign on, and adaptive authentication.



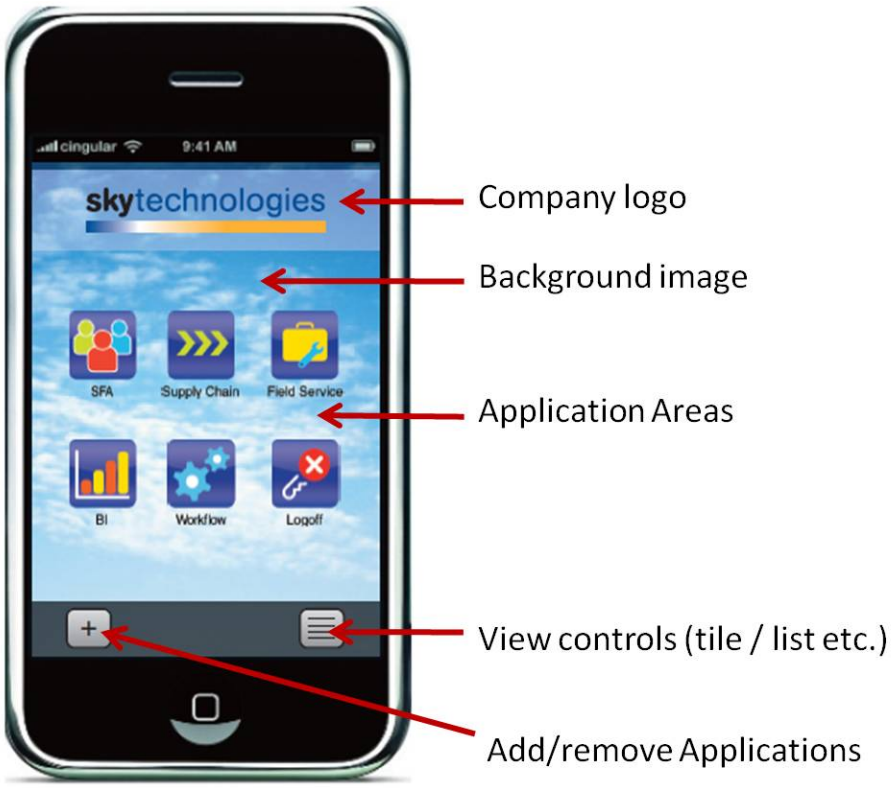
Role based provisioning:

When the user / device is identified and authenticated, all the applications and data profiles that are associated with the assigned roles are automatically synchronized with the container. Only the applications and data that the user / device is authorized to, is available. You may configure applications to “on demand”, that is, download at the user discretion. All the container configurations and profiling is maintained centrally in the MEAP server.



The application console:

The application console is the first display that a user sees after successfully authenticated by identity management and role based provisioning. It provides a central point from which to manage all applications on the device that the user is authorized to use. From here, users may add and remove the applications that they are authorized to, and launch them for execution. All applications are assigned to roles that are then assigned to specific users / devices. All these configurations are maintained in the central MEAP server. The console can support multiple users on the same device, using a multi-tenant option.



Multi-tenant data segregation:

One of the main issues with multiple users sharing the same device is how to effectively segregate the data between each user. The container utilizes a single database and so you must build the differentiator between the users into the data definitions. Kony for SAP supports multi-tenant applications and in this case, you can assign each user a unique tenant id that is used to effectively "partition" the data between users. An example of this is a device pool that is shared between field service contractors. Each different contractor can have a unique profile of applications and an administrator manages data centrally.

Decommissioning and black listing:

At any stage, you may blacklist an entire container or specific user. This means that the next time that the container starts and has network access; all the relevant applications and data is automatically removed from the device, that is, reset back to its initial provisioning state. This functionality is essential if a device is lost or stolen.

Device lock:

You may “lock” a container to a specific device, that is, if it is illegally copied to another device; it does not start, and can configure to automatically remove all data. This prevents any unauthorized backup or replication of the container data.

Configure Once, Run Anywhere

There is a native secure container available for each of the major mobile platforms: IOS, Android, BlackBerry and Windows. Because the container abstracts the application away from the device operating system, you no longer have to have to code different versions of the same application for each of the different platforms; that is, Objective C for IOS, Java for Android, C++ for Windows, and HTML variations per browser type. With Kony for SAP, you may configure the application just once and then run it on any of the major mobile platforms without change. In some cases where the user experience is totally different, for example, tablet versus smart phone, Kony for SAP can automatically ‘profile’ the correct application definition to use, depending on the type of device. This means that the same application definitions can run across all major mobile device types without change, that is, a true “configure once, run anywhere” capability. All the application and data definitions are stored securely in one secure place instead of fragmented and exposed. This is a major advantage over having to commission specialist programmers to re-write and maintain the application for each different platform, for example, Objective C for IOS, Java for Android and BlackBerry, and C++ for Windows. In addition, all the applications within the container can share the same secure database.

Change management:

You handle all application changes internally within the framework, that is, you do not require additional device management software. You may use MDM (Mobile Device Management) software or the public application stores to implement new versions of the container software.

Native, HTML and third party application integration:

Integration with native applications, such as email, contacts, calendar, maps, camera, and phone is built into the container and so no coding is necessary. This means that users may seamlessly navigate with the built-in device applications without having to leave the container. The secure container also supports the ability to launch third party applications and browser applications from within its framework. This offers users total flexibility and enables the container to use as a central secure point for all business applications. You can also imbed HTML browser applets directly into applications through its “web window” facility, providing complete control and application flexibility.

Security

The primary benefit of the secure container is total security of all its applications and data on the device. The following is a summary of the security features:

- You can control initial provisioning of the container itself through the use of trusted “white lists”, profiles and passwords.
- All configurations, application definitions and data are encrypted. Even if the device is hijacked, jail broken or the container is copied, the contents are protected.
- All data transmissions over the network are encrypted.
- You can lock the container to a specific device, meaning that it does not start if copied to another device.
- You may "black list" the container, that is, all applications and data is automatically removed if an attempt is made to connect to the host.
- You can use a range of identity management options to authenticate user access to the container through standard directory services, third party security applications, and custom functionality.
- Users can only access the applications and data that they are authorized to. The role based provisioning is strictly controlled through the user profiling facility on the central MEAP server.
- You may configure the container to automatically shut down if idle for a period of time or if the device goes into sleep mode.
- You can securely execute HTML inside of the container without the risks associated with a browser.
- All provisioning and access requests are audited.

FAQ

What mobile platforms are supported?

A native secure container is available for all the major mobile platforms, that is, Apple iOS (iPhone, iPad, and iTouch), Android, BlackBerry and Windows.

Can I run the same Kony for SAP application on any platform?

Because Kony for SAP applications are abstracted away from the specifics of the device operating system, you can run them without change on any of the supported platforms. There is no code, that is, you configure the layouts, and logic in the central MEAP server.

Can multiple users log on to the same container on the device?

Multiple (different) users may log on and each only has access to the applications that they are authorized to use. If the users share the same application, you must segregate the data using the multi-tenant option or another custom method. This is because a single container only supports a single database that is shared across all users.

Can users still log on and use the container if “offline”?

You can configure the Secure Container to automatically cache the last known user credentials so that they may log on even though there is no network connection, that is, they are totally “offline”. As soon as a network connection is available, the credentials are re-checked to make sure they are valid. If not, the user is logged off immediately. The container supports both real-time and offline applications.

Can the container be secured to a specific device?

You may configure the container to be locked to a specific device. This means that if it is copied or restored from a backup to another device, you cannot start it. All data is encrypted, meaning that you cannot view the contents using external tools.

Can the container be automatically decommissioned?

You can blacklist the container using the central MEAP. This means that all applications and data are automatically removed from the device the next time an attempt is made to connect.

Identity Management

As companies deploy more mobile business applications, across a range of mobile devices, integrated with a variety of back-end systems, it is becomes more and more important to properly authenticate remote users as to their identity and role, that is, what applications and data can they can access and use? Identity Management deals with identifying users within a system and controlling access to resources. In terms of Kony for SAP, Identity Management consists of User authentication and the retrieval of key information about the user from a variety of standard directory services and other sources. It is important to remember that the Secure Container runs locally on the device; providing a "secure environment" within which all applications are run and all data is stored. Kony for SAP has its own "role based" security model that is used to control the applications that a user may run and the data that is synchronized with the device.

For a more detailed explanation on the concepts of user authentication and role based application and data management, click here.

Overview

The first thing that you must decide is which identity management service to implement. Kony for SAP provides some base services that you may use out of the box, but if these do not provide the required functionality, then you may implement your own custom module into the framework. It is important to note that the heart of identity management is implemented in the Access Gateway and that the Secure Container acts a basic client and thus needs some minor configuration. In this way, you can change identity management in the gateway without impacting the remote Secure Containers.

The standard supported services at this stage are:

Service	Description
LDAP	Performs a basic LDAP user authentication.
SKY	Makes a call to the MEAP Server to validate a user ID and password against the user database. If successful, any associated language, tenant, work group, work area and parameters 1-8 are returned and the connection details are updated.
SAP	This option performs a SAP RFC call and returns the users log on language ID.

Service	Description
Custom	You may implement your own custom module written in Java into the Access Gateway. For more information on how to do this, click here.

Once you decide which identity services you require, you need to make the following changes:

- Update the Access Gateway configuration
- Update the Secure Container configuration to activate identity checking

Access Gateway Configuration

Identity Service Configuration Sections

Example LDAP service

```
SERVER.IDENTITYSERVICE:LDAP.DOMAIN = LDAP domain name
SERVER.IDENTITYSERVICE:LDAP.OFFLINELEASEHOURS = 1
SERVER.IDENTITYSERVICE:LDAP.PORT = LDAP server port
SERVER.IDENTITYSERVICE:LDAP.SERVER = LDAP server IP address
SERVER.IDENTITYSERVICE:LDAP.SERVICETYPE = LDAP
```

Example SAP service

```
SERVER.IDENTITYSERVICE:SAP.HOSTINTERFACE = SAP Host Interface ID
SERVER.IDENTITYSERVICE:SAP.OFFLINELEASEHOURS = 1
SERVER.IDENTITYSERVICE:SAP.SERVICETYPE = SAP
```

Example Sky service

```
SERVER.IDENTITYSERVICE:SKY.HOSTINTERFACE = SAP Host Interface ID
SERVER.IDENTITYSERVICE:SKY.OFFLINELEASEHOURS = 1
SERVER.IDENTITYSERVICE:SKY.SERVICETYPE = SkyMobile
```

Identity Port Configurations

In most cases, identity management calls from the Secure Container is directed through a normal gateway port. You may also configure a separate identity port if required.

Assigning Gateway Port Identity Service entry

```
SERVER.XMLGATEWAYPORT:PORT10000.DEFAULTIDENTITYSERVICE = name of the
default identity service to use if not specified by the container
```

Dedicated Identity port

```
SERVER.IDENTITYPORT:PORT15082.DEFAULTIDENTITYSERVICE = name of the
default identity service to use if not specified by the container
SERVER.IDENTITYPORT:PORT15082.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.IDENTITYPORT:PORT15082.HANDSHAKEENCRYPTIONALGORITHM =
SERVER.IDENTITYPORT:PORT15082.HANDSHAKEENCRYPTIONKEY =
SERVER.IDENTITYPORT:PORT15082.HANDSHAKEENCRYPTIONKEYFILE =
SERVER.IDENTITYPORT:PORT15082.HANDSHAKEENCRYPTIONKEYHEXENCODED =
SERVER.IDENTITYPORT:PORT15082.PORT = unique port number
SERVER.IDENTITYPORT:PORT15082.USEENCRYPTION = false
SERVER.IDENTITYPORT:PORT15082.XMLDEBUG = false
```

For more information on implementing encryption, refer implementing network encryption in the management security section.

Back to the #top

Secure Container Configuration

To enable identity management checking in the Secure Container, you need to implement a client IdentityCheck section that defines which Access Gateway port to connect to, the service to use and specific overrides on user prompt text labels. For a full description of all the configuration options available, refer to client configuration type in the installation guide. Typically, the Secure Container configuration is managed using the server profiler and pushed out to all the remote secure containers and Sky recommends this approach.

Client Identity Check Section

```
CLIENT.IDENTITYCHECK.LABEL:FOOTERTEXT = optional footer text on the
identity pop-up
CLIENT.IDENTITYCHECK.PORT = access gateway port number
CLIENT.IDENTITYCHECK.SERVER = access gateway IP address
CLIENT.IDENTITYCHECK.SERVICE = sky
```

User Authentication and Role based Provisioning

The section describes the high level concept of Identity Management and role based provisioning in terms of how users and devices are authenticated and how applications and data are profiled according to the users role in the organization.

- User Authentication (and the retrieval of key information about the user)
- Role based Application and Data Management
- FAQ
- Common Terms of Reference

It is important to remember that the Application Server runs locally on the device; providing a “secure container” within which all applications are run and all data is stored. Kony for SAP has its own “role based” security model that is used to control the applications that a user may run and the data that is synchronized with the device. This section provides a high level description of the components and processes.

5.10.5.9 User / Device Authentication

This process is all about identifying the user and the device against a central directory service and then optionally returning key attributes about the user that you may use in application processing, for example, work group, division, and employee ID. The authentication check occurs before the application server itself starts, thus preventing any synchronization of data from occurring until the user properly identifies. The following diagram identifies the key components of the authentication service.

1. Application Server "Secure Container"

The Application Server is provisioned onto the device. It provides a "secure container" environment within which all applications are run and data is stored. When the server first starts, it analyzes its local configuration to determine if identity management is enabled and which Access Gateway(s) to connect to. The gateway provides information on all the requirements, that is, and what information to prompt for. In this way, the gateway manages all identity management configurations centrally and securely. If single sign-on or "silent authentication" modes, you may inherit the user / device attributes from the operating system or other third party software on the device or, the user is prompted for the required attributes, for example, user ID and password. It is important to note that the system is entirely configurable as to what attributes are required; for instance, you may want the user to enter an active directory domain. You may also configure the system to maintain an "offline cache" so that a user may still log on with their last known credentials, even if the device disconnects from the network.

2. Access Gateway "Identity Manager"

Once you capture the required user / device attributes, a call is made to a central Access Gateway (usually behind a firewall) to perform the actual authentication. If the user is offline, the local encrypted cache is checked for the last known status. The call to the gateway is through a direct TCP / IP socket and the message content is encrypted. Once the gateway Identity Manager received the request, it analyzes its own configuration to determine which external "directory" service to invoke.

3. Built-in Directory Services

Identity Management supports a number of standard directory services that you may use to authenticate the user and optionally return key attributes. You implement these as industry standard APIs (Application Programming Interfaces) that should cover most situations, but in case they do not, you may implement your own custom functionality to support other systems or cater for differences in protocols and releases. The following table lists the standard built-in services:

Service	Description
Active Directory	Active Directory (AD) is a directory service that Microsoft created. It uses a number of standardized protocols to provide a variety of network services, including: • Lightweight Directory Access Protocol (LDAP) • Optional Kerberos-based authentication • DNS-based naming and other network information.
LDAP	The Lightweight Directory Access Protocol (LDAP) is an application protocol for reading and editing directories over a TCP / IP network. The latest version of LDAP is Version 3.
SkyMobile	The internal SkyMobile user profile engine.
SAP logon	Standard SAP RFC logon check.
Custom	Custom program using a supplied API.

4. External Service

The role of the external directory service is to:

- Authenticate the user / device credentials. In a “silent” mode configuration, you may need to derive the user credentials using the device attributes, that is, they are previously authenticated and Kony for SAP needs to inherit these details.
- Manage password expiry and changes (optional)
- Provide additional attributes about the user / device (optional).

Flexibility

The Identity Management functionality is highly configurable to enable you to:

- Dynamically define which fields are displayed and processed as part of the authentication (logon) process.
    - Return additional attributes about the user that the application and data synchronization requests can use.
    - Support password expiry warnings and changes.
    - Implement multiple types of directory service check, for example, a mix of Active Directory and LDAP.
    - Support high availability and DRP implementations.

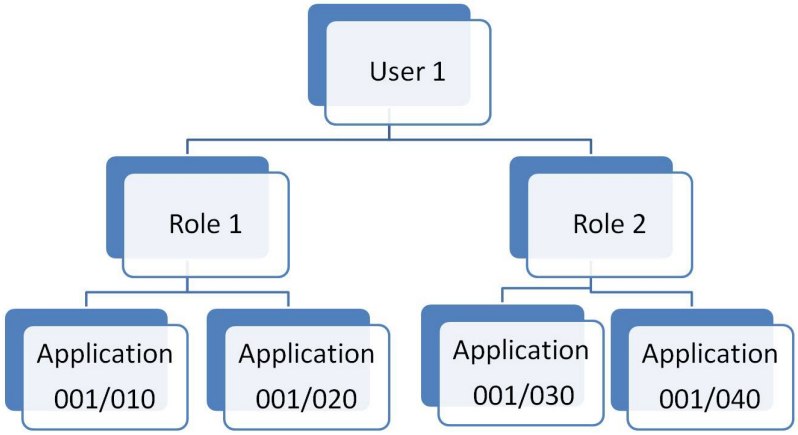
Silent Authentication

In some scenarios, the user does not physically log on to the Kony for SAP Container when it first starts, but is already pre-authenticated through the establishment of a secure network connection that establishes a trusted relationship between a specific user and device. In this case, Kony for SAP need to perform a silent lookup to “inherit” the user credentials from the associated authentication service using a generated certificate or the device technical details. Another example is single sign-on (SSO) where Kony for SAP “inherits” the user credentials from the device operating system, third party client application or a back-end host service. At the time of writing, there is no industry standard implementation of SSO across all the major mobile device types, and third parties thus client side SSO is not yet possible. However, you can implement a back-end host SSO service using the custom module functionality of the Gateway.

Role based Application and Data Management

Part of the provisioning process is to determine what applications and data the user is authorized to. You manage this using the “User Profiler” function of the Kony for SAP IDE (Integrated Development Environment) that enables you to define roles and allocate applications and data profiling rules. You can provision users when the Application Server is first provisioned, or when the user is successfully authenticated for the first time. In either case, the server checks its own local database to see if the user is provisioned; if not, a call is made to the “user profiler” on the central MEAP server. The user

profiler checks its configuration, that is, which roles are associated with the user, and automatically generates the host application cross reference and data object synchronization rules. A run-time cross reference for the user is also generated and is “pushed” out to the application server on the device and the host.



Dynamic data profiling rules

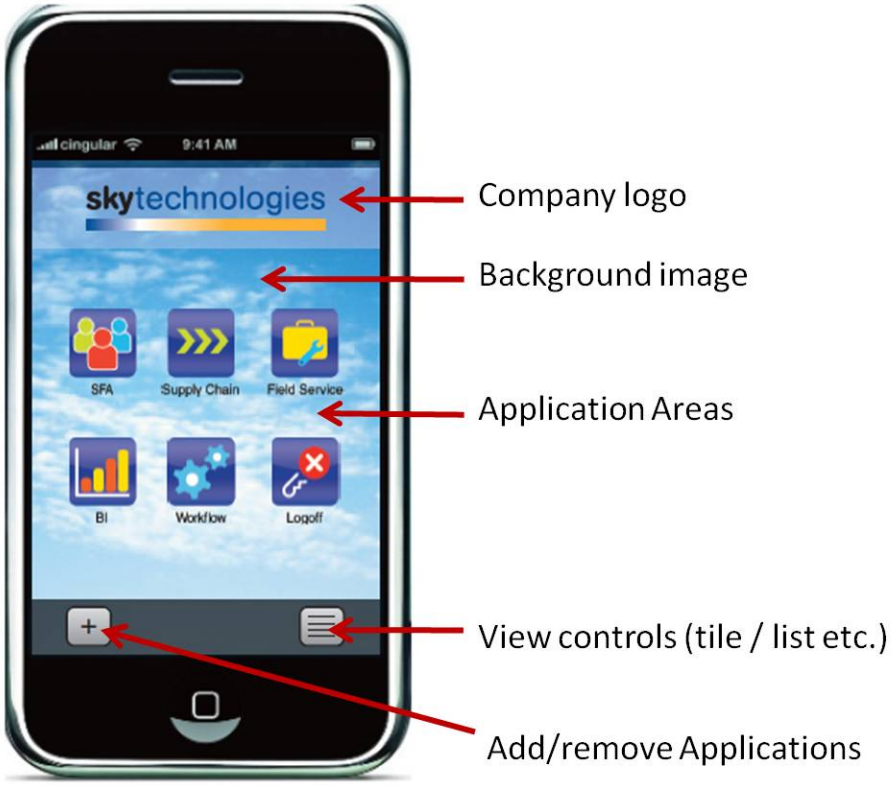
As mentioned above, you can automatically generate the data profiling rules for a user, so that they only receive the data that they are authorized to. To make this as dynamic as possible, you can associate attributes with the user that can then automatically substitute into data selection criteria, for example, sales division, work group, and region code. With Kony for SAP, you may associate up to eight 60 character attribute values with a user that you may then reference in any data selection criteria. For example: "SALES_REGION = &P1" where &P1 is the first attribute associated with the user that is either returned from a external directory look up or configured directly against the Kony for SAP user definition. This prevents having to configure static conditions in the Kony for SAP data profiler for each user.

Mobility on demand

Sky provides a default Application Console application that you may use as a launch pad for applications. The console provides the framework for “mobile applications on demand”, that is, you can add and remove applications from an authorized list. The user may also perform some basic

customization (if allowed), such as change the background image, assign different colors, manage a favorites list, and toggle the list view. You can fix all or part of the user interface, for example, the company logo and standard colour scheme. The application console also provides extra functionality to easily navigate between modules using the standard navigation bar that is present on every screen.

The user logs onto the console, using Sky standard Identity Management functionality. You use the user id is then to determine the available applications and groups. The mobile applications are organized within their application group (if any). The first console screen shows the group and all applications that are not explicitly assigned to a group. You can then easily expand to display the related applications. You can also add and remove applications that you are authorized to from the applications list.



Multi-tenant data segregation

One of the main issues with multiple users sharing the same device is how to effectively segregate the data between each user. Each Application Server shares a single database and so you must build the differentiator between the users into the data definitions. Kony for SAP supports multi-tenant applications and in this case, you can assign each user a unique tenant id that is used to “partition” the data between users and automate the data synchronization process.

FAQ

Can I plug-in my own directory service?

Kony for SAP uses the industry standard APIs for its built-in interfaces, you should require no modifications . However, if the release of your external directory software is not forward or backward compatible with the release used in Kony for SAP, then you may have to implement a “custom function” to handle the request.

Can I implement multiple directory services?

Identity Management functionality is very flexible, allowing you to implement multiple different directory services on a single Access Gateway and even multiple Access Gateways to support high availability and DRP requirements.

Can multiple users logon to the same Application Server?

Multiple different users may logon and each only has access to the applications that they are authorized to. If the users share the same application, you must segregate the data using the multi-tenant option or another custom method. This is because a single application server only supports a single database that is shared across all users.

Is password reset supported?

If supported by the external directory service, the user authentication process may easily implement and honour password expiry warnings and changes.

Can users logon if “offline”?

You can configure the Application Server to automatically cache the last known user credentials so that they may logon even though there is no network connection, that is, they are totally “offline”. As soon as a network connection is available, you re-check the credentials to make sure they are valid. If not, the user is logged off immediately.

Can I configure different logon fields?

The definition of what fields to display and process during the logon process are totally configurable and are defined against the Access Gateway, that is, not on the device. This makes it flexible to include additional fields that are supported by the external directory service, for example, domain and user defined attributes, for example, work group.

Can I return additional data about the user?

Up to eight 60 character attributes may be returned from the external directory service. You may then subsequently use these in applications and data synchronization requests.

Common Terms of Reference

Term	Description
Access Gateway	Access Gateway provides a single secure access point into the Customers network and back-end systems. It provides centralized services such as: encryption, compression and identity management.
Active Directory	Active Directory (AD) is a directory service created by Microsoft. It uses a number of standardized protocols to provide a variety of network services.
Application Console	A built-in generic application used to display, manage and launch applications that are associated with a specific user.
Application Server	The Application Server is a “secure container” that is installed onto the device. It runs all the applications and stores all data in a central repository. It enables both online and offline capability and handles all application and data synchronization with the central host (MEAP server).
Data Object	A grouping of inter-related database tables and binary objects.

Term	Description
Identity Management	The framework used to authenticate a user on the mobile device using a variety of in-built mechanisms, for example, Active Directory, LDAP, Radius Server, Host system, and User table.
Integrated Development Environment (IDE)	Contains all the workbenches and tools to define mobile applications, data objects, and profile definitions. It also provides all the system administration and management. The IDE is located centrally within the MEAP server.
LDAP	The Lightweight Directory Access Protocol (LDAP) is a protocol for reading and editing directories over a TCP / IP network.
Profilers	Special workbenches that are used to define rules to determine what applications and data get provisioned onto a specific Application Server on the mobile device.
Radius Server	Remote Authentication Dial In User Service (RADIUS) is a protocol that provides centralized authentication of network services.
SkyMobile	A complete framework to deploy mobile applications across multiple device types integrated with one or more back end systems. This type of framework is commonly referred to in the industry as a MEAP (Mobile Enterprise Application Framework).
Single sign on (SSO)	A mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords. Single sign-on reduces human error, a major component of systems failure and is therefore highly desirable but difficult to implement (source: The Open Group: http://www.opengroup.org/security/sso/).

Infrastructure Level Options

Securing the Web Status Page

The Application Server Web status page has options to stop and restart the server and maintain the local database contents. By default, it is not protected, meaning that any unauthorized person who knows the IP address and HTTP control port of the Application Server can effectively manage it. You can specify a user id and password in the configuration file. When you specify, a screen requesting logon details is invoked upon connection. The following HTTP control port configuration options that are available:

- To specify a user ID and password, configure 'AuthorisationUsername' and AuthorisationPassword'
- To restrict online configuration change, configure 'AllowConfigChangesFromWebPage'
- To control local database data enquiry and/or maintenance, configure 'AllowLdbEnquiryFromWebPage' and 'AllowLdbUpdatesFromWebPage'
- To control remote file management, configure 'AllowFileEnquiryFromWebPage' and 'AllowFileUpdatesFromWebPage'

Application Server HTTP Control port configuration

```
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWCONFIGCHANGESFROMWEBPAGE =
false
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWFILEENQUIRYFROMWEBPAGE = false
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWFILEUPDATESFROMWEBPAGE = false
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWLDBENQUIRYFROMWEBPAGE = true
SERVER.HTTPCONTROLPORT:PORT5071.ALLOWLDBUPDATESFROMWEBPAGE = false
SERVER.HTTPCONTROLPORT:PORT5071.AUTHORISATIONPASSWORD =
SERVER.HTTPCONTROLPORT:PORT5071.AUTHORISATIONUSERNAME =
SERVER.HTTPCONTROLPORT:PORT5071.HTMLBUTTONLOCATION =
SERVER.HTTPCONTROLPORT:PORT5071.HTMLPAGESIZE =
SERVER.HTTPCONTROLPORT:PORT5071.PORT = 5071
SERVER.HTTPCONTROLPORT:PORT5071.SECURE = false
SERVER.HTTPCONTROLPORT:PORT5071.SHOWLDBROWCOUNTS = true
```

```
SERVER.HTTPCONTROLPORT:PORT5071.SOCKETCLOSEDELAY =
SERVER.HTTPCONTROLPORT:PORT5071.SUPPRESSLOGO =
SERVER.HTTPCONTROLPORT:PORT5071.USEHTMLCOMPRESSION =
```

Securing the Configuration File

The `SkyMobile.cfg` file contains connection information that may be sensitive. You can automatically encrypt the configuration file using the web status page configuration maintenance facility. You need to have this only one and deploy to all servers, or you may do for each unique server. Refer to Viewing and maintaining the configuration file for more information.

Application Server / Gateway User Access Permissions

The user name under which the server runs should, if possible, have restricted access permissions on the host machine. You should only permit it to access files and directories that are directly related to the functioning of the SkyMobile system itself. You should never set the Application Server to run as a super user, administrator, or root account.

Using the Gateway to Restrict Access

You may use the SkyMobile Gateway as an application firewall, that is, execute in an isolated, secured environment; permitting access to specific ports and back end SAP host systems. In this way, the remote Java Servers need not know the specific SAP connection details, just an IP address and Port. You may also secure access to the gateway through a proxy server.

Restricting the IP Addresses that can Connect to SkyMobile

Either the Application Server and / or Gateway can restrict the IP addresses that may connect to it. You may specify specific or wildcard values using the 'ValidIpAddresses' and / or 'InvalidIpAddresses' options in the network or port sections of the configuration files. You may use this option to control what may, or may not, connect to the server or specific port on the server. Refer to network configuration section for more details. The same applies to the Gateway configuration file.

Application Server Network configuration

```
SERVER.NETWORK.ACCEPTTIMEOUT =
SERVER.NETWORK.BINDTIMEOUT =
SERVER.NETWORK.CHECKFORHOSTADDRESSCHANGES =
SERVER.NETWORK.HOSTNAMERESOLUTION =
```

```
SERVER.NETWORK.INVALIDIPADDRESSES =
SERVER.NETWORK.LISTENBACKLOG =
SERVER.NETWORK.NETWORKERRORSAMPLESIZE =
SERVER.NETWORK.NETWORKERRORTHRESHOLD =
SERVER.NETWORK.PREFERREDIPADDRESSES =
SERVER.NETWORK.READTIMEOUT =
SERVER.NETWORK.USENATIVESOCKETS =
SERVER.NETWORK.VALIDIPADDRESSES =
```

Using SAP Security to Restrict RFC Access

The Application server and Gateway both connect to SAP using a specific SAP user and password. You should set up this user in SAP as a CPIC type logon. This means that you may not use it to directly log on to SAP as a dialog user. You should also set up a specific profile to control what this CPIC user may do in the SAP system. A full example of a profile and authorization objects is given in the SkyMobile general installation guide. This profile only allows you to access certain function groups, that is, should limit calls to just SkyMobile, so that you cannot use the same CPIC user to call other SAP functions through RFC.

Data Encryption

You may configure data encryption at either the network transfer or database levels. To encrypt the data transmitted over the network, specify the EncryptionAlgorithm, EncryptionKey, EncryptionKeyFile and EncryptionKeyHexEncoded attributes in the host interface section of the configuration file. The gateway server that receives the data must also have its corresponding encryption attributes set. Various in-built encryption algorithms are supported and you may also implement your own. Also refer to the Encryption key generator section on what is supported and how to generate encryption keys.

Application Server Host Interface encryption configuration

```
SERVER.HOSTINTERFACE:XML.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:XML.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.HOSTINTERFACE:XML.ENCRYPTIONKEYSTRENGTH = 2048
SERVER.HOSTINTERFACE:XML.HANDSHAKEENCRYPTIONALGORITHM = RSA
SERVER.HOSTINTERFACE:XML.HANDSHAKEENCRYPTIONFILE = rsa_pub_key
SERVER.HOSTINTERFACE:XML.HANDSHAKEENCRYPTIONKEYHEXENCODED = true
```

```
SERVER.HOSTINTERFACE:XML.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:XML.MAXIMUMNUMBERHANDLERS =
SERVER.HOSTINTERFACE:XML.MAXIMUMOPTIMIZATIONLEVEL =
SERVER.HOSTINTERFACE:XML.MINIMUMNUMBERHANDLERS = 1
SERVER.HOSTINTERFACE:XML.USECOMPRESSION = true
SERVER.HOSTINTERFACE:XML.USECONDENSEDXML =
SERVER.HOSTINTERFACE:XML.USEENCRYPTION = true
SERVER.HOSTINTERFACE:XML.USETRANSFERFIELDS =
SERVER.HOSTINTERFACE:XML.XMLDEBUG = false
SERVER.HOSTINTERFACE:XML.XMLGATEWAYHOSTS = demo.skytechnologies.com
SERVER.HOSTINTERFACE:XML.XMLGATEWAYHOSTSELECTION =
SERVER.HOSTINTERFACE:XML.XMLGATEWAYPORT = 40002
SERVER.HOSTINTERFACE:XML.XMLGATEWAYRESPONSETIMEOUT = 60
SERVER.HOSTINTERFACE:XML.XMLGATEWAYSTALLTIMEOUT = 60
```

Application Level Options

SAP User IDs and Passwords

You require a SAP CPIC or Service user to enable Access Gateway to connect to SAP. This general 'system' user is used for all internal heartbeat, download, refresh and upload operations. This user is also used as the default for direct application connections (host functions). If you require to post transactions in SAP using the log on credentials of the users, then you must perform a "user switch". You may use Identity Management to perform a SAP user log on check and then use the user ID for all SAP transactions initiated by the device. Some customers pass in the SAP user ID/password as attributes returned from a active directory authentication. For details on how to implement SSO with SAP, contact Kony for SAP.

Active Directory Log on

Kony supports the use of Active Directory authentication, including the encrypted caching of user / password details remote stand alone applications. You implement these as part of the standard Identity Management functionality.

Data Encryption

You may configure Kony for SAP to encrypt all data traffic and / or database storage on the device.

Using SAP Security to Restrict Access

You may implement an ABAP authorization exit using the SAP administration console (system defaults). This exit is called for every SAP transaction and user exit call. You may then develop and implement a comprehensive check process, whereby the custom exit accepts calls from Kony for SAP and executes SAP authorization object calls to determine whether the function may use. For example, cancel an active session or execute screen function 'CANCEL ORDER'. The plant, work area, user ID, server-ID, mode and call-type attributes are all available for checking. A sample authorization exit /SKY/YVTIAUTH is provided with the installation and you may copy to another custom development class to used as a basis.

Restricting Applications to Specific Users

You may restrict the users to use applications using role based provisioning of user profiler.

Password Protecting Workbench Applications (lock)

You may specify a password (PIN) against a Kony for SAP application version that you need to specify in-order to maintain it in the workbench, and export / import it. This helps prevent any unauthorized changes from being made. Refer the SkyMobile Workbench guide for details. You can reset the forgotten passwords using the systems management console **Utility** > **Remove version lock**.

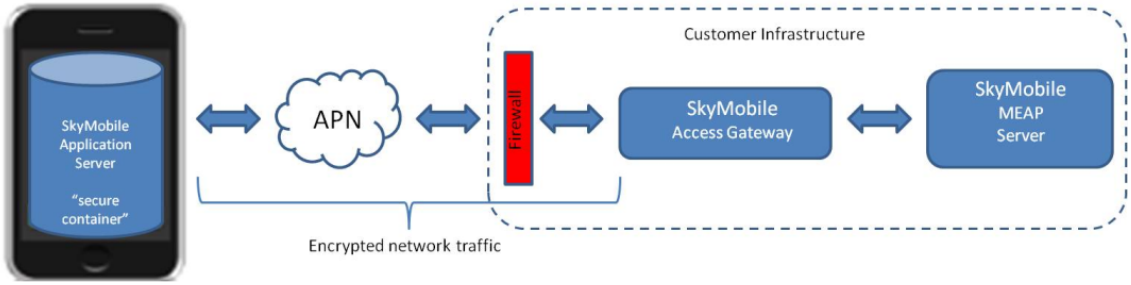
Making the Workbench Display Only

You can globally restrict the Kony for SAP Workbench to display only, that is, you may make no definition changes, except for change management export / import and application synchronization operations. This setting is designed to help lock down production environments for change management. You configure this option globally through the system configuration option on the System Management Console (YVTI).

Implementing Network Encryption

The topic of encryption is an extensive and complex one. However, it is helpful to understand some of the basic concepts when configuring Kony products to make use of encryption. Broadly speaking, there are two major types of encryption: symmetric and asymmetric. Symmetric encryption uses a single key both to encrypt and decrypt data. It is relatively fast, but management of the single key in use often becomes an issue (because it allows for both encryption and decryption of the data). Examples of some widely used symmetric encryption algorithms include AES, DES, Triple DES, and Blowfish. Asymmetric encryption uses a linked pair of keys (sometimes referred to as private and public keys). Asymmetric encryption is much more computationally intensive than symmetric encryption, but it leads to fewer key management issues because you can make one key (the "public" key) freely available as long as the other is kept secure. The most widely used asymmetric encryption algorithm is RSA. In order to overcome the limitations of each of these two types of encryption, it becomes common for a hybrid approach to adopt. This typically involves using asymmetric encryption to secure the exchange of a temporary "session key". Once the session key is generated and is known to both parties, all subsequent communications are then symmetrically encrypted using the session key. Hybrid approaches such as this, remove the key management issues surrounding symmetric encryption (because the session key is a temporary one, and is secured by the asymmetric encryption used during the "handshaking" process), and for the most part still, enjoy the associated performance benefits.

The following diagram shows the main components of Kony for SAP. The network encryption is implemented between the Application Server on the device and the Access Gateway that typically resides centrally behind a firewall. The objective is to effectively secure all mobile communications over the external network. By using internal encryption services, you no longer require specialized "secure tunnel" software and can safely use public network services, for example, the internet. No additional software is required. The Application Server is effectively a "secure container" on the device within which all applications are run and all data is stored.



Kony for SAP provides support for a number of different encryption modes:

- Exclusive use of symmetric encryption
- Exclusive use of asymmetric encryption
- A hybrid approach.

For a combination of performance and key management reasons, the hybrid approach is recommended for customers planning to implement encrypted communications. At present, the encryption algorithms supported are AES, DES, triple DES and Blowfish (for symmetric encryption) and RSA (for asymmetric encryption). Support for additional encryption algorithms is planned for future releases.

You include the table below for reference and provide a summary of the possible streams of encryption that Kony for SAP utilizes and configures:

	Endpoint	Relevant Configuration Setting	Comments
Provisioning	Client		Initial request is unencrypted, however you may redirect provisioning o an encrypted port using an encryption object.

	Endpoint	Relevant Configuration Setting	Comments
	Gateway	SERVER.XMLGATEWAY:xxx	Gateway configuration. You can still accept an initial unencrypted request if "AllowUnencryptedProvisioning" is enabled.
Identity Management	Client	CLIENT.IDENTITYCHECK	Client configuration. You can specify this through a provisioning instance and the encryption object, if specified overrides.
	Gateway	SERVER.XMLGATEWAY:xxx	Gateway configuration
Host Interface	Application Server	SERVER.HOSTINTERFACE:xxx	Client configuration. You can specify this through a provisioning instance and the encryption object if specified overrides.
	Gateway	SERVER.XMLGATEWAY:xxx	Gateway configuration
Client	Client	CLIENT.SERVER	Client configuration
	Application Server		Client configuration

To implement encryption to secure communications over a mobile network, perform the following:

- Generate the encryption keys using Kony for SAP EKG utility
- Configure an encrypted port on the Access Gateway (or multiple gateways)

- Install and configure the Application Server on the device:
    - Automatically using Server Profiling
    - Manually configure the options

Encryption Key Generator

The Encryption Key Generator (EKG) tool is a command line driven utility to generate encryption keys suitable for use with Kony for SAP. It is installed as a selected component of Kony for SAP Windows expert installer and you can find in a directory called `EKG`, below the Kony for SAP software installation directory. The EKG utility is invoked from a DOS or UNIX command shell, through the `ekg.bat` or `ekg.sh` file (respectively, as appropriate to the operating system in use). It accepts input parameters to control what type of key to build and generates a private and / or public key file with the naming convention: "{algorithm}.public/private_encryption_key".

Command Line Execution

Windows: ekg.bat -a <algorithm> -k <key strength> [-h]
Unix:ekg.sh -a <algorithm> -k <key strength> [-h]

Parameters

-a <algorithm> specifies the encryption algorithm (currently either BLOWFISH or RSA)
-k <key strength> specifies the key strength in bits
-h specifies that the generated key should be hexencoded

Output

Following a successful invocation, the EKG utility writes the randomly generated encryption key(s) as files, outputting them into the current directory. Asymmetric encryption algorithms such as RSA require both a private and a public key, in which case two files, each containing one of the key pair, is generated. Symmetric encryption algorithms such as Blowfish only generates one key. The naming convention of the generated key files is "{algorithm}.public/private_encryption_key".

Example

```
> ekg.bat -a RSA -k 2048 -h
File 'rsa_public_encryption_key' was successfully written.
File 'rsa_private_encryption_key' was successfully written
```

Example Implementation

The following examples of Access Gateway and Application Server configurations show the options to implement a hybrid asymmetric / symmetric encryption approach, that is, the initial "handshake" between the server and gateway is performed using RSA asymmetric encryption that generates a unique key that is then subsequently used to implement symmetric encryption for all communications. It is important to note that each Application Server platform, that is, Windows, BlackBerry, IOS and Android all have their own configuration editors. Refer to the Installation guide for more details and a more comprehensive description of all the available options.

Generate the Encryption Key Files

You do this using Kony for SAP EKG utility program. Here you generate a public and private key file for the RSA encryption algorithm with a key strength of 2048 bits.

Example

```
> ekg.bat --a RSA --k 2048 -h
```

```
File 'rsa_public_encryption_key' was successfully written.
File 'rsa_private_encryption_key' was successfully written.
```

5.10.5.10 Access Gateway Port Configuration

To implement an encryption port on Access Gateway, copy the rsa_private_encryption_key file into the Gateways own directory. Maintain its configuration as per below:

Access Gateway port configuration

```
SERVER.XMLGATEWAYPORT:PORT22000.COMPRESSIONALGORITHM = GZIP
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYFILE =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYHEXENCODED =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYSTRENGTH =
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONALGORITHM = RSA
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEY =
```

```
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEYFILE = rsa_
private_encryption_key
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEYHEXENCODED =
true
SERVER.XMLGATEWAYPORT:PORT22000.HOSTINTERFACE = SKQ
SERVER.XMLGATEWAYPORT:PORT22000.PINGCHECKIDLESECONDS =
SERVER.XMLGATEWAYPORT:PORT22000.PINGCHECKTIMEOUT =
SERVER.XMLGATEWAYPORT:PORT22000.PORT = 22000
SERVER.XMLGATEWAYPORT:PORT22000.USECOMPRESSION = true
SERVER.XMLGATEWAYPORT:PORT22000.USEENCRYPTION = true
SERVER.XMLGATEWAYPORT:PORT22000.XMLDEBUG = false
```

Note: You do not require an encryption key file for the Blowfish algorithm, because the Blowfish session keys are randomly generated.

Application Server Configuration

Note: You can automatically implement this configuration and key file using automatic server provisioning.

This example shows the options to set if you configure the server manually.

To implement encryption on the Application Server, first copy the `rsa_public_encryption_key` file into the `SkyMobile (VTI)` directory. Then maintain the configuration options as per below:

Application Server configuration

```
SERVER.HOSTINTERFACE:SKQ.COMPRESSIONALGORITHM = GZIP
SERVER.HOSTINTERFACE:SKQ.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.HOSTINTERFACE:SKQ.ENCRYPTIONKEYSTRENGTH = 2048
SERVER.HOSTINTERFACE:SKQ.HANDSHAKEENCRYPTIONALGORITHM = RSA
SERVER.HOSTINTERFACE:SKQ.HANDSHAKEENCRYPTIONFILE  = rsa_public_
```

```
encryption_key
SERVER.HOSTINTERFACE:SKQ.HANDSHAKEENCRYPTIONKEYHEXENCODED = true
SERVER.HOSTINTERFACE:SKQ.HOSTINTERFACETYPE = XmlClient
SERVER.HOSTINTERFACE:SKQ.MAXIMUMNUMBERHANDLERS =
SERVER.HOSTINTERFACE:SKQ.MAXIMUMOPTIMIZATIONLEVEL =
SERVER.HOSTINTERFACE:SKQ.MINIMUMNUMBERHANDLERS = 1
SERVER.HOSTINTERFACE:SKQ.USECOMPRESSION = true
SERVER.HOSTINTERFACE:SKQ.USECONDENSEDXML =
SERVER.HOSTINTERFACE:SKQ.USEENCRYPTION = true
SERVER.HOSTINTERFACE:SKQ.USETRANSFERFIELDS =
SERVER.HOSTINTERFACE:SKQ.XMLDEBUG = false
SERVER.HOSTINTERFACE:SKQ.XMLGATEWAYHOSTS = 192.168.2.62
SERVER.HOSTINTERFACE:SKQ.XMLGATEWAYHOSTSELECTION =
SERVER.HOSTINTERFACE:SKQ.XMLGATEWAYPORT = 22000
SERVER.HOSTINTERFACE:SKQ.XMLGATEWAYRESPONSETIMEOUT = 60
SERVER.HOSTINTERFACE:SKQ.XMLGATEWAYSTALLTIMEOUT = 60
```

Note: You do not require an encryption key file for the Blowfish algorithm, because the Blowfish session keys are randomly generated. Also, note the presence of the "EncryptionKeyStrength" parameter that indicates that you should generate a 2048bit Blowfish session key for use with symmetric encryption taking place after the initial handshake.

Automatic Server Provisioning

It is possible to fully automate the implementation of encryption to a remote Application Server using the Server Profiler, that is, over the air. You configure the provisioning definitions and rules using the Server Profile workbench.

> ***Note:*** Refer to IDE (Integrated Development Environment) guide for a full description of the Server Profiler workbench functionality, options and terminology.

This section describes how to configure an encrypted port on an Access Gateway and then configure and provision an encryption object. Kony for SAP recommends implementing a 'hybrid' approach for performance and security reasons. This is where the initial handshake between the Application Server on the device and the Access Gateway is performed using a asymmetric encryption mode and then all subsequent communications are performed using a symmetric encryption mode. This method ensures that the actual keys used are randomly generated.

The Server Profiler provides a central location where all configurations, application cross referencing, and data profile rules are automatically generated and "pushed" out to the Application Server on the device. The idea is that you implement the Application Server and then it automatically provisions itself when it connects for the first time. This becomes a challenge when implementing encryption, that is, the Application Server must know the encryption configuration, mode, and key in order to connect with Access Gateway and the back end host. To enable this, the Server Profiler has encryption object definitions that describe how to implement the encryption. This encryption object then gets attached to the server profile either explicitly or implicitly through a "white list" entry.

There are two possible provisioning modes:

- Using a white list entry (based on unique device type and id) in Kony for Cloud (recommended approach)
- Associated with a server profile definition (on premise)

Using a Sky Cloud white list means that you do not have to expose an unencrypted port on the customers Access Gateway on premise, that is, the device receives its encryption configuration automatically the first time it starts; and is then able to connect to the customers Kony for SAP access gateway encrypted port. If you use the server profile definition (on premise) option, the Application Server must first communicate with the back end MEAP server using an unencrypted port, receive the encryption configuration and then switch over to use an encrypted port, that is, the un-encrypted port is only ever used to support the initial provisioning of the device, that is, no application data.

The following example steps assume the use of a white list:

1. Generating and Loading the Encryption Key Files
2. Configuring Access Gateway
3. Creating the Encryption Object
4. Provisioning the Device (the Process)

5.10.6 Generating and Loading the Encryption Key Files

The first thing that you need to do is generate the public and private encryption keys that you use to encrypt the data that is transferred between Application Server and Access Gateway. Kony for SAP provides a Windows utility program as a component of "Windows expert installer" (see the installation guide for more details). Once installed, the Encryption key generator utility program is run from the command line to automatically generate the encryption key files. For example:

Example

```
> ekg.bat -a RSA -k 2048 -h
File 'rsa_public_encryption_key' was successfully written.
File 'rsa_private_encryption_key' was successfully written.
```

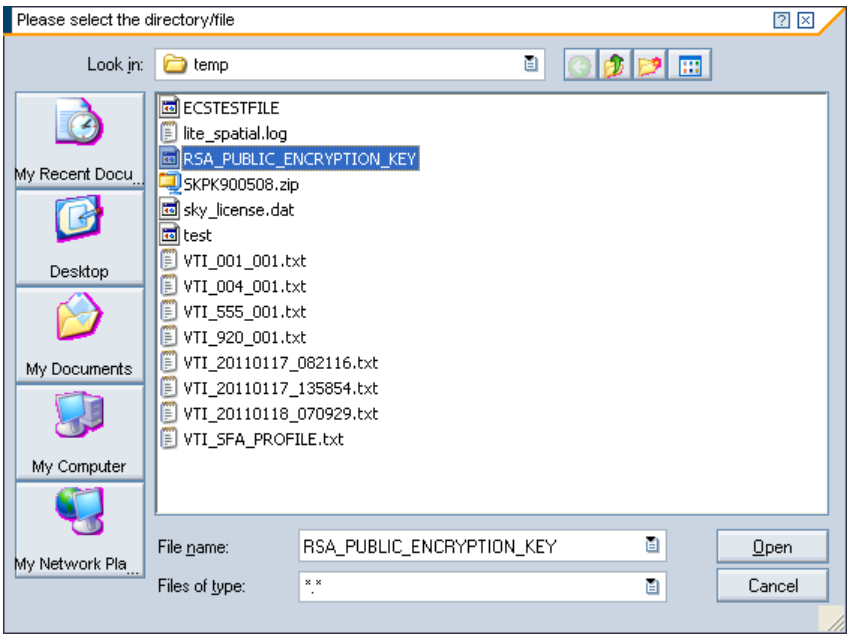
> ***Note:*** The RSA encryption algorithm is used with a key strength of 2048.

The -h parameter indicates that the value is hex encoded, meaning that it is both portable and somewhat human readable. There is also no extension on the file.

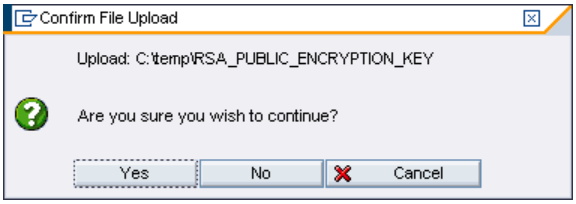
Once generated, upload the key files into the Binary Repository. This is performed in SAP using Kony for SAP IDE. Type in transaction code "/n/sky/yvtb" (or yvtb for short) to start the binary object manager and execute past the initial selection screen. Use the upload file utility to load the file into the central binary repository on the host. The IDE binary object guide describes how to do this in detail, but the following are the basic steps:

Once you start the binary manager, click the 

icon in the application toolbar. You are prompted for the file that you wish to upload. Navigate to the key files that you generated with the Sky EKG utility (as above). Select one and click open.

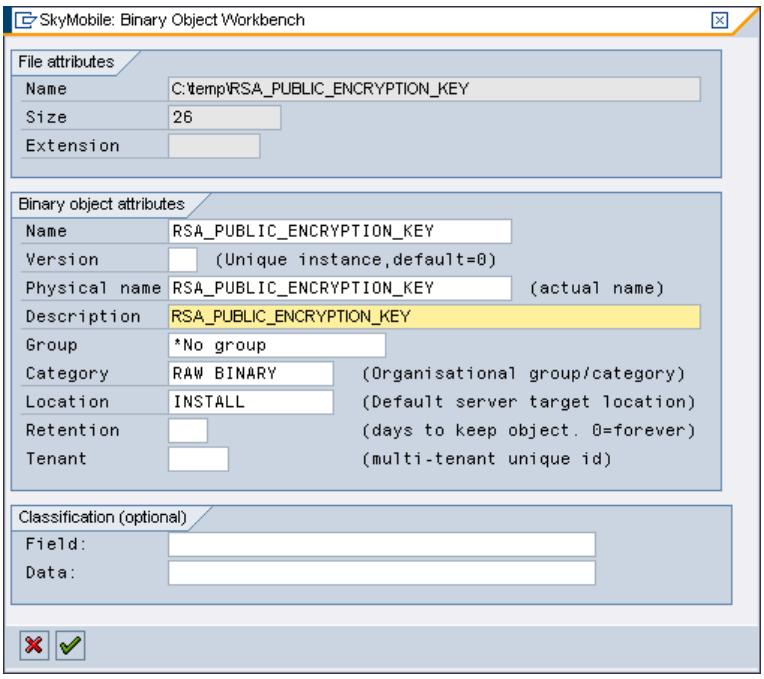


Select **Yes** on the confirmation.



Fill out the other attributes as required.

> ***Note:*** You can have a different name and description. It is a good idea to keep the names the same.



Repeat the process for the other key file, that is, both public and private. Once you upload the files, you may now wish to remove the generated files from your PC.

Once the key files are loaded into Kony for SAP binary repository, you may now associate them with an encryption object.

5.10.7 Configuring Access Gateway

The following configuration section implements an encrypted port into either a new or existing Access Gateway. In this example, port 22000 is exposed. The initial handshake is using RSA encryption with reference to the "rsa_private_encryption_key" file, as generated by Kony for SAP EKG utility. You must copy this file into the main `VTI` directory. After the initial handshake is performed using RSA asymmetric encryption, all subsequent calls are made using Blowfish symmetric encryption using a key strength of 2048.

rsa_private_encryption_key

```
SERVER.XMLGATEWAYPORT:PORT22000.COMPRESSIONALGORITHM = GZIP
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONALGORITHM = BLOWFISH
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYFILE =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYHEXENCODED =
SERVER.XMLGATEWAYPORT:PORT22000.ENCRYPTIONKEYSTRENGTH =
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONALGORITHM = RSA
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEY =
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEYFILE = rsa_
private_encryption_key
SERVER.XMLGATEWAYPORT:PORT22000.HANDSHAKEENCRYPTIONKEYHEXENCODED =
true
SERVER.XMLGATEWAYPORT:PORT22000.HOSTINTERFACE = SKQ
SERVER.XMLGATEWAYPORT:PORT22000.PINGCHECKIDLESECONDS =
SERVER.XMLGATEWAYPORT:PORT22000.PINGCHECKTIMEOUT =
SERVER.XMLGATEWAYPORT:PORT22000.PORT = 22000
SERVER.XMLGATEWAYPORT:PORT22000.USECOMPRESSION = true
SERVER.XMLGATEWAYPORT:PORT22000.USEENCRYPTION = true
SERVER.XMLGATEWAYPORT:PORT22000.XMLDEBUG = false
```

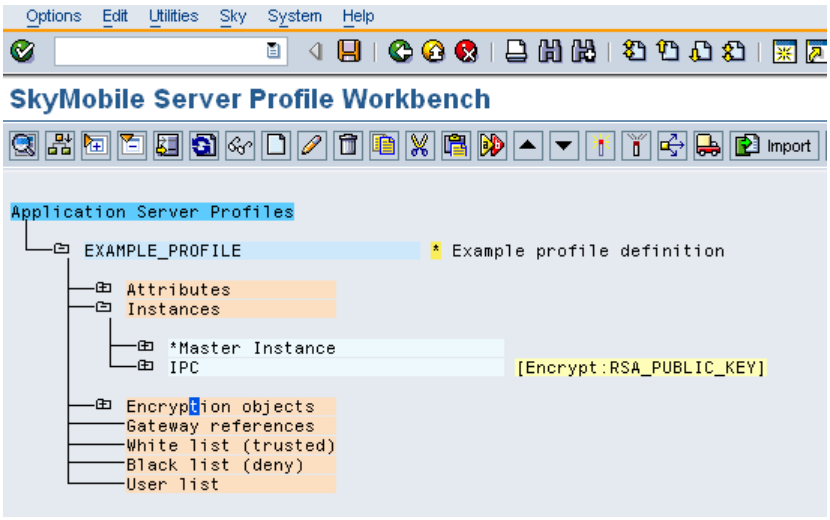
Once the Gateway restarts, the encrypted port is now available for Application Servers to connect to.

> ***Note:*** You may configure any port number.

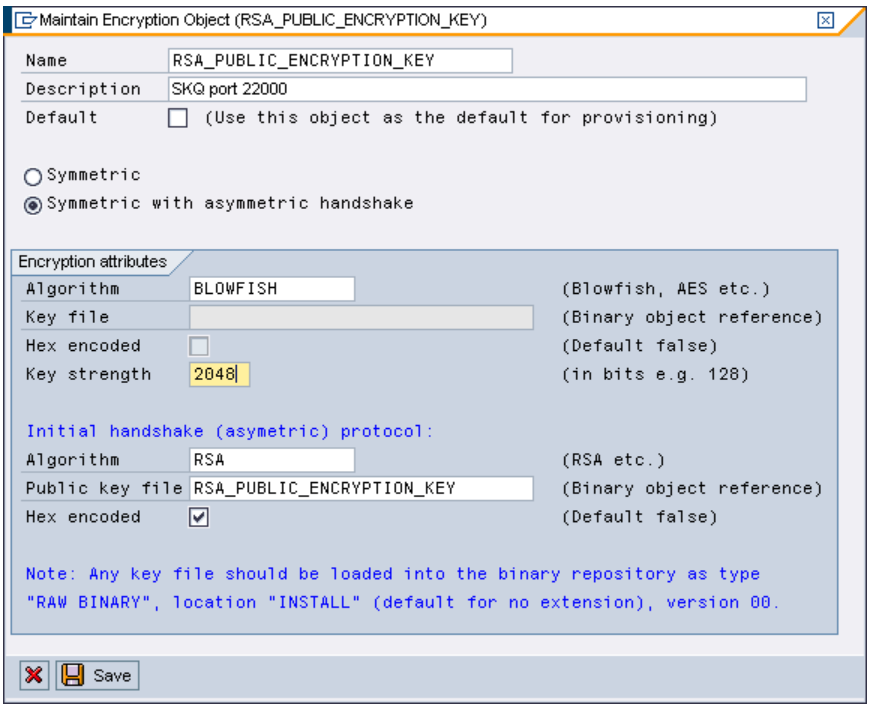
5.10.8 Creating the Encryption Object

> ***Note:*** The main discussion of how to use the Server Profiler is in the Integrated Development Guide (IDE).

This section assumes that a profile is already configured and just shows the creation of a white list entry for an iPhone device. In the following "example profile", position your cursor on the Encryption Object node and click the **Create** icon.



The following pop-up appears:



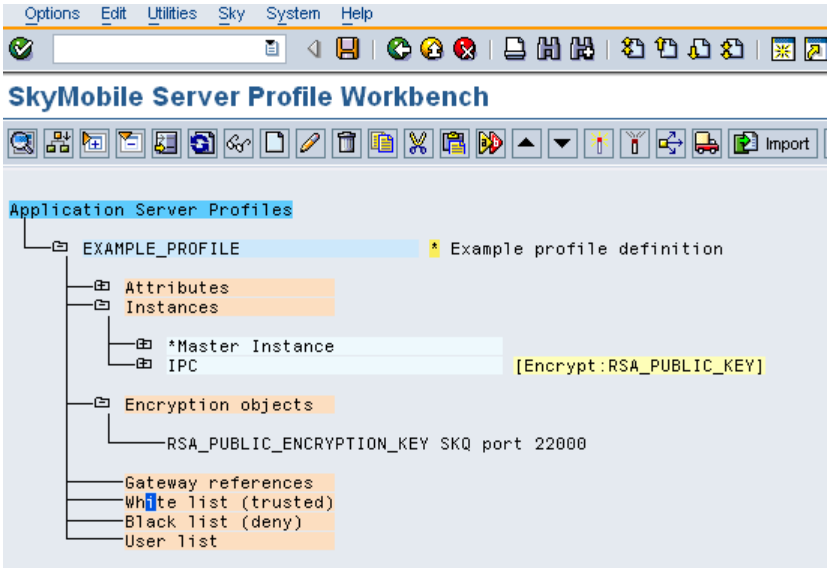
It is important to specify the same hex encoded option and key strength as the generated key files. The key file reference is the name of the uploaded object in the binary repository, that is, the binary object name.

5.10.9 Creating a White List Entry

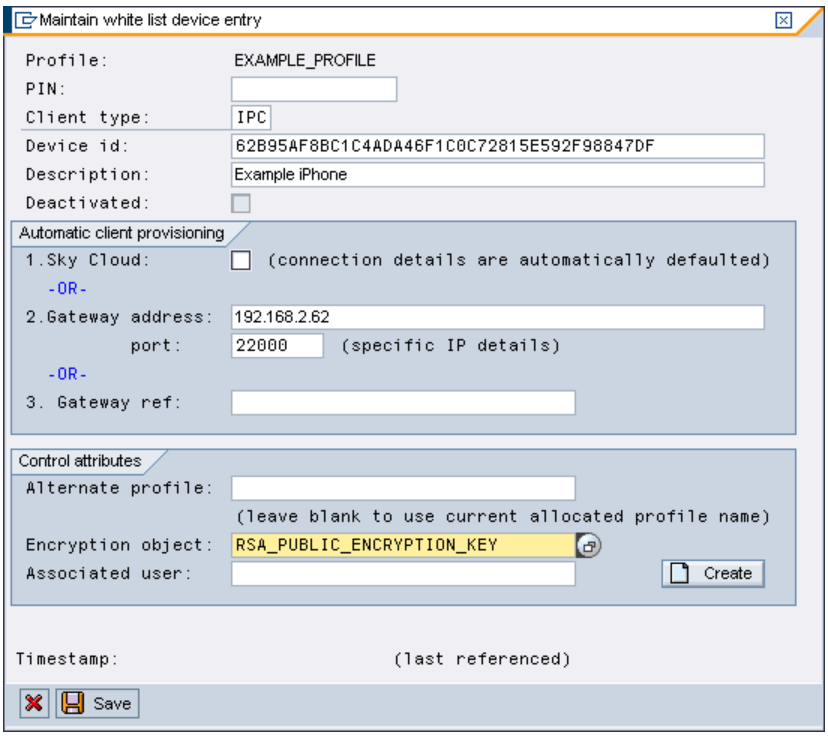
A white list entry identifies a unique device by its type and ID and provides automatic provisioning information, so that when the user starts the Application Server for the first time on the device, it knows which provisioning profile to use and which Access Gateway to connect to. Since white list entries are global and are not connected to a specific customer, they are defined centrally in the Sky Cloud and the Application Servers check here first as part of the initial provisioning process. You can associate an encryption object (as defined in the previous step) with the white list definition so that the Application Server can connect to the target gateway.

Contact Kony for SAP for details on maintaining white list entries in the Sky Cloud.

To create a white list entry in the Server Profiler, position your cursor on the White List node and click the **Create** icon.



The following pop-up appears:



The above white list entry is for iPhone UDID "62B95AF8BC1C4ADA46F1C0C72815E592F98847DF" and directs it to connect to an Access Gateway 192.168.2.62 port 22000. It also is associated with the encryption object "RSA_PUBLIC_ENCRYPTION_KEY" that automatically generates the correct encryption configuration and key in order to connect to the gateway port.

5.10.10 Provisioning the Device

Now that the gateway is configured and a server profile white list entry is configured for the device, everything is in place for an Application Server to automatically provision onto the device without the user performing any configuration.

The steps for this "white list" example are as follows:

- Download the Application Server "secure container" onto the iPhone from the iStore (see the installation guide for details on the options available)
- Once you download and install, start Kony for SAP
- The server connects to the Sky Cloud using client type IPC and UDID "62B95AF8BC1C4ADA46F1C0C72815E592F98847DF"
- An activated white list entry is located and the gateway and encryption details are passed back to the server
- The server connects to the SkyMobile Access Gateway port 22000 using RSA asymmetric encryption, that is, initial handshake
- A unique session key is generated by the application server
- The server communicates with the back end through gateway port 22000 using Blowfish symmetric encryption to download data, and synchronize application definitions.
- Once ready, the primary Kony for SAP application is launched (as configured) and control is passed to the user.

If you do not use the "white list" method, you need to perform a on premise provision, that is, the user need to specify the gateway IP address and port, and the profile to use. In this case, the encryption object is associated against the profile definition and you must make the first call to the back end MEAP server (on premise) through an un-encrypted port to retrieve its configuration. Once the server receives the configuration, it then automatically implements the encryption and directs all subsequent requests against the encrypted port on Access Gateway.

> ***Note:*** You ever use the un-encrypted port only for provisioning requests to the back end.

You cannot use this port for any other purpose and you may further secure by applying IP address ranges, relayed through a proxy server.

Mobile Infrastructure Hardening

Secure Communications

- Deploy proxy in DMZ
- Secure the communications between device and gateway using HTTPS
- Use client side certificate for device authentication with proxy.

Secure Storage

- Encrypt DB storage
- Use container provisioning exit to generate unique key per device.

Restrict Host Access

- Only permit RFC calls for devices with a current heartbeat.

Mobile Single Sign On

About

The section provides an overview of the single sign-on (SSO) options that SAP provides and describes how you can achieve SSO on mobile devices.

Key Topics

- Overview
- Prerequisites
- Configuration
- Authentication Process
- Connection Pool Management
- Security Considerations

Overview

SAP provides a number of options for SSO including:

- SSO from a Windows desktop into SAP ABAP systems (for example, ERP, CRM) by leveraging the Windows Active Directory Kerberos capabilities
- SSO from a Windows desktop into SAP web applications (for example, Portal) through SPNEGGO
- SSO from SAP Portal into SAP ABAP systems through SAP SSO tokens

Deploying SAP SSO can make life for end users simpler, in that they potentially only need to know they Windows AD credentials to access the SAP environments. But when it comes to deploying mobile applications it can make things more difficult. For example, with SSO from the desktop to ERP enabled, the user may not know their SAP User ID and password, and yet for audit reasons the organisation may require all transactions uploaded from mobile devices to post under the SAP User ID of the user who created them.

The SAP SSO Identity Module is designed to support this type of configuration. This identity service enables mobile device users to authenticate against an SAP Java system (for example, SAP Portal) and receive an SAP SSO token if successful. You can use this SAP SSO token to access SAP ABAP environments.

Click here to go back to the top.

Prerequisites

To use the SAP SSO Identity Module, you need the following items in place:

- An SAP Java environment (for example, Portal) available and configured to issue SAP SSO tokens
- The SAP ABAP system (s) (for example, ERP, CRM) configured to trust tokens issued by the SAP Java system
- Sky Gateway running v24.03.00 or later
- SkyMobile or SkySync v24.03.00 or later

Configuration

To make use of the SAP SSO Identity Module, you need to configure a number of parameters on the Sky Gateway. The key items are listed in the table below. For full details, refer to the configuration section.

Configuration Option	Purpose
PORTALHOST	The IP address of the SAP Java stack that performs the authentication and issue the SSO token
PORTALPORT	The IP port of the SAP Java stack
SERVICETYPE	SAPSSO

Example SAP SSO IdM Service Configuration

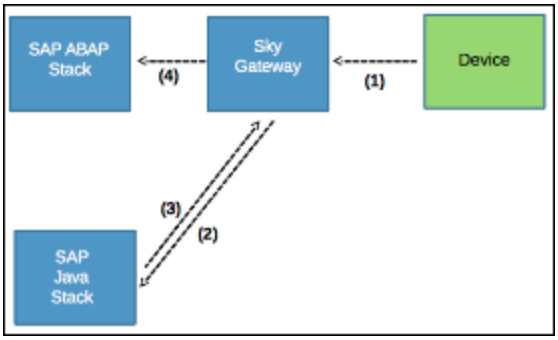
```
SERVER.IDENTITYSERVICE:SSO.HOSTINTERFACE = SKDRFCCLIENT
SERVER.IDENTITYSERVICE:SSO.PORTALHOST = 192.168.2.42
SERVER.IDENTITYSERVICE:SSO.PORTALPORT = 50000
SERVER.IDENTITYSERVICE:SSO.REGISTERSESSIONS = true
SERVER.IDENTITYSERVICE:SSO.SERVICETYPE = SAPSSO
```

Click here to go back to the top.

Authentication Process

The process flow for a user authenticating using the SAP SSO Identity Module is shown in the diagram below. The steps are:

1. The user starts the mobile application on the device and the device sends an identify message to the Sky Gateway server
2. The gateway passes this to the SAP SSO Identity Module that forwards the user credentials on to the configured SAP Java environment through HTTP(S).
3. If authentication is successful, the SAP Java environment returns an SAP SSO token.
4. The Sky gateway presents the SAP SSO token to the backend SAP ABAP environment to achieve a logon for the mobile device user.



Click here to go back to the top.

Connection Pool Management

Normally, the Sky Gateway uses a CPIC or Service user to establish a connection pool with the ABAP environment. This has the advantage of avoiding the performance overhead involved in logging on/off repeatedly. The downside is that the mobile device connections to SAP are all done under the one User ID (unless you use user switching functionality).

With the use of the SAP SSO Identity module, the Gateway still establishes a generic connection pool using the configured CPIC / Service user. When it receives an incoming transaction with an attached SSO token, it delegates a transaction handler thread to process the transaction. In order to do this, any existing connection as the default user closes and a new one opens as the user to whom the SSO token belongs. This does incur a small performance overhead that is the inevitable consequence of individualized connections (generic user connection pooling is always going to perform better, simply because there is less work involved in switching between users).

The Sky Gateway does incorporate sophisticated algorithms to minimize switching between users, by looking ahead at upcoming transactions in the queue and using analysis of these transactions to balance the users to whom its transaction handler threads are assigned. Non-active connections assigned to individual users are reassigned or closed after a short period of inactivity.

Click here to go back to the top.

Security Considerations

Once an SSO login takes place, any auditable transactions sent to the SAP system executes as the associated SAP user. Auditable transactions currently include the following:

- Download / refresh / upload of local database tables.
- Download / refresh / upload of data objects.
- Online (realtime) calls made directly against SAP from the application.

In order to execute as the associated SAP user, the SSO token is sent from the device to the Sky Gateway. The SSO token is embedded inside the XML that forms the message body and hence is compressed and / or encrypted in the same way as the rest of the message body.

There are security ramifications to using SSO token as you can use the SSO token as a means to access the SAP system outside the controls of the SkyMobile environment (albeit one with a limited lifespan), there are security ramifications to using it. You should always enable HTTPS and / or SkyMobile encryption in a production environment, but this becomes even more important when SSO is in use, to prevent the SSO token to pass between the device and the Sky Gateway in the clear.

To illustrate the nature of what is sent between the device and the Sky Gateway, some example messages are below.

The following example shows a local database table download request, without the use of SSO.

```
Example Download Request (No Compression/Encryption)
<?xml length="1308" ?>|
<SAPREQUEST FUNCTIONNAME="/SKY/VTI_DOWNLOAD_LOCAL_DATA"
STOREANDFORWARD="N">
<STRUCT NAME="I_HEADER" DIRECTION="IN">
... more XML ...
```

```
</STRUCT>
```

Example Download Request (With Compression/Encryption)

```
<?xmlcompressed algorithm="GZIP" encryption="BLOWFISH" length="405"
?>
... 405 bytes of encrypted/compressed binary data ...
```

The same download request looks somewhat different when SSO is in use, as shown below.

Example Download Request With SSO (No Compression/Encryption)

```
<?xml length="2802"
authsig="FD1DE7456020C2018E0C13A856D38C0FF47593CB83C988FEBA8D6BE76C9
57AA3" ?>

<SAPREQUEST FUNCTIONNAME="/SKY/VTI_DOWNLOAD_LOCAL_DATA"
SSO_
```

```
TOKEN="AjExMDAgABFwb3J0YWw6YW5keS5zb3V0aIgAB2RlZmF1bHQBAApBTkRZLlNPV
VRIAgADMDAwAwADQVNKBAAMMjAxMzA1MzAwMDI
1BwAEAAAACgoACkFORFkuU09VVEj/AQUwggEBBgkqhkiG9w0BBwKggfMwgfA
CAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHATGB0DCBzQIBATAiMB0xDDAKBgNVBAM
TA0FTSjENMAsGA1UECxMESjJFRQIBADAJBgU
rDgMCGgUAoF0wGAYJKoZIhvcNAQkDMQsGCSqGSIb3DQEHATAcBgkqhkiG9w0BCQUxDxc
NMTMwNTMwMDAy
NTMzWjAjBgkqhkiG9w0BCQQxFgQU5eM!Iij31zARo6FF8j4QzG2snBkwCQYHKoZIzjgE
AwQvMC0CFEpKh3kLbxLpw28QKWDViaP7FMLUAhUAu9PoT0vv33
1U1oCKUsgsx0sOjA0=" STOREANDFORWARD="N">

<STRUCT NAME="I_HEADER" DIRECTION="IN">

... more XML ...

</STRUCT>
```

Example Download Request With SSO (With Compression / Encryption)

```
<?xmlcompressed algorithm="GZIP" encryption="BLOWFISH" length="810"
authsig="FD1DE7456020C2018E0C13A856D38C0FF47593CB83C988FEBA8D6BE76C
957AA3" ?> ... 810 bytes of encrypted/compressed binary data ...
```

Important: **Authorization Signature:** The "authsig" (authorization signature) attribute is used in the XML header when you use SSO. This is a "fingerprint" of the user's login credentials (that is, the SSO token). The Sky Gateway uses it to efficiently optimize allocation of connection pool resources, without the need to immediately decrypt, decompress, and parse all incoming transactions. The authorization signature is a salted SHA256 hash of the user login credentials and you can publicly expose without presenting any real security risk. It has no bearing on the actual processing of the transaction other than to assist the gateway in its management of the connection pool. Actual sign-on always takes place using the SSO token embedded in the message, and the authorization signature is not used as part of this process.

5.11 Utilities

As well as the standard built-in functions, a number of optional utilities are provided that you may use to help implement particular functionality or resolve issues.

5.11.1 Key Topics

Global Command Processor
The Switchboard Facility
Printer Interfaces
Device Interfaces

5.11.2 Global Command Processor

The global command processor enables commands to direct to Java Servers and Gateway processes on the same machine. The primary advantage of GCP is to abstract the SAP system management processes from having to know the exact directory structure or port configuration of the servers and gateways. It is also useful to enable commands to issue from the command line. You can also use GCP to manage multiple instances on Java Servers on either UNIX or Windows servers.

Example call to GCP from the Unix command line

```
CD …/<<SKYHOME>>
./gcp.sh VTI status
```

Example call to GCP from a Windows DOS command window

```
CD …\Sky Technologies\v15.00.00\GCP
gcp.bat VTI status
```

> ***Note:*** See the starting, stopping, restarting and monitoring of VTI Java servers earlier in this section for examples of issuing GCP commands. Since GCP uses the VTI Java server command port, all command port options are available. See "The VTI Java server command port" earlier in this section for details.

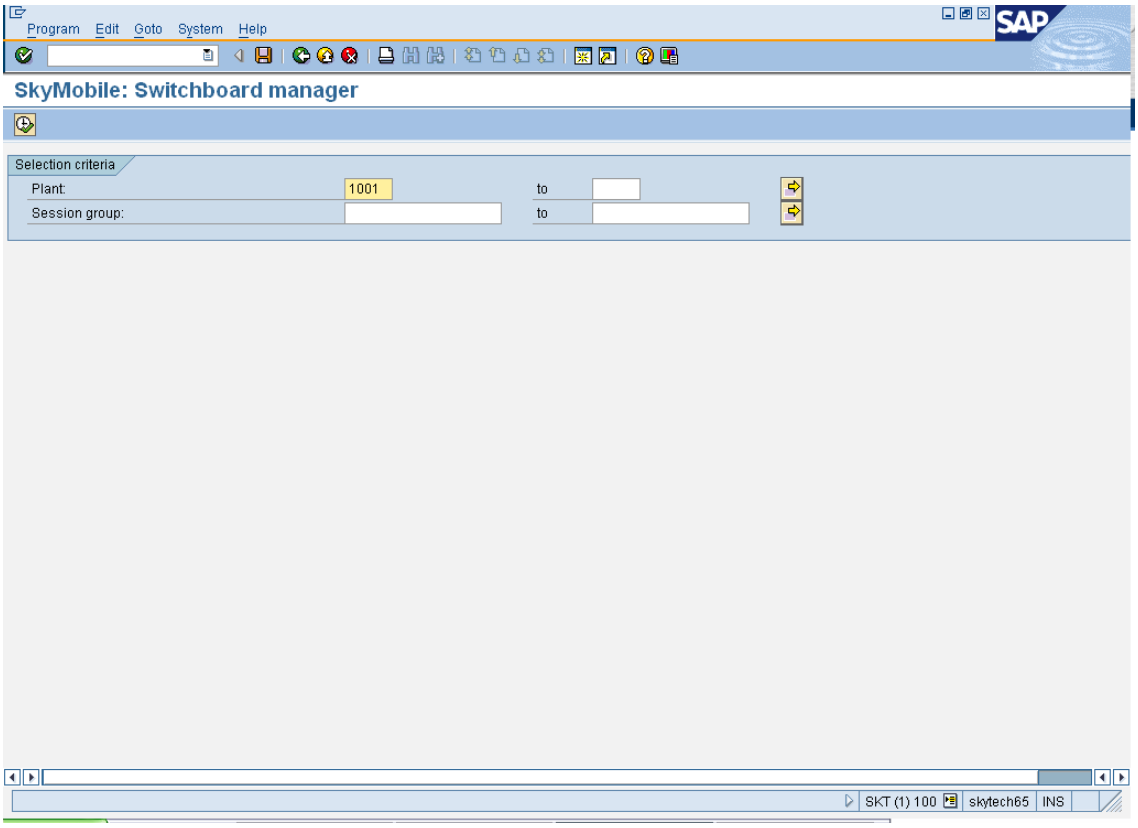
5.11.3 The Switchboard Facility

The switchboard facility is only available for direct SAP sessions, that is, are not currently supported for local stand alone sessions. The switchboard allows you to save screens, or a position in an application, for later use. This allows operators to effectively switch from one function to another without having to navigate through menus. For example, an operator who performs receipting, put away and delivery picking can have all three screens active and just switch between them, without going through the menus.

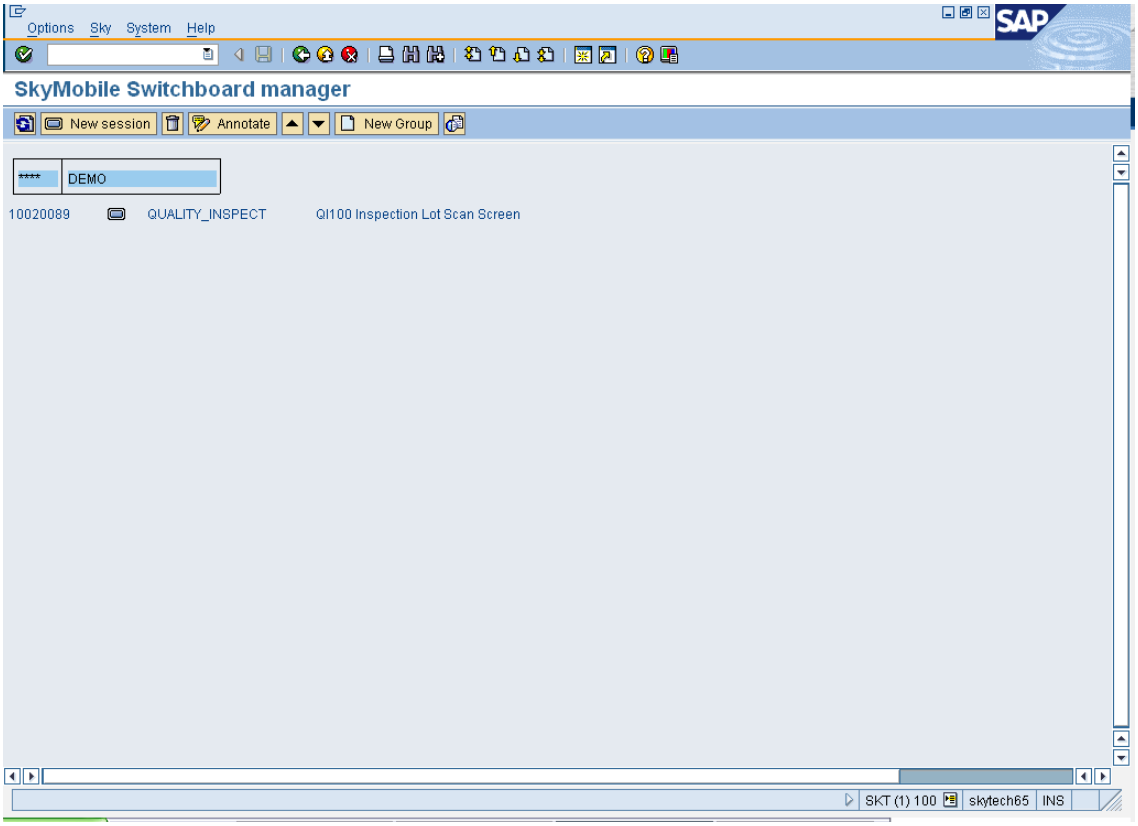
You can create Switchboard sessions either directly in SAP using the switchboard manager, or through special internal VTI function keys assigned to the application.

5.11.3.1 The SAP Switchboard Manager

You start the switchboard manager from the VTI SAP administration console (YVTI). The option is available from **Utilities** > **Switchboard Manager**. The following selection screen appears:



You may select all switchboard sessions for a specific plant or session group. The session group is assigned when a switchboard session is created. The main switchboard screen appears, showing all selected switchboard sessions within session group.



You can perform the following tasks from this screen:

- Create a new switchboard session in the current group. If there is no switchboard group, you are prompted for a new session group. If you wish to create a new group, use the new group function.
- Delete a switchboard session.
- Annotate a switchboard session, that is, describe its usage.
- Create a standard text template to automatically generate annotation text for a session function with substitution variables. This is so switchboard users can see what the session is for.
- Create a new session group.

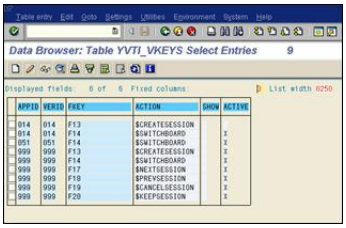
- Maintain SAP emulation options, that is, specify a new application.
- Re-start a switchboard session.

> ***Important:*** Each session line has a 'terminal' icon, that is, a hotspot. If you click this, the switchboard session restarts. Similarly, you may double-click the session line.

> ***Note:*** To maintain the switchboard session context, click cancel or close the pop-up. If the user logs off, the switchboard session is terminated.

5.11.3.2 Enabling Switchboard Functionality from the User Interface

To perform switchboard functionality from the user interface, you must populate the yvti_vkeys (internal VTI function key action list) with the function key and special switchboard action. You then use these function keys in the screen to process switchboard functions. You must specify the function key combinations for every application version combination. Use se16 to change the table values. You must also install the internal application '999' (defaulted with the SkyMobile installation).

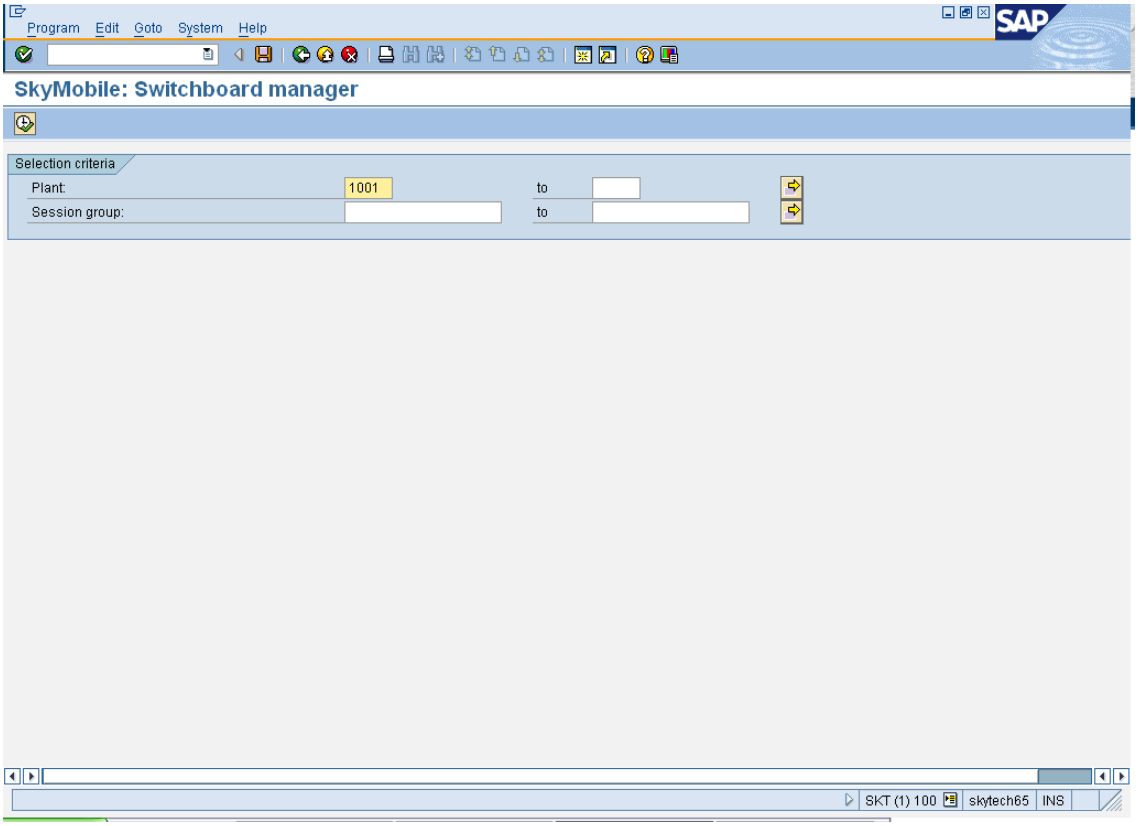


> ***Note:*** The SkyMobile internal function keys are included in each screen function, by default. If you wish to suppress them, check the suppress internal function keys check box on the SAP emulator settings. You should not allocate these internal key settings to any 'normal' application keys, else the application key takes preference.

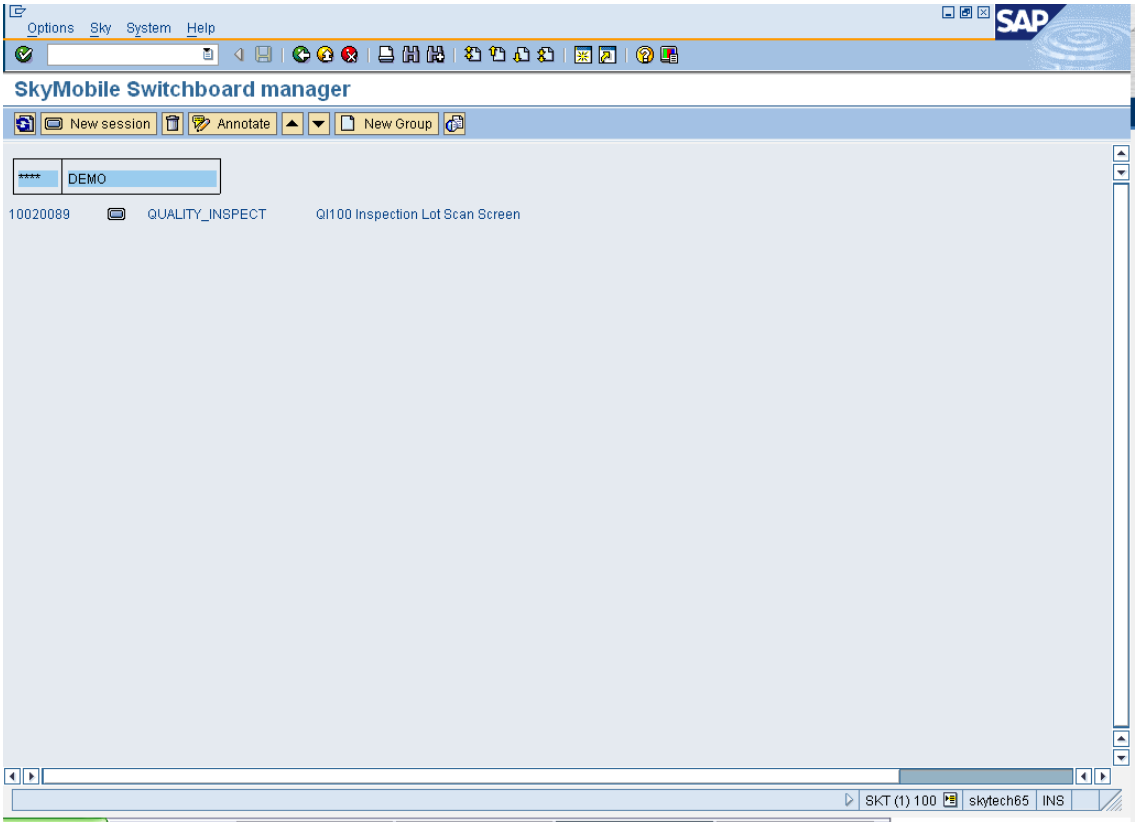
Once this is in place, you may execute the switchboard functionality directly from the user interface. You may change the default switchboard screens, by changing application 999 version 999.

5.11.3.3 The SAP Switchboard Manager

You start the switchboard manager from the VTI SAP administration console (YVTI). The option is available from **Utilities** > **Switchboard Manager**. The following selection screen appears:



You may select all switchboard sessions for a specific plant or session group. The session group is assigned when a switchboard session is created. The main switchboard screen appears, showing all selected switchboard sessions within session group.



You can perform the following tasks from this screen:

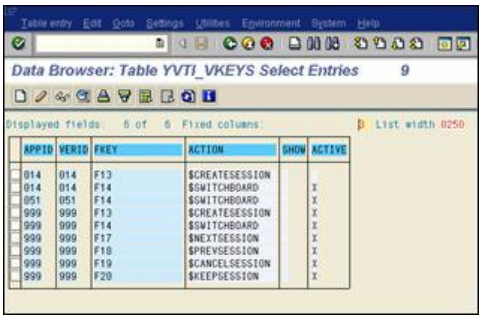
- Create a new switchboard session in the current group. If there is no switchboard group, you are prompted for a new session group. If you wish to create a new group, use the new group function.
- Delete a switchboard session.
- Annotate a switchboard session, that is, describe its usage.
- Create a standard text template to automatically generate annotation text for a session function with substitution variables. This is so switchboard users can see what the session is for.
- Create a new session group.
- Maintain SAP emulation options, that is, specify a new application.
- Re-start a switchboard session.

Note: Each session line has a 'terminal' icon, that is a hotspot. If you click this, the switchboard session is restarted. Similarly, you may double-click the session line.

Important: To maintain the switchboard session context, click **Cancel** or close the pop-up. If the user logs off, the switchboard session terminates.

5.11.3.4 Enabling Switchboard Functionality from the User Interface

To perform switchboard functionality from the user interface, you must populate the yvti_vkeys (internal VTI function key action list) with the function key and special switchboard action. These function keys are then used in the screen to process switchboard functions. You must specify the function key combinations for every application version combination. Use se16 to change the table values. You must also install the internal application '999' (defaulted with the SkyMobile installation).



> ***Important:*** The SkyMobile internal function keys are included in each screen function, by default. If you wish to suppress them, check the suppress internal function keys check box on the SAP emulator settings. You should not allocate these internal key settings to any 'normal' application keys, else the application key takes preference.

Once this is in place, you may execute the switchboard functionality directly from the user interface. You may change the default switchboard screens, by changing application 999 version 999.

5.11.4 Printer Interfaces

Kony for SAP has two printer interfaces that can help automate the production and printing of labels, invoices, and receipts. Common functions are provided in SAP that may be called by ABAP programs to load print template, substitute values and direct the result to a printer. Alternatively, the Secure Container has a built-in printer interface that you may use to produce barcode labels, and invoices. The configuration for this is in the Secure Container configuration file (`skymobile.cfg`). See the options in the Java Server configuration section for details.

5.11.4.1 Print Template

The basic principle is to read in a printer template from a designated directory. A java exit then uses standard APIs to substitute values and issue an external command to direct the result print file to either a local or networked printer. The Kony for SAP Java API contains all the necessary calls and considerably simplifies the programming involved. For examples of programming printer interfaces, see the Java programming guide for details.

Example Configuration

```
[PrintTemplate:SamplePrintTemplate]
TemplateFileName = template.txt
WorkDirectory = c:\temp
ShellCommand = cmd.exe /c
PrintCommand = type &filepath& > LPT1
PrintCommandTimeout = 10
LogPrintCommands = true
DeleteWorkFiles = false
WorkFileExpiryDays = 1
```

5.11.4.2 Template File

The template contains the layout of the report or label with substitution variables (place holders), that is, where values are to be placed by the Java exit and printer control sequences for graphicstc. Most printers supply their own specific template layout generators, there are third party packages or you may manually construct them.

5.11.4.3 Print Command

The command used depends on the operating system used and whether the printer is local or networked. The following table shows some examples:

Windows - parallel printer	type &filepath& > LPT1
Windows - network printer	print /D:{print share} &filepath&
Unix	lp --d {print definition} &filepath&

5.11.4.4 The SAP Printer Interface

This interface constitutes of a suite of common functions that are invoked by custom ABAP programs. You must optionally import the YVTP development class to enable this functionality. This development class is not implemented as part of the standard SAP add-in and you must request separately from Kony for SAP.

5.11.5 Device Interfaces

The Java programming SDK supplies a complete set of classes to interface directly with serially connected and TCP / IP aware devices. This enables the Java Server to interact directly with peripheral devices such as: card scanners, gantry systems, weighbridges, text display systems, and specialised printers. For examples of programming device interfaces, see the Java programming guide for details.

6. Interfacing

6.1 About

Kony for SAP provides a comprehensive suite of tools and utilities to effectively monitor and manage interfaces, periodic processing and background transactions in SAP. It provides a single point from which to organize, monitor, document and analyze all background processing. The key areas of functionality are:

Process Control

In this context, process control is the ability to define the sequence, rules and dependencies to execute programs in background. It also allows the structuring of individual technical executions into logical business processes.

Monitoring

SAP provides basic tools to check the status and output of job steps and IDOCs. It does not provide tools to monitor in the context of a logical process or show the status of all background processing "real time" or at a given point in time. Kony for SAP provides a central point to monitor all background processing in either "real time" or exception reporting modes.

Notification

Kony for SAP provides functionality to notify users and external systems of any failure or completion using a variety of methods, for example, Internet, SAP mail, Workflow, external monitor (HP Openview / CA-Unicentre). Kony for SAP also provides controls to restrict the number of notifications issued within a specified time frame; and an escalation procedure, should a notification fail to acknowledge in a timely fashion.

Statistics

Kony for SAP keeps statistics on successful and failed runs and provides options to download these statistics to MS-Excel and /or MS-Access for analysis reporting. Comprehensive trend analysis is possible on the frequency, volume and duration of background processing. SAP provides some basic data on job and IDOC execution, but has no context of a logical business process and no easy way to summarize and report the data. Kony for SAP provides extensive trend reporting to quickly highlight problem areas and map the overall efficiency of the system.

Data management

SAP provides no mechanisms to effectively manage external files or track interface data. Kony for SAP provides a range of functionality to manage files and data passed into and out of interfaces.

Scheduling

SAP provides basic scheduling functionality, but does not have the ability to define complex multilevel dependencies. It also lacks some basic scheduling options such as 'last day of the month'. There is also no easy way to view the overall periodic processing schedule in terms of historical, current and future. The ECS scheduling option provides the following functionality:

- Single point to view the status of all periodic processing
- Simple or complex multi-level scheduling
- Full integration with the SAP calendar and SAP job management system
- Comprehensive scheduling functions
- Ability to configure exclusions, for example, run every day except Saturdays, only run between 8 am-6 pm.
- Calendar / chart reporting
- Ability to transport and compare schedule definitions across SAP environments

System Management

Kony for SAP provides facilities to shut down and restart background processing in a controlled fashion. When you shut down SAP, any active background tasks are left in a 'zombie' state, that is, flagged as active but are non-existent. The Kony for SAP system management allows to defer any started tasks until the system is available once more.

Online Documentation

SAP provides some basic facilities to document background processes, these use the sap script editor and are generally unsuitable for technical and end-user documentation. In addition, there is no mechanism to document a logical group of tasks such as a business process or a schedule. Kony for SAP provides a facility to document background processes and schedules using an external word processor such as MS-Word. ECS also provides a mechanism to annotate runs of a background process. There is no facility to annotate background transactions in SAP.

Restart/Recovery

Kony for SAP provides a standard framework to define all the rules for the restart and recovery of background processes. This enables business owners to manage interface and periodic processing and not by IS. Kony for SAP provides facilities to successfully restart background processing after failure and provides an audit trail of all processing.

High Volume Background Performance

Large number of background tasks (such as jobs, BDC sessions and IDOCs) can cause severe performance problems. SAP provides no alternative to the problem and the solution is usually a complete re-design or to upgrade the hardware. Kony for SAP provides a facility, the ECS background server, that can dramatically speed up processing of high volume background transactions through intelligent workload balancing and execution techniques.

6.2 Key Topics

Interfacing Overview
Interface Workbench
Monitoring

6.3 Overview of Interfacing

6.3.1 About

This section describes the main concepts of SkyConnect and the terminology that you use throughout this manual. Once a basic understanding of the architecture of SkyConnect and how its process control works is attained, it becomes evident how logical and essential SkyConnect is to effectively control all background processing.

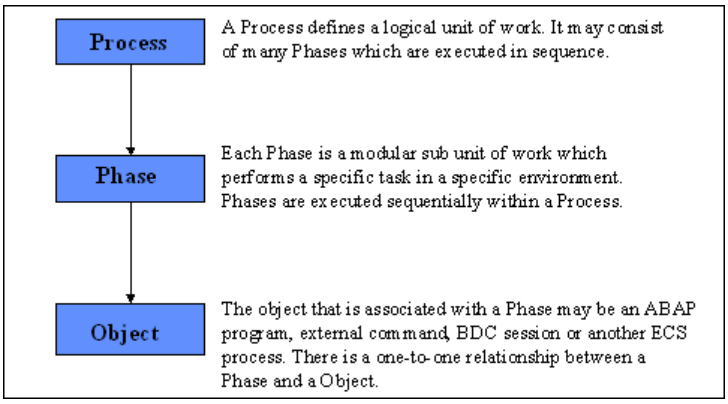
6.3.2 Key Topics

Concepts and Terminology
Navigation
Defaults and Preferences
Sample Processes
Security

6.3.3 Concepts and Terminology

6.3.3.1 Processes, Phases and Objects

A process is a logical grouping of executions (Phases) that together make up a single logical unit of work (run). The executions can be disparate (that is, internal to SAP R/3, external to SAP R/3), utilize different techniques (that is, custom ABAP program, and SAP R/3 standard programs) that collectively constitute a business process. ECS provides all the controls necessary to effectively execute, monitor and restart interfaces and periodic processing. You use the ECS Workbench is to define Process and Phase definitions.



6.3.3.2 Execution of a Process

You may initiate an ECS Process in many ways, explicitly from within a program, manually using the **Start Process** option in ECS, through SAP R/3 event, and a file arriving in a directory. Execution always starts with the 1st Phase in the sequence and then subsequent phases. You may assign run time attributes to the Process definition that govern how it is executed and how you monitor and control the Phases.

6.3.3.3 Execution of Phases within a Process

Phases are executed in a sequential fashion. You may explicitly start phases 'programmatically' or automatically by ECS when the previous Phase is completed. In the ideal case, Phases are started and ended explicitly by calls to ECS within the executable program or script. This gives the programmer full control over start and end phases with specific parameter, enqueue and interval

options. You may also configure phases to kick off automatically once the previous phase successfully completes. For example, once phase-1 generates the BDC session, start phase-2 to process it.

6.3.3.4 Groups

Group definitions are optional and you use to categorize ECS Processes and Schedules for easy identification. Groups are useful to restrict reporting criteria. You may also use groups to secure areas, for example, Basis, Accounts Payable, and Accounts Receivable. You may only assign a Process or Schedule to one group.

6.3.3.5 Filter fields and values

It is possible to further categorize Process "Phase" runs at a low level (for example, plant). You may assign filter fields and populate at run time. These are useful to segregate Process based on the transaction data, for example, display all the warehouse management transactions for plant 1234. They are also useful to direct notifications to specific users or groups of users, for example, if material movement fails for plant 1234, notify John Smith.

6.3.3.6 Queuing Controls

ECS provides many controls to effectively manage concurrent processing and avoid failures due to the exclusive control of SAP resources. You may effectively serialize processing with logical locking and configure automatic checks to prevent accidental executions.

6.3.3.7 Serialization

The ECS enqueue manager provides locking mechanisms that 'single thread', the execution of ECS Processes and Phases. This is necessary to prevent the failure of conflicting programs over the same resource. For example:

- Only one task may update a materials stock within the same plant at a time, therefore we need to check whether the plant / material is updated. If it is, you suspend the execution of the Phase until the update is completed. In-order to detect that the plant and material is currently enqueued (locked), ECS publishes the enqueue data provided by the program and checks to see if it is currently 'owned' by another task. Thus an enqueue is effective across all active Phases. In contrast to enqueue, serialization only affects a specific Process definition. It may be necessary

to 'single thread' an entire Process. that is, all other Process runs are suspended while there is one active. ECS release subsequent runs, once active runs complete.

- In order to avoid SAP 'resource locking' failures, an interface to upload bill of material and routing data is serialized in ECS so that only one Process may execute at a time.

6.3.3.8 Single Instance and Frequency Controls

It is often required to ignore the execution of a process if one already runs, or abort the execution if a run already occurred. For example:

- We do not want to start another invoicing run if one already executes, because the data is picked up and processed by the current run. You want to ignore the submission of another task.
- Do not allow the monthly material roll to proceed if one already run today.

ECS enables these controls to configure for Process definitions. You may need the controls to prevent duplication and accidental submission of interfaces and / or periodic processing.

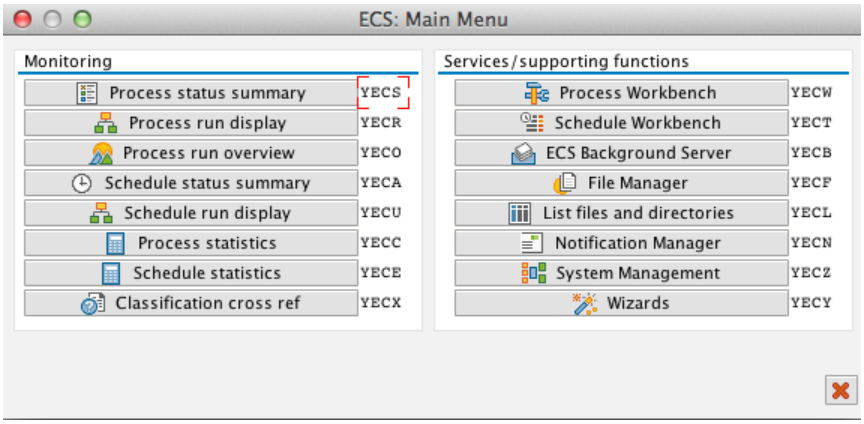
Click here to go back to the top.

6.3.4 Navigation

The ECS online system functionality is divided into 15 functional areas, as listed in the short cut transaction codes. Transfer to any of these areas through the short cut transaction or the goto menu on any of the main screens. A main menu is provided that you may start through transaction YECM .

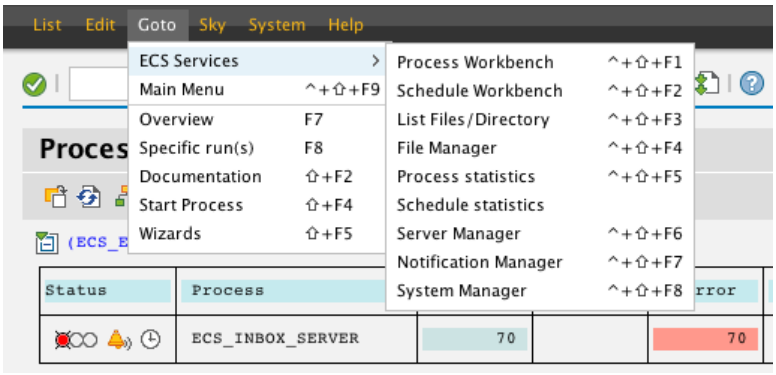
6.3.4.1 Main Menu

The ECS main menu appears using transaction YECM. This transaction is multi-functional and is used to manage and monitor all defined processes within the system. You may also use multiple shortcut transactions to go directly to the area of interest.



6.3.4.2 GOTO Menu

All the main ECS screens have a GOTO menu that is useful to branch from the current function to other areas of the system.



6.3.4.3 Transaction Codes

Code	Description
YECB	ECS background server manager
YECO	Process run overview
YECC	Process statistics
YECR	Process run selection
YECE	Schedule statistics
YECS	Process status summary
YECF	File manager
YECT	Schedule manager
YECI	Check Installation
YECW	Process workbench
YECL	List directory and file contents
YECX	Switchboard
YECM	Main menu
YECY	Wizards
YECN	Notification manager
YECZ	Systems management

Click here to go back to the top.

6.3.5 Defaults and Preferences

6.3.5.1 System Defaults

System defaults may be explicitly overridden by user defaults. You must set up the constants as external with reference to the CORE component.

Command	Flag
YECS_SAPGUI_MESSAGES	Display ECS sapgui status messages YES/NO

6.3.5.2 User Defaults

You may configure user specific default values and certain ECS preferences by way of SAP PID (**System** -> **User profile** -> **Own data** -> **Parameters** menu option).

Command	Description
YECS_GROUP	Default ECS group
YECS_FILTER_FIELD	Default ECS filter field name
YECS_FILTER_VALUE	Default ECS filter field value
YECS_SAPGUI_MESSAGES	Display ECS sapgui status messages YES / NO

Click here to go back to the top.

6.3.6 Sample Processes

ECS comes with a suite of sample processes that test that the basic functionality is working correctly.

- Logon to the Client to test
- Execute transaction YECS to display process statuses
- Specify group ECS
- Select the **Show ECS samples** check box on the selection screen
- Execute the selection and the main display process status screen appears
- Start the ECS sample process that is tested by placing the cursor on the name and clicking start
- A start process pop-up box appears,click **Execute** to continue
- You should submit a run to execute that should result in a green light.

Test the following sample processes to ensure that the basic functionality is operational:

- ECS_TEST_ABAP
- ECS_TEST_BDC
- ECS_TEST_EVENT
- ECS_TEST_SERVER_BDC

If all of these processes complete with a green light, ECS works correctly.

The **check install** option (transaction YECI) automatically starts four ECS sample test processes to test that the basic ECS functionality works correctly.

6.3.7 Security

6.3.7.1 About

The security of applications and data is important to most organizations when deploying mobile devices that store data locally on the device and transmit data across wireless networks.

This section describes the various mechanisms available to secure the infrastructure and mobile applications in general.

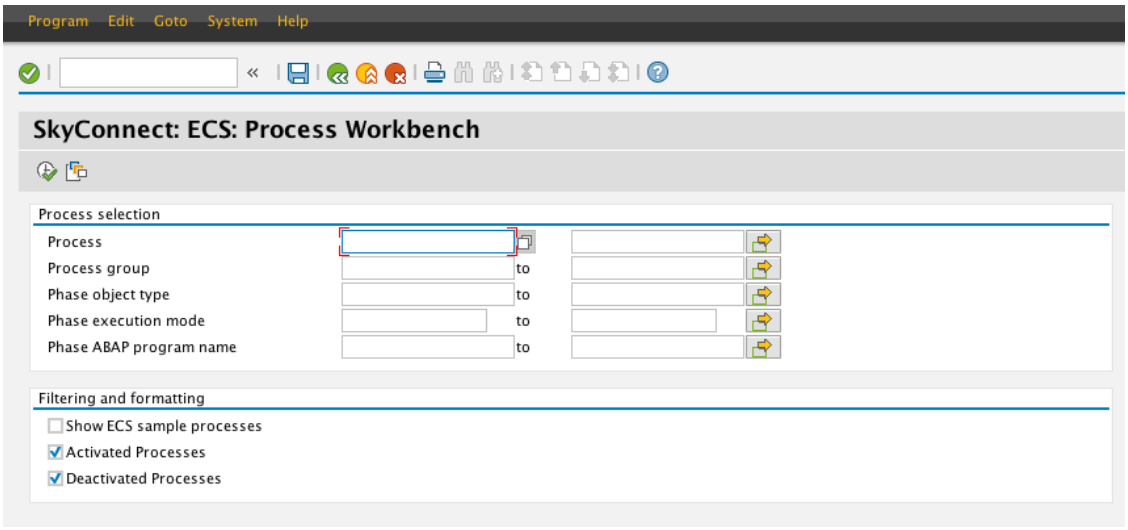
6.3.7.2 Key Topics

The Secure Container Concept
Identity Management
Infrastructure Level Options
Application Level Options
Implementing Network Encryption
Mobile Infrastructure Hardening
Mobile Single Sign On

6.4 The Interface Workbench

6.4.1 About

In ECS terms, process control is the ability to define the sequence, rules and dependencies to execute tasks in the background. ECS provides a comprehensive online workbench with which you can define Process definitions and their dependencies. Schedule definitions are created through the Schedule manager. The ECS workbench is used to create and maintain ECS process and phase definitions. The workbench has an intuitive interface that lists the processes and phases using a tree structure. The phase node (+) can be clicked on to explode all the related phases.



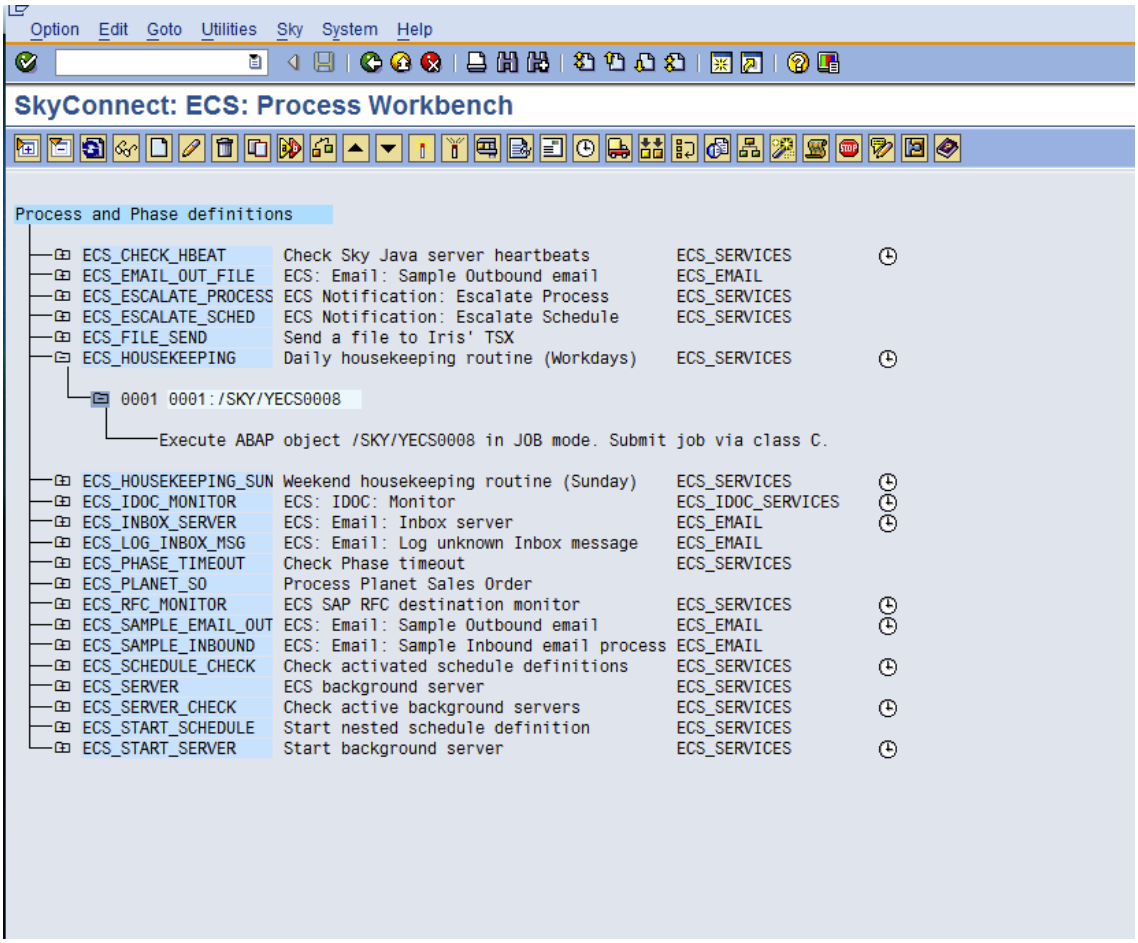
To start the workbench, execute the YECW transaction or select the **Workbench** option from the **GOTO** menu on any of the main screens. You may restrict the processes listed at a time by entering a pattern, for example, Z* or a range. Additionally, you may restrict the display to a specific group. Once you enter the selection criteria for the list (if any), execute the report and the Workbench screen appears.

6.4.2 Key Topics

Interface Workbench
Maintaining Processes
Maintaining Phases
Dependency Controls
Wizards

6.4.3 The Interface Workbench Screen

The ECS Process definitions appear in a hierarchical tree display. You may expand the tree nodes or collapse by clicking the "+" or "-" boxes. There are also icons on the application toolbar to choose a specific process, permanently expand a definition, and refresh the display. The user interface is very much 'position cursor' and click an option.



You may maintain definitions by positioning the cursor on a node and executing an application toolbar option. For instance:

- To create a new Process, position the cursor on the top level node "Process and phase definitions" and click the create icon in the application toolbar.

- To maintain a Process, position the cursor on the process and click the change icon in the application toolbar.
- To create a Phase definition, position the cursor on the position where you want to insert the phase and click the create icon in the application toolbar. This may be on the Process name if you want to insert the phase as the first Phase, or on an existing Phase to insert 'after'.

6.4.3.1 From this screen

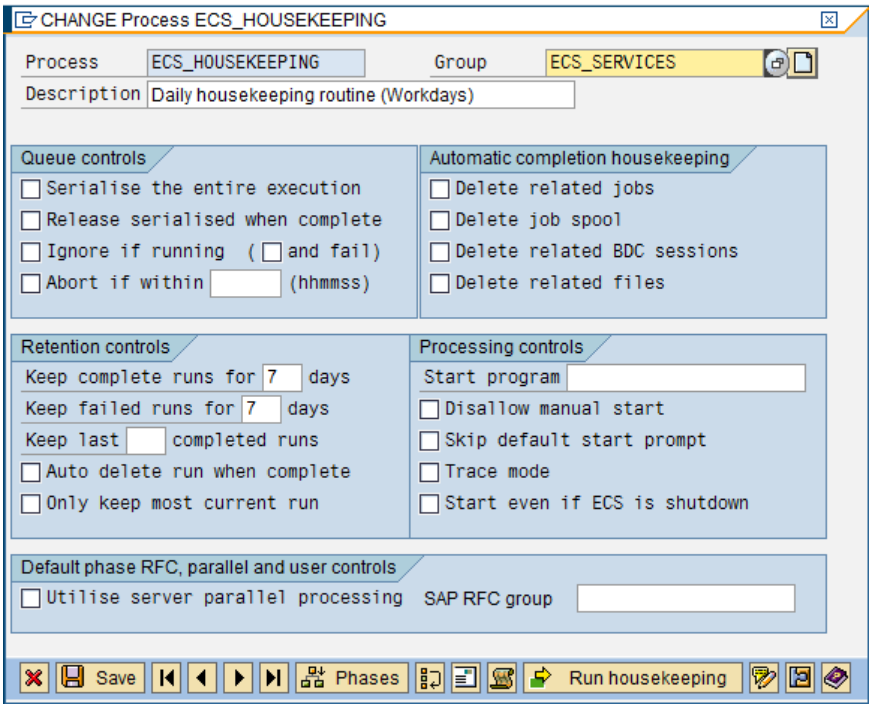
- Create and maintain Process and Phase definitions
- Enable or Disable Process definitions
- Start a Process
- Maintain Process documentation
- Maintain Process notifications
- Maintain Schedule definitions
- Execute utilities, such as transport and compare
- Configure Phase: condition code and message evaluation

6.4.3.2 Phase Sequence

The sequence of Phases is important within a Process. The order they appear in is the order they are executed in. See the section " Creating and maintaining Phase definitions" for more details. You can use the up and down arrows to dynamically promote or demote Phases within a Process.

6.4.4 Maintaining Process Definitions

The same screen is used to either create or change an ECS process definition. To create a process, position the cursor on the first node of the tree, or on another process, and click the **Create** icon. To change a process, position the cursor on the process requiring change and click the **Change** icon. The following pop-up screen appears:



Field	Description
Process Name	Unique name for the process. The process name must be unique across the ECS system. This is disabled in change mode. ***Note:*** Process names starting with ECS are reserved for internal Sky use and you may overwrite on upgrade.
Group	You may allocate the process to a group. You are recommended this to help organize definitions and restrict displays through selection criteria.

Field	Description
Description	A meaningful description of the process.
Queue Controls	• Serialize the entire execution: This option instructs ECS to single thread all executions of the process, so that only one may run at a time. The release next serialized run option instructs ECS to only release the next serialized run for execution when the current run completes successfully, that is, the next run is still held if the current run fails. • Release serialized when complete: is additional to "Serialize the entire execution". If you select this check box, the next process run is only released when the current process completed successfully. • Ignore run if one already running: If an instance of the process is already in-progress, the start of a new process run is ignored. This option is useful to save system resources when "generic" type batch programs are submitted. • Abort if run within hhmmss: Fail the run if an instance of the process has already run within the configured timeframe (hhmmss). This option is useful to prevent duplications of a run.

Field	Description
Automatic Completion Housekeeping	These options are initiated when the entire process completes. By complete, you mean completes on its own or force complete after a failure. The delete options are mainly used to automate housekeeping in situations where there is a large volume of background job processing. This helps to keep the size of the SAP database to a minimum. • Delete related jobs: all asynchronous jobs submitted on behalf of the user and the ECS monitoring jobs are deleted from SAP R / 3. • Delete job spool: ECS deletes any spool definitions associated with the run. • Delete BDC sessions: any BDC sessions associated with the run are deleted. • Delete related files: any input or output files referenced by any of the Phases are deleted. Use this option with care and only in situations where the files are not required or successfully archived.

Field	Description
Retention Controls	• Keep completed runs for x days: Configure the number of days that completed runs is kept in the system before housekeeping removes them. • Keep failed runs for x days: The maximum number of days failed runs are kept before the ECS housekeeping automatically 'force completed'. This option is optional, the default is to keep them indefinitely, or as long as defined in the system defaults. • Keep last x completed runs: retains only the latest x completed runs for a process. The others are automatically deleted. This option is useful to restrict displays for frequent / high volume runs. • Auto delete run when complete: ECS automatically deletes the run definition when it completes successfully. Only use this option in situations where the run data is never referenced, that is, no audit trail is necessary. • Only keep most current run: Only one run, the latest, is retained. Any others are automatically deleted. This option is useful in situations where only the latest run is relevant, for example, a monitoring task that re-evaluates the criteria each time it is run.

Field	Description
Processing Controls	• Start program: Design and implement a "front end" start program to execute in place of the standard ECS start routine. This feature is useful to provide customized screens that capture specific detail and provide extra functionality and validation. By providing custom start screens, you can use ECS as the central tool to start and monitor all interfaces. A user defined start program is a useful tool to perform some "interactive interfacing" such as downloading a PC file and then kicking off an ECS process to continue execution of the interface in background. You should not confuse this feature with a Phase start exit program. • Disallow manual start: Users may start processes manually from either the Display Process status screen or from the Workbench. If this flag is set, it is impossible to start the process manually (default). This control is intended to prevent users from inadvertently starting processes that are designed to initiate by other means. • Trace mode: Allows trace messages ('T') to write to the run log. If set at the process level, you assume all phases are to be in trace mode. You may specify trace mode at the phase level. • Start event if ECS is shutdown: Instructs ECS to execute the process, even if an ECS shutdown is issued. Normally, the start request is queued until ECS successfully re-starts.
Default phase RFC, parallel and user controls	ECS background servers can support SAP parallel processing, that is, execute multiple phase runs off the queue at the same time. These controls allow parallel run configuration for the process. For further information see the details on parallel processing. • Utilize server parallel processing: Enable parallel processing for this process (assuming parallel processing enabled in the ECS background server) • SAP RFC group: The RFC group to use for parallel processing to manage dialog resources consumed.

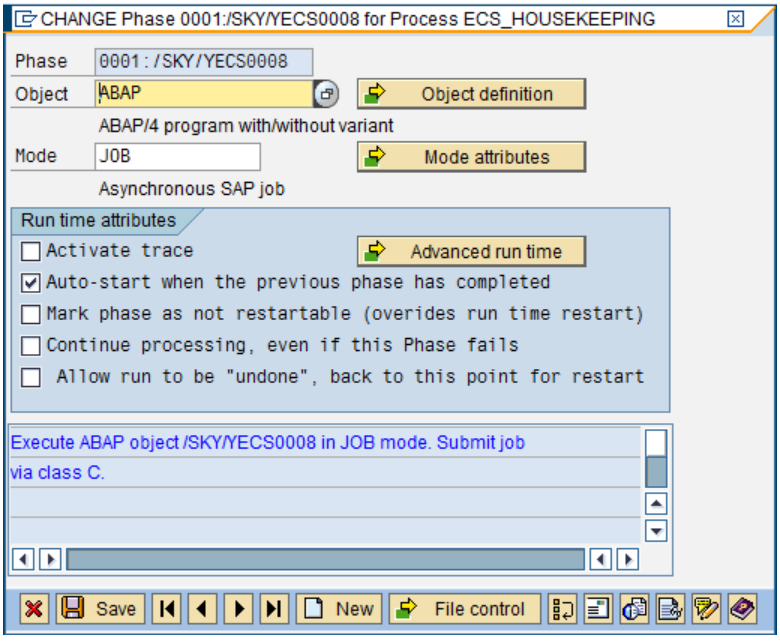
Field	Description
Arrow keys	These buttons allow the selection of the first, last, next and previous processes. This can be more convenient than returning to the Workbench each time.
Phase button	Selecting the **Phases** button in the application toolbar can effect the creation and maintenance of the process phases. This is more convenient than returning to the Workbench each time.

6.4.5 Maintaining Phase Definitions

Phases are created and maintained for a specific process. The sequence of phases is important since the order in which they are listed is the order in which they are executed. To create an initial phase, position the cursor on the process and click the create button. If additional phases are required after an existing phase, position the cursor on the preceding phase and click the create button.

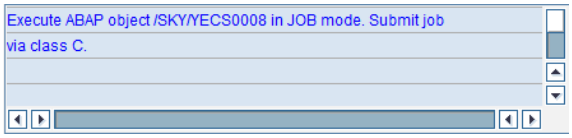
> ***Note:*** If the phases are out of sequence, position the cursor on the phase that you need to move and click the up / down arrows on the application toolbar.

To change the attributes of an existing phase, position the cursor on it and click the change icon. The same screen as create appears with the Process name disabled.



Field	Description
Phase name	Unique name for the phase within the process. Phase names are qualified by the Process name, therefore you may use the same phase name in other process definitions. This is disabled in change mode. Process names that start with STARTED and COMPLETE are reserved for internal use.

Field	Description
Object	The type of object that the Phase is to execute. ECS Core and extensions provide standard object types, such as ABAP, BDC, COMMAND, EVENT, PROCESS, and SCHEDULE. Select the object type through the 'drop down' search help. The attributes of the object are selected by clicking the object definition button. Click here for more details.
Mode	This option describes how ECS executes the object. There may be restrictions imposed for object types, depending on the mode. You specify further attributes for the mode by clicking the mode attributes button. Click here for more details.
Activate trace	Enables tracing for the phase so that any calls of the ECS trace macro get logged.
Auto-start when the previous phase has completed	To configure ECS to automatically start this Phase when the previous Phase completes, select this check box. This setting is not applicable for the 1st Phase in a Process that starts automatically regardless. If you do not specify auto-start, you must start the phase through an ECS command in an ABAP program. See the "Programming guide" for details.
Mark phase as not restartable	This setting makes the Phase definition "non restartable" from the ECS run data display. It overides any explicit ECS restart commands and restart configuration. This is useful to prevent users from automatically restarting failed Phases that may have complex considerations.
Continue processing, even if this phase fails	This is an extension of "conditional Phase processing" where Processing may continue with the next Phase in sequence, even if it has failed. Conditional Phase processing is discussed in depth later in this section.

Field	Description
Allow run to be "undone", back to this point for restart	
Advanced run time	The advanced run time button displays a list of attributes that you may use to further control the execution of the Phase. Click here for more details.
Text Box	The text display gives a meaningful interpretation of the Phase definition. It is intended to summarise the object and mode attributes and any problems and warnings with the definition. 
Arrow keys	These keys allow the selection of the first, last, next and previous Phases. This can be more convenient than returning to the Workbench each time.
New	Resets the screen as if a new phase is created from scratch. The new phase follows the last one that is created.
File control	All Phases may perform automatic file control; such as copy, rename, move and / or archive the file contained in the ECS 'file name' associated with the Phase. You may imbed substitution variables in the names that the ECS resolves at run time. You may find a complete description of how you can use ECS to automate external file management in the file management section. Click here for more details.

6.4.5.1 Phase Object Types

You may select a list of object types to choose from the drop down search help. The object type may need to have some attributes defined, such as name. Use the **Object definition** button to prompt for extra attributes. Further details on each object type is provided below:

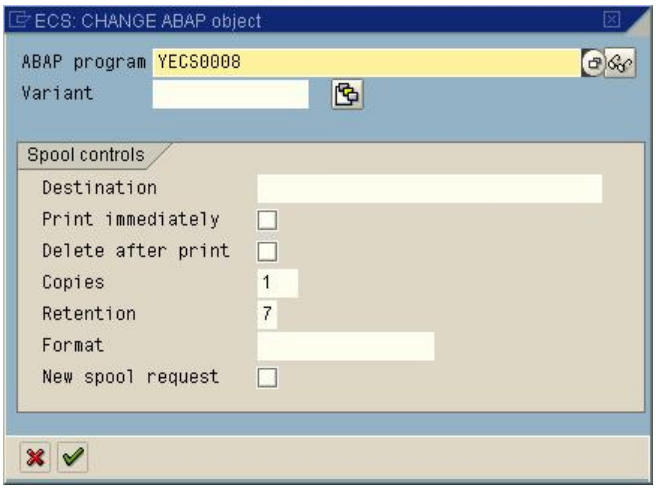
- ABAP
- BDC
- COMMAND
- CONFIRMATION
- COPYFILETOHOST
- COPYFILETOPC
- DUMMY
- EVENT
- NONE
- PROCESS
- SCHEDULE

ABAP

Term	Description
ABAP Program	The name of the ABAP program to execute. You may use the glasses icon to display the code using the ABAP editor.
Variant	A variant to use for the ABAP program. You may use the maintain variant icon to create or display existing variants.

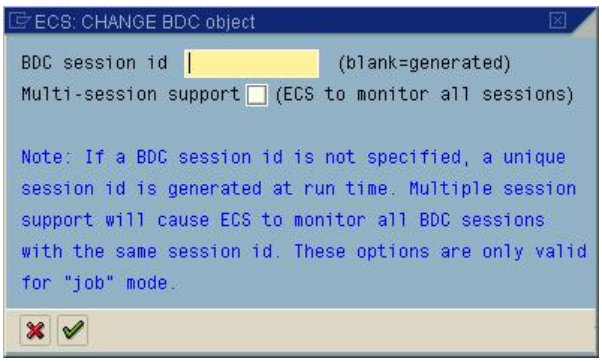
Term	Description
Spool controls	You may use these to control any spool that the program produces when it is executed.

Execute an ABAP/4 program with optional variant and spool controls.



BDC

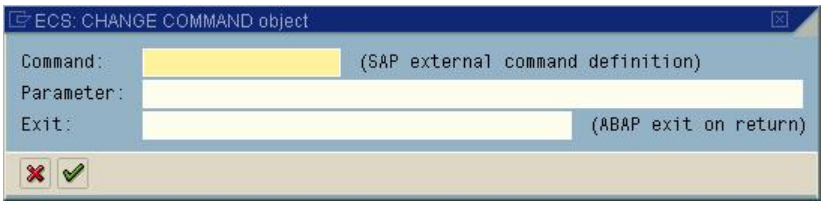
Execute a BDC session. Refer The BDC programming interface section for more details.



Term	Description
BDC session id	JOB mode only. The name of a specific or generic BDC session to execute. If you leave this blank, any BDC name passed from the previous phase executes. You may specify a generic BDC session ID, for example, ABC*. You may also dynamically substitute the BDC ID at run using the ECS and VAR1/2/3 user defined variables.
Multi-session support	Instructs ECS to submit and monitor all BDC sessions with the specified name. You use the monitor 'spawned' jobs feature to do this.

COMMAND

Issue an external OS command referencing a SAP logical command definition. You may specify a parameter string that may contain dynamically substituted variables.

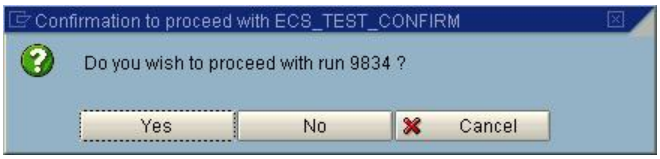


Term	Description
Command	The name of a SAP logical command definition (SM59). The command may only have one parameter.
Parameter	A parameter string to substitute into the command. The parameter may contain the following ECS substitution variables: &rdata, &file_name, &rnumber, &yyyymmdd, &hhmmss.
Exit	The name of an ABAP program to invoke on return from the command execution. The program is passed the resulting output from the command. See the documentation on executing external commands in the ABAP programming guide.

CONFIRMATION

The confirmation object executes only in a synchronous mode to prompt the user whether to continue with the next phase or not. It is designed to implement a controlled break in the process, to allow a manual confirmation

to continue, for example, you need to check the automatically imported file before the data conversion runs. If the phase executes in background, the phase run automatically fails. Upon re-process, or if executed for the first time in dialog mode, the following pop-up appears:



See ECS sample Process ECS_TEST_CONFIRM for an example.

COPYFILETOHOST

The PC file upload object is designed to execute synchronously in dialog mode. If executed in background, the phase run fails and you need to re-process in dialog mode. This object prompts the user for a file name that resides on the PC (Desktop) and automatically initiates a file download to a automatically generated file name on the host. You may alter the file name using phase file controls and you pass on to any subsequent phase through the v_ecs_*file_name parameter.

> ***Note:*** See sample Process ECS_TEST_COPYFILHOST for an example.

COPYFILETOPC

The PC file download object is designed to execute synchronously in dialog mode. If executed in background, the phase run fails and you need to re-process in dialog mode. This object prompts the user for a file name to create on the PC (Desktop). It automatically initiates a download from a file name on the host. You may alter the file name using phase file controls and is pass it to this phase through the v_ecs_*file_name parameter.

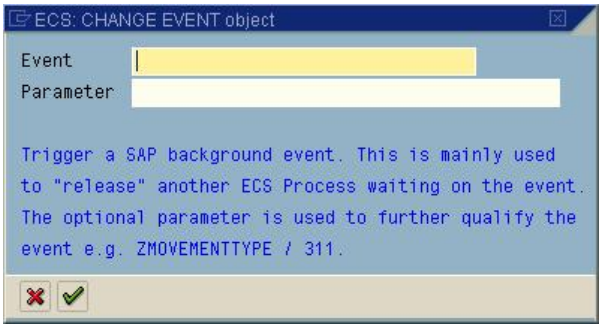
> ***Note:*** See sample Process ECS_TEST_COPYFILEPC for an example.

DUMMY

The dummy object executes an ABAP/4 program (YECS9000) that performs no action. It is useful in-order to specify a phase that is configured to perform other functions, such as file control and notifications without having to execute anything.

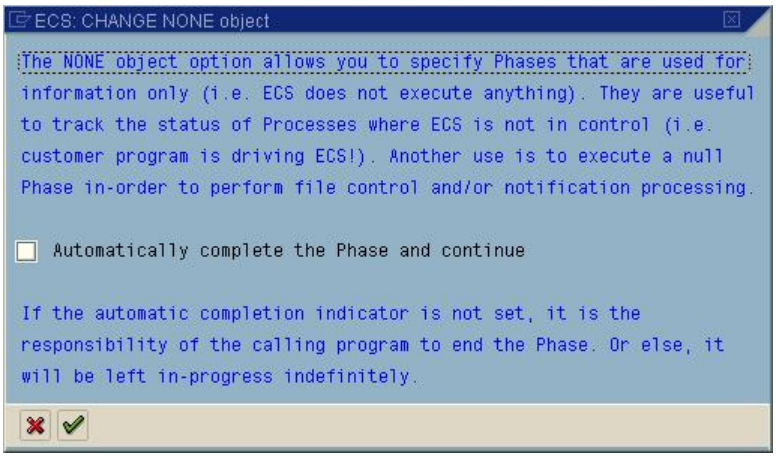
EVENT

Trigger a SAP background event.



Term	Description
Event	The name of a SAP background event to trigger. This is used mainly to release another ECS process that is suspended, waiting on the event.
Parameter	This optional parameter data is used to further qualify the event.

NONE

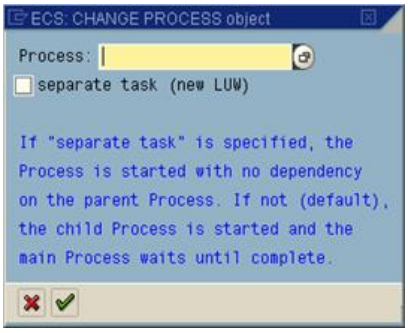


Term	Description
Automatically complete the Phase and continue	NONE object types, indicate to ECS that the phase is controlled from within an ABAP program. If you set this check box, ECS automatically proceeds with the next phase when this phase completes. If you do not set this check box, it is the responsibility of the controlling program to start the next phase through the 'ecs_ start_next_phase' command.

PROCESS

Start another ECS process (nested process call). The current process may either wait for the process to

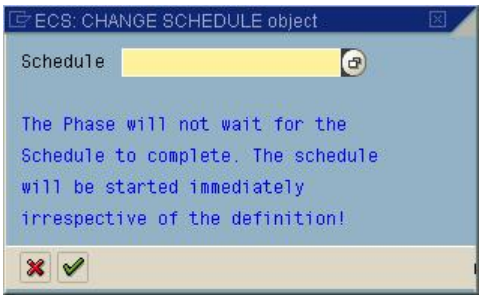
complete or continue processing the next phase.



Term	Description
Process	The name of the ECS process to start. ***Important:*** ECS does not check for recursive calls.
Separate task	If specified, the process starts as a separate task with no dependency on the current process. If not, the current process waits until the process completes.

SCHEDULE

Start an ECS schedule. The phase does not wait for the schedule to complete, but continues on with the next phase. Note that the Schedule starts, irrespective of its schedule option code, and any Schedule exceptions are taken into account.



Term	Description
Schedule	Start the specified ECS Schedule definition immediately. ***Important:*** The Schedule option code is ignored (for example, last day of every month), but exceptions are taken into account.

Utility Objects

This collection of objects is designed to make interfacing simple. Standard objects are provided to perform the most common functions, such as data transfer and starting applications.

EXECPCPGM

The PC execute program object is designed to execute synchronously in dialog mode. If executed in background, the phase run fails and need to re-process in dialog mode. This object executes any program that resides on the PC (Desktop), or network, either by the registry reference or by file association. This is useful to automatically start MS-Word, MS-Excel, Netscape, and Explorer from within SAP.

EXECPCFILE

Execute the PC application associated with a file type, for example, fred.doc normally invokes MS-Word. The file name is passed using v_ecs_*file_name. This object is designed to execute synchronously in dialog mode. If executed in background, the phase run fails and needs to re-process in dialog mode.

LISTFILE

List the contents of a file. This is useful for diagnosing problems or reviewing the contents of a file. If executed in dialog mode, the object invokes the ECS list file routine, else simply lists the first 255 characters of each record.

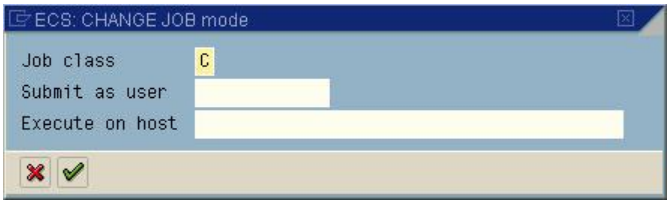
6.4.5.2 Phase Execution Modes

The execution mode dictates how ECS is to execute the object. You may specify further attributes by selecting the **Mode Attributes** button to prompt for extra attributes. More details on each mode are provided below:

- JOB
- SERVER
- SYNCHRONOUS

JOB

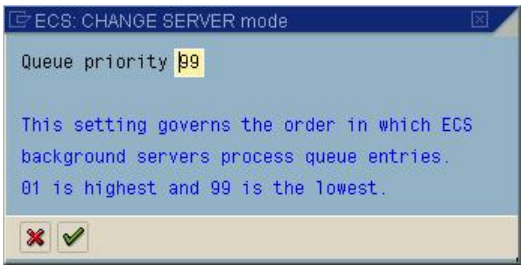
Execute the phase object as a SAP background job.



Term	Description
Job class	The SAP background class, for example, A, B, C to assign to the job.
Submit as user	A specific user ID to use for background execution
Execute on host	Nominate a specific host machine where you need to execute the job

SERVER

Execute the object asynchronously through the ECS background server



Term	Description
Queue priority	The server queue priority is defaulted to 99. This priority governs the entries that the servers to process first in a busy system (01=highest and 99 is the lowest).

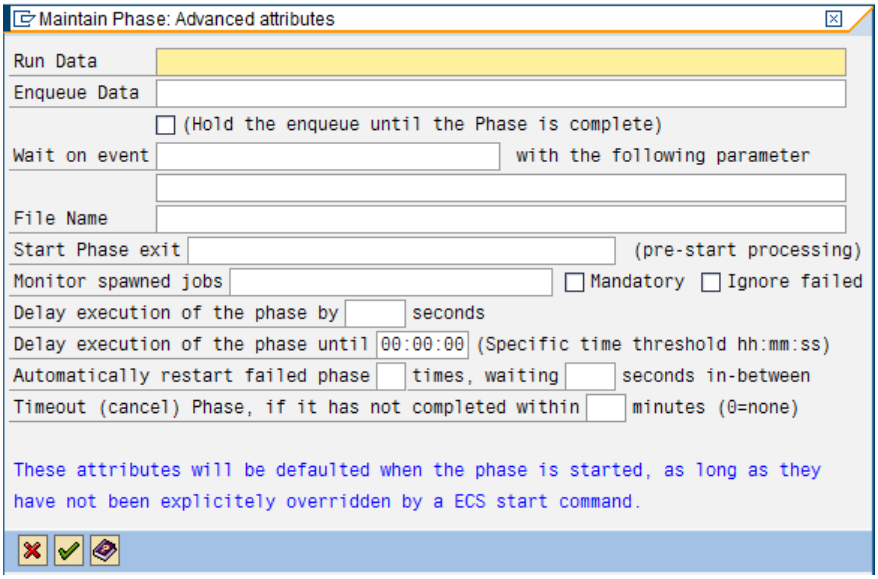
SYNCHRONOUS

The object is executed directly by ECS. This option is designed to execute programs in the foreground that require user input (dialog).

> ***Important:*** If the program aborts (short dumps), ECS has no way to detect this since the program is called directly.

6.4.5.3 Advanced Attributes

Advanced run time attributes allow further options to control and manage the phase execution. These options are overridden if explicitly used when starting a phase programmatically.



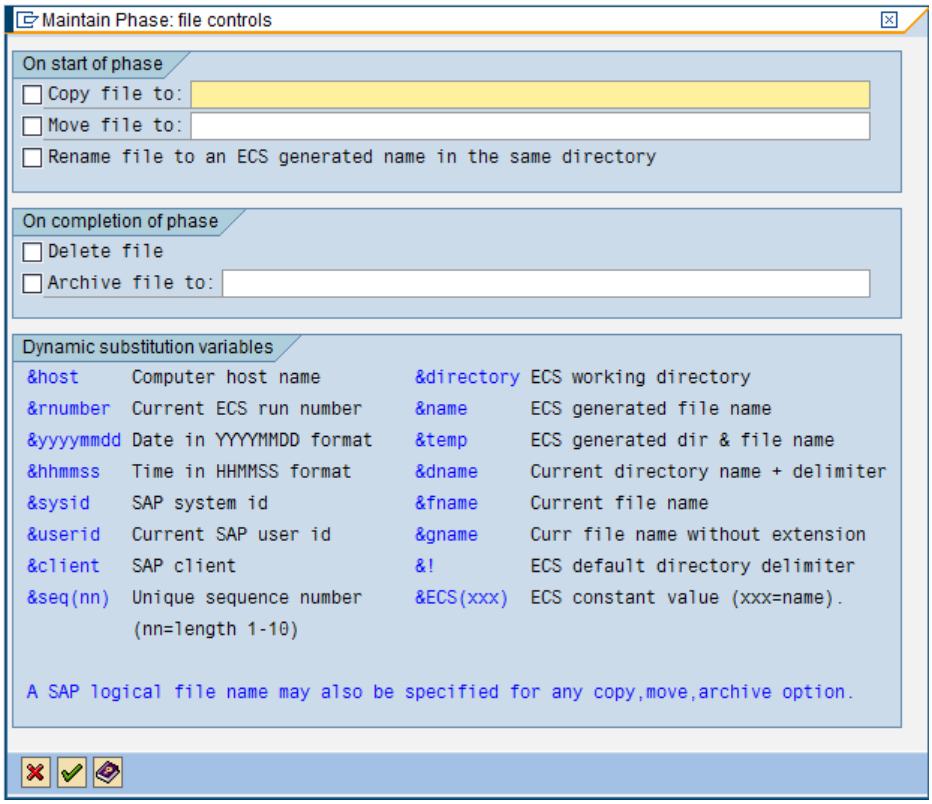
Attribute	Description
Run data	Fixed string up to 255 characters long, that is accessible through the v_ecs_*rdata variable following an 'ecs_import_parameters' command.
Enqueue data	Fixed value used to enqueue the Phase run.
Wait on event	Suspend the Phase on a SAP background event and only release when the event is triggered.
File name	Fixed file name that is accessible through the v_ecs_*file_name following an 'ecs_import_parameters' command.

Attribute	Description
Start phase exit	Invoke a user defined program when starting the phase. Click here for more information. This is typically used to perform extra validation before starting the Phase run. The exit program may alter certain run time attributes. You may also abort the start.
Monitor spawned jobs starting with	JOB mode only. If a job (typically standard SAP) starts other jobs, ECS may monitor the progress of these 'spawned' jobs. The next Phase does not start until all the spawned jobs are completed. Progress of all jobs is annotated in the ECS run log. The name may be generic, that is, all jobs start with ABC*. ***Note:*** All jobs started since the Phase started are monitored, not just the jobs started directly by the Phase. If the Phase ends manually, the monitoring of the spawned jobs is ignored.
Delay execution of the Phase by	This option delays the start of the Phase by xxx seconds. This may be useful in cases where you may need to pass a small amount of time to allow processing to take place, for example, update task to complete.
Delay execution of the Phase until	Suspends the execution of the Phase until the specified time. If the time passes in the current date, the Phase starts immediately. ***Note:*** Another technique for an 'operation mode change' is to start waiting on an event that is scheduled to trigger at a specific time.
Automatically restart failed Phase	You may use this option to automatically re-start a failed Phase up to a maximum number of times with a wait interval in between. This option is intended for situations where Phases normally fail due to 'resource' locking and you may restart and re-process with no risk. ***Note:*** You should design the Phase to be top down restartable for this option.

6.4.5.4 Phase File Control

You use file control options to automatically copy, move, rename, delete and/or archive the current file name specified in `v_ecs_*file_name`. ECS supplies some standard substitution variables that are substituted at run time. You may specify either a physical or SAP logical name. If you use a logical name, you may specify up to two parameters enclosed in <….> after the name.

Refer File Management for more details.



Command	Description
Copy file to	At the start of the Phase, copy the current file to the name specified
Move file to	At the start of the Phase, move the current file to the name specified

Command	Description
Rename file to..	At the start of the Phase, rename the current file
Delete file	On successful completion of the Phase, delete the current file
Archive file to	On successful completion of the Phase, archive the current file to the name specified.

You can use a range of substitution variables for file control. Click here for more details.

6.4.5.5 Conditional Phase Processing

You may configure phases to execute conditionally based on the value on a condition code that the system set automatically, or programmatically through the 'ecs_set_condition_code' command. In this way, you may determine execution of Phases at run time. If a Phase is bypassed, the condition code set by the last Phase executed is true if the Phase to test is '*'. You may configure a phase to continue, even though they failed by using the **Continue processing even if this Phase fails** option on the Phase definition. In this case, the condition code is not reset. All condition code tests and results are annotated in the ECS run log.

Further details on each mode is provided below:

- Internal ECS Condition Codes
- Defining Condition Codes

Internal ECS Condition Codes

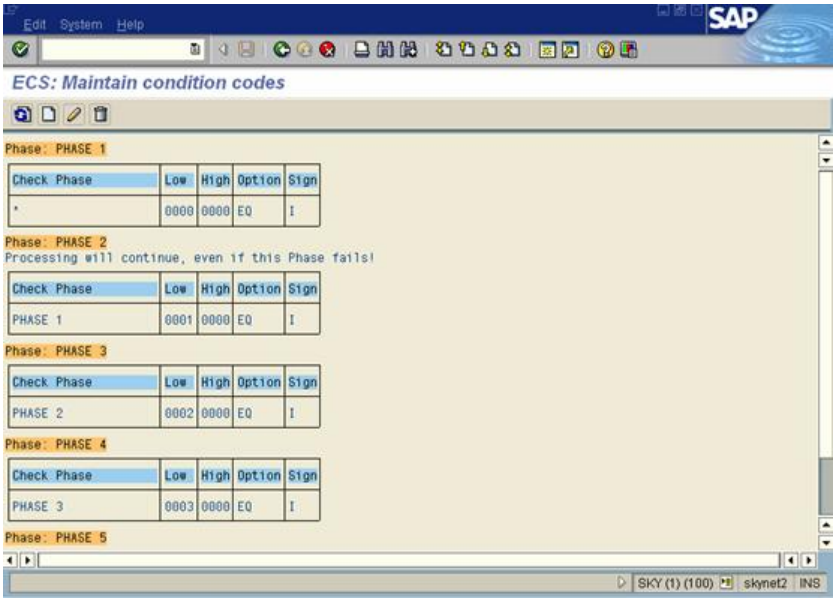
The following codes are reserved from use:

Code	Description
0000	Configure if the Phase completes successfully and no condition code is explicitly configured
9998	Configure if the Phase is bypassed due to condition
9999	Configure if the Phase failed and no condition code is explicitly configured

Defining Condition Codes

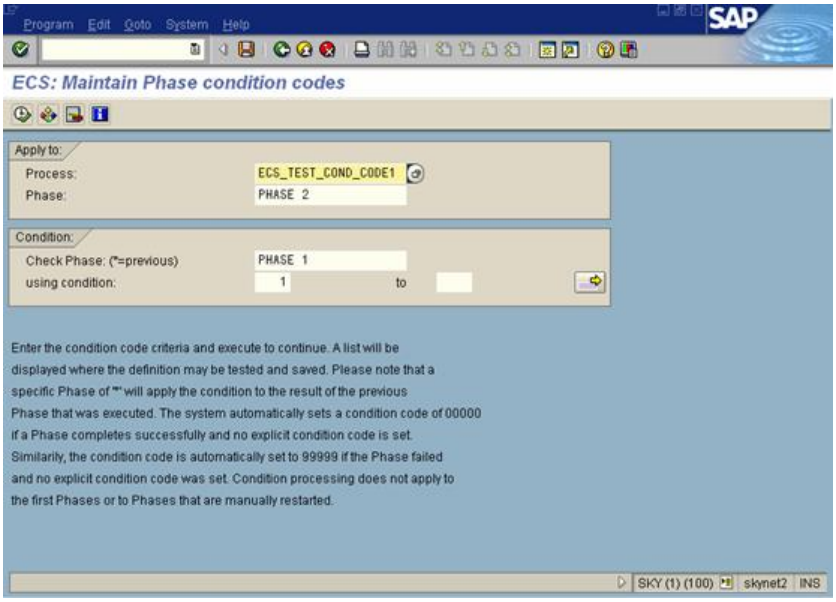
ECS utilizes the select option (range table) feature in SAP to configure simple or complex check conditions. Use the **condition** option on the **EDIT** menu from the Workbench, or the **Condition** button 

on the Process or Phase definition screen to define condition codes. Conditions are also clearly identified on the Process 'tree' display as a hotspot icon. Use the following screen to maintain condition codes:



The Process Phases are listed in the order of execution and any conditions appear as range tables. To create a condition for a Phase, position your cursor on the Phase name and press the create icon. To maintain a condition for a Phase, position your cursor on the condition and press the change or delete icon.

When creating or changing a condition, the following screen appears:



> ***Note:*** The condition code logic is captured using the SAP selection screen.

You may specify as many condition checks as you require using the selection option interval and extension prompts. You may specify a specific Phase, or the last Phase executed ('*'). Once complete, click **Execute** to list and test the condition. You may test the condition, by entering a value in the input field provided.

6.4.5.6 Phase Message Evaluation

Phase message evaluation enables you to perform 'low level' monitoring of spool, job logs, and BDC logs that are associated with a specific Phase run. For example, you may write error messages to the spool by a program. If you detect these messages, you should fail the run in-order to notify support. In the same way, you may scan other sources of information, such as job and BDC logs for the existence of messages. You may configure multiple definitions for any Phase. This feature is originally designed to perform low level monitoring of standard SAP periodic processing, such as monthly material roll, where you write critical error conditions to the spool and cannot easily monitor.

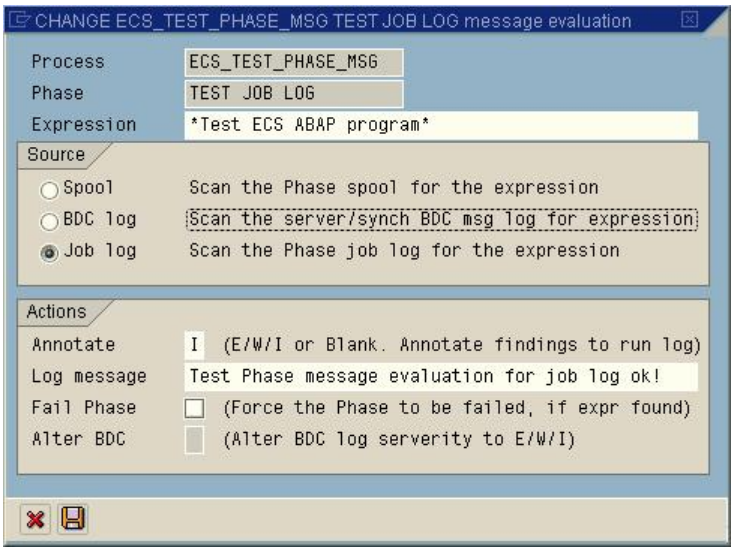
Further details on each mode is provided below:

- Configuring Phase Message Evaluation Definitions

Configuring Phase Message Evaluation Definitions

You may configure phase message evaluation for any Phase by selecting the  icon from the workbench or Phase maintenance pop-up screens. Use the following screen to maintain Phase message evaluation configuration:

The Phase message evaluation definitions are listed in Process / Phase / Source order. Source determines what Phase output to evaluate, for example, SPOOL, JOB LOG, BDC LOG. To create, change, display, copy or delete a definition; position your cursor on a relevant area on the screen and click the icon on the application toolbar. Position on the screen is important in-order to default Process, Phase, Source information.
When you create or change a definition, the following screen appears:

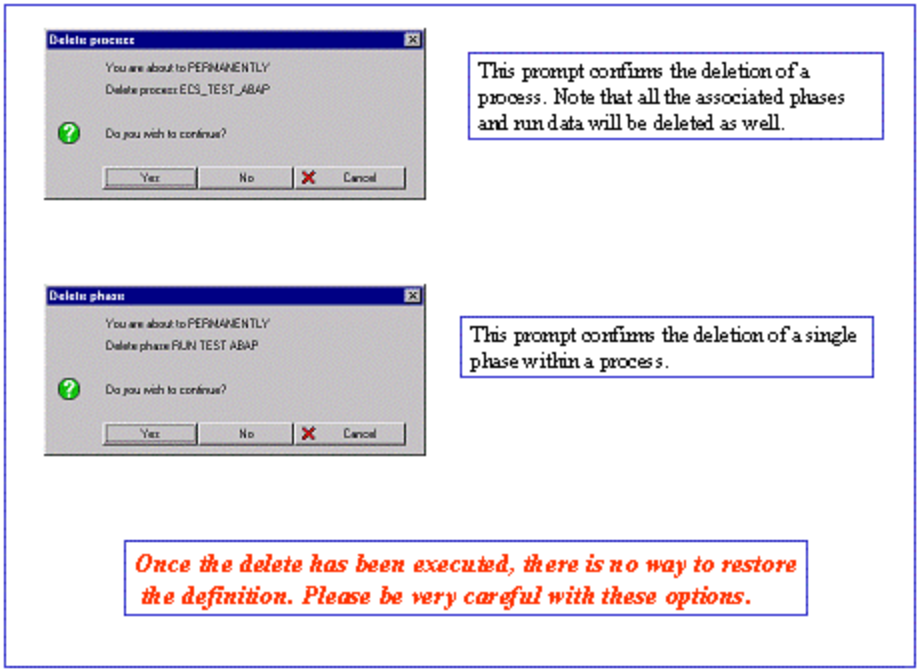


This screen prompts for the message expression to scan for, the source of the Phase messages and the action to perform when the expression is located. The expression is located through pattern match and thus should locate wild card characters, such as '*', if the exact location is not known or contains variable components. If a message is located in the source, you may perform the following actions:

Term	Description
Annotate ECS run log	The findings are annotated to the ECS run log using the specified severity I-information, W-warning and E-error. Leave blank to suppress.
Log message	This is an optional message to annotate to the run log to help further explain the action. Leave blank to suppress.
Fail Phase	If the expression is found, automatically fail the Phase.
Alter BDC log	If the BDC session is executed in ECS background server or synchronous modes, you may dynamically alter the severity of the located message, that is, errors may be demoted to warnings, and information messages promoted to errors. ***Note:*** This option is not available for BDC sessions executed by SAP.

Deleting Process and Phase definitions

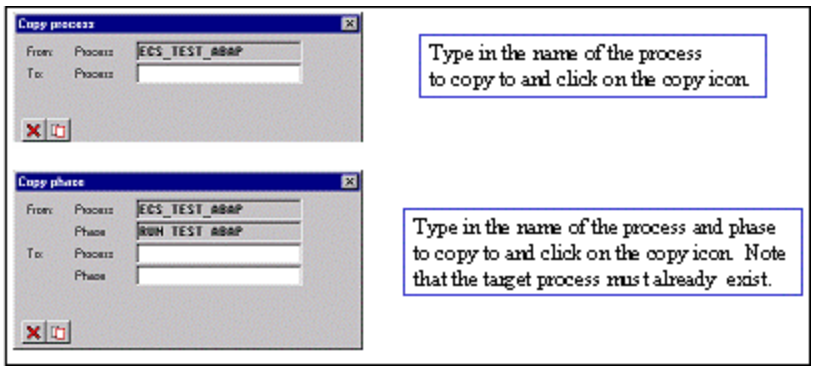
To select a process or phase definition for deletion, position the cursor on the name and click the Delete icon (rubbish bin). One of the following confirmation screens appear.



Copying Process and Phase Definitions

Copying existing definitions is a very quick way to construct new Processes and Phases. Position the cursor on the definition that you need to copy and click the **Copy** icon.

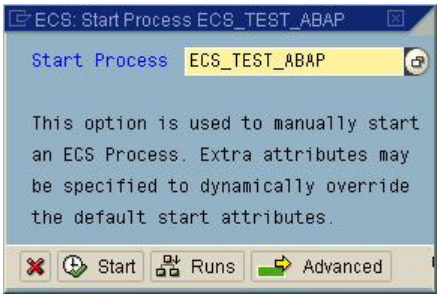
> ***Note:*** The "from" Process and Phase must exist when you copy Phases.



Starting a Process from the Workbench (Manual Start)

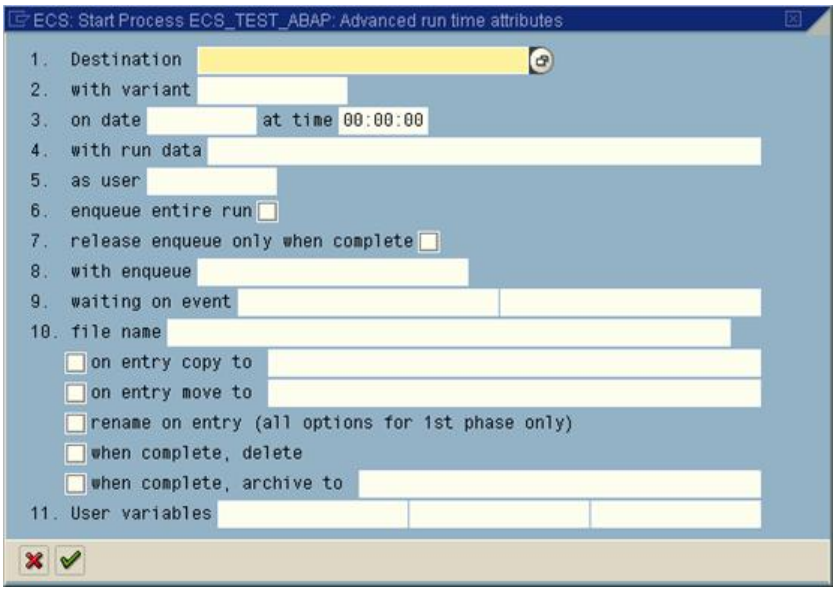
A convenient way to test whether a process executes correctly is to execute it from the Workbench using the Start Process option. Alternatively, you can use this option to start manual processes. Position the cursor on the process that you need to start and click the **Start Process** icon. The following pop-up appears:

> ***Note:*** You must configure the process to allow it to start manually. You perform this through the Workbench "maintain Process" function. The default is to disallow manual start.



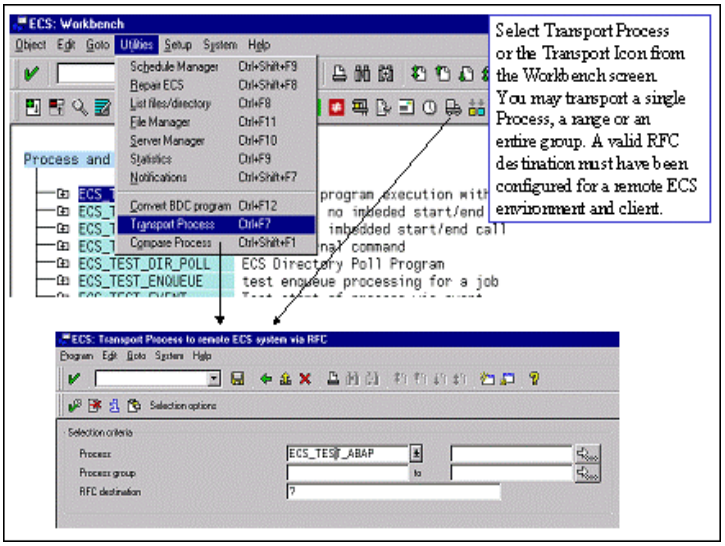
> ***Note:*** You may define a user defined start program for the process. If this is the case, ECS starts the user-defined program in place of the default start screen. See the sample process ECS_TEST_USER_START and the program YECS9009 for more details. This feature is useful to provide custom front ends to the ECS start process functionality that perform extra validation, authorisation or functionality other than that the standard ECS start process provides.

You may specify advanced runtime attributes through the advanced button:



Transporting Process Definitions

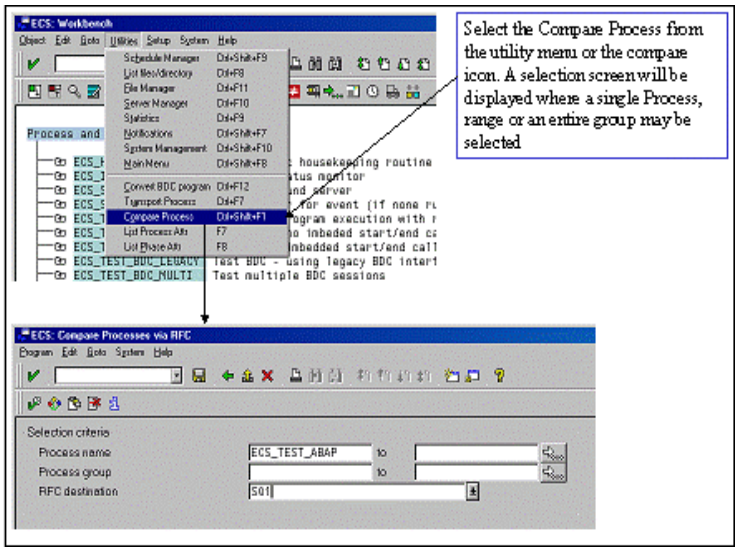
You can transport the ECS Process definitions to another ECS environment on a remote SAP R/3 system. This is performed realtime using SAP R/3 -> SAP R/3 RFC connection. You must configure all the RFC connections available using transaction SM59. It is possible to transport a single Process, a range, or an entire group. Any existing Process definitions are replaced.

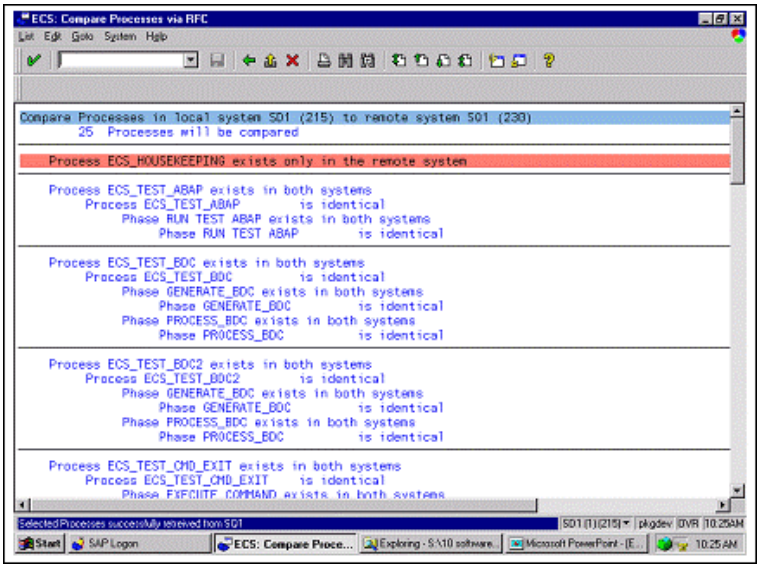


> ***Note:*** The ECS Process transport facility does not use the SAP R/3 correction and transport system. This is because logic, as well as data definitions, is required when moving Process definitions between ECS environments.

Comparing Process Definitions

ECS provides the ability to compare ECS process definitions between ECS environments on remote SAP R/3 systems. This is performed realtime using SAP R/3 -> SAP R/3 RFC connection. You must configure all the RFC connections available using transaction SM59. It is possible to compare a single Process, a range of Processes, or an entire group. A report detailing the differences appears:





6.4.6 Dependency Controls

6.4.6.1 About

ECS has comprehensive controls to help organize and better manage processing. Typically, you may configure these controls 'statically' against a Process or Phase through the Process Workbench or dynamically at run time through ECS ABAP / 4 commands.

6.4.6.2 Key Topics

6.4.6.3 Serialization

Sometimes, you must execute processes in a sequential fashion, that is, only one instance of a process at a time. This is normally due to exclusive control conflicts over resources, locking, performance, and badly designed interfaces. To instruct ECS to serialize the execution of a Process, simply check the **Serialise the entire execution** check box on the Process attribute maintenance screen through the Process Workbench; or through the ECS_ENQUEUE_FOR_ENTIRE_RUN start parameter with ECS_SET_ENQUEUE {process name}. This effectively issues an ECS enqueue with the name of the Process on the first Phase.

> ***Important:*** The first Phase must not have any ECS enqueue processing associated with it.

Further controls are available to only release the enqueue when the current Phase is completed successfully. Select the **Release next serialised run** check box on the Process attribute maintenance screen or issue the ECS_REMOVE_ENQUEUE_ON_COMPLETE start parameter.

Click here to go back to the top.

6.4.6.4 Enqueuing

If you require only one instance of a Phase to execute at a time, you can use the ECS enqueue facility to suspend execution of a Phase until the current Phase completes execution. You may hard code enqueue values or specify dynamically at run time. Either specify a fixed value using the enqueue data parameter on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench. You may specify an enqueue value dynamically at run-time using the ECS_SET_ENQUEUE command.

An extension to the Phase level enqueue, is the ability to retain the enqueue for the life of the run, that is, until the run ends either in failure or is successful. In addition, it is possible to only release the enqueue only when the run ended successfully. You may set these options dynamically at run time using the ECS_ENQUEUE_FOR_ENTIRE_RUN and ECS_REMOVE_ENQUEUE_ON_COMPLETE commands.

Click here to go back to the top.

6.4.6.5 Ignore if Already Running

In cases where a program processes 'all available work', it is not necessary to start another iteration of the program since one already executes. Typically, you trigger these processes automatically by events, for example, delivery goods issue. This option ignores starting another Process run if one is in-progress. Check the **Ignore run if one already running** check box on the Process attribute maintenance screen through the Process Workbench.

> ***Note:*** There is a small chance that the program that currently executes may be in the process of exiting when the check is made. Therefore, processing may be left outstanding.

Click here to go back to the top.

6.4.6.6 Delay Execution

It is normally bad design to make assumptions on dependencies, but sometimes you have no choice. This option delays the execution of a Phase by a number of seconds. It is used mainly to allow enough time for other dependent processing to have completed (for example, Update task to end). Specify this value using the **Delay execution of the Phase by nnn seconds** option on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench; or through the ECS_DELAY_START_IN_SECONDS command.

Click here to go back to the top.

6.4.6.7 Delay Execution Until

This option is designed to suspend processing until a specific time threshold reaches. This may be useful to 'collect' Processes during the day and then release them for execution at night. Specify this value using the **Delay execution of the Phase until ...** option on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench.

Click here to go back to the top.

6.4.6.8 Using SAP Events

SAP background events are a useful way to suspend the execution of a Phase until the event is triggered by either another Phase, an event external to ECS or manually. A SAP background event may consist of either an event id (unique name) or an event id qualified by a event parameter. In the second case, you may specify a value of up to 64 characters to uniquely identify an instance of an event. Specify an event (with optional parameter) using the **'Wait on event ... with optional parameter'** option on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench; or through the ECS_SET_EVENT command. You may configure either SAP or Customer events in this way.

> ***Note:*** Event processing is only available for Phases executed in job mode.

Click here to go back to the top.

6.4.6.9 Do not Start the Process if one Has already Run within hhmmss

You may prevent unexpected duplicate runs by specifying this option. Specify this value using the **Abort if within hhmmss** option on the Process maintenance screen through the Process Workbench. This check is only valid if there is at least one successful run.

Click here to go back to the top.

6.4.6.10 Automatically Restart Failed Phases (nnn Times waiting nnn Seconds in-between)

If a Phase fails, this option automatically re-starts it, again a configurable amount of times, waiting a configurable number of seconds in-between processing. This option is used to automate cases where Phases are 'top down restartable', for example, rerun BDC that failed due to locking conflicts (xyz in use). Specify this option using the **Automatically restart failed Phase** option on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench.

Click here to go back to the top.

6.4.6.11 Monitoring Spawned Jobs (Dependent Jobs Started from a Phase)

This function is designed specifically for standard SAP conversion programs that submitted batch jobs. You need to monitor these 'spawned' child jobs along with the Phase that started them. You may specify either a specific or generic (ABC*) name of the job. There is no direct connection as to the parent of a job, therefore ECS searches for all jobs, with the name specified that were started since the start of the ECS Phase. If you find one or more jobs, ECS monitors the progress of the spawned jobs as part of the Phase run. The details of the spawned jobs are annotated to the run log. Specify this option using the **Monitor spawned jobs starting with**' option on the 'advanced run time' screen invoked from the Phase maintenance screen through the Process Workbench.

Click here to go back to the top.

6.4.6.12 Conditional Execution of Phases

You may execute phases within a Process conditional through the use of condition code logic.

This is explained in detail in Conditional Phase processing.

This powerful functionality enables complex conditional execution based on a mix of configuration and programming commands.

Click here to go back to the top.

6.4.6.13 Continue Processing, even if the Phase Fails

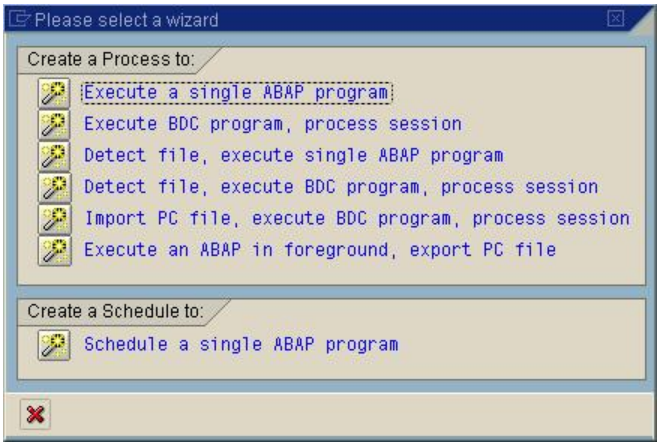
Phases within a process are normally executed sequentially, that is, one after another when each completes successfully. In cases where it is important to continue execution of the process even though a phase has fails, set the **Continue processing even if this Phase fails** option on the Phase maintenance screen through the Process Workbench.

> ***Note:*** The failed Phase is completely bypassed, that is, the run may complete successfully regardless and is therefore not re-processable.

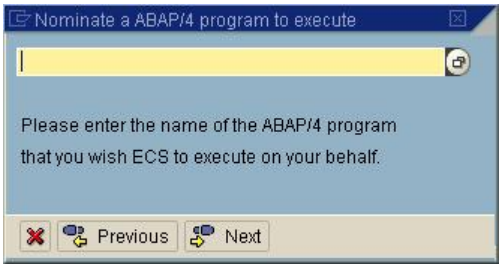
Click here to go back to the top.

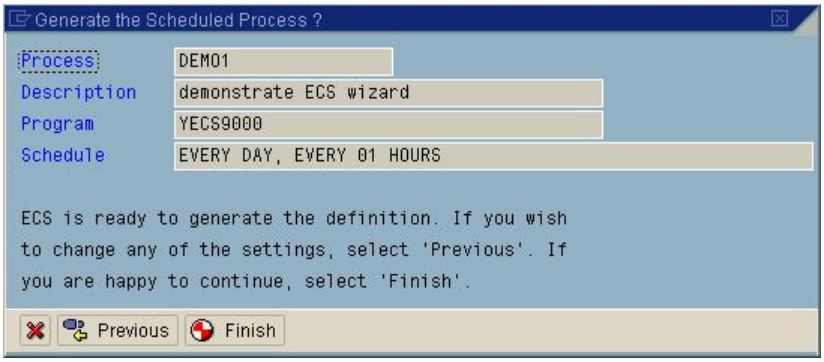
6.4.7 Wizards

ECS provides a wizard facility to guide users, step by step, through defining common Process and Schedule definitions. Click the icon on any of the main screens, or select the **Wizards** option from the **GOTO** menu, to invoke the wizard menu:



Wizards guide the user though a sequence of screens, until finally generating the definition. You may use the previous and next buttons to navigate to either the next or subsequent screens.





6.5 Monitoring

6.5.1 About

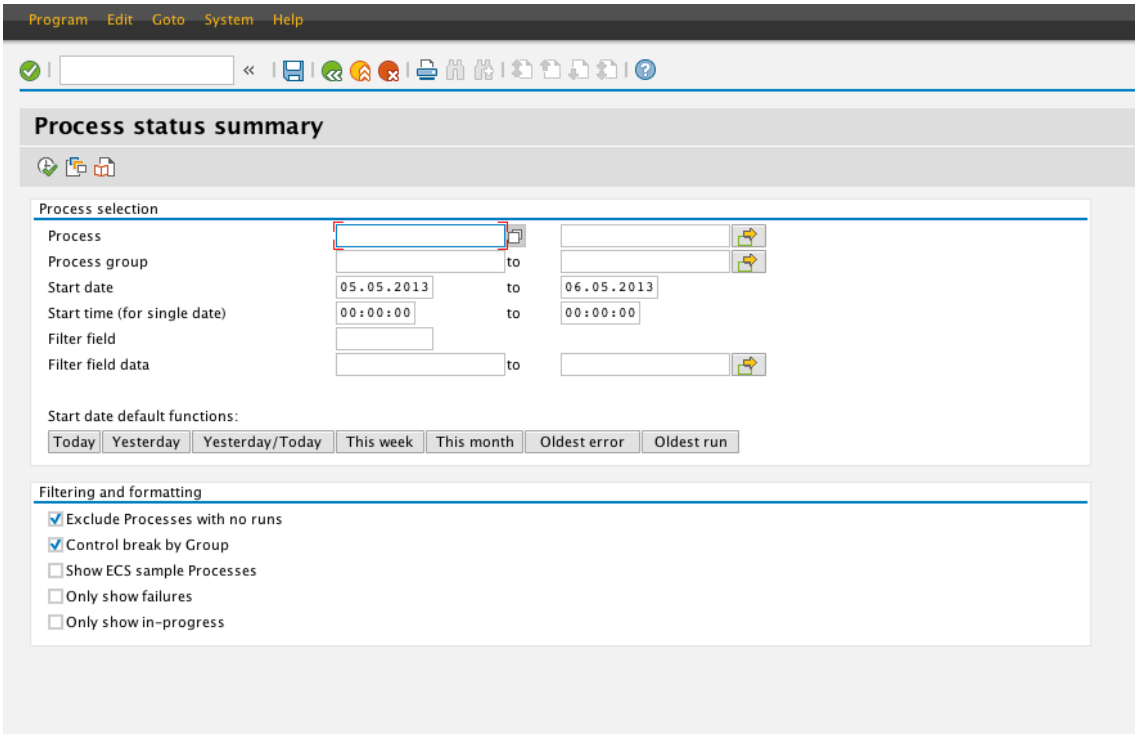
ECS provides a number of different options for monitoring process runs.

6.5.2 Key Topics

Process Status Summary View
Process Runs View
Process Runs Overview

6.5.3 Process Status Summary

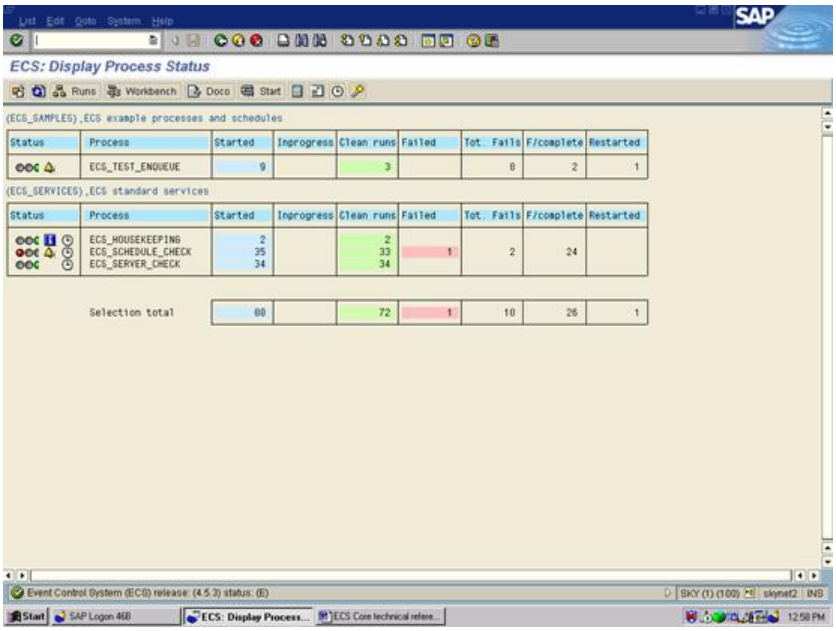
The report provides a high level summarized view of all the Process runs. You may then 'drill down' on a specific area to display individual runs. Execute the YECS transaction code. The following selection screen appears:



This screen allows you to limit the display to a certain range of processes and / or a process group. You can use the filter field and values to further restrict the list to filter criteria specified at run time. The start date range does not limit the list of processes that appear, but the period of the status check.

Field	Description
Process	Filter by a specific process
Process Group	Filter by a process group

Field	Description
Start Date & Time	The date range defaults from yesterday, to today. If you specify a start time, you must specify a 'single' specific date.
Exclude Processes with No Runs	Only show Processes that recorded runs against them for the selection criteria entered.
Control Break by Group	Displays all the processes according to their assigned ECS group. This helps clarify all processing within a logical area.
Show ECS Sample Processes	The ECS sample processes, provided with the system and referenced by this documentation, are normally suppressed. Select the **ECS samples** check box to view. Once you enter the selection criteria for the list (if any), execute the report and the following screen appears:



6.5.3.1 Screen fields and icons

The Processes are listed in alphabetical order within ECS group (If control break by group is selected). The high level status of each Process is represented on the left hand side by a traffic light. The statistics, that is, total runs, and failures appear on the right. You may 'double click' each line to display all associated runs, or alternatively selected columns contain hotspots (cursor changes to a hand!). These hotspots are useful to restrict the runs listed by context, for example, Selecting the hotspot in column 'Failures' may only display failed runs.

Status

The traffic light status. Green indicates that all the processes completed successfully, Red indicates that at least one run failed whilst Yellow indicates that no runs failed but at least one run is still in progress. The information, warning ? and error icons indicate that at least one of the logs has a run log entry associated with it. A clock indicates that one or more of the Process runs are associated with a ECS schedule definition. Each icon is context sensitive, that is, has a hotspot to drill-down or branch to the relevant area for more information.

> ***Note:*** The high level icon status is determined in the order of failures, in-progress then complete (that is, red, yellow, green).

Process

The process name is listed alphabetically unless you select the control break by group option; in this case the process names are listed alphabetically within the group.

Started

Number of runs that are started.

In Progress

Number of runs that are currently in progress (active).

Clean Runs

Number of runs that completed successfully from beginning to end without any failures.

Failures

Indicates that there are runs that failed and need attention.

Tot. Fails

The total number of failures that occurred. This number may exceed the total number of started runs. This is because a single run may have multiple failures, caused by the multiple restarting of failed runs.

F/Complete

Total number of runs that are either force completed at the run level or at the phase level.

Restart

Number of runs that restart either manually or automatically by ECS.

From this Screen

- View the high level status of each process selected
- Drill-down and display each run by double-clicking the process. The 'In progress', 'Clean Runs' and 'Failures' columns are hotspots that only display in progress, completed or failed runs for the Process line.
- Refresh the display
- Display a Status overview report of all the runs by status.
- Select and view a specific run by its run number
- Maintain ECS process and phase definitions through the process workbench
- Maintain Process documentation
- Start a Process
- View Process and Phase statistics

- Goto other ECS functions such as: The Workbench, Schedule Manager, Server Manager, File manager.

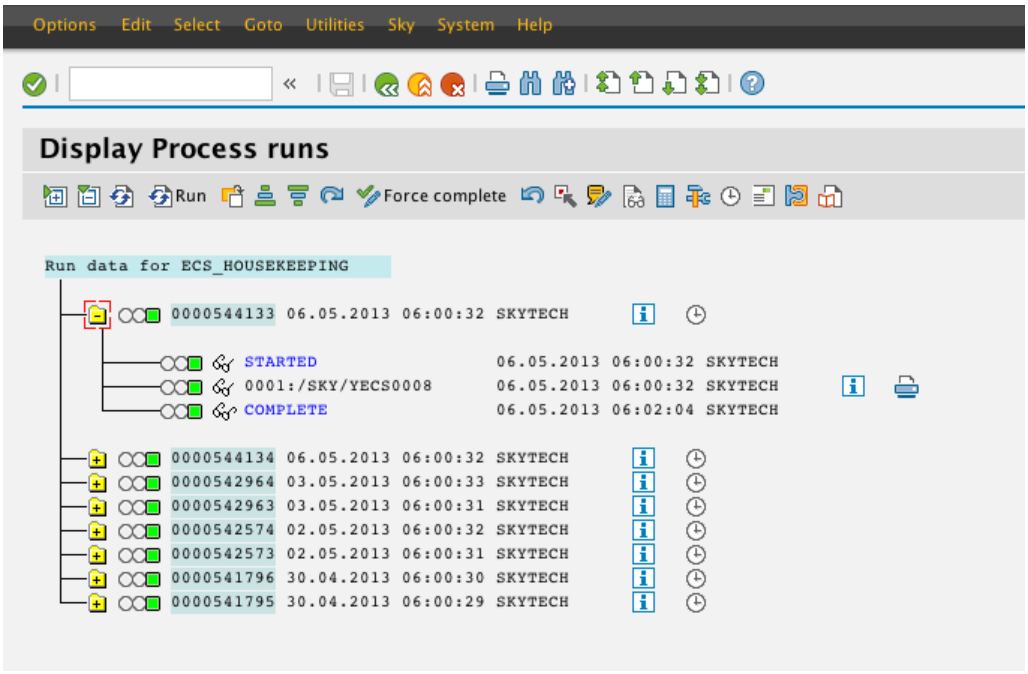
6.5.3.2 Performance

In some cases where there are large volumes of runs, it is significantly faster to restrict the criteria through the "Specific run" screen, or list the runs through the "Process Overview" screen. If a specific run number is known from the outset, you can access the run details through YECR entering the run number. In a busy system, ECS may have to scan large volumes of status data for the displays. Use the date range, Process name and specific Status selection options to improve performance.

6.5.4 Display Process Runs

This report shows runs that occur for a specific process. To run the report, double-click the process line or select "all runs", the runs listed is for the same selection criteria as that you use to list the processes, that is, date range. If you require a more specific view, use the "specific run" screen to specify the selection criteria. You may execute this report directly using transaction YECR.

> ***Note:*** Use the date range, specific run numbers and specific status selections to restrict the criteria to a manageable number of runs. Large numbers of runs can impact performance since ECS has to scan all the status records recorded.



6.5.4.1 From this Screen

- Expand and collapse the phase information linked to each process run.
- View all the details associated with the execution of the phase.
- Refresh the list, or a specific run.
- Force complete a Phase or entire run.

- View the run log associated with the run or phase.
- View the spool generated for a Phase.
- Annotate a run.
- Maintain Process documentation using MS-Word.
- View Process and Phase statistics.
- Cancel or delete a run definition.
- Goto other ECS functions such as: The Workbench, Schedule Manager, Server Manager or File manager.

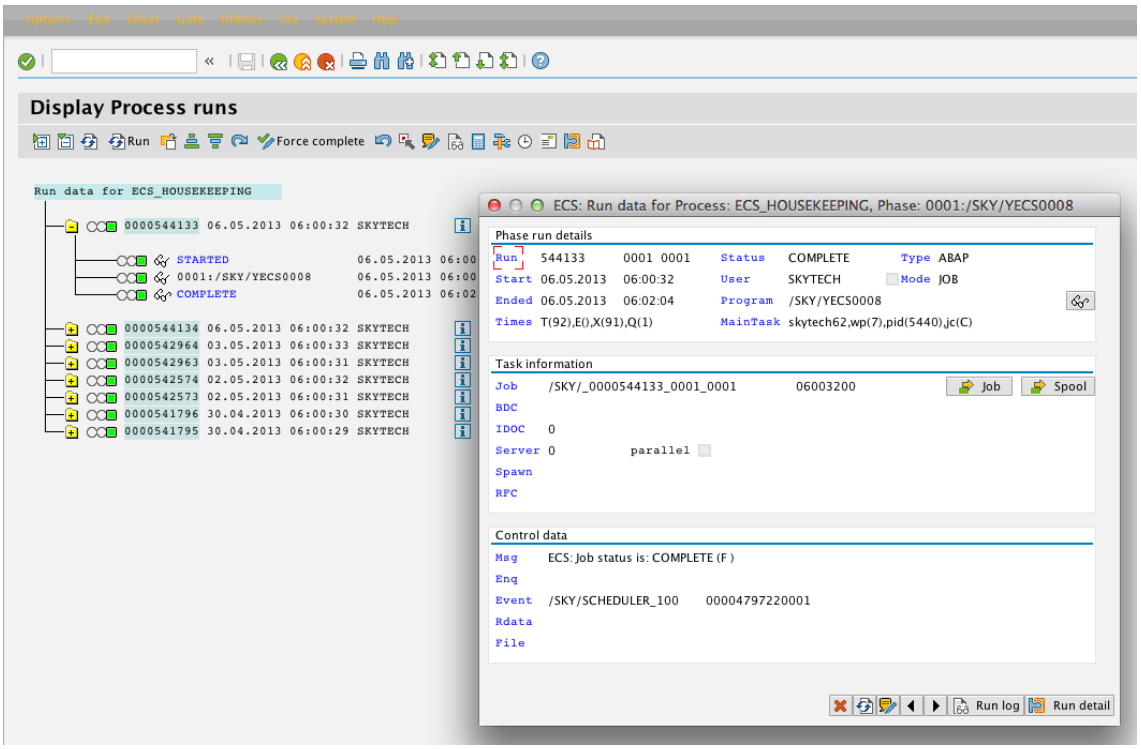
> ***Note:*** The started and completed phases are internally generated by ECS to "sandwich" the other phases executed.

6.5.4.2 Reprocessed Phases

If a phase is reprocessed, it displays multiple times; a phase number within the phase node uniquely identifies each iteration.

6.5.4.3 The Run Data Display Screen

This screen shows all the attributes and detail associated with the execution of a specific phase. It is invoked by clicking the "glasses" icon on the run tree report.



6.5.4.4 From this Screen

- Display all the attributes associated with the execution of a phase
- Display job details submitted by ECS
- Display BDC session details
- Display IDOC data
- Display spool data
- Display file contents and / or ECS File manager references
- Reprocess a failed phase
- Reprocess failed BDC sessions
- Display the run log entries or annotate the phase
- Display all dependencies of a run, for example, enqueues and events that it may wait on

> ***Note:*** The green arrows that appear adjacent to job, BDC and IDOC indicate that more details are available, for example, SAP display job details (SM37), BDC session details (SM35) or IDOC details (WEDI).

6.5.4.5 The Run Log Report

This report displays all the run log entries for an entire run, or a specific phase.

Refer Using the ECS run log for details on how to use this feature.

From this Screen :+

- View the run log entries.
- Reset the category of the run log from Warning or Error (alarm) to information (I).
- Refresh the run log entries. This may be useful, as the run is in-progress to view what it is doing.
- Add annotations to the run log.

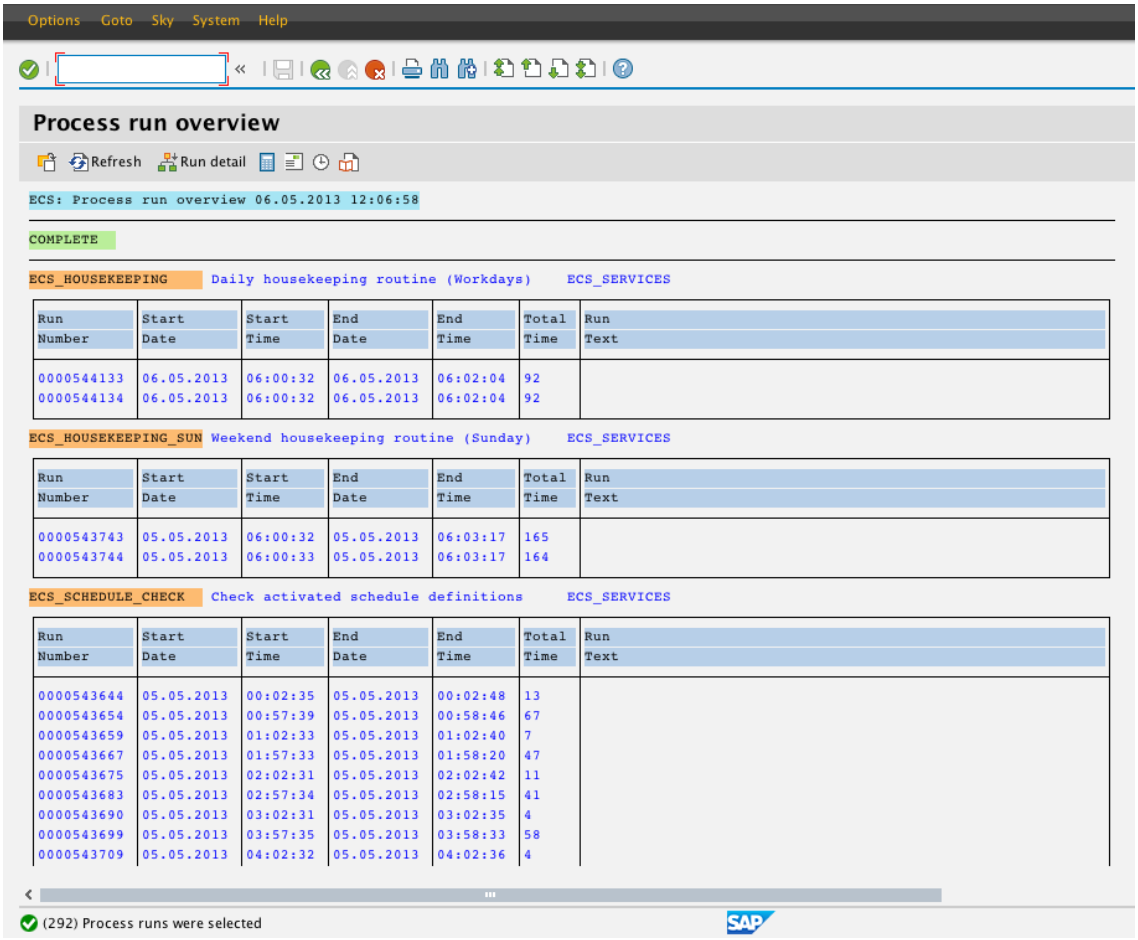
6.5.4.6 Annotating the Run

The annotate icon appears on the Run Tree and run log reports. It is a pop-up that allows five lines of free format text to append to the run log of a specific run. You can use this facility to log extra information about a specific run, and how you can manually fix it...Run log data is associated with a specific run and is deleted when the run passes its designated retention period.

6.5.5 Process Run Overview

This report offers an alternate view to the "run tree" report and offers easy drill-down facilities to narrow down the display. You can also schedule the report to run periodically as an exception report to highlight runs that fail. You may execute this report using the YECO transaction.

> ***Note:*** This option is a faster way to view the statuses of a large number of runs.



6.5.5.1 From this Screen

- Restrict the list by clicking status and / or process details.
- View the run detail by clicking the run number, or positioning the cursor and clicking the run detail icon.
- Execute utilities such as: The Workbench, Schedule Manager, Server Manager, and File manager.

6.5.5.2 Scheduling the Status Overview Report

Create a variant on the specific selection criteria for the report. Create a Process definition to execute program YECS0009 with the variant and a Schedule definition to execute the Process. You should direct the output to a printer and implement a process for the report distribution and checking.

6.6 Statistics

6.6.1 About

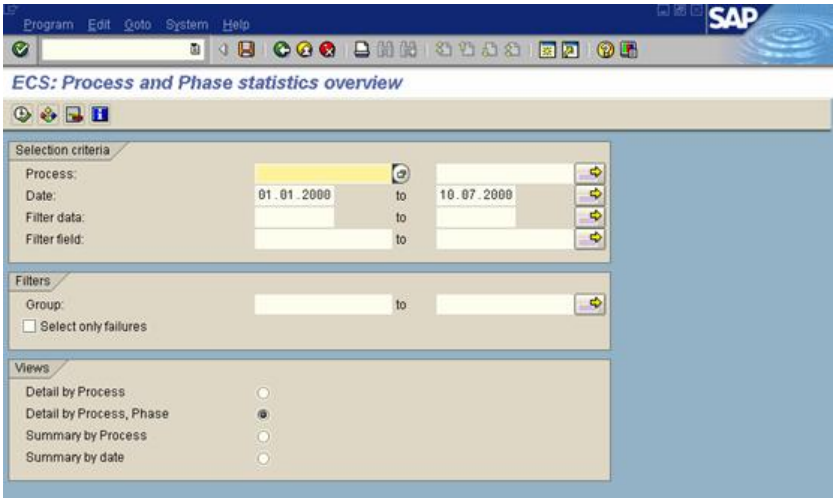
ECS keeps run time statistics for all executions at both the Schedule and Process levels. You may access the statistics manager through the YECC transaction shortcut, the **GOTO** > **ECS services** > **statistics** menu path, or the calculator icon on any of the main screens. ECS provides some report views, graphical statistical views and the option to download statistical data in CSV format for reporting in MS-Excel.

6.6.2 Key Topics

Process and Phase Statistics
Schedule Statistics

6.6.3 Process and Phase Statistics

The Process and Phase summarize the statistics by date. There is a simple view of the statistics screen available from the "Display Process Status", "Display run tree" and "Overview" screens. To cater for more sophisticated reporting and trend analysis, you can download statistics in a CSV format and import into a reporting tool. Select the calculator icon from any of the main report displays and the following selection screen appears:

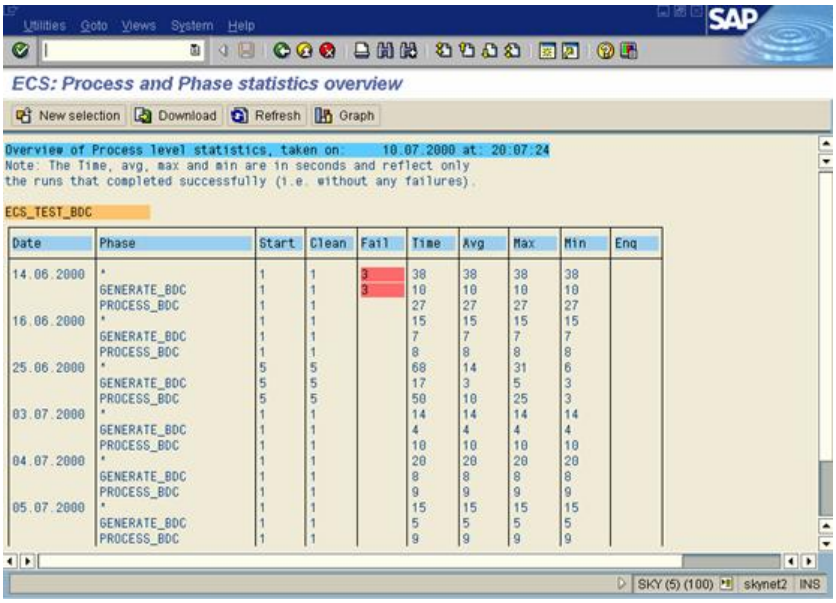


This selection screen allows you to restrict the selection of statistical data and an initial view. You may change the view dynamically at any time.

> ***Note:*** You may select statistics on specific filter data. Once you execute, the following views are available:

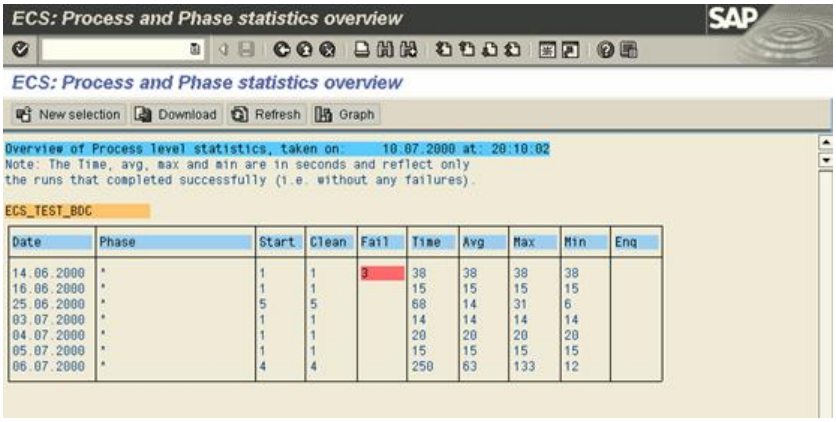
6.6.3.1 Detail by Process, Phase view

This view summarizes the run and time statistical for each Process and Phase by date.



6.6.3.2 Detail by Process View

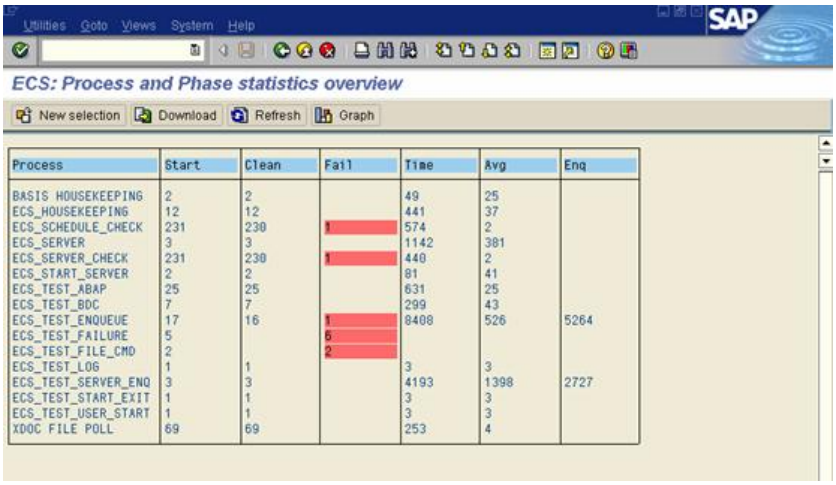
This view summarizes the run and time statistical data for each Process by date.



> ***Note:*** This view is identical to the Detail by Process, Phase view except that the Phase level statistics are suppressed.

6.6.3.3 Summary by Process View

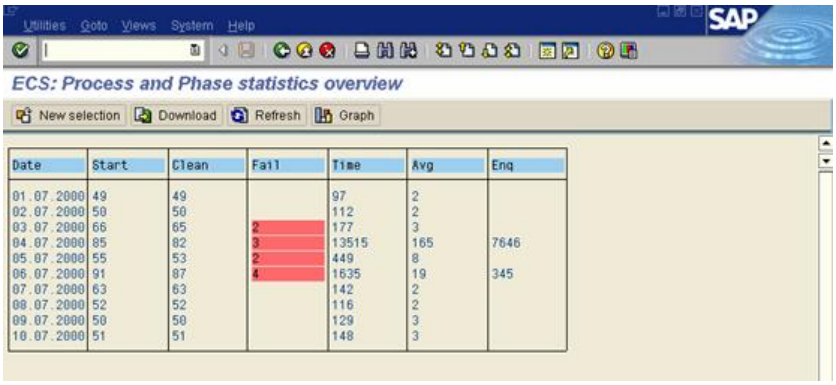
All run and time statistics are summarized, for the selection criteria, at the Process level. This is useful to highlight potential performance bottlenecks and problem areas.



Process	Start	Clean	Fail	Time	Avg	Enq
BASIS HOUSEKEEPING	2	2		49	25	
ECS_HOUSEKEEPING	12	12		441	37	
ECS_SCHEDULE_CHECK	231	230	1	574	2	
ECS_SERVER	3	3		1142	381	
ECS_SERVER_CHECK	231	230	1	440	2	
ECS_START_SERVER	2	2		81	41	
ECS_TEST_ABAP	25	25		631	25	
ECS_TEST_BDC	7	7		299	43	
ECS_TEST_ENQUEUE	17	16	1	8408	526	5264
ECS_TEST_FAILURE	5		5			
ECS_TEST_FILE_CMD	2		2			
ECS_TEST_LOG	1	1		3	3	
ECS_TEST_SERVER_ENQ	3	3		4193	1398	2727
ECS_TEST_START_EXIT	1	1		3	3	
ECS_TEST_USER_START	1	1		3	3	
XDOC FILE POLL	69	69		253	4	

6.6.3.4 Summary by Date View

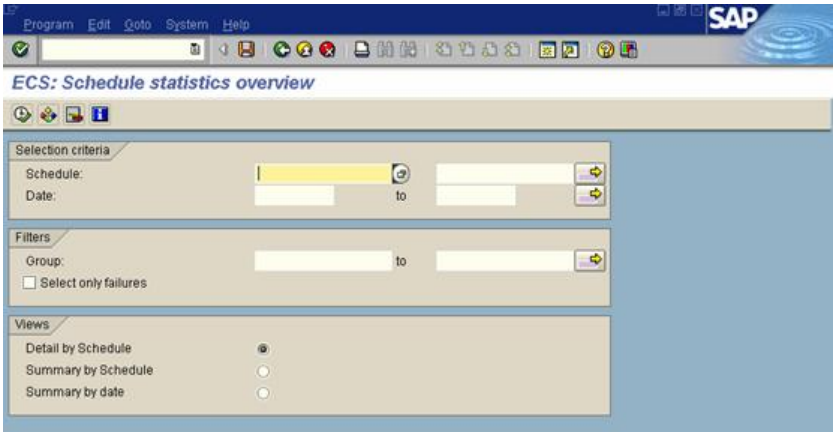
All statistics are summarized to display a trend by date.



Date	Start	Clean	Fail	Time	Avg	Enq
01.07.2000	49	49		97	2	
02.07.2000	50	50		112	2	
03.07.2000	66	65	2	177	3	
04.07.2000	85	82	3	13515	165	7646
05.07.2000	55	53	2	449	8	
06.07.2000	91	87	4	1635	19	345
07.07.2000	63	63		142	2	
08.07.2000	52	52		116	2	
09.07.2000	50	50		129	3	
10.07.2000	51	51		148	3	

6.6.4 Schedule Statistics

The statistics are summarized by Schedule by date. There is a simple view of the statistics screen available from the "Schedule Manager" and "Schedule run tree". To cater for more sophisticated reporting and trend analysis, you can download statistics in a CSV format and import into a reporting tool. Select the calculator icon from any of the main report displays and the following selection screen appears:

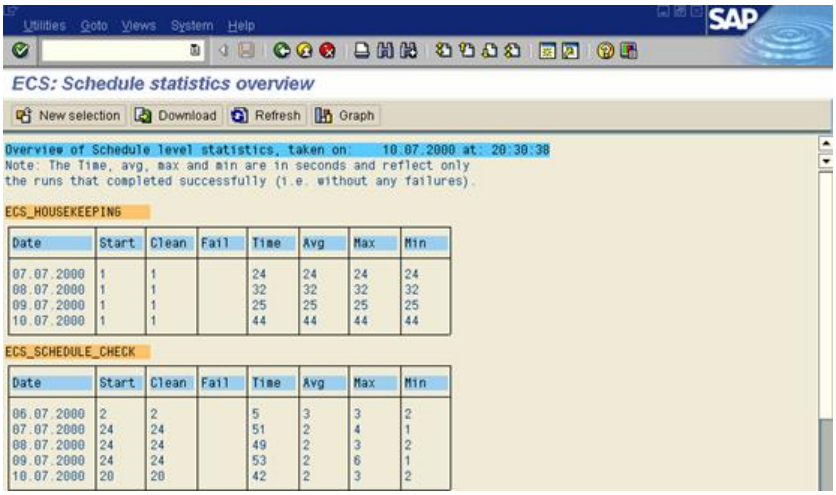


This selection screen allows you to restrict the selection of statistical data and an initial view. You may change the view dynamically at any time.

> ***Note:*** You may select the statistics on specific filter data. Once you execute, the following views are available:

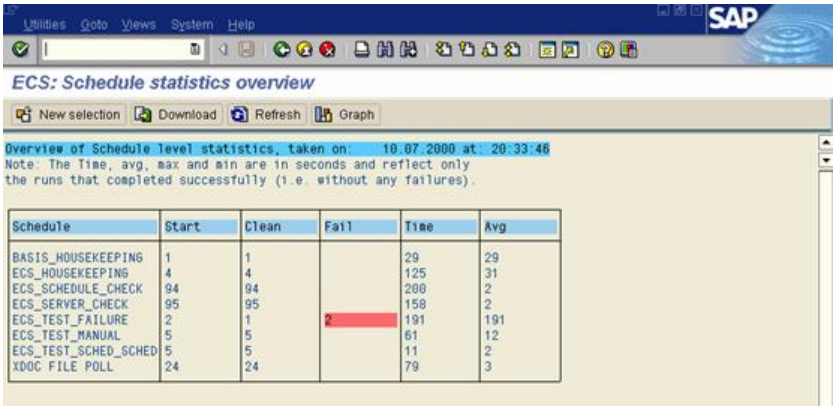
6.6.4.1 Detail by Schedule View

This view summarizes the run and time statistical data for each Schedule by date.



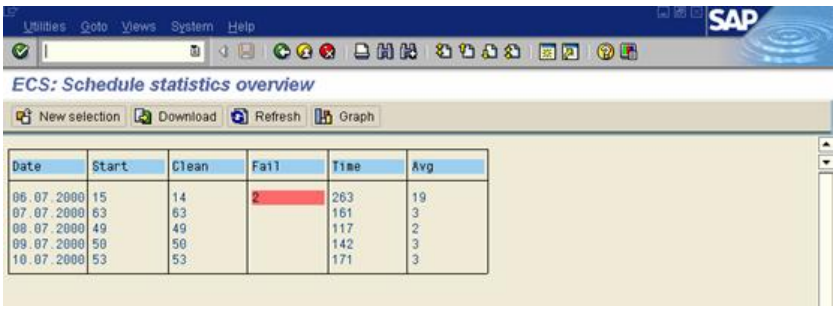
6.6.4.2 Summary by Schedule View

All run and time statistics are summarized, for the selection criteria, at the Schedule level. This is useful to highlight potential performance bottlenecks and problem areas.



6.6.4.3 Summary by Date View

All statistics are summarized to display a trend by date.

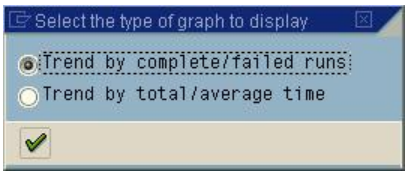


6.6.5 Graphical Statistic Analysis

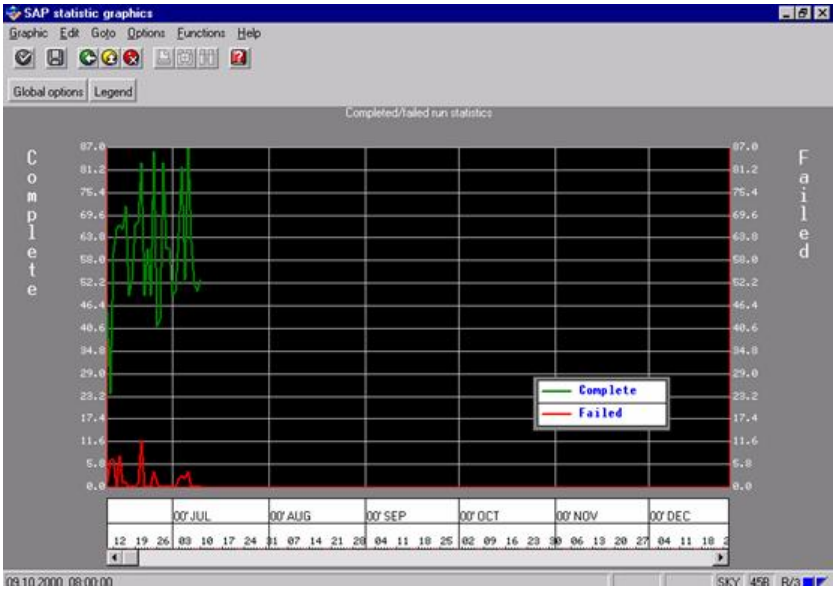
You may also use many views o generate into statistical graphs that are useful to show trends. The Graph icon is used to automatically invoke the generation of a graph.

> *Important:* Not all views have the graph option.

If you can generate a trend graph, you use issue the following prompt:



This prompt determines whether the graph trend is to by run, or time values. Once selected, a trend graph is generated. For example:



You may change the graph attributes, that is, colors, scale, dynamically through the options menu, for example, you may use **Options >Time units** to adjust the scale.

6.6.6 Statistics Download

List the statistics that require download on the run statistics report. Select the **Download** icon 

on the application toolbar of the list display. A dialog to create the file name on the PC or Network appears. The statistics are extracted and written out in CSV (comma delimited) format. The first line contains the column field names. The format is as follows:

6.6.6.1 Example Format of Process Summary Statistics

Column Name	Description
Process	Process name
Date	Run date CCYYMMDD
Filter_field/data	Filter field definition and data. ***Note:*** You may generate multiple lines of statistics for each unique occurrence of the filter field definitions.
Total_ complete/error	Total number of completed runs and failures
Total/avg/max/min times	The total, average, maximum and minimum times in seconds
Total_enqueue	Total time the process is enqueued in seconds

6.6.6.2 Example Download File

Example

```
process,date,filter_key,filter_data,total_complete,total_error,total_
time,avg_time,max_time,min_time,total_enqueue
```

```
ECS_POLL_DIRECTORY,19990507,,,000000000000008,000000000000001,
000000000000051,000000000000006,000000000000013,
000000000000003,0
ECS_POLL_DIRECTORY,19990508,,,000000000000022,000000000000000,
000000000000147,000000000000007,000000000000050,000000000000001,
0
ECS_POLL_DIRECTORY,19990509,,,000000000000005,000000000000000,
000000000000123,000000000000025,000000000000109,000000000000002,
0
ECS_SERVER,19990501,,,000000000000041,000000000000003,000000000029521
,000000000000720,000000000001764,000000000000391,0
ECS_SERVER,19990502,,,000000000000008,000000000000000,000000000003269
,000000000000409,000000000000820,000000000000309,0
ECS_SERVER,19990503,,,000000000000050,000000000000002,000000000062181
,000000000001244,000000000018034,000000000000308,0
ECS_SERVER,19990504,,,000000000000048,000000000000000,000000000076753
,000000000001599,000000000018081,000000000000313,0
ECS_SERVER,19990505,,,000000000000047,000000000000000,000000000080231
,000000000001707,000000000041545,000000000000318,0
ECS_SERVER,19990506,,,000000000000046,000000000000009,000000000071860
,000000000001562,000000000015672,000000000000306,0
ECS_SERVER,19990507,,,000000000000052,000000000000007,000000000084930
,000000000001633,000000000013377,000000000000308,0
```

6.7 Online Documentation

It is important to document ECS Process and Schedule definitions in both business and technical terms. It is equally important to document what to do when something goes wrong. You can document any aspect of an ECS Process or Schedule definition using an external editor (for example, MS-Word). You can create, maintain and display this documentation from within the ECS application in SAP/R3, that is, ECS invokes the desktop editor. You may hold documentation may only at the Process and Schedule levels and is not available for sub-levels, that is, Phases and Schedule levels / items.

6.7.1 Specifying a Documentation Directory and Editor

Before you can document Processes and Schedules in this manner, you must configure a documentation directory and documentation editor in the ECS system defaults. You may change the directory name at any time. It is important to consider the sharing of documentation across a network. The documentation directory name is specified in the 'System configuration' menu option from the setup menu on the main ECS screen. If no directory is specified (blank), the documentation feature is disabled. The documentation editor is configured in table YECS_WINRG, that is, the Windows registry definition (MS-Word is provided by default).

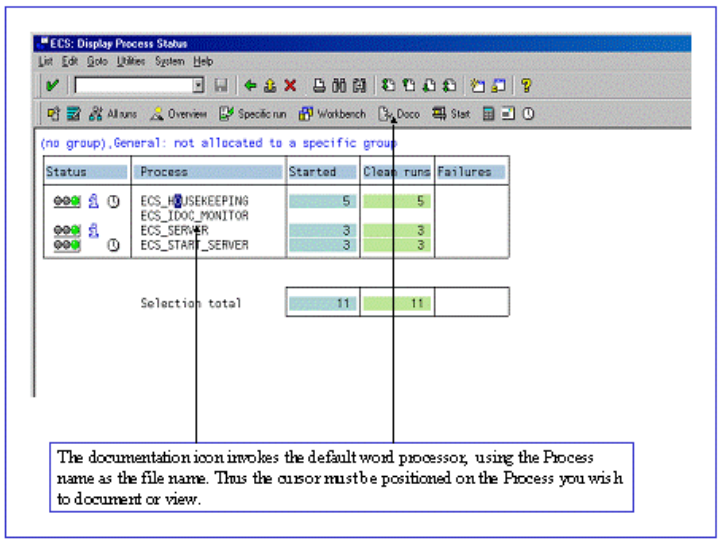
6.7.1.1 Naming Conventions

You may specify any valid directory name. The naming convention for each file created, is {Process name}.{default extension}.

> ***Note:*** You may have a Process or Schedule name up to twenty characters long. Because of this, only windows environments and word processing programs that handle long files names are supported.

6.7.2 Using Documentation

You invoke all maintenance of the documentation from within SAP R/3. The **Documentation** icon is on the application toolbars of the Process status, Run tree, Workbench and Schedule Manager screens. Just position the cursor on the name that require documentation and click the **Documentation** icon.



6.8 File Management

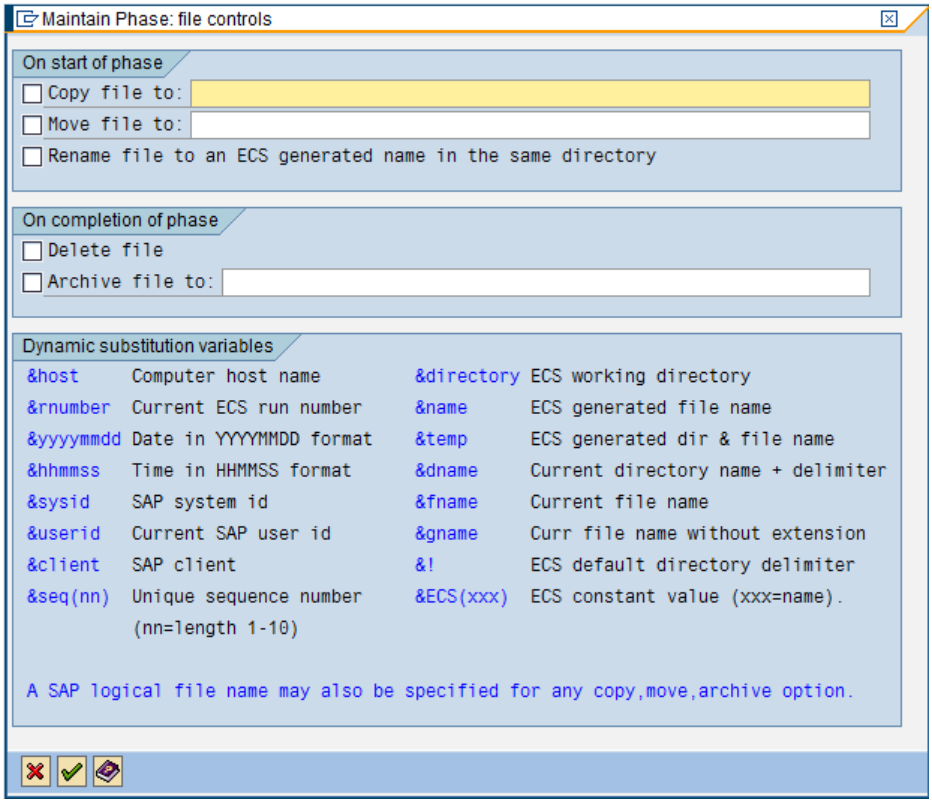
ECS provides a number of utilities to manage and access files that the interfaces use.

6.8.1 Key Topics

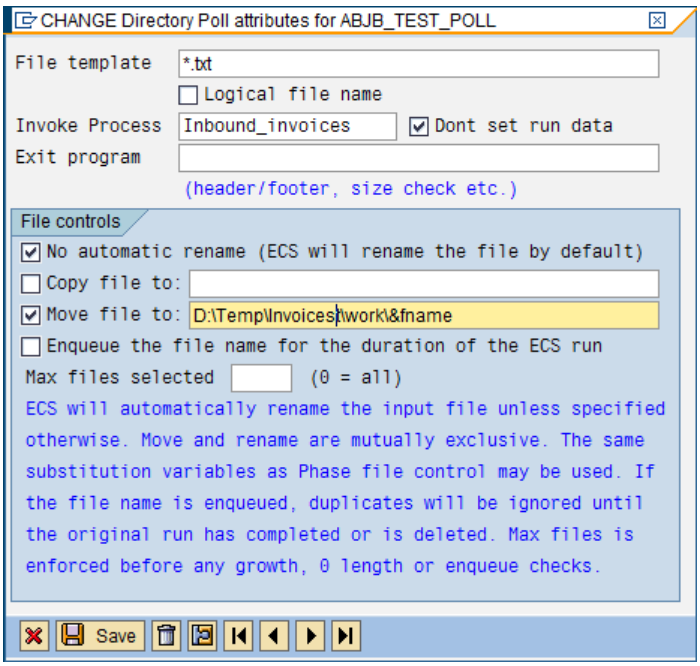
Automatic File Management
File Manager
Browsing Files
Poll Directories

6.8.2 Automatic File Management

You may configure ECS to automatically copy, rename, move and delete the primary file when a Phase is started and/or ended. See the File control options on the phase definition and the ABAP programming guide for more details. In addition, you may configure the ECS Directory Poll definitions to perform automatic copies and moves, before a Process starts.



> ***Note:*** You may configure a Directory Poll attribute to copy or move files similar to Phase entry and completion file controls. By default, files are automatically renamed unless configured otherwise.



6.8.2.1 Dynamic Variable Substitution in Logical or Physical File Names

You may imbed the following variables in any of the file names and the ECS file manager can dynamically substitute at run time.

> ***Note:*** You may specify either a SAP logical file definition or a direct physical file in any ECS file manager function. You do not require a special identifier for logical files, the ECS file manager always attempts to resolve the name as a SAP logical file name first.

Variable	Description
&HOST	SAP instance hostname
&RNUMBER	Current ECS interface run number
&YYYYMMDD	Current date in YYYYMMDD format
&HHMMSS	Current time in HHMMSS format

Variable	Description
&SYSID	SAP System ID
&USERID	Current SAP User ID
&CLIENT	SAP Client number
&DIRECTORY	ECS working directory
&NAME	ECS generated filename
&TEMP	ECS generated directory and filename
&DNAME	Current directory with trailing directory delimiter
&FNAME	Current filename
&GNAME	Current filename without extension
&!	ECS default directory delimiter
&SEQ(NN)	Unique sequence number (nn = length 1 - 10)
&ECS(XXX)	ECS constant value (xxx=name)

6.8.2.2 Logical File Names

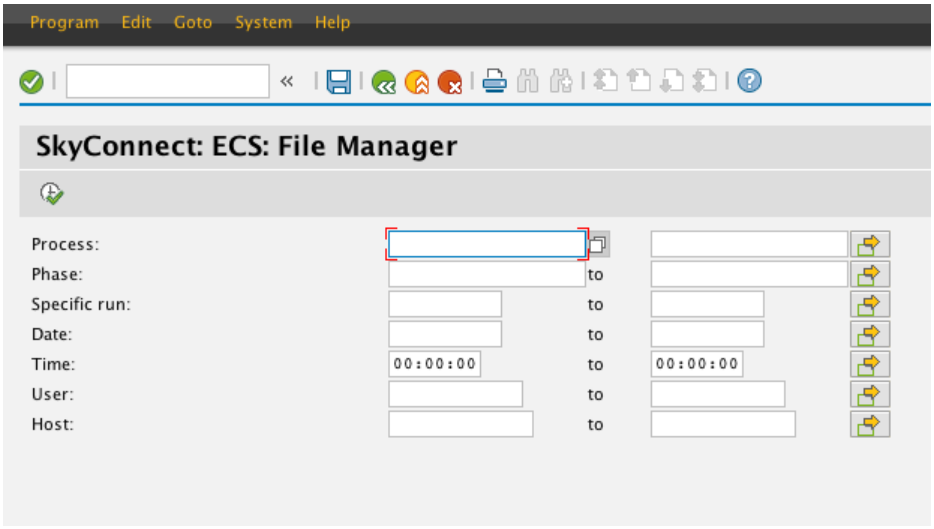
You may optionally specify up to two parameters for a SAP logical file name. These are identified by the '<' and '>' delimiters, for example, TEST_LOGICAL<parm1><parm2>. You may specify the standard ECS substitution variables within the parameters, for example, TEST_ LOGICAL<&runtime>.

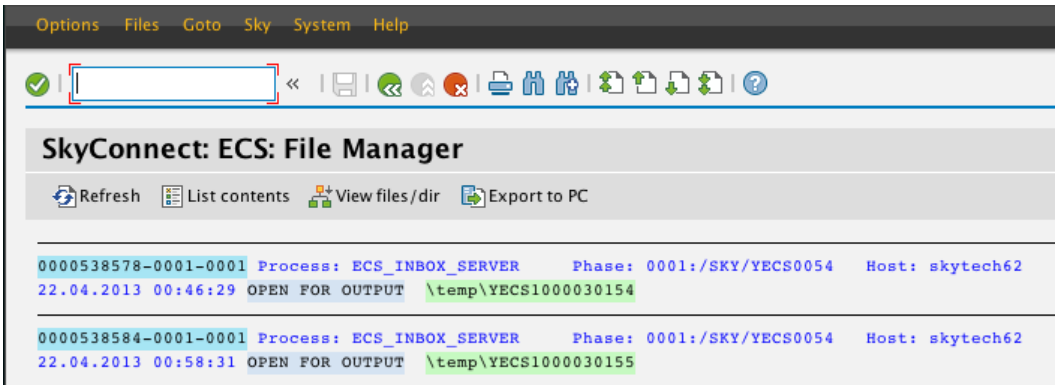
6.8.3 ECS File Manager

When you use the ECS external file commands (see the section External File Management), or configure the automatic file rename and archive options for phase definitions, ECS keeps track of the input and output files. You may display these file relationships using the File Manager. In addition, the File Manager provides a central point to view the contents of files / directories and download files from the Host to a PC.

> ***Note:*** All files are expected to be in "in text mode" (that is, fixed record format). If the file contains variable length records, you may unpredict the results. In a multi-instance SAP R/3 environment, the file creates on the machine where the program executes. In order for subsequent processing to be possible, either make sure that the files directory is shared across all the machines, or nominate a specific host for all processing against the file. If the ECS File Manager is used, the you must be able to access the file from all machines where batch and dialog processing is possible.

Select the **File Manager** option on the **Utilities** menu on any of the main screens, or proceed directly through transaction YECF.





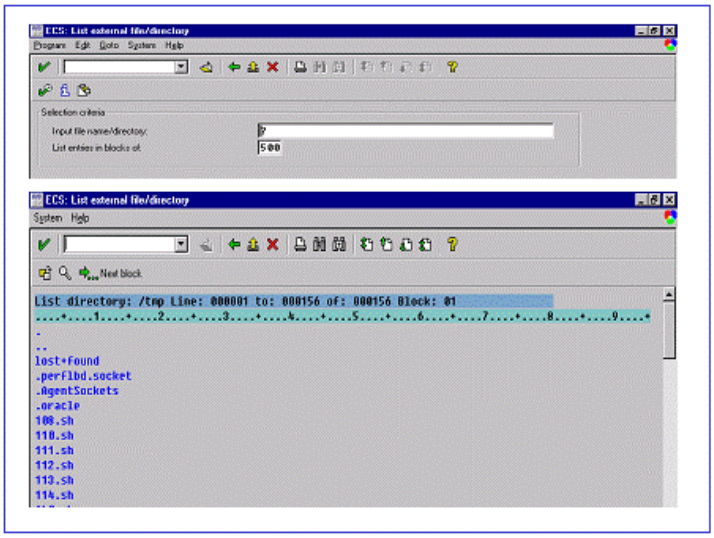
6.8.3.1 Downloading Files to PC

To download any file from the File Manager screen to PC, position the cursor on the file that require download and select the download to PC icon. A dialog for the name of the PC file to create appears and the contents are subsequently written to the PC.

> ***Note:*** This facility is not designed for high volumes of data. For large files, a file transfer utility is a more efficient option.

6.8.4 Listing File and Directory Contents

ECS provides a facility to navigate external directories and list file contents. You may start this utility from the File Manager using the **View files/Dir** icon or through the **List Files/ Directory** option on the **Utilities** menu from any of the main screens. To navigate to the next level in a directory or list a file contents, double-click a line in the list. The list displays contents in blocks. The number of entries in each block is specified on the selection screen. The next block appears by clicking the next block icon.



Navigate to the next level of a directory or list the contents of a file by double-clicking the line. The displays are stacked, so "back" returns to the previous display. You may specify wildcard names in the selection criteria to further limit the display. For example, /tmp/abc* lists all files in `/tmp` starting with "abc". In addition to ' * ', you may use ' + ' as a generic placeholder, for example, /tmp/abc+++g* lists all files in `/tmp` that start with 'abc' and have a 'g' in position 7.

6.8.5 The Directory Poll Program

ECS has a standard function that you can configure to monitor one or more directories for file names and automatically start a Process if a name matches a specified pattern. This feature is especially useful for inbound and outbound interfacing with SAP R/3 using files.

> ***Note:*** You may configure the Directory Poll program to automatically copy, rename or move the file when starting the associated Process.

6.8.5.1 Concept

A "poll definition" is created that:

- Defines the directory to list
- Includes the file name patterns that are to be checked for
- Specifies the ECS Process that starts the processing of the file

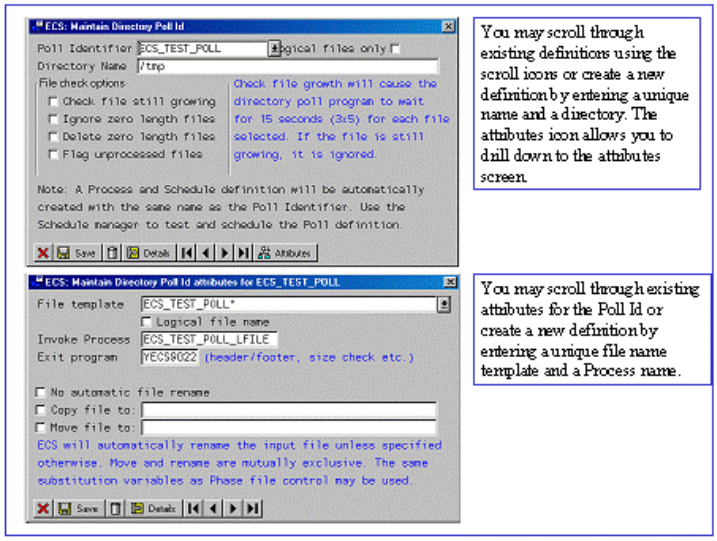
Once the "poll definition" is created or changed, a ECS Process and Schedule definition is automatically maintained with the same name as the poll definition. The Schedule definition is only created once and is ignored for any change operations.

> ***Note:*** There is a one-to-one relationship between a poll definition and a directory, however, you may configure many file name patterns.

6.8.5.2 Creating and Maintaining Poll Definitions

A poll definition consists of a poll id record and one or more poll id attribute records. You may create these definitions at any time through the **Directory Poll Defn** option on the **Setup** menu on the ECS Workbench screen.

Note: The Poll identifier is the key of the first screen (Maintain Poll ID) and just entered on its own, retrieves any existing directory name data. Similarly, the Poll Identifier and File template make up the key of the second screen (Maintain Poll attributes) and if just entered individually, retrieves any existing Poll Attribute data. Use the scroll icons to navigate and the attribute icon to drill-down.



6.8.5.3 Screen Flow and Fields

You need to create the Poll id record, first. This defines the host directory to monitor. Then one or more Poll id attributes records are created to define what the poll directory program does with any files located in the directory. You may define one or more of these attribute records for any single Poll id.

Poll Identifier

This is a unique name up to 20 characters long.

Directory Name

This is the host directory to monitor files. A full directory name is required.

File Template

Each attribute record is uniquely identified by a file template. The file template may be a specific name or contain a pattern, that is, * stands for any string and + for any character. For example, ECS* selects all files that start with the name ECS. GL+++SAP* selects all files that start with the letters GL in position 1-2 and SAP in positions 6-8. File names with a "." (dot) prefix are ignored.

Invoke Process

This is the Process that automatically starts and passes the full file name when a match is found. The file name is passed to the Process using both the run data string and file name parameters.

No Rename

The default is to rename the input file to a ECS generated name in the same directory when the Process is started. You may avoid this by specifying "no rename".

> ***Note:*** If the file is not renamed, it may be re-processed (duplicated) the next time the directory is polled.

Copy and Move

You may use these external file management options to automatically copy or move the input file to another destination. You may use dynamic substitution. See External File management for details.

Displaying Poll ID Details

It is difficult to analyze all settings using the entry maintenance screens. Use the **detail** option to list the attributes of the specified Poll id definition.

6.8.5.4 Scheduling the Directory Poll Program

An ECS Process and Schedule definition is automatically maintained whenever the poll definition is maintained. The Schedule definition is only created once. The ECS Process and Schedule definitions are generated with the same name as the Poll definition. You need to configure the Schedule definition and activate using the ECS Schedule manager.

Important: Do not serialize the Directory Poll Process definition. You cannot schedule the processes that are serialized. See the sample Poll id ECS_TEST_POLL that monitors directory `/tmp` and start process ECS_TEST_POLL_LFILE for each file found that contains 'ECS'. The sample Schedule definition ECS_TEST_POLL executes the directory poll program every 15 minutes.

6.8.5.5 Monitoring and Error Processing

The directory poll program YECS0010 writes a summary of actions and details of any error to the run and job logs. If a serious error is encountered, all subsequent processing is ignored. Once a Process starts for a file, any subsequent processing must be monitored in ECS.

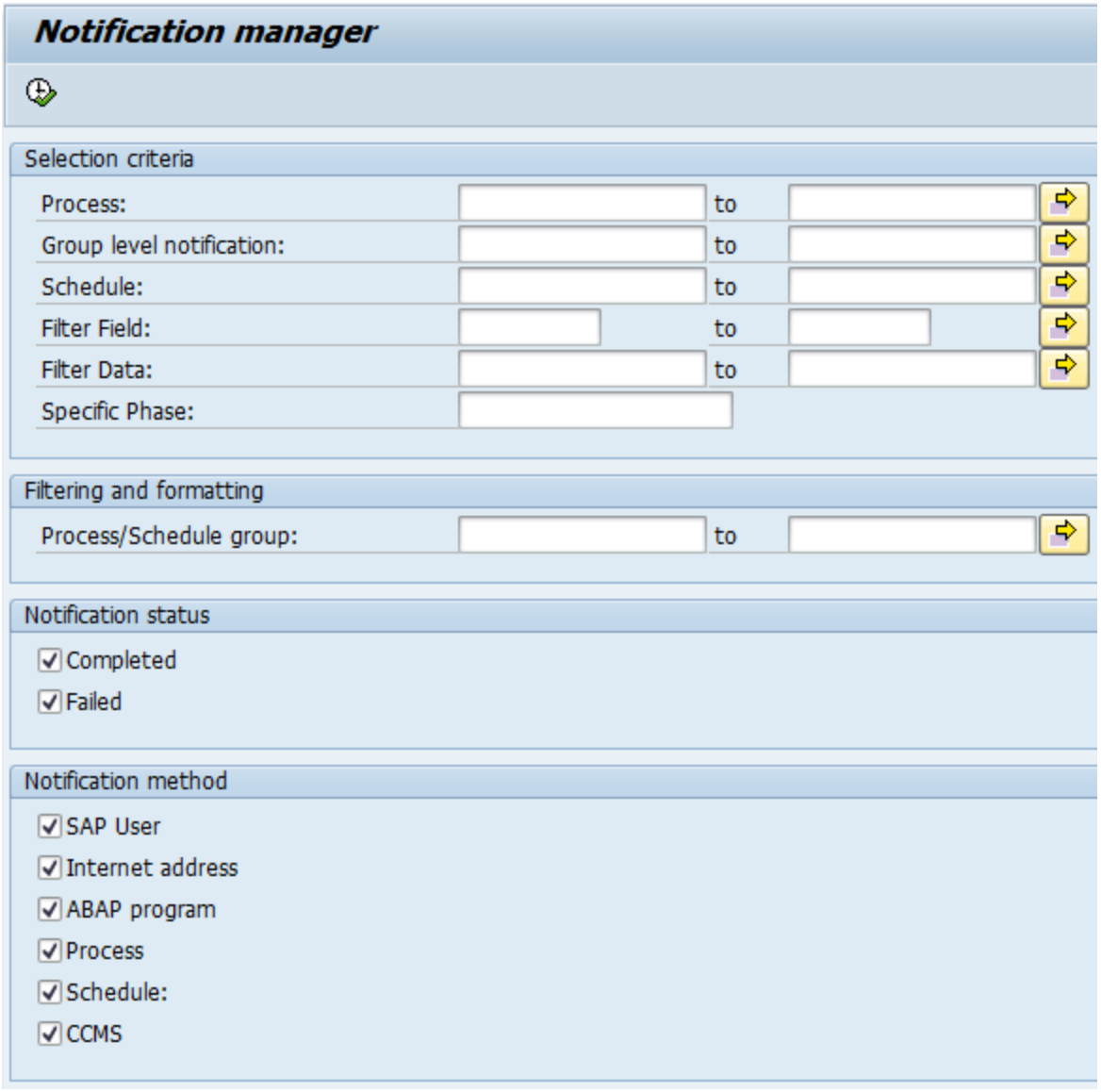
6.9 Notification Manager

The Notification Manager performs a variety or actions when components of a Process or Schedule fail or complete successfully. You may use the following notification methods:

- Send an email message
- Call a specialized ABAP/4 program with the details
- Start an ECS Process
- Start an ECS Schedule
- Publish message to CCMS (Activate monitoring to publish messages to CCMS)

The Notification Manager allows you to configure a list of users that are to receive notifications, the method used to notify them and the conditions under which the notification is triggered. Notifications are evaluated by ECS end Phase processing that invokes the Notification manager. It is important to remember that whilst notifications are used primarily to communicate failures, you may also configure them for successful completions.

To start the Notification Manager, execute the YECN transaction or select the option from the 'GOTO' menu on any of the main screens.



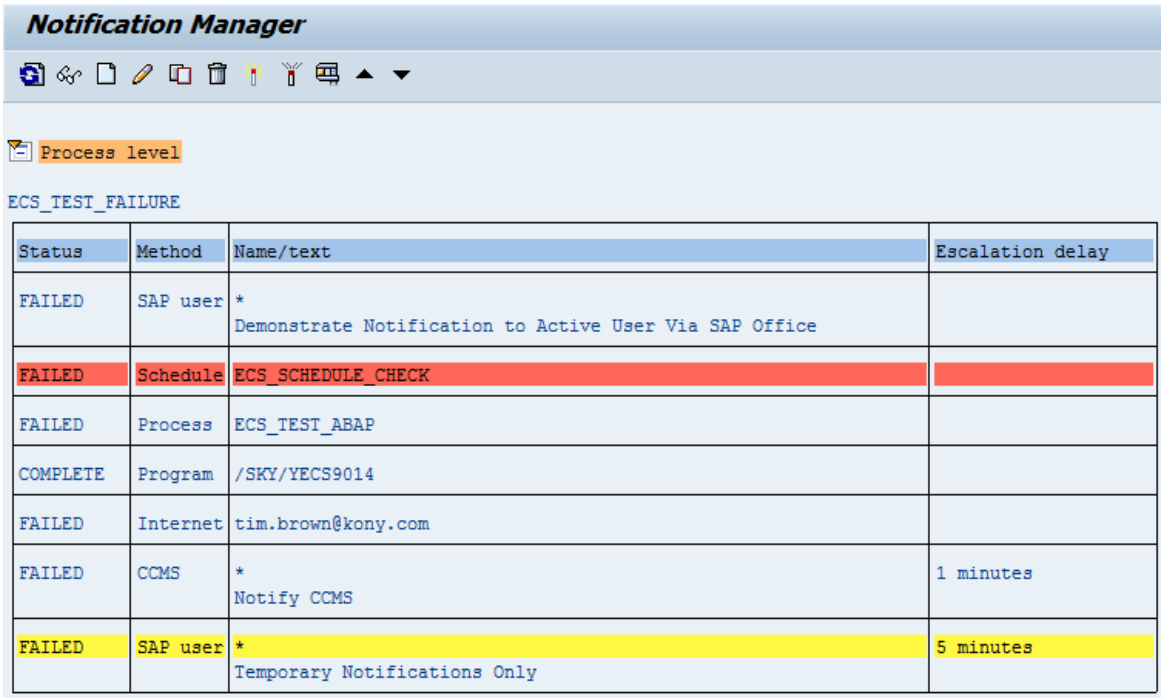
6.9.1 Key Topics

Maintaining Notifications

Notifications Exit

6.9.2 Maintaining Notification Entries

The Notification maintenance screen allows you to create, change and delete notification entries for a Group, Process or Schedule. Position the cursor on the required item on the list and maintain using the icons on the application toolbar. A pop-up then appears to capture the details of the notification rule, that is, the level of the notification, the status that the Notification is to trigger upon, the method for the notification.



6.9.2.1 Screen Fields and Icons

The list has control breaks on change of Group, Schedule and Process. The summary list displays the trigger status, notification method and the attributes. The screen is context sensitive. You position your cursor on a specific notification entry or control break on the list and click the appropriate icon on the application toolbar. For example, to change an existing notification entry, place your cursor anywhere on the list line and select the change icon.

List Colours

Entries highlighted in yellow are temporary notification entries that did not perform an action yet. Red lines are disabled notification entries.

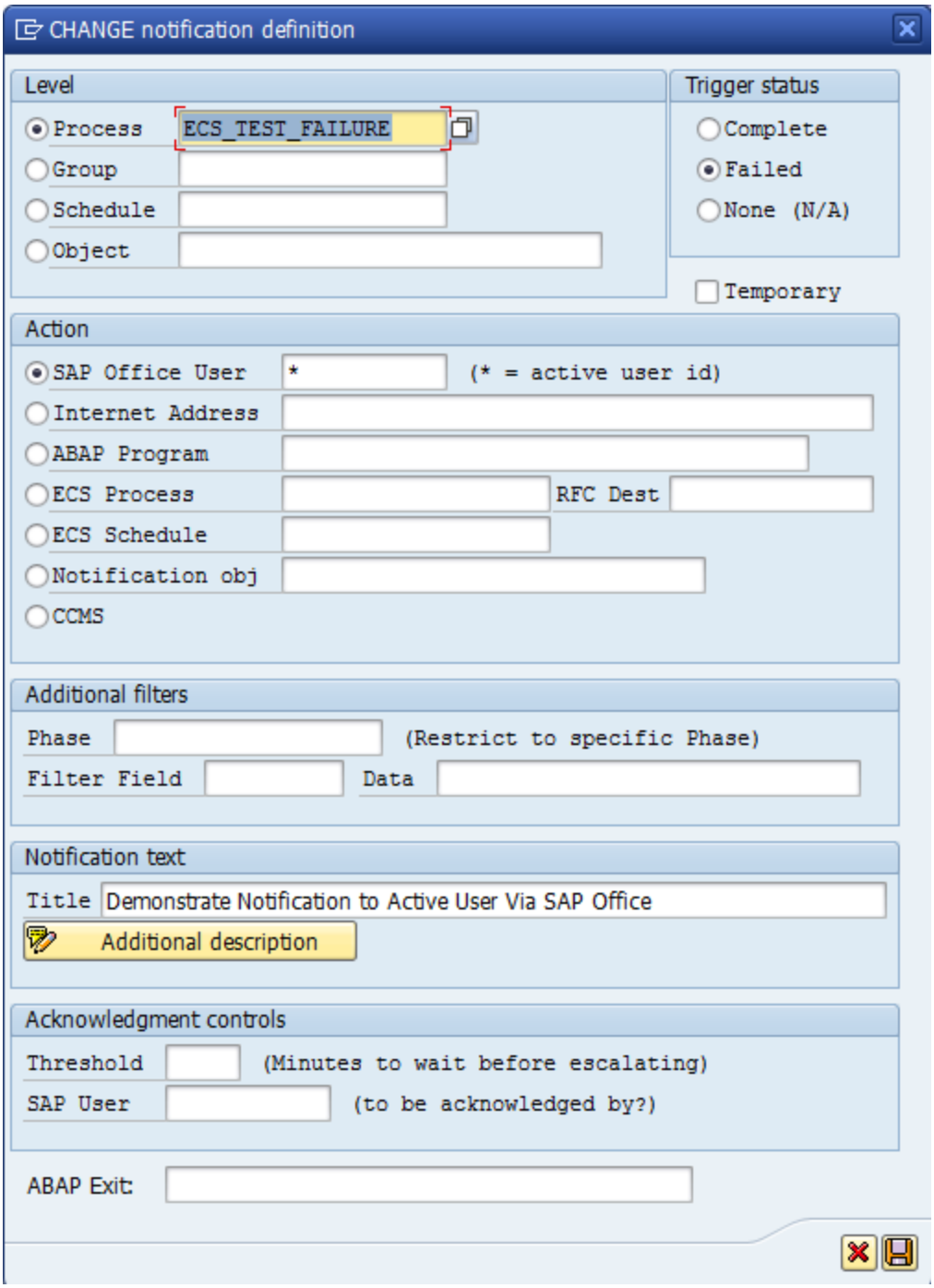
Enable / Disable

You may disable notification entries from use, that is, kept but ignored by the Notification manager. This is useful in cases where notifications are optional and are activated manually when required. The Notification manager automatically disables temporary notifications, once they are processed.

Testing Notifications

You may test an individual notification by positioning the cursor on the item in the list, and selecting the test icon from the application toolbar. The notification manager processes the appropriate method with test data.

6.9.2.2 Pop-up Maintenance Window



6.9.2.3 Screen Fields and Icons

Level

You may specify notifications at various levels, for example, if an ECS group is specified, any Process or Schedule allocated to the group is considered. Schedule level notifications take an action, either when a failure occurs within the schedule, or when the overall schedule is completed. A notification may also be restricted to a specific Phase within a Process.

Temporary

The **Temporary** check box indicates that you need to process the notification once only. After which, the Notification manager automatically disables the definition.

Trigger Status

You may issue a notification either when the selected level completes successfully, or fails. In most cases, you use notifications for failures, but it is important to note that you may use notifications to automate processing whenever a level successfully completes.

Method

Indicates to the Notification manager, what to process. For example:

- Send a message to a specific SAP user, (or current '*'), through SAP Office
- Send an Email message to an internet address (you must configure SAP to send Email)
- Execute a Notification exit ABAP/4 program. See following section, external program or API.
- Start another ECS Process (possibly on another R/3 system)
- Start another ECS Schedule
- Publish message to CCMS (Monitoring need to be active to publish messages to CCMS)

> ***Important:*** If using Email to an internet address, the user ID that the Process executes under must have an address type of email maintained in SAP office.

Filter Field

If you configure specific filter field and data, only notifications that are intended for all or those that match the filter specified are issued. This is an effective way to direct notifications to certain areas, for example, plant, business area. Notifications for specific filters are highlighted in the display.

Text

This is a free format description, used mainly to describe the context of the Notification definition.

6.9.3 Notifications Exit

ECS provides a programming exit to enable more complex interfacing with external third party monitoring products such as HP-Openview, and CA-Unicentre. You must write the exit in ABAP/4 and is entered as an option on the notification maintenance pop-up. In the future, we plan to provide standard ECS exit programs for all the major monitoring products.

6.9.3.1 Coding the ABAP/4 Exit Program

The following parameter structure is passed to the exit program using local memory ID ECS_ NOTIFICATION_EXIT. The program must import the following two structures:

YECS_NFYPG

Message line	CHAR 255	Return code from exit
RETURNCODE	Sy-subrc	ECS: Process name
INAME	CHAR 20	ECS: Phase name
PNAME	CHAR 20	ECS: Run number
RNUMBER	NUMC 10	ECS: Status code
STATUS	CHAR 10	ECS: Filter field
FILTER_FLD	CHAR 10	ECS: Filter data
FILTER_DTA	CHAR 32	ECS: Schedule name
SCHED_NAME	CHAR 20	ECS: Schedule run number
SCHED_RNUM	CHAR 10	Return code from exit

> ***Note:*** You can have STATUS either COMPLETE or FAILED. The schedule name and process name are mutually exclusive.

Declaring and importing the parameters

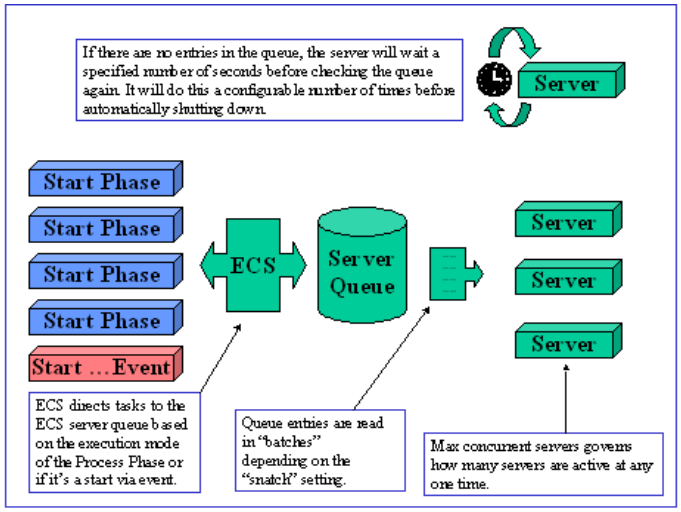
```
DATA:V_PARMLIKE YECS_NFYPG,
IT_MESSAGELIKE SOLI OCCURS 5 WITH HEADER LINE.

IMPORT PARAMETER TO V_PARM
MESSAGE TO IT_MESSAGE
FROM MEMORY ID 'ECS_NOTIFICATION_EXIT'.
```

Refer the sample ABAP/4 program YECS9014 for an example on how to accept parameter data from the Notification manager. You may then perform any type of processing required.

Take special care when coding the exit, as any abnormal failure fails the ECS end phase processing. Any performance problems with the exit impacts the response of the overall end phase processing.

6.10 ECS Background Server

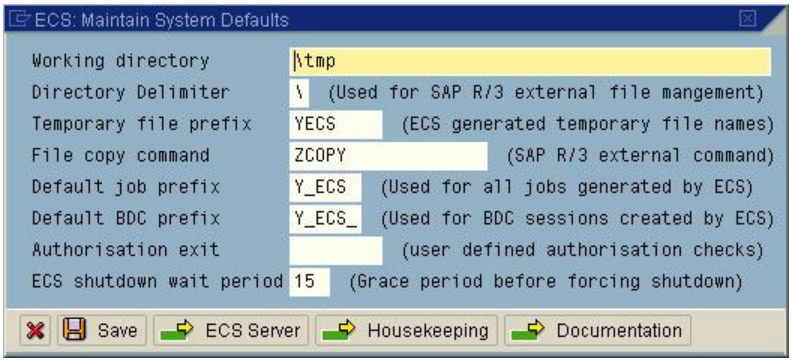
The ECS background server option is designed to process background tasks more efficiently. It is designed for environments that have a medium to high number of batch jobs that have a small execution time. The server function works by having one or more continuous processes running in the background that execute tasks that are placed in a queue. If there are no entries in the queue, the server waits and checks the queue periodically for work. This feature can dramatically reduce the number of jobs executed in the system and improve the performance of batch, because all the overhead of job submission, and scheduling is removed. ECS starts one or more servers automatically (depending on the workload) and automatically shut themselves down if there is no work to do. You can configure the maximum number of concurrent server processes and the amount of time to wait (idle time).



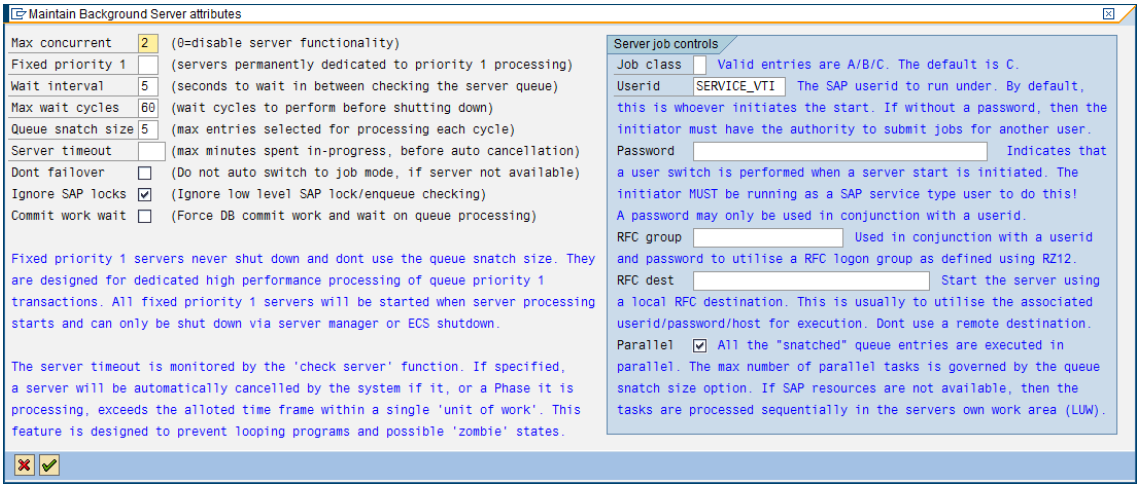
6.10.1 Configuration

6.10.1.1 How to Configure Servers

There are settings in the system defaults that govern the behavior of the ECS Server function. You may specify these through the **system defaults** option on the setup menu on the Workbench screen.



The configurations for the ECS background server are clearly marked at the bottom of the screen. Click this and the following pop-up appears.



6.10.1.2 Screen Fields and Icons

Max Concurrent

This configuration specifies the maximum number of ECS servers that may execute at any point in time. If you specify zero (0), this disables ECS server functionality and phases that are configured for the ECS Server are switched to run as asynchronous jobs. You may increase or decrease this value at any time.

Wait Interval and Max Wait Cycles

These two parameters determine how long each server has to wait for more work, before they automatically shut themselves down. The servers starts up automatically depending on the workload or if none are running. You are recommended to wait for an interval of 10 seconds and a wait cycle of 60 (10 minutes).

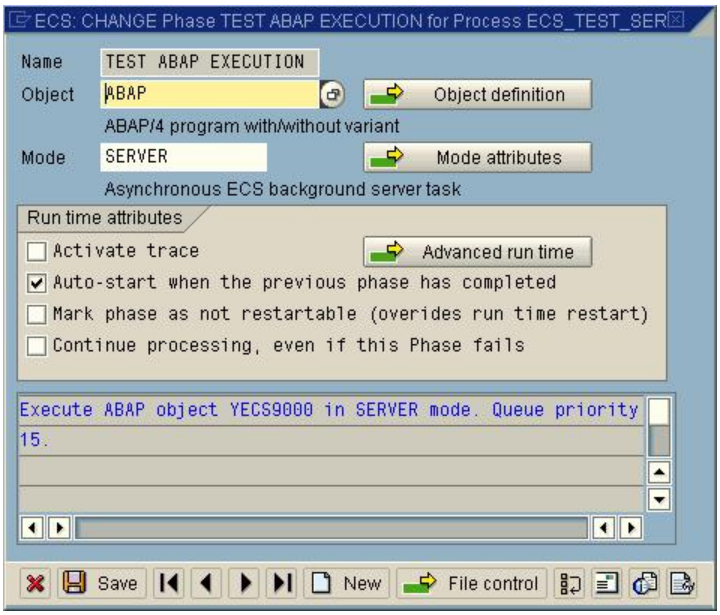
Queue Snatch Size

This is the maximum number of queue entries that each server reserves for processing at any one time. Servers can process tasks both sequentially (one after each other) or in parallel, thus to load balance the queue entries across multiple servers, it is possible to limit the number of entries that you can allocate and reserve. You are recommended a value of *5*. Increase this value if the tasks are short and are processed quickly without delay. Decrease if some tasks may delay processing. The queue snatch size determines the maximum number of parallel executions that are possible in each processing cycle.

> ***Note:*** You may also use queue priority configurations to load balance server tasks.

6.10.1.3 How are Process Phases Directed to an ECS Server?

Once you configure the ECS server function, you may configure those process phases that require processing by an ECS background server. You can do this by the specifying a execution mode of SERVER on the phase.



Configure the execution mode to ECS server. On pressing **Enter**, the queue control button appears. This allows a server queue priority to maintain (the lower the number, the higher the priority). Use the queue priority to make sure that high priority time critical requests are processed before low priority (near enough time) requests.

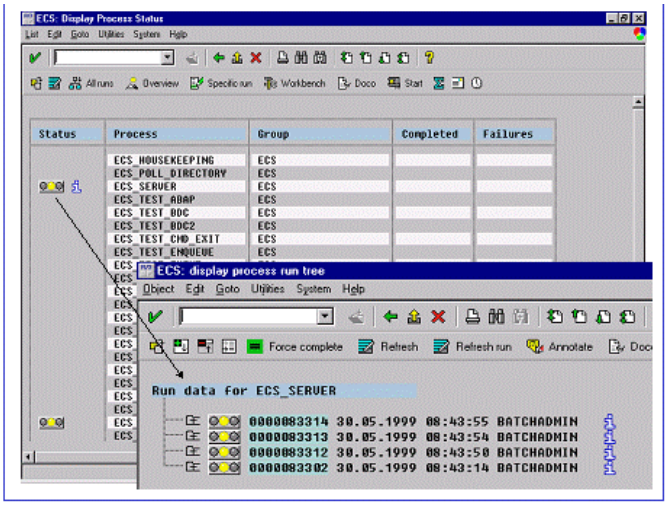
> ***Note:*** If you disable the server functionality, the Phases configured to execute by ECS background server are executed as asynchronous jobs. If this fault tolerance is handled automatically, no further action is required.

6.10.1.4 ECS Event Processing

ECS background server functionality is used internally to automatically start ECS Processes that are initiated using the ECS_START_PROCESS_VIA_EVENT command. This is mentioned here, because you may use the Server Manager to view and diagnose event queue requests. There is no configuration required for event queue processing, but you may decide to implement ECS server functionality or increase the number of servers to cater for situations where there is a high workload of event processing.

6.10.1.5 The ECS Server Process Configuration

ECS background servers are executed as normal ECS Processes. The Process name is ECS_SERVER and you may view its runs and manage just like any other ECS Process.



Important: Because the server executes in the background and may process many different types of transaction, it must have the necessary authorization in SAP to perform all the tasks allotted to it. By default, the server takes on the authority of the user id that issued the ECS_START_PROCESS that caused the server to start. You are highly recommended that you configure a specific user id for the server to use for this processing. To do this, maintain the RUN SERVER PROGRAM phase of the ECS_SERVER process, specifying the user ID to use through the job attributes. If the server performs dynamic SAP user switching, then it must run under a SAP service type user.

6.10.2 Parallel Processing

ECS background servers can support SAP parallel processing, that is, execute multiple phase runs off the queue at the same time. To do this, you must first enable parallel processing in the background server configuration and then flag individual processes and phases (server mode attributes) as parallel capable. You can configure an RFC processing group optionally to help limit the scope of the SAP resources (dialog processes) to specific SAP instances. Read SAP documentation on defining and using RFC groups for more information.

> ***Note:*** Parallel processing takes up SAP dialog processes and thus suits short sharp transactions. You can achieve Significant performance gains with this feature, that is, utilizing the distributed processing power of the whole SAP NetWeaver environment to process interface entries. Like anything, you must think through and design properly the usage and impact. It is a good idea to include basis in this process.

The maximum number of parallel tasks that a server kicks off at any one time is determined by the configured "snatch size", that is, how many queue entries the servers read off the queue per processing cycle. Parallel runs then start first, followed by any serial runs, that is, executed sequentially by the server one after the other. If there are not enough SAP resources available (dialog processes), a message is issued to the run log to this effect and the run executes serially in the servers own LUW as per the default mode.

> ***Important:*** If you intend to use parallel processing, ensure that the SAP system has enough free dialog processes to meet the configured "snatch size". Lack of SAP dialog processes do not cause the runs to fail, but defeats the purpose and the performance gains are not met.

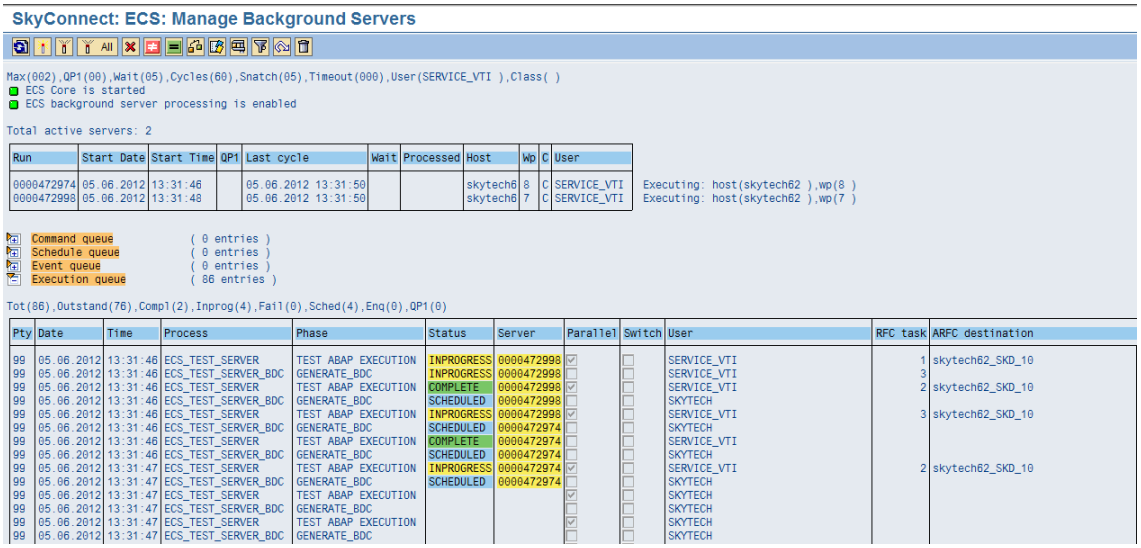
6.10.3 Switching SAP Users in Server Mode

The ECS background servers support the dynamic switching of SAP users. To do this, you must start the ECS background servers as a SAP service type user and a valid SAP userid and password, and you must supply when starting the process or phase. To perform a user switch at run time, the server spawns an asynchronous RFC task and thus requires a free SAP dialog task to do this. If there are no SAP resources available at the time, the run is fails with log messages to that effect.

> ***Note:*** If you are using a significant amount of user switching, ensure that there are enough SAP dialog processes available. If you use in conjunction with parallel processing, this is generally not a issue as the user switch occurs in the same parallel task. If not, at least one free dialog task must be available.

6.10.4 The Server Manager

Start the server manager through the **Server Manager** option on the **Utilities** menu on any of the main ECS screens or through transaction YECB. The Server manager displays the number of servers that are active, any active queue requests, any outstanding commands that are issued against the servers as well as allowing the ability to control server processing. The active server commands start, stop, enable, disable and cancel are available from this utility.



6.10.4.1 Active Servers

This list is of the current ECS_SERVER Processes that are active (in-progress). Double-click the line to display the run details for a particular server. The run number of the server is used to identify queue entries that are currently processed. Commands are issued against a particular server by placing the cursor on a specific server and clicking the relevant icon in the toolbar.

6.10.4.2 Queue Entries

The queue entries appear in priority order, that is, the order in which they are processed. The Process and Phase run details appear by double-clicking the queue item line. The current status of the entry may have the following values:

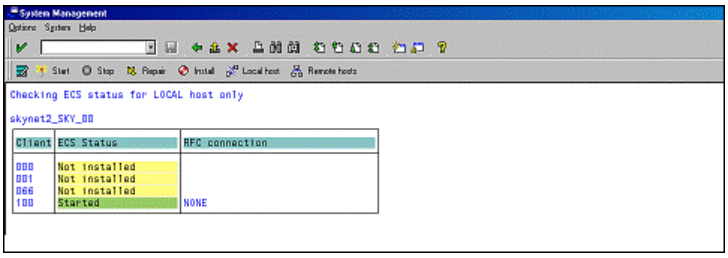
Value	Description
INPROGRESS	The task is currently executed
SCHEDULED	The task is scheduled for execution (snatched)
ENQUEUED	The task is dependent on a enqueue
[blank]	The task is not yet detected by a server
FAILED	The task is failed by a server. See the annotation logs of the server and the associated run for more details.

6.11 System Management

The ECS system management options are used to control the availability of the ECS system and repair ECS processes after an abnormal system failure. You can use the startup and shutdown options in conjunction with SAP startsap and stopsap processing to ensure the integrity of all ECS background processing. Access the **ECS System Management** menu from the **System Management** option from the **Utility** or **Main** menu or through transaction YECZ.

6.11.1 The System Management Menu

You can access the **System management** menu using the YECX transaction, the ECS main menu or through the **System management** option on the **Utilities** menu.



6.11.1.1 Shut Down ECS

This procedure disables ECS processing. Any phases that started after shutdown are deferred in a queue (yecs_defer). The deferred entries are automatically re-submitted when ECS restarts. The shutdown process disables ECS background server functionality causing any active servers to shut down. You may configure a "grace period" in the ECS system defaults (ECS shutdown wait period) that suspends the shutdown process to allow currently active processes to end. After the "grace period", any active ECS processes are canceled.

> ***Note:*** Phases that execute in "synchronous" or "none" modes are not deferred, since ECS has no control over the LUW (logical unit of work). The error processing and subsequent restart logic are the responsibilities of the programmer.

6.11.1.2 Start Up ECS

This option enables the ECS system for background processing after ECS is shut down. It automatically enables ECS background server processing and re-submits any deferred runs that are queued whilst ECS was shut down. The ECS repair process also starts to check the integrity of the ECS system.

6.11.1.3 Repair ECS

The repair process may run manually at any time to:

- Check the integrity of the ECS database.
- Check the integrity of the run-time environment, that is, the status of active process and schedule definitions.

A report of the current status and all actions performed is produced. The repair is a cut down version of the normal ECS housekeeping process. A repair executes automatically as part of the ECS startup process. This is run asynchronously as an 'ECS_HOUSEKEEPING' process so as not to delay the startup process.

6.11.2 Shutting Down and Starting ECS from an External Program or Script

Shutting down and starting ECS from an external program or script involves calling the Y_ECS_SHUTDOWN or Y_ECS_STARTUP functions in SAP R/3 through the documented SAP R/3 RFC (remote function call) method. Consult the SAP R/3 help on RFC for details. The simple method is to execute the SAP supplied STARTRFC program that you can find in the rfcsdk library.

Example to shut down ECS from a Unix script

```
/.../startrfc              -h {host} -s {sys id} -u {user} -p
{password} -c {client}

                                  -f Y_ECS_SHUTDOWN
```

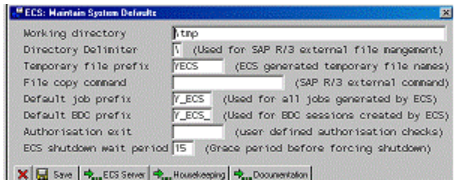
Example to start up ECS from a Unix script

```
/.../startrfc              -h {host} -s {sys id} -u {user} -p
{password} -c {client}

                                  -f Y_ECS_STARTUP
```

> ***Note:*** The location of the "startrfc" program may vary from site to site.

6.11.3 ECS System Management Configuration

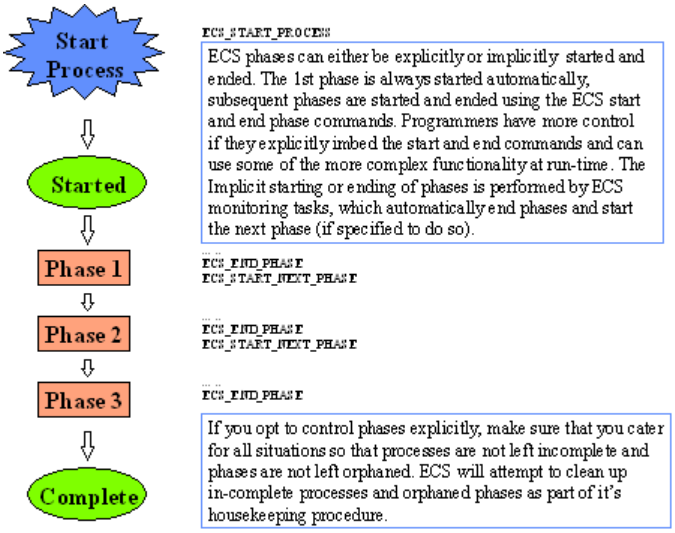
You may configure a "grace period" for the ECS shutdown in the ECS system defaults.



The grace period is configured in seconds. This parameter causes the ECS shutdown process to wait the specified number of seconds before you cancel any active ECS processes. This "grace period" is to allow all active processing to end before you force a shutdown.

6.12 ECS Programming Guide

There are many ways to initiate and control ECS Processes and Phases. This section describes each method in turn. It is up to the programmer to decide on the best method for the application. You must also decide, whether the program is to control ECS through imbedded calls to ECS, or whether ECS controls the execution of the Process and automatically starts Phases when pre-requisites are met; or a mixture of both. Automation though ECS is the preferred method since there is less effort involved and tends to be more modular. Imbedded calls to ECS may be necessary to take advantage of some of the more complex functionality.



6.12.1 The Basics

You may use ECS to execute ABAP/4 programs without any code change, but if you require interaction with the ECS monitoring facilities, logs, and pass parameters between Phases, you need to use ECS commands. ECS provides a powerful suite of ABAP/4 commands that perform a wide range of functions. For example:

ECS Command	Description
Include yecssdk	Contains all the ECS command definitions
Ecs_import_parameters	Imports the ECS parameter definitions from the start or previous Phase
Ecs_end_phase_as_ failed	Ends the current Phase as failed
Ecs_write_log	Writes a message to the ECS 'run' log
Ecs_export_parameters	Export the current ECS parameter values for the next Phase

Example

```
REPORT  DEMO1.

  INCLUDE YECSSDK.
........
  ECS_IMPORT_PARAMETERS.       " get parameters from start or previous
Phase
........
  IF SY-SUBRC NE 0.
     ECS_WRITE_LOG 'E' '1' TEXT-001.
     ECS_END_PHASE_AS_FAILED.  " indicate failure to ECS
```

```
    LEAVE PROGRAM.
  ENDIF.
.......
  ECS_EXPORT_PARAMETERS.       " Save parameters for next Phase
```

6.12.1.1 The ECS SDK

Before you can utilize the embedded ECS commands with an ABAP/4 program, you must include YECSSDK, for example, INCLUDE YECSSDK. ECS commands and variables follow a strict naming convention so as not to impact on the program.

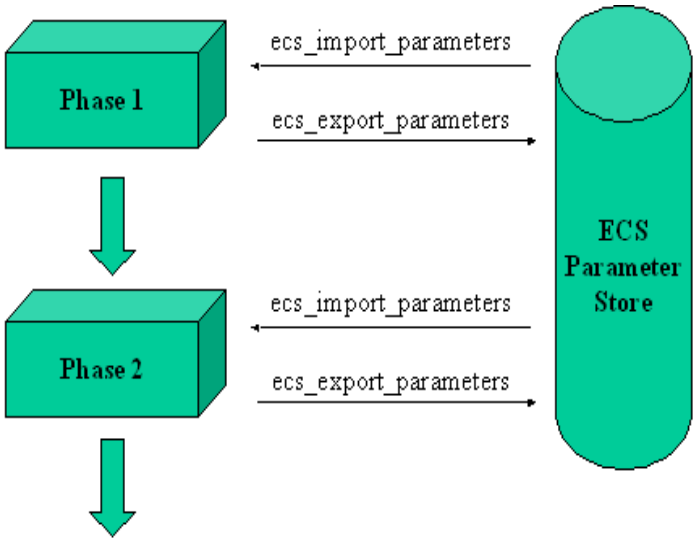
6.12.1.2 Important Note about using the ECS Commands

ECS commands are implemented using ABAP/4 macros. Macros are a convenient way to provide a simpler interface to the programmer with reduced coding required. However, there are some rigid rules about macro calls:

- The YECSSDK must be specified. This contains all the documented ECS macros.
- All parameters must be specified. If a macro requires three parameters, three must be passed.
- Parameters must be separated by spaces not commas.
- Either literals or variables may be used as parameters. Literal values must be enclosed in single quotes.
- ECS provides the result of the macro call in V_ECS_*RC (a good result is 0). This is automatically cleared and set within the macro, and consequently requires no manual intervention, however, error handling need to be catered for by the programmer.

6.12.1.3 Importing and Exporting ECS Parameters

ECS provides a standard mechanism to pass parameters between different Phases of a Process. In order to issue ECS commands to control the run, you must issue a ECS_IMPORT_PARAMETERS command. If you wish to pass data to the next Phase, you need to perform an ECS_EXPORT_ PARAMETERS command to save the current parameter data. The next Phase may then access the parameter data by performing an ecs_import_parameters.



> ***Note:*** The various mechanisms to maintain parameter data is fully explained in the section 'Passing data between Processes and Phases.

6.12.1.4 Internal Variables

The ECS SDK contains a number of internal variables that are both maintained automatically by ECS and may set programmatically. These variables follow a strict naming convention so as not to clash with any programming convention. All variables start with v_ecs_*...

> ***Note:*** See the section Internal variables and data types for a full list.

Variable	Description
V_ECS_*RC	The internal ECS return code. You should evaluate this after each ECS command. 0=ok.
V_ECS_*MSG	A formatted message, used to explain errors or provide extra information.
V_ECS_ *RNUMBER	The current ECS run number

Variable	Description
V_ECS_ *RSEQUENCE	The Phase number within the current run (rnumber)
V_ECS_ *PHASE_NUM	The interation of the Phase (rsequence). This uniquely identifies each execution of the same Phase, that is, re-starts.

When you perform an ECS_IMPORT_PARAMETERS, the ECS internal variables are automatically populated. ECS automatically maintains these variables as commands are performed. In this way, programming ECS controls is dramatically simplified.

6.12.1.5 Checking Return Codes and Basic Error Processing

The results of every ECS command are recorded in V_ECS_*RC. Zero (0) indicates the command is successful and any other value indicates the command failed. V_ECS_*MSG contains the message.

6.12.1.6 Failing the Run

If processing fails unexpectedly and the program imports ECS parameters using ECS_IMPORT_ PARAMETERS, you may fail the current ECS run from within your program. You may achieve this using the following commands:

- ECS_END_PHASE_AS_FAILED
- ECS_END_PHASE_AS_REPROCESSABLE
- ECS_END_PHASE {status}

> ***Note:*** Refer Starting/ending Processes and Phases for more details.

6.12.1.7 Annotation

You may supply more information about the progress or problem with a run by using the ECS annotation feature. Simple commands are provided, so that you may construct an audit trail and view along with the ECS run. There are many options to format data, automatically copy list output and messages, indicate severity and structure the annotation log messages. Example commands:

- ECS_WRITE_LOG
- ECS_REFRESH_LOG
- ECS_APPEND_LOG
- ECS_COMMIT_LOG
- ECS_COPY_MESSAGE_TO_LOG.

> ***Note:*** See the section: Using the ECS run log to annotate, audit and trace runs for more details

6.12.1.8 Example Definitions and Code (Samples)

ECS provides many example 'sample' Process and Schedule definitions that you use to demonstrate features and coding techniques. These samples are referenced throughout this manual. Use the Process workbench (YECW) to view the ECS samples.

> ***Important:*** Remember to select the **Show ECS samples** prompt on the selection screen.

You may also restrict the display to just the ECS samples by selecting the ECS_SAMPLES group.

Sample Process	Description
ECS_TEST_ABAP	Execute a single ABAP/4 program.
ECS_TEST_BDC	Execute a BDC session.
ECS_TEST_BDC2	Demonstrate execution of a BDC session with embedded start / end commands.
ECS_TEST_BDC_ LEGACY	Execute a BDC with standard BDC interface calls.
ECS_TEST_BDC_ MULTI	Test multiple BDC sessions.

Sample Process	Description
ECS_TEST_ CHECKPOINT	Demonstrate checkpoint restart techniques.
ECS_TEST_CMD_ EXIT	Demonstrate the use of an external command exit.
ECS_TEST_ COND_CODE1	Demonstrate condition code techniques.
ECS_TEST_ CONFIRM	Demonstrate a confirmation Phase.
ECS_TEST_ COPYFILEPC	Copy a host file to the PC.
ECS_TEST_ COPYFILHOST	Copy a PC file to the host.
ECS_TEST_ ENQUEUE	Demonstrate enqueue techniques.
ECS_TEST_EVENT	Execute a process by triggering and event.
ECS_TEST_EXEC_ CMD	Execute an external command.
ECS_TEST_ FAILURE	Demonstrate an ABAP/4 program failure with short dump.
ECS_TEST_FILE_ CMD	Demonstrate file manager techniques.
ECS_TEST_LOG	Demonstrate the use of the ECS annotation log.

Sample Process	Description
ECS_TEST_ MULTI_STEP	Execute multiple ABAP/4 programs
ECS_TEST_ PARAMS	Demonstrate the various techniques for passing parameters between Phases.
ECS_TEST_ PARM_TBL	Demonstrate passing internal tables between Phases (not recommended (Use export / import data instead).
ECS_TEST_ PHASE_MSG	Demonstrate Phase Message Evaluation.
ECS_TEST_POLL	Demonstrate ECS Poll Directory definition.
ECS_TEST_POLL_ LFILE	List file contents (invokes by ECS_TEST_POLL).
ECS_TEST_POLL_ LOGICL	ECS Poll Directory definition with logical files
ECS_TEST_ SERIAL	Demonstrate Process serialization
ECS_TEST_ SERVER	Execute ABAP/4 program via ECS background server
ECS_TEST_ SERVER_BDC	Execute BDC through ECS background server
ECS_TEST_ SERVER_ENQ	Demonstrate enqueue techniques through ECS background server.
ECS_TEST_ SERVER_FAIL	Demonstrate an ABAP/4 program failure with short dump through ECS background server.

Sample Process	Description
ECS_TEST_ SPAWNED_JOB	Demonstrate spawned job monitoring.
ECS_TEST_ START_EXIT	Demonstrate the use of a Phase start exit.
ECS_TEST_ SYNCHRONOUS	Execute an ABAP/4 program in synchronous fashion.
ECS_TEST_USER_ START	Demonstrate an user defined Process start exit.
ECS_TEST_ VARIANT	Execute an ABAP/4 program with a variant.

6.12.2 Initialization and Termination

If a program is submitted from ECS, it needs to execute the ECS_IMPORT_PARAMETERS macro to retrieve ECS control data and any parameters passed to the program. The "import parameters" function configures the ECS session variables such as the current Process, Phase, and run number. It is only necessary to execute the command if the program expects parameters from ECS, or intends to execute ECS commands.

Note: Refer "ECS internal variables and data types" in the appendix, for descriptions of the ECS command variables, their values and their usage.

Command	Parameters	Description
ECS_ INITIALISATION	[no parameters]	Clear all the ECS working variables.
ECS_IMPORT_ PARAMETERS	[no parameters]	Retrieve the ECS Current Process and Phase run attributes. This command is required if it is intended to execute ECS commands or retrieve parameter data. Refer Note 1 in the below Note.
ECS_EXPORT_ PARAMETERS	[no parameters]	Set the ECS run attributes. Mainly used to set parameter values to pass to a subsequent phase that is automatically started by ECS. Refer Note 2 in the below Note.
ECS_SAVE_ SESSION	[no parameters]	Save the current session attributes to memory. You may restore these at a later stage to "re-position" on a previous Process. The saved session is only kept for the duration of the current program execution.
ECS_SAVE_ SESSION_ID	{Session id}	Save the current session attributes to a specific memory id. This enables to save multiple sessions. You may restore the specific saved session at a later stage using ECS_ RESTORE_SESSION_ID. The saved session is only kept for the duration of the current program execution.

Command	Parameters	Description
ECS_ RESTORE_ SESSION	[no parameters]	Restore attributes from a previously saved session. This command is used in conjunction with ECS_SAVE_ SESSION. You may perform a restore may only within the scope of the current program.
ECS_ RESTORE_ SESSION_ID	{Session id}	Restore a specific saved session. This command is used in conjunction with ECS_SAVE_SESSION_ID. You may perform a restore only within the scope of the current program.

Note:

1. There is no need to perform ECS_IMPORT_PARAMETERS if the program was not submitted by ECS. The same is true, if the program does not need any parameter data or does not issue any ECS commands.
2. ECS_EXPORT_PARAMETERS should only be used in circumstances where parameter data (run data string / file name) needs to be passed to a subsequent phase that is being started automatically by ECS once the current phase has completed. If the next phase is being explicitly started through ECS_START_NEXT_PHASE, the parameters can be set and passed via the ECS_SET_RUN_DATA and ECS_SET_FILE_NAME commands. In this case, ECS_ EXPORT_PARAMETERS is not required.

The following table demonstrates the use of the ecs_export_parameters command to set the run data and file name parameters for a next Phase that is automatically started by ECS. In the case of a Phase start that is controlled programmatically (that is, Manual), the ecs_start_next_phase command performs the export automatically.

Automatic Phase start	Manual Phase start
ecs_import_parameters.	ecs_import_parameters.

Automatic Phase start	Manual Phase start
v_parm = v_ecs_*rdata.	v_parm = v_ecs_*rdata.
…..	…..
ecs_set_run_data v_parm.	ecs_set_run_data v_parm.
ecs_export_parameters.	ecs_start_next_phase.

Note: The run data and file name parameters are automatically inherited from the previous phase if you do not re-specify them; you do not require the export in the automatic Phase start if you make no parameter changes.

6.12.3 Starting and Ending Processes and Phases

These commands are the basic set, more specific options and attributes are documented under:

- Starting a Process through ECS event
- Starting a Process through RFC
- The BDC programming interface
- Using the run log to annotate and audit runs
- Processing external files

You use the following commands to start processes and phases from within an ABAP program. You need to configure "Runtime" attributes prior to each call since they are initialized after wards.

Command	Parameters	Description
ECS_START_PROCESS	{process name}	Initiates the ECS process and starts the first phase
ECS_START_NEXT_PHASE	[no parameters]	Initiates the next phase in sequence for the current process.
ECS_END_PHASE_AS_COMPLETE	[no parameters]	Ends the current phase as complete
ECS_END_PHASE_AS_FAILED	[no parameters]	Ends the current phase as failed and automatically configures the reprocessable flag, that is, you can rerun.
ECS_END_PHASE_AS_REPROCESSABLE	[no parameters]	Ends the current phase with the specified status. The status must be either COMPLETE or FAILED. You may use the constants C_ECS_*COMPLETE and C_ECS_*FAILED.

Command	Parameters	Description
ECS_END_PHASE	{status}	Ends the current phase with the specified status. The status must be either COMPLETE or FAILED. You may use the constants C_ECS_*COMPLETE and C_ECS_*FAILED.
ECS_START_PROCESS_VIA_EVENT	{process name}	Start the specified process through the triggering of an event. See the section "starting a process through event" for more details. There is a restriction on the attributes that you can use with this option.

You use the following commands to configure "run time" attributes for the start Process / Phase:

Command	Parameters	Description
ECS_SET_RUN_DATA	{data}	Free format data string, up to 255 char, to pass to an ABAP program associated with the phase. Stored in variable v_ecs_*rdata.
ECS_SET_RUN_FILTER	{field} {value}	Populate user-defined field and value definitions used for reporting, for example, WERKS 1234 would identify the run for plant 1234.
ECS_SET_BDC	{id} {queue id}	Specifies a BDC ID and queue name.
ECS_SET_START_DATE_TIME	{date} {time}	Schedule the start for a specific date and time.
ECS_SET_USER	{user id}	A specific user ID to use for the run.

Command	Parameters	Description
ECS_SET_EVENT	{id} {parameter}	Specify an event and associated parameter, up to 64 char.
ECS_SET_ENQUEUE	{enqueue data}	Specify an enqueue string, up to 80 char.
ECS_ENQUEUE_FOR_ENTIRE_RUN	[no parameters]	Retain the enqueue until the run completes successfully or failed (see next command).
ECS_REMOVE_ENQUEUE_ON_COMPLETE	[no parameters]	Only release the enqueue when a phase completes successfully.
ECS_SET_VARIANT	{variant name}	Specify a variant to use when executing a ABAP program. This overrides any variant specified at the Phase level.
ECS_DON'T_KEEP_BDC	[no parameters]	Used to indicate that the BDC session and logs are not kept once the BDC is processed. Refer The BDC programming interface for more details.
ECS_SET_IDOC	{IDOC number}	Specify the IDOC associated with the Phase.
ECS_SET_RFC_DEST	{RFC dest}	Refer Starting a Process through RFC for details

Command	Parameters	Description
ECS_ RESERVE_ RUN_NUMBER	[no parameters]	Pre-allocate an ECS run number. Normally, the run number is allocated as part of the start process. This means that there is no link between the 'start via event' and the actual run. By using this command, the run number is pre-allocated by ECS and stored in the internal variable v_ecs_*rnumber_reserve. ***Note:*** This variable is cleared after any start is performed.
ECS_DELAY_ START_IN_ SECONDS	{seconds}	Delay the start of the Process / Phase by nnn seconds. You can use this option as a last resort to allow enough time for dependencies to complete before continuing.
ECS_SET_ PHASE_NOT_ RESTARTABLE	[no parameters]	Marks the Phase as not re-processable under Any circumstances. Overrides all restart options and configuration.
ECS_SET_ USER_VAR1	{variable/literal}	Configure a user defined variable to pass to the next phase. You may specify up to 32 characters.
ECS_SET_ USER_VAR2	{variable/literal}	Configure a user defined variable to pass to the next phase. You may specify up to 32 characters.
ECS_SET_ USER_VAR3	{variable/literal}	Configure a user defined variable to pass to the next phase. You may specify up to 32 characters.
ECS_SET_ FILE_NAME	{file name}	Specify a file name to be pass to the process or phase. Stored in variable v_ecs_*file_name. This also sets the primary file name for use with external file commands.
ECS_IGNORE_ SHUTDOWN	[no parameters]	Start the Process / Phase even if ECS is shutdown.

Command	Parameters	Description
ECS_SET_ EXEC_MODE	{mode}	Override the execution mode of the Phase, that is, JOB, SERVER or SYNCHRONOUS .

You use the following commands to help automate external file processing. They apply to the 'primary' file name as set through the ECS_SET_FILE_NAME command, or on the Phase. Refer File management and Processing external files for more information.

Command	Parameters	Description
ECS_MOVE_ FILE_AT_ START	{template/name}	Automatically move the primary file when the process starts.
ECS_COPY_ FILE_AT_ START	{template/name}	Automatically copy the primary file when the process starts.
ECS_DELETE_ FILE_ON_ COMPLETE	[no parameters]	Automatically delete the primary file when the process completes.
ECS_ ARCHIVE_ FILE_ON_ COMPLETE	{archive directory}	Automatically copy the primary file to the `archive directory` and delete the original, when the process completes successfully.

The following commands are used to configure "run time" attributes for the end phase:

Command	Parameters	Description
ECS_SET_ PHASE_AS_ REPROCESSABLE	[no parameters]	Mark a failed phase as reprocessable.

Command	Parameters	Description
ECS_SET_SYST	[no parameters]	Pass the current SYST (that is, SY-) internal variable configurations to ECS. ECS references some of the variables to document message details and current status of the program.
ECS_IGNORE_IF_ ALREADY_ENDED	[no parameters]	Do not fail the end phase, if it is already complete.

Special utility commands:

Command	Parameters	Description
ECS_WAIT_IN_ SECONDS	{seconds}	Forces a wait for the specified number of seconds
ECS_EXECUTE_ COMMAND	{logical cmd} {parm}	Executes the specified logical SAP command definition passing the parameter data.

6.12.4 Passing Data between Process and Phases

ECS provides six methods to pass data to a Process and between Phases within a Process. See sample programs YECS9005 and YECS9006 for examples on how to apply these techniques.

Item	Description
Simple string	A single free format string of up to 255 characters
User defined variables	An unlimited number of user defined variables up to 255 characters
Exported data items	Any variables, structure, internal table may be stored.
3 static User variables	You may populate three "static" user variables. You may store a maximum of 32 characters in each.
External file	This option is suitable for large volumes and /or complex data structures.
Constants	Retrieve pre-defined_ fixed value settings that you configure in ECS. This prevents having constant or literal information “hard coded” in programs.

Support option to pass up to two internal tables, but you are not recommended this 'legacy' method due to the more complex programming and restrictions in functionality. Use the more flexible **Exported data items** option instead.

6.12.4.1 Simple Parameter String

The programmer may pass a parameter string of up to 255 characters to the Phase tarted using the ECS_SET_RUN_DATA command. The receiving program must issue an ECS_IMPORT_ PARAMETERS command to receive the data string into the V_ECS_*RDATA variable.

6.12.4.2 User Defined Variables

You may define an unlimited number of user defined variables using the ECS_SET_USER_ VARIABLE, and can retrieve using the ECS_GET_USER_VARIABLE commands. Each variable may be a maximum of 255 characters. The variables are automatically passed between phases. You may clear / initialize a specific user defined variable using the ECS_REFRESH_USER_VARIABLE or you may clear ALL user variables using the ECS_REFRESH_USER_VARIABLES commands.

6.12.4.3 Exported Data Items

You may export any variable, structure or internal table from one phase and imported into another. You use the ECS_EXPORT_DATA command to store the data items, and you use the ECS_ IMPORT_DATA command to retrieve the data item.

> ***Note:*** The sending and receiving data item name and structure need to be identical.

These data items are maintained for the duration of the ECS run and are automatically housekept when the run is deleted.

> ***Note:*** Each stored data item is unique across the entire run, that is, versions are not kept for the different phases. You need to consider this for checkpoint / restart processing, since recovery may be limited if the one and only original is incorrectly modified during the run.

6.12.4.4 Three Static User Variables

You may populate up to three pre-defined user defined variables (character 32) to pass data between Phases. You use the commands ECS_SET_USER_VAR1/2/3 to set the values prior to starting a Process or a Phase. The receiving program must issue an ECS_IMPORT_PARAMETERS command to receive the data string into the V_ECS_*USER_VAR1/2/3 variables. This option does

not offer much advantage over using "User defined variables", other than they are automatically retrieved for you as part of the ecs_import_parameters processing.

6.12.4.5 External File

You may pass data to a process or between phases by first writing the data to an external file, then specifying the file name to ECS and then reading the file from within a program. The name of the file is passed to the phase started using the ECS_SET_FILE_NAME command. The ECS_IMPORT_PARAMETERS command returns the name of the file in the V_ECS_*FILE_NAME variable. The file name is passed automatically between phases and you can use the ECS_DELETE_FILE_ON_COMPLETE command to instruct ECS to automatically delete the file once the process successfully completes.

> ***Note:*** See process ECS_TEST_PARAMS (programs YECS9005 and YECS9006) for an example on how to pass data string and file data between two phases within a process. You may mix the simple parameter string and external file methods. Only one ECS_IMPORT_PARAMETERS command is required to return all parameters. If you pass parameters to a subsequent phase and the next phase is not explicitly started, (that is, ECS is starts automatically), it is necessary to perform an ECS_EXPORT_PARAMETERS command to save the definitions.

6.12.4.6 ECS Constants

You do some ECS configuration by way of ECS Constants. The use of these constants gives flexibility over ECS functions by allowing users to define the settings in one place. Each constant has a value, validity begin date and a value. The inclusion of a date allows the flexibility of setting in advance new values without having to worry about the need to change the ECS Constant value manually.

6.12.4.7 Example

From January first, 2000, you need to use the FTP script Y_ECS_FTP_SCRIPT_ONE. On June first 2000, you require a new script Y_ECS_FTP_SCRIPT_TWO due to a change in the login arrangements.
You create two Constants as follows:

Constant	Begin Date	ECS Component	Value
FTP_COMMANDFILE_ STANDARD_TEXT	20000101	FTP	Y_ECS_FTP_ SCRIPT_ONE
FTP_COMMANDFILE_ STANDARD_TEXT	20000601	FTP	Y_ECS_FTP_ SCRIPT_TWO

6.12.4.8 Usage of ECS Constants

For details of the name of the ECS constants to configure see the relevant section of the ECS manual. Each part of ECS using the command reads the table and use the value relating to the relevant date. The relevant date is normally the system date but in some cases, it can be the date that the run started, or another processing date.

6.12.4.9 Dates and Flags

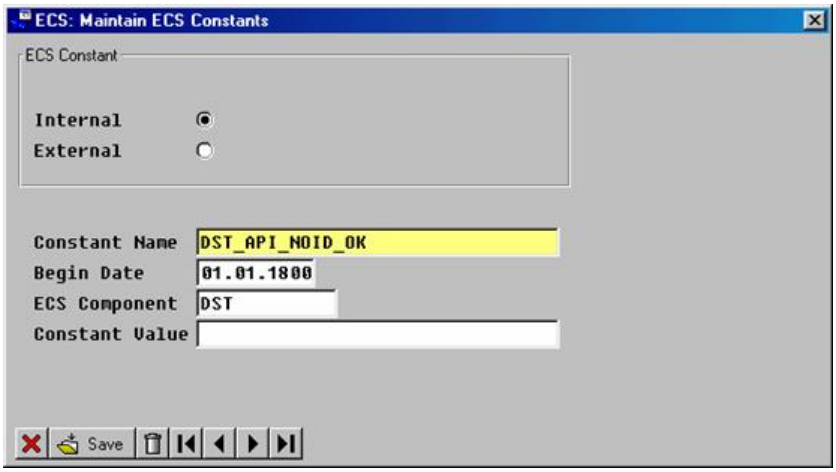
If the constant expects a date, you need to enter the date in the format CCYYMMDD. For example, 19990201 for the second of February 1999. If it expects a true / false, type flag then, X is true and space (or blank) is false.

6.12.4.10 ECS Component

The ECS component field is an information only field and indicates the software component of ECS that reads the constant. For example, DST (DataStream), FTP (FTP server), CORE (core functionality).

6.12.4.11 Configuring ECS Constants

In order to configure ECS Constants in the system, select the Workbench (transaction YECW) and select the ECS Constants on the **Setup** menu. Enter the information in the corresponding fields of the following screen:



> ***Note:*** ECS constant name and Begin date are mandatory. The 'ECS_INTERNAL' field is defaulted to external constant. You rarely require the creation or deletion of internal ECS constants.

6.12.4.12 Internal Table through Start Phase (superseded by Exported Data Items)

You are not recommended this legacy option due to the more complex programming involved and restrictions in ECS functionality. It is retained for backward compatibility reasons only. It is used to pass up to two internal tables between Phases of a Process through the Y_ECS_START_PHASE function call. The following restrictions apply:

- Phases may only execute in Asynchronous job mode
- Internal tables may not be sent through RFC
- The receiving program may not be started automatically by ECS. In addition, it may not be started by ECS command (for example, ECS_START_PHASE). It must be started by function call Y_ECS_START_PHASE.
- The receiving program may not import parameters using ECS_IMPORT_PARAMETERS. Instead, it must issue function call Y_ECS_IMPORT_PARAMETERS.

> ***Note:*** See the sample Process ECS_TEST_PARM_TBL (programs YECS9012 and YECS9013) for an example on how to pass two internal tables as parameters through start phase between two programs.

6.12.5 Internal Variables and Data Types

In some cases, you may need internal ECS information to access, or need work variables to declare based on ECS variables. The SDK provides internal variables containing data about the execution of a Process Phase. In addition, a number of common data types are provided for easy variable declaration.

6.12.5.1 Internal Variables

This is a list of the more common ECS SDK internal variables. Programmers need to reference them in some cases to retrieve values such as the run data parameter, and current file name. These parameters are normally populated using ECS_SET commands, but you may retrieve them or configure in the program code.

Variable	Description
v_ecs_*bdc_id	BDC session ID (this may be automatically generated by ECS)
v_ecs_*bdc_qid	The unique queue id allocated to the SAP BDC session
V_ecs_*checkpoint_data	Value of the last checkpoint operation
V_ecs_*cond_code	Condition code
v_ecs_*enqueue	Enqueue data string
v_ecs_*eventid	Event ID
v_ecs_*eventparm	Event parameter
v_ecs_*file_name	The current file name
v_ecs_*idoc	Internal SAP IDOC number
v_ecs_*iname	Process name

Variable	Description
v_ecs_*key_data	Filter data
v_ecs_*key_field	Filter field
v_ecs_*msg	A formatted error message
v_ecs_*phase_num	Unique instance of the Phase within Phase sequence (that is, restarted)
v_ecs_*pname	Phase name
v_ecs_*rc	The result of the ECS command. 0-ok, any other value is a failure
v_ecs_*rdata	Run data parameter
v_ecs_*rnumber	Unique number identifying the run
V_ecs_*rnumber_ reserve	Reserved run number
v_ecs_*rsequence	Unique Phase sequence number
V_ecs_*trace_ind	Phase trace indicator
V_ecs_*user_var1/2/3	User variables 1, 2 and 3

6.12.5.2 Standard Types

From time to time, programmers need to declare variables for use with ECS commands and functions. The following types are provided as part of the ECS SDK:

Variable
ECS_RUN_NUMBER
ECS_PROCESS_NAME
ECS_PHASE_NAME
ECS_RUN_DATA
ECS_ENQUEUE_DATA
ECS_FILE_NAME
ECS_EVENT_ID
ECS_EVENT_PARM
ECS_LOG_DATA
ECS_USER_VARIABLE_NAME
ECS_USER_VARIABLE_DATA
ECS_CONDITION_CODE

6.12.6 Start Phase Exit Program

You may implement a custom program at start Phase time to perform additional processing. It is called after start Phase validation and before execution. The program may also alter certain run time attributes and pass them back to start Phase processing. Start phase exits are useful points to perform specific checks that are not possible with ECS. For example, check whether a SAP R/3 lock is in effect, substitute run parameter data, and specify an enqueue key. In addition, the exit program may set a return code to abort the start phase.

6.12.6.1 Configuration

A Start Phase exit program is configured in the advanced section of the Phase definition through the Workbench. You need to code this custom program in ABAP/4. It is passed a standard structure YECS_STRTX that you may optionally modify and pass back to ECS. The exit is invoked after validation of the start parameters and before the Phase is initiated. You may then freely apply any code. Keep in mind that the exit program is called directly, thus any abnormal termination halts the start Phase processing. Additionally, any performance problems affect the overall start Phase performance. The exit program may reject the start by setting the return code to any value other than zero (0).

6.12.6.2 Parameters

Item	Parameters	Description
RETURNCODE	INT 4 (char 10)	Return code (default 0)
INAME	CHAR 20	Process name (non modifiable)
PNAME	CHAR 20	Phase name (non modifiable)
RDATA	CHAR 255	Run data
RUSER	CHAR 12	Session: SAP user from SAP log
FILE_NAME	CHAR 128	File name

Item	Parameters	Description
START_DATE	DATS 8	Phase start date
START_TIME	TIMS 6	Phase start time
KEY_FIELD	CHAR 10	Filter field
KEY_DATA	CHAR 32	Filter data
ENQUEUE	CHAR 80	Enqueue value

Sample code Example

```
data: v_strtx like yecs_strtx.

  import yecs_strtx to v_strtx
         from memory id 'ECS_START_PHASE_EXIT'.

  v_strtx-enqueue = ...
  ..................
  ..................

* the following export is only necessary if you wish to modify the
* parameters. Note: iname and pname modifications are ignored.

  export yecs_strtx from v_strtx
         to memory id 'ECS_START_PHASE_EXIT'.
```

6.12.7 Starting a Process through ECS Event

ECS supports the starting of a Process through an interim event queue. This queue is effectively a table that contains deferred ECS start process entries. Why the interim process? That is, why not just call ECS directly? The reason is that there is a limitation on the ABAP/4 commands that you can execute in some user exits, that is, those that are performed in update tasks. For example, it is not possible to call another ABAP program or perform a function from within an update task. Since most user exits in SAP R/3 are very useful points from which to initiate processes, the ECS "start process via event" functionality is a convenient and easy way to accomplish this. The following command is used to start a process through an ECS event from within an ABAP program. You need to configure "Runtime" attributes prior to each call since they are initialized afterwards.

Command	Parameters	Description
ECS_START_PROCESS_VIA_EVENT	{process name}	Start the specified process through the triggering of an ECS event.

You use the following commands to set "run time" attributes for the start through an event command. A more complete description of these parameters is available under the starting / ending a phase section. Not all start Process attributes are supported for this option.

Command	Parameters
ECS_SET_RUN_DATA	{data}
ECS_SET_RUN_FILTER	{field} {value}
ECS_SET_USER	{user id}
ECS_SET_ENQUEUE	{enqueue data}
ECS_ENQUEUE_FOR_ENTIRE_RUN	[no parameters]
ECS_REMOVE_ENQUEUE_ON_COMPLETE	[no parameters]

Command	Parameters
ECS_SET_EVENT	{id} {parameter}
ECS_SET_FILE_NAME	{file name}
ECS_RESERVE_RUN_NUMBER	[no parameters]
ECS_DELAY_START_IN_SECONDS	{seconds}
ECS_RENAME_FILE_AT_START	[no parameters]
ECS_COPY_FILE_AT_START	{copy to directory}
ECS_MOVE_FILE_AT_START	{move to directory}
ECS_DELETE_FILE_ON_COMPLETE	[no parameters]
ECS_ARCHIVE_FILE_ON_COMPLETE	{archive directory}
ECS_SET_USER_VAR1	{variable / literal}
ECS_SET_USER_VAR2	{variable / literal}
ECS_SET_USER_VAR3	{variable / literal}

Example ABAP Code

```
ECS_SET_RUN_DATA v_rdata.
ECS_SET_RUN_FILTER 'WERKS' v_werks.
ECS_ENQUEUE_FOR_ENTIRE_RUN.                  " Allow only one process
to run at a time

ECS_RESERVE_RUN_NUMBER.                          " Get ECS to pre-
allocate the run number
V_rnumber_reserve = v_ecs_*rnumber_reserve.      " Store allocated
```

```
run number for audit log.

ECS_START_PROCESS_VIA_EVENT 'ECS_TEST_ABAP'.
if v_ecs_*rc ne 0.
   .... error processing .....
endif.
```

Note: See the Process ECS_TEST_EVENT (program YECS9003) for an example on triggering the example 'ECS_TEST_ABAP' Process through an ECS event.

6.12.8 Starting a Process through RFC

RFC stands for Remote Function Call. This option enables an ECS process to start in one SAP system from an external script, external program or from another SAP system.

6.12.8.1 External Script or Program (non-SAP)

You can cancel the function /SKY/RFC_START_PHASE using a SAP RFC call, or through the STARTRFC supplied program. In order to code a user RFC client program, read the Remote Function Call documentation on how to do this. You may find the documentation on how to call the STARTRFC.C standard SAP supplied program in the RFC SDK library ...../rfcsdk/text.

STARTRFC

```
startrfc      -h {host name} -s {system id} \
-u {userid} -p {password} -c {client} \
-f /SKY/RFC_START_PHASE \
-e PROCESS=ECS_TEST_ABAP \
-e FILTER_FIELD=WERKS \
-e FILTER_DATA=P001  \
-e RUN_DATA=testing 123
```

6.12.8.2 ABAP Program

You use the following command to start a process through RFC from within an ABAP program. You need to configure "Runtime" attributes prior to each call since they are initialized afterwards. The only difference between a normal ECS_START_PROCESS and through RFC is the use of the ECS_SET_RFC_DEST command to specify a destination. There are also some restrictions on the attributes that you may specify for an RFC start Process.

Command	Parameters	Description
ECS_SET_RFC_DEST	{Destination}	This sets the RFC destination for the ECS_START_PROCESS

Command	Parameters	Description
ECS_START_ PROCESS	{process}	Start the specified process through RFC call.

The following commands are used to configure "run time" attributes for the start through RFC option. A more complete description of these parameters is available under the starting/ending phase's section. Not all start Process attributes are supported for this option.

Command	Parameters
ECS_SET_RUN_DATA	{data}
ECS_SET_RUN_FILTER	{field} {value}
ECS_SET_USER	{user id}
ECS_SET_ENQUEUE	{enqueue data}
ECS_ENQUEUE_FOR_ENTIRE_RUN	[no parameters]
ECS_REMOVE_ENQUEUE_ON_COMPLETE	[no parameters]
ECS_SET_EVENT	{id} {parameter}
ECS_SET_FILE_NAME	{file name}
ECS_DELAY_START_IN_SECONDS	{seconds}
ECS_RENAME_FILE_AT_START	[no parameters]
ECS_COPY/MOVE_FILE_AT_START	{target directory}
ECS_DELETE _FILE_ON_COMPLETE	[no parameters]
ECS_ARCHIVE_FILE_ON_COMPLETE	{archive directory}

Command	Parameters
ECS_SET_START_DATE_TIME	{date} {time}
ECS_SET_USER_VAR1/2/3	{variable/literal}

Example

```
ECS_SET_RUN_DATA v_rdata.
ECS_ENQUEUE_FOR_ENTIRE_RUN.                " Allow only one process to
run at a time
ECS_SET_RFC_DEST 'PRD_SD'.                      " Configure the RFC
destination
ECS_START_PROCESS 'ECS_TEST_RFC' .
if v_ecs_*rc ne 0.
   ....
```

6.12.9 The BDC Programming Interface

Batch Input Processing is originally designed as an interfacing tool to upload legacy data from external systems into SAP R/3. It therefore supports only in-bound interfaces and is dependent on SAP R/3 transaction sequences and screen layouts. This makes this technique vulnerable to any changes made to screen layouts in new releases of SAP R/3. Non the less, the cost of re-working BDC scripts for new releases may out-weigh the cost of using and supporting some of the more elegant interfacing techniques such as ALE or BAPIs. The Batch Input Processing interface comprises of the programmer creating a sequential "script" of transactions and their screen field attributes that mimic exactly what a user may enter through the SAP R/3 screens. Once the script (BDC session) is created, you can execute in two different ways; by either attaching the BDC script to a "call transaction" that executes the BDC in an interactive (dialog) mode or in background as a separate task. ECS supports the background method that is better scheduling, error processing and recovery.

6.12.9.1 Creating BDC Sessions

ECS caters for many different methods for processing BDC sessions. At the end of the day, an ABAP program generates one or more BDC scripts that are then subsequently executed and monitored by BDC phases. ECS provides a complete set of commands to create BDC sessions, however, you may convert SAP generated (that is, SHDB) or custom BDC data processing to use ECS with minimal impact.

There are four basic steps to generate a BDC session:

1. Create a BDC session name
2. Generate a BDC script (BDCDATA)
3. Insert the BDC script into the session
4. Close the BDC session.

> ***Note:*** You may perform steps 2 and 3 repeatedly.

6.12.9.2 Creating a BDC Session Name

The first step in creating the BDC session is to create (declare) the session name. This name is used to uniquely identify the BDC to be executed for monitoring, restart and reporting purposes. Another facet of creating the BDC session is to work out the execution mode of the next BDC phase. This is important to determine how to physically store the BDC scripts. The ecs_bdc_open_group generates a unique name by analyzing the BDC ID of the first BDC phase after the current phase and applying the following logic:

1. If the BDC ID is blank, generate a unique session name using the BDC prefix from the session defaults suffixed by a unique number.
2. If the BDC ID is &VAR1, &VAR2, &VAR3. (Same result as 1.) This indicates that the BDC session name is dynamically substituted using the ECS user variables. The programmer is responsible for setting the variables using the ecs_set_user_var1/2/3 command.
3. If a template BDC ID is specified containing &yyyymmdd, &hhmmss, &number and/or &rnumber; the variables are substituted and the BDC ID is returned.
4. A fixed BDC ID may be specified that is used as is.

6.12.9.3 Specifying a Custom Session Name

In the event that an ECS automatically generated BDC session name(as above) is not required, it is possible to specify a customized BDC session name using the **ecs_bdc_open_custom_group** command. An alternative to this is to hard code the BDC session name in the BDC ID of the BDC phase. However there are many cases, such as standard SAP conversion programs where the BDC session name is fixed or is specified as part of an input file header.

6.12.9.4 Generating BDC Scripts

You may use ECS commands to define screen, field, okcode and cursor definitions in BDCDATA format. ECS commands make this process easier, but you may use other functions and then the data copied to the ECS internal BDCDATA table through the **ecs_copy_bdcdata** command. The ECS

commands to create BDC scripts are easy to use. There is also a conversion utility available that converts BDC sessions as generated through the SHDB utility (BDC recorder) directly into ECS BDC commands. Refer Converting BDC scripts as generated by the SAP BDC recorder (SHDB) for more details.

6.12.9.5 Inserting BDC Data

Once the BDC script is generated for a transaction, it is inserted into the BDC session using the **ecs_bdc_insert** command, specifying the associated transaction. Once the script is inserted, it is deleted from the ECS (BDCDATA) internal table. This command expects the BDC script to be in internal table it_ecs_bdcdata. If the script is generated into a different BDCDATA table, you can use the *ecs_copy_bdcdata command, specifying the name of the internal table to copy.

> ***Note:*** The contents of the internal table are cleared after the data is copied.

6.12.9.6 Closing the BDC Session

Once the BDC session name is created and the BDC script is generated, the session is closed using the ecs_bdc_close_group command. This makes the BDC session available for processing.

6.12.9.7 Generating Multiple BDC Sessions from a Single Program

It is possible to execute up to three separate BDC sessions that are created from a single ABAP program. To do this, multiple BDC phases are defined with a BDC ID of &VAR1, &VAR2 or &VAR3 that refer to the ECS user defined variables for the BDC session name. The programmer is responsible for setting the user defined variables using the ecs_set_user_var1/2/3 commands. Typically, this is done directly after the BDC session is created. For example,

Example

```
Ecs_bdc_open_group.
Ecs_set_user_var1 v_ecs_*bdc_id.
….
Ecs_bdc_close_group.

Ecs_bdc_open_group.
Ecs_set_user_var2 v_ecs_*bdc_id.
```

```
….
Ecs_bdc_close_group.

Ecs_bdc_open_group.
Ecs_set_user_var3 v_ecs_*bdc_id.
….
Ecs_bdc_close_group.
```

The Process Phases following this program refer to &VAR1, &VAR2 and &VAR3 respectively.

6.12.9.8 Catering for Multiple BDC Sessions with the same Name

In some cases, such as standard SAP conversion programs, multiple BDC sessions are generated with the same name. Under normal circumstances, ECS fails the BDC processing phase if more than one BDC session is found with the same name. If there is a need to process and monitor all existing BDC sessions with a specific name, check the **multiple session support** option on the BDC phase definition. This option is valid only for asynchronous job processing.

6.12.9.9 BDC Command Summary

The following section explains how to use ECS commands to construct a BDC session. ECS not only provides the commands for ease of use, but to link the BDC session with ECS and uniquely identify each BDC session though a uniquely generated name.

The following set of commands from the ECS SDK (YECSSDK) are used to manage BDC sessions:

Command	Parameters	Description
ECS_BDC_ OPEN_ GROUP	[no parameters]	Generate a unique BDC session ID and create it.
ECS_BDC_ BEGIN	{program} {screen}	The program and screen ID that the following field /okcode entries are to process against.

Command	Parameters	Description
ECS_BDC_FIELD	{field} {value}	Specify a BDC field assignment
ECS_BDC_OKCODE	{okcode}	Specify a BDC okcode assignment
ECS_BDC_CURSOR	{field}	Specify a BDC cursor assignment
ECS_BDC_INSERT	{transaction}	Specify the transaction that BDC is to execute and copy the BDC script in it_ecs_*bdcdata to the BDC session.
ECS_BDC_CLOSE_GROUP	[no parameters]	Close the current BDC session, ready for execution.
ECS_BDC_GENERATE_GROUP_ID	[no parameters]	Generate a unique BDC group name which is returned in v_ecs_*bdc_id. ***Note:*** No open is performed
ECS_BDC_OPEN_CUSTOM_GROUP	{session name}	Create a unique BDC session with the name specified.
ECS_DON'T_KEEP_BDC	[no parameters]	Specify before any of the open options to indicate that the generated BDC session and any logs are not to be kept once the BDC is successfully processed.

Command	Parameters	Description
ECS_COPY_ BDCDATA	{internal table}	Copy the BDC script from a user defined BDCDATA Internal table to the ECS internal table it_ecs_*bdcdata. ***Note:*** The user defined internal table is cleared.

6.12.9.10 ECS Process and Phase Requirements

Once the BDC generation program is complete and ready to execute, you should construct the ECS Process and Phase definitions that are going to control and run the BDC interface.

1. Create a Process for the BDC generation and execution
2. Create a Phase to execute the ABAP program
3. Create subsequent phases to execute BDC sessions.

Refer the previous section Creating a BDC session name for details on what to specify in the optional BDC ID field in the phase definition. You may repeat step 3 up to three times to execute multiple BDC sessions.

Note: You are recommended that ECS generates the BDC session name, that is, leave the BDC ID blank.

Important: ECS analyses the BDC ID and execution mode of the first BDC phase following the BDC generation program for the ecs_bdc_open_group command. This is important when you generate multiple BDC sessions from a single program.

6.12.9.11 Execution of BDC Sessions

The method used to execute the BDC session depends on the execution mode of the first BDC Process Phase following the program that is generating the session. If asynchronous job, then the BDC session is stored and executed through the SAP BDC background processing system (RSBDCSUB, and SM35). If in ECS Server or synchronous mode, then the BDC session is stored internally and executed using the call transaction method.
If no BDC session is found, the phase is completed normally assuming that none is generated. This is to cater for situations where a BDC session may be optionally generated by a program.

6.12.9.12 Converting BDC Scripts as Generated by the SAP BDC Recorder (SHDB)

The SAP R/3 SHDB function is a convenient way to capture the keystrokes used to process a SAP transaction and subsequently generate a program that executes the BDC script. ECS provides a utility program to convert SHDB generated BDC script to ECS commands. In most cases, no additional change is required. Follow the following steps:

- Execute the Convert BDC program from the **Utilities** menu in the Workbench screen.
- Enter the program name to convert and execute the program.
- A report lists the result. The specified program is converted and saved.

6.12.9.13 Example 1: - Automatic BDC Session Name Generation and Execution

In this example, we use the ECS_TEST_BDC sample process. The BDC generated here is very basic and is not a true reflection of an actual business requirement.

Process:	CS_TEST_BDC	
Phase 1:	GENERATE_BDC	ABAP program YECS9001
Phase 2:	PROCESS BDC	BDC (the BDC session details are passed automatically)
		* Auto start is enabled

When process ECS_TEST_BDC starts, it automatically starts phase 1 that generates the BDC session. Once complete, control is passed back to ECS that checks that phase 1 completed ok and if so, automatically starts Phase 2. All the BDC session details are passed automatically to Phase 2 that submits the BDC session for execution. The progress of the BDC session is monitored and control is passed back to ECS on completion. Sample ABAP code for Phase 1:

Example

```
REPORT  YECS9001.

  INCLUDE YECSSDK.

  ECS_IMPORT_PARAMETERS.

  ECS_BDC_OPEN_GROUP.
  ECS_BDC_BEGIN 'SAPMSEU0' '0200'.
  ECS_BDC_FIELD 'RSEUX-CK_VALUE' 'YECS'.
  ECS_BDC_OKCODE 'OBJA'.
  ECS_BDC_BEGIN 'SAPMSSY0' '0120'.
  ECS_BDC_OKCODE '/12'.
  ECS_BDC_BEGIN 'SAPMSEU0' '0200'.
  ECS_BDC_OKCODE '/12'.
  ECS_BDC_INSERT 'SE80'.
  ECS_BDC_CLOSE_GROUP.
```

6.12.9.14 Example 2: Programmed Phase Start and End

In this example, we use the ECS_TEST_BDC2 sample process. The BDC generated here is identical to that used in 'Example usage 1'. The difference between the two is that in this example, Phase 1 is ended explicitly through an ECS_END_PHASE command and Phase 2 started explicitly through an ECS_START_NEXT_PHASE command.

Process:	CS_TEST_BDC	

Phase 1:	GENERATE_BDC	ABAP program YECS9002
Phase 2:	PROCESS BDC	BDC (the BDC session details are passed automatically)
		* Auto start is enabled

When process ECS_TEST_BDC2 starts, it automatically starts phase 1 that generates the BDC session. Once complete, it explicitly ends Phase 1 and starts Phase 2. All the BDC session details are passed automatically to Phase 2 that submits the BDC session for execution. The progress of the BDC session is monitored and control is passed back to ECS on completion. Sample ABAP code for Phase 1:

Example

```
REPORT YECS9002.
 INCLUDE YECSSDK.

 ECS_IMPORT_PARAMETERS.

 ECS_BDC_OPEN_GROUP.
 ECS_BDC_BEGIN 'SAPMSEU0' '0200'.
 ECS_BDC_FIELD 'RSEUX-CK_VALUE' 'YECS'.
 ECS_BDC_OKCODE 'OBJA'.ECS_BDC_BEGIN 'SAPMSSY0' '0120'.
 ECS_BDC_OKCODE '/12'.
 ECS_BDC_BEGIN 'SAPMSEU0' '0200'.
 ECS_BDC_OKCODE '/12'.
 ECS_BDC_INSERT 'SE80'.
 ECS_BDC_CLOSE_GROUP.

* end the current phase we are in.
 ECS_END_PHASE_AS_COMPLETE.

* start the next phase in the sequence.
ECS_START_NEXT_PHASE.
```

6.12.9.15 Example 3: Custom BDC Session Name and Generation of Multiple Sessions

This example contains two examples of how to invoke the ECS BDC interface with minimal change to the BDC code. In this example, you use BDC sessions generated using SHDB. The basic changes are to replace the BDC open and close group with ECS equivalents. The examples also demonstrate how to pass the BDC session (group) names to a phase to execute them through the ECS user defined variables. This technique is necessary if you generate multiple BDC sessions from within one program.

Process:	ECS_TEST_BDC_LEGACY	
Phase 1:	GENERATE_BDC	ABAP program YECS9023
Phase 2:	PROCESS_BDC	BDC id = &VAR1
Phase 3:	PROCESS_BDC2	BDC id = &VAR2

Example

```
 INCLUDE YECSSDK.
DATA: V_BDC_ID(12) TYPE C.
DATA: BEGIN OF BDCDATA OCCURS 0.
 INCLUDE STRUCTURE BDCDATA.
DATA: END OF BDCDATA.

* Get ECS runtime parameters.
ECS_IMPORT_PARAMETERS.
* ---------------------------------------------------------------------
----
* Example 1. Let ECS determine the name and populate the
*            ECS user variable 1 which is automatically substituted
*            in the BDC phase of the process with a BDC id of &VAR1.
* ---------------------------------------------------------------------
----
```

```
* Get ECS to generate a unique session name.
ECS_BDC_OPEN_GROUP.
* std SAP BDC as generated via SHDB
perform bdc_dynpro      using 'SAPMSEU0' '0200'.
perform bdc_field       using 'BDC_OKCODE' 'OBJA'.
perform bdc_field       using 'BDC_CURSOR''RSEUX-CK_VALUE'.
perform bdc_field       using 'RSEUX-CK''X'.
perform bdc_field       using 'RSEUX-CK_VALUE''$tmp'.
perform bdc_field       using 'RSEUX-EP''X'.
perform bdc_dynpro      using 'SAPMSSY0' '0120'.
perform bdc_field       using 'BDC_OKCODE' 'TRRT'.
perform bdc_field       using 'BDC_CURSOR' '02/03'.
perform bdc_dynpro      using 'SAPMSEU0' '0200'.
perform bdc_field       using 'BDC_OKCODE' 'TRRT'.
perform bdc_field       using 'BDC_CURSOR' 'RSEUX-CK_VALUE'.
perform bdc_transaction using 'SE80'.

* copy the SAP standard BDCDATA internal table to ECS
 ECS_COPY_BDCDATA BDCDATA.
 ECS_BDC_CLOSE_GROUP.


* save the generated BDC id in a user variable to be dynamically
* substituted in the BDC id of the phase with a BDC id of &VAR1.
ECS_SET_USER_VAR1 V_ECS_*BDC_ID.

* ---------------------------------------------------------------------
----
* Example 2. We dictate the BDC session name to ECS and populate the
*            ECS user variable 2 which is automatically substituted
*            in the BDC phase of the process with a BDC id of &VAR2.

* ---------------------------------------------------------------------
```

```
----
* create a custom BDC id and tell ECS to use it as the session name.
CONCATENATE 'MYBDC' SY-UZEIT INTO V_BDC_ID.
ECS_BDC_OPEN_CUSTOM_GROUP V_BDC_ID.
* std SAP BDC as generated via SHDB
perform bdc_dynpro      using 'SAPMSEU0''0200'.
perform bdc_field       using 'BDC_OKCODE''OBJA'.
perform bdc_field       using 'BDC_CURSOR''RSEUX-CK_VALUE'.
perform bdc_field       using 'RSEUX-CK''X'.
perform bdc_field       using 'RSEUX-CK_VALUE' '$tmp'.
perform bdc_field       using 'RSEUX-EP' 'X'.
perform bdc_dynpro      using 'SAPMSSY0' '0120'.
perform bdc_field       using 'BDC_OKCODE' 'TRRT'.
perform bdc_field       using 'BDC_CURSOR' '02/03'
perform bdc_dynpro      using 'SAPMSEU0' '0200'.
perform bdc_field       using 'BDC_OKCODE' 'TRRT'.
perform bdc_field       using 'BDC_CURSOR' 'RSEUX-CK_VALUE'.
perform bdc_transaction using 'SE80'.

* copy the SAP standard BDCDATA internal table to ECS
ECS_COPY_BDCDATA BDCDATA.
ECS_BDC_CLOSE_GROUP.

* save the generated BDC id in a user variable to be dynamically
* substituted in the BDC id of the phase with a BDC id of &VAR2.
 ECS_SET_USER_VAR2 V_BDC_ID.
* save the user variables that have been set, so they may be
referenced
* by subsequent phases.
ECS_EXPORT_PARAMETERS.
```

6.12.10 Using the ECS Run Log to Annotate Audit and Trace Runs

The ECS log is used to add information about a specific phase of a run. Programmers may write text information from their ABAP programs to provide more information for users to determine what happened. This text may provide an audit trail of the programs processing, warn the user about certain conditions or provide detailed instructions on how to resolve and error. If a log for the run phase exists, the severity of the entries (info, warning, or error) is highlighted with icons in the online ECS reporting system and the exception report. In addition, the log may be used as a trace facility, that is, messages are only displayed if the trace option has been activated at either the Process or Phase levels.

6.12.10.1 What is the Difference between the Run Status and the Log Status?

It is important to recognize the different context between the run Phase status (In-progress, failed, complete) and the status of log messages (info, warning and error). The Phase run status determines whether ECS can proceed with the next Phase of a Process. In some cases, it may be ok to proceed with the next Phase, even if there are recorded errors in the log. However, these errors must still be highlighted to the user. An example may be to execute as much of an input file as possible and highlight erroneous entries that require manual intervention.

6.12.10.2 Internal Use of the Log

ECS uses the log internally to provide information about the execution of a run. This is useful to support staff to help diagnose any system problems or ECS behavior. Normally, these messages are written against the current run Phase.

6.12.10.3 Severity and Level Attributes

Each entry written to the log is assigned a severity (info, warning, error or trace) and a level (1,2,3, and 45). You use these attributes to both designate the importance of the information and provide a 'depth' filter to the amount of information written to the log. As mentioned previously, the fact that a log exists, and the status of the data within it is highlighted by the ECS online and background exception reports. The level is set by the programmer to provide a hierarchical structure to the log entries. This is useful to provide an indented display of the data and use the level selection prompt in the online log display to limit the information displayed, for example, only display level 1 and 2 entries. You can configure a log level and severity at the phase level to limit the level of log entry and severity written to the system, for

example, only writes level 1 entry or warnings and errors. Any entries that fall outside of the configuration are ignored. This feature is optional and is useful to capture varying degrees of log entry to help diagnose what the program did. The program always calls ECS with log entries, but ECS may or may not write out the entry depending on what filter information is coded at the phase level. The internal variable v_ecs_*log_severity contains the maximum severity to date.

Attribute	Description
I	Informational
W	Warning
E	Error
T	Trace

Important: The maximum severity written so far is stored in variable v_ecs_*log_severity. You may use this from within a program to determine whether to fail the run or not. Trace entries are only written to the log if the ECS trace is activated at the Process, Phase or through ecs_set_trace_on command.

6.12.10.4 Writing Entries to the Log

In order to record log information, you must allocate an ECS run number as well as the run sequence number. If a re-run situation exists, the phase number is also necessary. It is possible to write a single message, or batch up messages and commit them to the log at a pre-defined point in the program. You are recommended that messages are batched wherever possible. 'Batched' messages are only committed to the log using an ECS_COMMIT_LOG command. Therefore, in cases where the log messages are critical, it is a good idea to use the 'direct write message' technique (ECS_WRITE_LOG) that commits the message to the log immediately. Be aware that using the direct write message technique may affect the performance of the application if a high volume of messages are written.

6.12.10.5 Programming "Batched" Log Messages

Type	Parameters	Description
ECS_ REFRESH_ LOG	[no parameters]	Clears the internal table used to store all the log entries. This is not necessary in most cases, since the ECS_COMMIT_LOG command performs this anyway. It may be useful in some cases to disregard log messages and start again. This command does not delete the ECS log table entries.
ECS_ APPEND_ LOG	{severity} {level} {message}	Appends a message to the internal table used to 'stack' all entries in the batch. The severity must be (I-info, W-warning , E-error , T-trace) and the level can be any number from 1 to 9. The message may be any text string from 1 to 255 characters.
ECS_ APPEND_ LOG_ BASIC	{message}	Appends a message to the internal table used to 'stack' all entries in the batch. The severity is set as info and the level is 1. The message may be any text string from 1 to 255 characters.
ECS_ APPEND_ LOG_ ULINE	[no parameters]	Write an underline
ECS_ APPEND_ LOG_SKIP	[no parameters]	Skip a line
ECS_ COMMIT_ LOG	[no parameters]	Calls the Y_ECS_WRITE_LOG function to update the ECS log table and then performs an ECS_REFRESH_LOG.

Example 1

```
ECS_REFRESH_LOG.
ECS_APPEND_LOG 'I' '1' 'Example line 1'.
ECS_APPEND_LOG 'I' '1' 'Example line 2'.
ECS_COMMIT_LOG.
```

Example 2

```
ECS_REFRESH_LOG.
ECS_APPEND_LOG v_severity v_level v_message.
ECS_COMMIT_LOG.
```

Important: The data string is a single argument of maximum 255 characters. Use structures, field symbols, write or concatenate to structure a string with multiple values. The log entries are only committed to the ECS log database table when the ECS_COMMIT_LOG is executed. Be aware that you may loose any batched log entries held in the internal table in-between log commit operations.

6.12.10.6 Programming "direct write" Messages

Type	Parameters	Description
ECS_ WRITE_ LOG	{severity} {level} {message}	Inserts a message into the ECS log table. The severity must be (I-info, W-warning , E-error, T-trace) and the level can be any number from 1 to 9. The message may be any text string from 1 to 255 characters.
ECS_ WRITE_ LOG_ ULINE	[no parameters]	Write an underline

Type	Parameters	Description
ECS_WRITE_LOG_SKIP	[no parameters]	Skip a line

Example

```
ECS_WRITE_LOG 'I' '1' 'information... this is a level 1 message'.
ECS_WRITE_LOG 'W' '2' 'warning... this is a level 2 message'.
ECS_WRITE_LOG 'E' '3' 'error... this is a level 3 message'.
```

Important: The data string is a single argument of maximum 255 characters. Use structures, field symbols, write or concatenate to structure a string with multiple values. Each ECS direct log write performs a synchronous insert into the ECS log table and then performs a commit work. If many entries are written to the log, the 'batched' mechanism is more efficient. It is possible to call the /SKY/WRITE_LOG function directly, but this is not recommended because the internal call structure may change between ECS releases. By using the macro calls, the programmer is shielded from the 'under the cover' workings.

6.12.10.7 Writing Report List Details to the Log

It may be convenient for the programmer to use the ABAP WRITE statement to list prospective log entries. ECS provides a command to copy the current list (dictated by SY-LSIND) to the ECS log.

Note: The ECS log has a limit of 255 characters, so anything that exceeds this, is truncated.

Type	Parameters	Description
ECS_COPY_LIST_TO_LOG	[no parameters]	Copies the contents of the current list to the log for the current run instance.

Type	Parameters	Description
ECS_COPY_LIST_ LINE_TO_LOG	[no parameters]	Copies the current list line to the log.

Important: Only the 1st 255 characters of each line are copied. Positioning on the correct list is the responsibility of the programmer. The current list as dictated by SY-LSIND is copied. ECS do not delete the list once it is copied. This is the responsibility of the programmer.

6.12.10.8 Writing Message Details to the Log

After a message is issued, that is, MESSAGE S001, the message may be automatically written to the ECS log. ECS takes the current message settings from SYST.

Type	Parameters	Description
ECS_COPY_ MESSAGE_TO_LOG	{severity} {level}	Copies the contents of the current list to the ECS log for the current run instance.

6.12.10.9 Writing Trace Entries to the Log

You may use the ECS log to write optional program trace entries, that is, the specified entries are only written to the ECS log when the trace option is activated at either the Process or Phase level. To specify trace entries, simply use a 'T' for the severity level for any of the ECS_APPEND_LOG, ECS_ WRITE_LOG and ECS_COPY_MESSAGE_TO_LOG commands. Checking the trace mode at either the Process or Phase level through the ECS workbench usually activates the trace facility. Alternatively, you may turn on off the trace programmatically.

Type	Parameters	Description
ECS_SET_TRACE_ON	[No parameters]	Activates the trace facility
ECS_SET_TRACE_OFF	[No parameters]	De-activates the trace facility

6.12.10.10 Generic Example of all Logging Functions

This example is based on sample program YECS9000 that is executed by ECS sample Process ECS_TEST_ABAP. It is a generic example of all the ECS logs options available.

Example

```
INCLUDE YECSSDK.
ECS_IMPORT_PARAMETERS.
```

```
batch up ECS log entries for the current run.
ECS_APPEND_LOG: 'I' '1' 'Test run details'.
ECS_APPEND_LOG: 'I' '2' V_ECS_*INAME.
ECS_APPEND_LOG: 'I' '2' V_ECS_*PNAME.
ECS_APPEND_LOG: 'I' '2' V_ECS_*RNUMBER.
ECS_APPEND_LOG: 'I' '2' V_ECS_*RSEQUENCE.
ECS_APPEND_LOG: 'I' '2' V_ECS_*RDATA.
ECS_APPEND_LOG_ULINE.
ECS_APPEND_LOG_SKIP.
ECS_APPEND_LOG: 'W' '1' 'Show warning'.
ECS_APPEND_LOG: 'W' '2' 'This is a test warning message'.
ECS_APPEND_LOG: 'W' '2' 'to demonstrate what it looks like !'.
ECS_APPEND_LOG_ULINE.
ECS_APPEND_LOG_SKIP.
ECS_APPEND_LOG: 'E' '1' 'Show error'.
ECS_APPEND_LOG: 'E' '2' 'This is a test error message'.
ECS_APPEND_LOG: 'E' '2' 'to demonstrate what it looks like !'.
ECS_APPEND_LOG_ULINE.
```

```
demonstrate trace facility (this may be activated optionally at the
process, phase or start process level)
ECS_SET_TRACE_ON.
ECS_APPEND_LOG_SKIP.
ECS_APPEND_LOG: 'T' '1' 'test append log trace level 1'.
```

```
ECS_APPEND_LOG: 'T' '2' 'test append log trace level 2'.
ECS_APPEND_LOG: 'T' '3' 'test append log trace level 3'.
ECS_APPEND_LOG_ULINE.
ECS_SET_TRACE_OFF.
```

```
now commit the batched up ECS log to the ECS database.
ECS_COMMIT_LOG.
```

```
demonstrate how to write direct log entries.
ECS_WRITE_LOG_SKIP.
ECS_WRITE_LOG 'I' '1' 'Direct log write level 1'.
ECS_WRITE_LOG 'I' '2' 'Direct log write level 2'.
ECS_WRITE_LOG 'I' '3' 'Direct log write level 3'.
ECS_WRITE_LOG_ULINE.
```

```
demonstrate trace facility using direct write technique.
ECS_SET_TRACE_ON.
ECS_WRITE_LOG_SKIP.
ECS_WRITE_LOG 'T' '1' 'test direct write trace level 1'.
ECS_WRITE_LOG 'T' '2' 'test direct write trace level 2'.
ECS_WRITE_LOG 'T' '3' 'test direct write trace level 3'.
ECS_WRITE_LOG_ULINE.
ECS_SET_TRACE_OFF.
```

```
write log to spool (just to demonstrate copy list to log!)
WRITE: / 'Write line 1 to spool'.
ECS_COPY_LIST_LINE_TO_LOG.
```

```
copy the entire list to the log
ECS_COPY_LIST_TO_LOG.
```

```
test copy message to log using trace.
MESSAGE S080 WITH 'Test message'.
ECS_COPY_MESSAGE_TO_LOG 'T' '1'.
```

6.12.11 Processing Files

ECS has commands and configuration options that make the handling of external files far simpler than native ABAP and issuing of external commands from SAP R/3. It is possible to make use of either the configuration options at the Phase and Directory Poll Attribute level or alternatively, the imbedding of ECS commands in ABAP programs to control external files. See the section on File management for more details on processing external files.

Command	Parameters	Description
ECS_ CREATE_ FILE	[no parameters]	Create a file name based on the system defaults. V_ECS_*FILE_NAME is set.
ECS_ GENERATE_ FILE_NAME	[no parameters]	Only generates the file name in the same format as ECS_CREATE_FILE. No file is created/ opened.
ECS_ CHECK_ FILE_ EXISTS	[no parameters]	Check that the primary file V_ECS_*FILE_ NAME exists
ECS_OPEN_ FILE_FOR_ INPUT	[no parameters]	Open the file name specified in v_ECS_*FILE_ NAME for input in fixed text mode.
ECS_OPEN_ FILE_FOR_ OUTPUT	[no parameters]	Open the file name specified in v_ECS_*FILE_ NAME for output in fixed text mode.
ECS_OPEN_ FILE_FOR_ APPEND	[no parameters]	Open the file name specified in v_ECS_*FILE_ NAME for appending in fixed text mode.

Command	Parameters	Description
ECS_ CLOSE_FILE	[no parameters]	Close the file name specified in v_ECS_ *FILE_NAME
ECS_SET_ FILE_NAME	{file name}	Sets the primary file name variable V_ECS_ *FILE_NAME.
ECS_SET_ FILE_NAME2	{file name}	Sets the secondary file name variable V_ECS_ *FILE_NAME2.
ECS_ SWITCH_ FILES	[no parameters]	Switches the primary and secondary file names. You may still use the old command, ECS_SWITCH_FILE_NAMES.
ECS_ WRITE_TO_ FILE	{data or variable}	Transfer data to file that is currently processed
ECS_ APPEND_ TO_FILE	{data or variable}	Transfer data to file that is currently processed
ECS_ DELETE_ FILE	[no parameters]	Delete the file currently that is processed
ECS_READ_ FROM_FILE	{into variable}	Read file currently that is processed into the specified variable or structure. ***Note:*** The file opens in text mode, therefore, the amount read is determined by the "end of line marker".

Command	Parameters	Description
ECS_COPY_TABLE_TO_FILE	{internal table}	Copies the contents of the specified internal table and writes each record to the file currently being processed.
ECS_COPY_FILE_TO_TABLE	{internal table}	Copies the contents of the file currently being processed to the specified internal table.
ECS_COPY_FILE	[no parameters]	Copies the file name specified in the primary file name V_ECS_*FILE_NAME to the secondary file name V_ECS_*FILE_NAME2. Note that the file names are switched after the command is executed successfully.
ECS_MOVE_FILE	[no parameters]	Moves the file name specified in the primary file name V_ECS_*FILE_NAME to the secondary file name V_ECS_*FILE_NAME2. Note that the file names are switched after the command is executed successfully.
ECS_MOVE_FILE_AND_REPLACE	[no parameters]	Same as ECS_MOVE_FILE except that the target file will be replaced if it exists already.

Note: You can handle only two files at a time using this method. The primary file name is stored in V_ECS_*FILE_NAME and the secondary file name is stored in V_ECS_*FILE_NAME2. These are set using the ECS_SET_FILE_NAME and ECS_SET_FILE_NAME2 commands. All files are processed in "text mode" (that is, fixed record format). If the file contains variable length records, the results may be unpredictable.

6.12.11.1 Specifying File Names (Logical or Physical)

Most of the ECS commands that are processing external files don't require a file name to pass as a parameter. Processing is almost always dependent on the v_ecs_*file_name/2 internal variables that are set. Only the ECS_SET_FILE_NAME/2 commands require for the file name to explicitly specify, otherwise the variables are automatically set through the ECS_IMPORT_PARAMETERS, ECS_CREATE_FILE and ECS_GENERATE_FILE_NAME commands. You may process up to two file names concurrently and pass between phases, the focus is switched between the two using the ECS_SWITCH_FILES command. The file name may be a SAP logical name or a direct physical name. You may also specify standard substitution variables that are resolved automatically by the ECS file manager. Refer Dynamic variable substitution in logical or physical file names in the File management section for details.

6.12.11.2 Error Processing

Return codes are recorded in the variable v_ecs_*rc and the message is returned in v_ecs_*msg. See the section Checking return codes for more details.

6.12.11.3 Generic Example of all Functions

These examples are based on sample programs YECS9019 and YECS9007 that are executed by ECS sample Process ECS_TEST_FILE_CMD.

You can handle only two files at a time using this method. The primary file name is stored in V_ECS_*FILE_NAME and the secondary file name is stored in V_ECS_*FILE_NAME2. You set these filenames using the ECS_SET_FILE_NAME and ECS_SET_FILE_NAME2 commands. All files are processed in "text mode" (that is, fixed record format). If the file contains variable length records, the results may be unpredictable.

Example

```
INCLUDE YECSSDK.
DATA: V_MSG(255) TYPE C,
V_REC(80) TYPE C,
V_RC(3) TYPE N.

ECS_IMPORT_PARAMETERS.
ECS_REFRESH_LOG.
ECS_APPEND_LOG 'I' '1' 'Testing external file management commands:'.
```

- ----------
- Test creation of file, writing data and close file sequence
- ----------

```
ECS_APPEND_LOG 'I' '2' 'Testing create, write to file and close:'.
```

- create new file.

```
ECS_CREATE_FILE.
V_RC = V_ECS_*RC.
CONCATENATE 'Create file, rc=' V_RC 'name=' V_ECS_*FILE_
NAME
INTO V_MSG SEPARATED BY SPACE.
ECS_APPEND_LOG 'I' '3' V_MSG.
```

- check file we have created exists.

```
ECS_CHECK_FILE_EXISTS.
V_RC = V_ECS_*RC.
CONCATENATE 'Check exists, rc=' V_RC 'name=' V_ECS_*FILE_
NAME
INTO V_MSG SEPARATED BY SPACE.
ECS_APPEND_LOG 'I' '3' V_MSG.
```

- write test records to file.

```
ECS_WRITE_TO_FILE 'test write to file record'.
V_RC = V_ECS_*RC.
CONCATENATE 'Write to file, rc=' V_RC 'name=' V_ECS_*FILE_
NAME
INTO V_MSG SEPARATED BY SPACE.
ECS_APPEND_LOG 'I' '3' V_MSG.
```


- close the file

```
ECS_CLOSE_FILE.
```

6.12.12 Checkpoint Restart

ECS provides a facility to help programmers control checkpoint restart from within their ABAP programs. The concept of checkpoint restart is to store a key (unique data string) that identifies a logical completion of processing. If the program should fail and re-start, the stored checkpoints are analyzed to identify the point that was last successfully processed and from which to re-start execution. For example: When processing an external interface file, each record number is check pointed ***after*** the record is successfully processed. If the run should fail and to re-process, the program skips past the records already processed until one is reached that is not check pointed. This is the position from which to re-start processing.

Command	Parameters	Description
ECS_INSERT_ CHECKPOINT	{variable/literal}	Insert a checkpoint data record (max 150 bytes)
ECS_LOCATE_ CHECKPOINT	{variable/literal}	Locate a checkpoint data record (max 150 bytes)

The result of these commands is returned in v_ecs_*rc. The last checkpoint data value is stored in variable v_ecs_*checkpoint_data.

6.12.12.1 Example: (Sample ABAP program YECS9018)

This is a very simple example of checkpoint restart where a Process (ECS_TEST_CHECKPOINT) is intentionally failed after a number of iterations within a loop. When the failed phase is re-processed, it progresses past the point of failure until a maximum of five iterations is exceeded, at which point the run is successful.

XXTITLEXX

```
INCLUDE YECSSDK.

DATA: V_LOG_DATA TYPE ECS_LOG_DATA,
V_INDEX(1) TYPE N.
```

```
ECS_IMPORT_PARAMETERS.
ECS_APPEND_LOG_ULINE.

DO 5 TIMES.
V_INDEX = SY-INDEX.


 • see if the checkpoint exists. If so, continue with next

iteration.
 ECS_LOCATE_CHECKPOINT V_INDEX.
 IF V_ECS_*RC = 0.
 CONCATENATE TEXT-001 V_INDEX INTO V_LOG_DATA.
 ECS_APPEND_LOG 'I' '1' V_LOG_DATA.
 CONTINUE.
 ENDIF.

 • insert checkpoint.

 ECS_INSERT_CHECKPOINT V_INDEX.
 CONCATENATE TEXT-002 V_INDEX INTO V_LOG_DATA.
 ECS_APPEND_LOG 'I' '1' V_LOG_DATA.

 • intentionally fail the phase.

 ECS_SET_PHASE_AS_REPROCESSABLE.
 ECS_END_PHASE_AS_FAILED.
 ECS_APPEND_LOG 'I' '1' TEXT-003.
 EXIT.

 ENDDO.

 ECS_COMMIT_LOG.
```

6.13 ECS Configuration

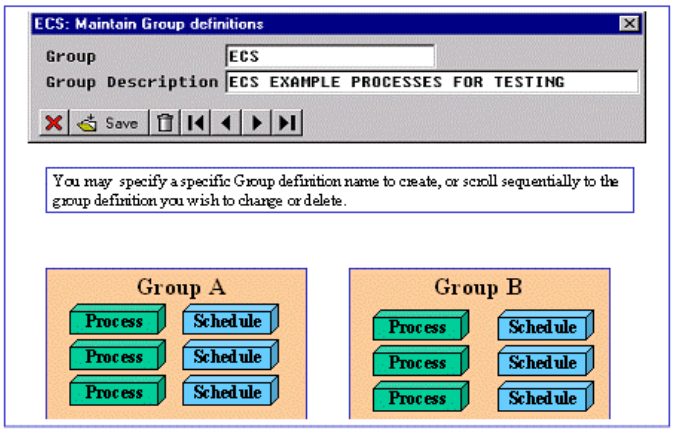
Apart from the ECS system configuration, there are optional configurations that you may use to help structure and organize the ECS Process and Schedule definitions. You can access these through the ***setup*** menu in the Process Workbench (***YECW***).

6.13.1 Configuring Group Definitions

Groups are used to segregate ECS Process and Schedule definitions. You may enter groups as selection criteria to effectively filter the content of online screens and reports. A group is free format and you may define at any time. However, you may delete it only if it is no longer referenced by any Process or Schedule definitions.

You may maintain group definitions through the Workbench **setup** menu option:

- Logon to the Client to configure
- Execute transaction YECW to start the ECS Workbench (execute the selection screen)
- Select the **group definitions** option from the **setup** menu.



> ***Note:*** It is not necessary to enter groups immediately. You may maintain them at any stage to help structure and categorize Process and Schedule definitions.

Groups are entirely optional and you should create on a "as needs" basis. You may not be able to delete a process group if it is still allocated to any Process or Schedule definition.

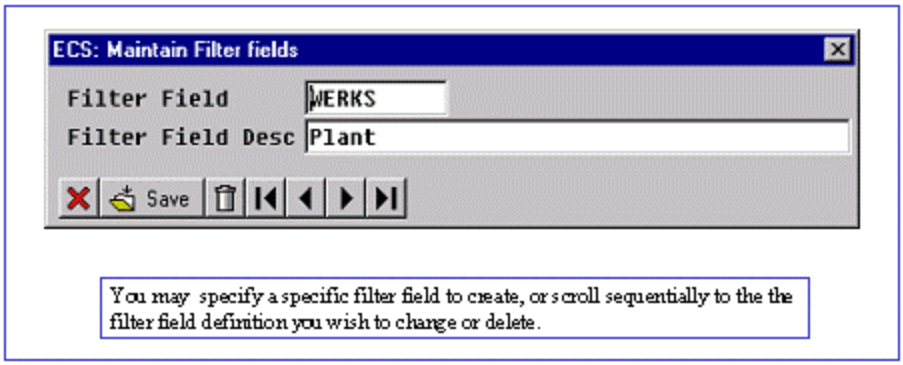
6.13.2 Configuring Filter Definitions

You use filter definitions to identify specific runs. You specify the filter value at run time and can use in report criteria to narrow the list contents. You may also use it in a security check to prevent run data from listed, for example, you may set up filter fields for Plant, Business area, and Division. There is a one-to-one relationship between a run and its filter data. Filter values are configured using the ecs_set_filter command (see the programming guide for details). The filter field and data is supplied at run time and recorded against the run.

For example: You may start a process to perform a material movement for plant P001. The filter field: WERKS and filter data P001 is provided to ECS through the ecs_set_filter command. If you want to view runs for plant P001, specify WERKS and P001 in the ECS filter selection criteria.

You maintain filter definitions through the Workbench **setup** menu option:

- Logon to the Client to configure
- Execute transaction YECW to start the ECS Workbench (execute the selection screen)
- Select the **Filter fields** option from the ***setup*** menu.



> ***Note:*** It is not necessary to enter filter fields immediately. You can maintain the filter fields at any stage to help structure and categorize runs. Filter fields and data are not validated.

Filter fields are entirely optional and you need to create on a "as needs" basis. Refer the ECS_SET_FILTER command in the "Programming guide".

6.14 ECS Appendix

6.14.1 Invoking an ECS Process through SAP Output Determination

In many cases, you may need a Process to invoke from within a user exit or through output determination. In normal cases, this is a simple ECS_START_PROCESS. However, if the invoking program executes in an Update task, there are restrictions with regard to what commands are possible, for example, COMMIT WORK. Another method available to trigger an ECS Process under these circumstances is to use ECS_START_VIA_EVENT. Example program using NAST:

```
REPORT Z???????.

* Tables
*-------
TABLES: NAST.

* includes.
* ---------
INCLUDE YECSSDK.

* Data Variables
*---------------
DATA: V_RUN_DATA LIKE YECS_EVENT-RDATA.

CONSTANTS: C_PROCESS_NAME LIKE YECS_EVENT-INAME  " ECS Process name
                              VALUE '????????????'.

*---------------------------------------------------------------------
----
*  This form is called by the standard output processing program to
*  initiate a ECS event to execute ?????????????? with the NAST
object
*  key (delivery, order)
```

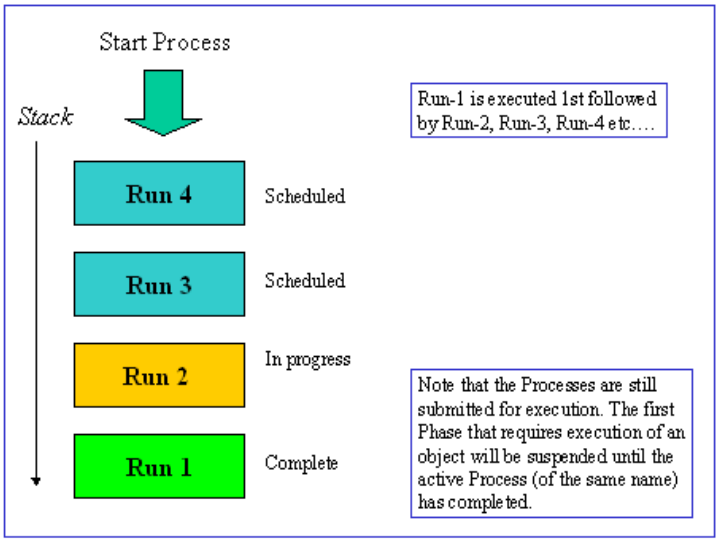
```
*---------------------------------------------------------------------
----
FORM ENTRY USING V_SUBRC V_SCREEN.

   CLEAR V_SUBRC.
   V_RUN_DATA = NAST-OBJKY.

*  Insert the entry into the event queue.
   ECS_SET_RUN_DATA V_RUN_DATA.
   ECS_SET_USER SY-UNAME.
   ECS_START_PROCESS_VIA_EVENT C_PROCESS_NAME.

ENDFORM.
```

6.14.2 Serialization of Processes

You may allow some Processes to execute in a sequential fashion (that is, one after the other). This is necessary to enforce an order of execution, for example, prevent exclusive control interlocks over system resources.

It is necessary to:

- Execute Processes sequentially in the order in which they are started.
- Delay the execution of the next process in line if the current process failed.
- Ignore a new execution of a Process if one already executes or executed in a certain time frame.



You may serialize each run of a process by either:

- Configuring the Process definition. Refer The Workbench for more details.
- Issuing ECS commands from an ABAP/4 program. Refer the section "Programming Guide" for more details.

You may ignore or prevent the starting of a new Process automatically by:

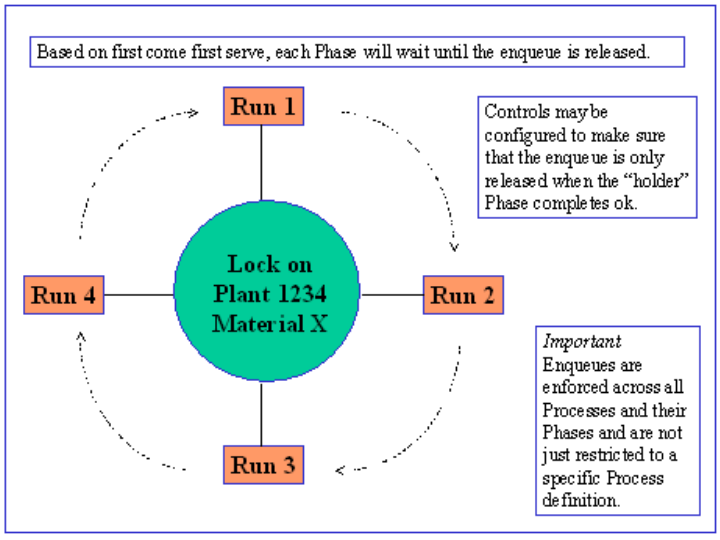
- Configuring the Process definition to ignore starting a Process if one is already in-progress.
- Configuring the Process definition to fail the start of a new Process if one already runs within a specified time frame.

> ***Important:*** Serialization of a Process and issuing an enqueue for a Phase within a Process are mutually exclusive since serialization issues an internal enqueue using the process name as the enqueue key.

6.14.3 Enqueuing Phases

> ***Note:*** The main difference between serialization of a Process and the enqueuing of a Phase, is that serialization forces each Process, in its entirety, to execute sequentially, whereas an enqueue only applies at the Phase level within a Process. In addition, an enqueue is enforced across all Phases that require It, whereas Process serialization only applies to a specific Process definition. Each enqueue is released when the Phase ends. You may also configure each enqueue that the enqueue is only released when the Phase completes successfully.

In a way, serialization of a Process is a type of enqueue that only permits one instance of an entire Process to execute at a time. However, it may be necessary to enqueue the execution of individual Phases based on other criteria such as business data. An enqueue ensures that only one execution occurs at a time to avoid any conflict. In order for enqueue to work, you need to publish the enqueue criteria when a Phase starts. The enqueue is only honoured by the subscribers, that is, those who request the same enqueue during execution of any Phase.
For example: It is necessary to enqueue material movements by plant and material because SAP R/3 locks the data at this level. In this case, all Phases that perform material movements are started with the plant and material as their enqueue key. ECS then ensures that all executions of Phases, with the same enqueue key, are serialized based on a "first come first serve" basis.



You may specify the ECS enqueue using the following methods:

- Configured against the Phase definition using the Workbench.
- Through embedded ECS commands in a ABAP/4 program. See the section "Programming guide".
- When the Process starts manually from within the ECS online system.

> ***Important:*** You may not specify an enqueue in conjunction with the serialization of a Process, since serialization already uses the enqueue option with the Process name as the enqueue key.

6.15 Job Scheduling

ECS job scheduling builds on some of the basic scheduling features and functionality already that SAP provided. The ECS Scheduler provides comprehensive scheduling functions, multilevel scheduling and complete integration with the SAP calendar and the job management system. ECS provides utilities to automatically convert existing SAP job definitions to ECS schedule definitions. The ECS schedule manager is used to define Schedule definitions

6.15.1 Key Topics

How the Scheduler Works
Schedule Workbench
Monitoring Schedules

6.15.2 How the Scheduler Works

As mentioned previously, a Schedule definition (made up of levels and items) determines the periodic timing, the order and the tasks to execute. When a schedule definition is ready, it is activated. The activation process submits a ***schedule execution job*** called {default job prefix}_{schedule name}_{client}. This job is submitted to execute using the schedule definition interval for example, every weekday every hour.

When SAP starts the scheduled job, it reads the schedule definition and starts the Process / Schedule defined against each schedule item. All items are released in the order of the level they belong to. ECS uses its start on event options to do this. ECS waits until all Process runs are executed and are at status COMPLETE before releasing processes at a lower level. If any processes have failed, ECS defers execution until they are successfully restarted, force completed or deleted.

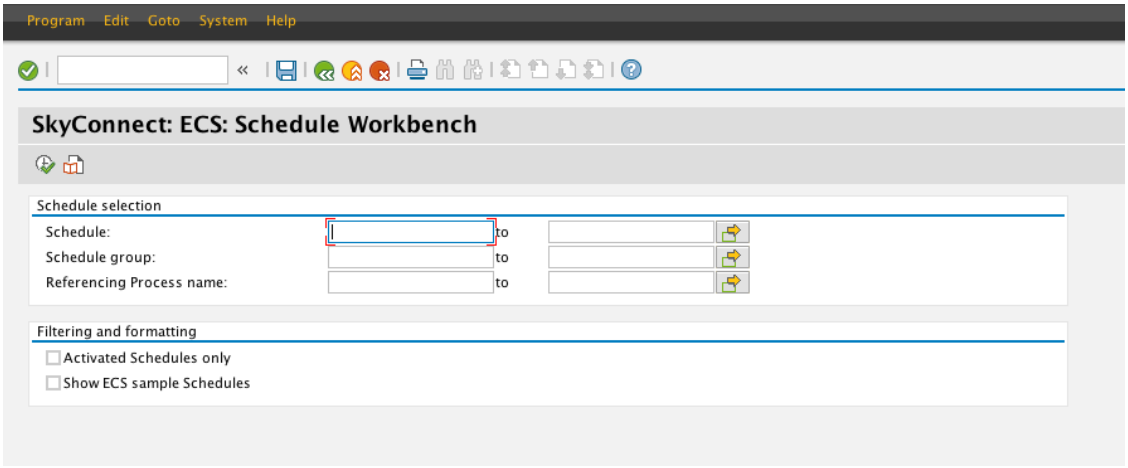
6.15.2.1 Nesting Schedules

Nested Schedules are by default considered part of the Master Schedule definition. Therefore, the Schedule only continues with the next level once the entire nested schedule are completed. The **no dependency (separate run)** option causes ECS to still start the Schedule, but processing does not wait for the spawned schedule to complete.

6.15.3 The Schedule Workbench

The Schedule Workbench allows to define periodic processing and dependency controls to execute ECS process definitions in the background. The Scheduler supports complex multi-level and nested schedule definitions. You may also configure low level exceptions (time ranges) to enable complete control.

Select **Schedule Manager** option from the ECS services menu from any of the **Goto** menus, or start it directly using transaction YECT. The following selection screen appears:



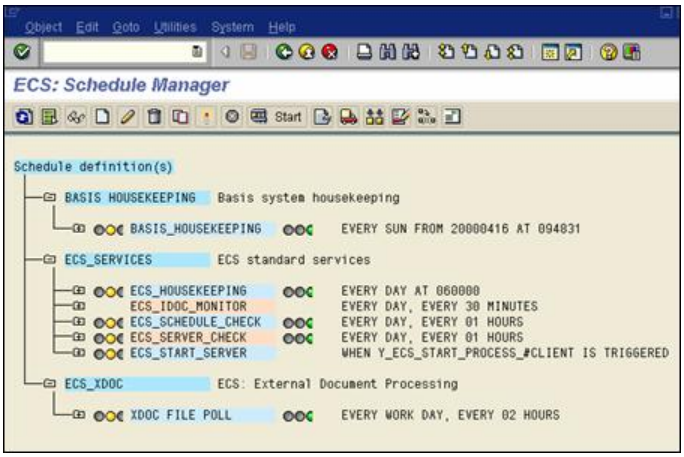
6.15.3.1 Key Topics

Displaying Schedules
Maintaining Schedules
Maintaining Phases
Dependency Controls
Wizards

6.15.3.2 Displaying Schedule Definitions

You can create and maintain Schedule definitions, levels and items through the main schedule screen. The screen lists all the selected definitions in a hierarchical tree structure showing each definition, the levels and items as separate nodes in the tree. Traffic lights and colors help to determine the current status "at a glance".

You may expand or collapse the tree nodes by clicking the " + " or " - " boxes. There are also icons on the application toolbar to choose a specific schedule, permanently expand a definition, refresh the display.



Screen Fields and Icons

The Schedules are listed in alphabetical order with any assigned ECS group. The tree structure is used to display the hierarchical levels and items. Each Schedule definition has two statuses indicated by traffic lights. The traffic light on the left hand side of the schedule name indicates the status of the Schedule job definition. The traffic light to the right of the Schedule name indicates the status of the Schedule runs.

Schedule Jobs

The selected schedules are listed in alphabetical order. If SAP job definitions exist for the Schedule, a traffic light appears on the left hand side of the name indicating whether the job is "in-progress", "complete" or failed. If the schedule job is in-progress, this takes precedence over any complete or failed jobs that may exist. If there are no jobs in-progress, failed jobs take precedence over completed

jobs. The traffic light also serves as a hotspot to easily display related schedule job definitions. The job definition is automatically maintained by ECS whenever a Schedule definition is activated. The jobs are automatically house kept by ECS.

> ***Note:*** Jobs are not the jobs to execute Process Phases, they are "bootstrap" jobs that are scheduled to start the Schedule definition at the defined interval.

Schedule Runs

A Schedule run definition is identified by a unique number and contains all the detail of the Schedule execution and its current status. If the traffic light is green, it means that all Schedule runs are completed successfully, yellow indicates that one or more runs are still in-progress and red indicates that one or more schedule runs failed.

Schedule Name Colour Codes

If you enable or activate a schedule, the name is highlighted in green. If you deactivate the Schedule, the name appears in purple. If there are failed jobs for the schedule, the name is highlighted in red.

Exceptions

If 'low level' scheduling exceptions are configured for the Schedule definition, they are indicated by an icon to the right of the Schedule run traffic light.

Maintaining Definitions

You maintain the schedule definitions by positioning the cursor on a node, or relevant area of the screen, and clicking an application toolbar icon. For instance:

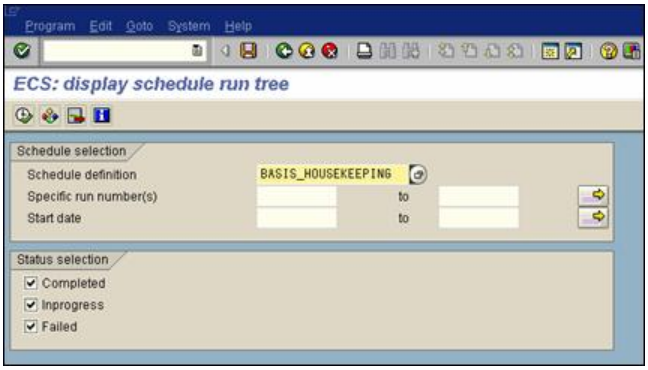
- To create a new Schedule definition, position the cursor on the top level node "Schedule definitions" and click the create icon in the application toolbar.
- To maintain a Schedule definition, position the cursor on the Schedule and click the change icon.
- To create a Schedule Level, position the cursor on the Schedule and click the create icon.
- To create a Schedule Item, position the cursor on the Schedule Level and click the create icon.

Displaying Schedule Job Definitions

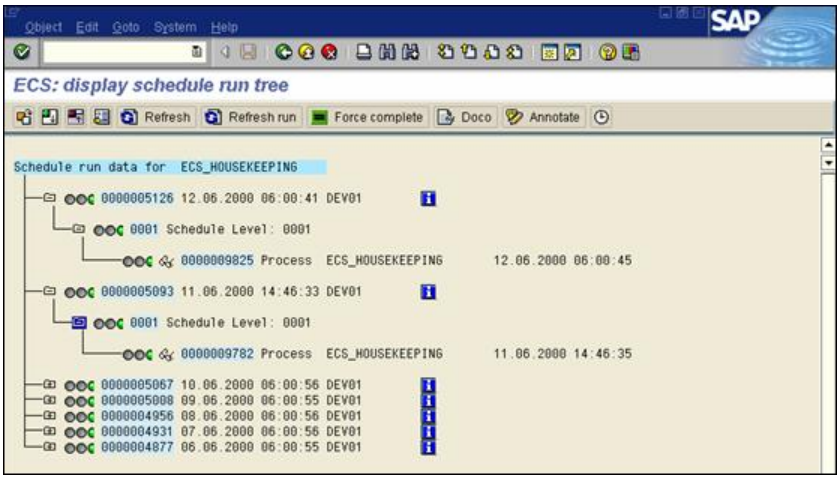
When you activate an ECS schedule, it generates a SAP job definition. An yellow traffic light should appear, indicating that the Schedule is in-progress. You may display the Schedule job definition by positioning the cursor on the icon and clicking. The SM37 job display is invoked.

Displaying Schedule Runs

Every execution of a Schedule is assigned a unique number that you use to track its status. If a Schedule has runs, a traffic light appears on the right hand side of the Schedule name. You may display the Schedule runs definition by positioning the cursor on the icon and clicking. The following selection screen appears:



You may specify additional criteria to further restrict the list. Select the **Execute** icon to display the Schedule run tree display.



This report displays the current status of each schedule run and allows you to drill down to the Schedule levels and items.

Screen Fields and Icons

Force Complete

You may 'force complete' a schedule run at any level. Force complete, overrides any recorded failures and flags the run as completed. The run must be in a failed state before it may be force completed.

Canceling and Deleting Runs (Edit menu)

You may cancel a schedule run at any level. The run must be in a status of in-progress. Only the entire run may be deleted, and only when the run is in a status of complete.

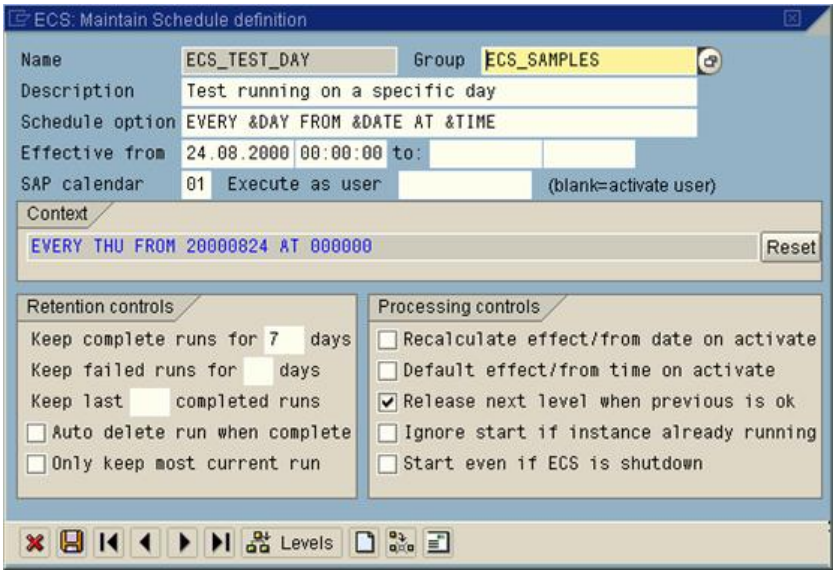
Annotation Log

Every Schedule run is annotated. The most severe status of the annotation log entries is shown by the information, warning or alarm icon on the right hand side of the Schedule run. The annotation log is only maintained at the run level. Manual entries are maintained through the annotate icon on the application toolbar.

6.15.3.3 Maintaining Schedule Definitions

The maintenance of Schedule definitions is invoked from either the main screen by positioning the cursor and clicking either the **Create** or **Maintain** icons on the application toolbar, or through the "drill down" level and item icons on the pop-ups. The scroll buttons are useful to position on a specific entry without having to return to the main schedule display each time.

> ***Important:*** You may not maintain active schedules, you need to deactivate them first. Be careful to save each time these screens are navigated. No confirmation prompts are issued.



This model screen is used to specify the name, description and schedule interval. The scheduling interval is selected from a list of predefined options. The user is then prompted for effective dates and times, number and day details where required. Refer Schedule options and How the Scheduler works for more details.

Screen Fields and Icons

Schedule Option

This field is key to the operation of the ECS scheduler. Instead of using complicated prompts and buttons, the approach is to assign 'schedule objects' that describe the schedule and contain substitution variables. Use the search help to view a list of available schedule options. You are then automatically prompted to supply any substitution variables that the option may need. The context display shows the complete schedule option with variables.

Effective from/to Dates

Indicates when the date and time range for when the schedule is allowed to execute. The effective to date and time is optional, initial indicates forever. You need to specify the effective from date and time and may automate using the **calculate the effective from date** and **set effective from time** options.

SAP Calendar

All scheduling options in ECS require a SAP calendar definition to be defined. This is to enable proper work day and holiday processing. If no schedules are specified, use the SAP default factory calendar 01.

Re-calculate effect/from Date on Activate

Select this option if you want ECS to automatically determine the next effective from date to use for the schedule option, when the Schedule is activated.

Default Effect/from time on Activate

Select this option if you want ECS to automatically default the effective from time to the current time, when the schedule is activated.

Release Next Level when Previous is Ok

This setting indicates that all Process and nested schedule items need to complete successfully before the next level of the Schedule is released for processing. If this option is set at the Schedule definition level, it is applied to all Schedule levels. If this option is not set at the Schedule definition level, you may specify it may independently for each schedule level.

> ***Note:*** If this option is not set at either the schedule or its levels, the schedule run completes successfully, even if schedule items failed.

Ignore Start if Instance Already Running

Do not start the schedule if a run is already active (in-progress). No run is created.

Start even if ECS is Shutdown

The Schedule is started, even if ECS is shutdown. If this is not set (default), the start is ignored until ECS is restarted. See the section on System Management for more details.

Retention Controls

The configurations dictate how ECS is to perform housekeeping on the Schedule runs:

- Keep complete runs for x days. Housekeeping automatically deletes complete runs when the specified number of days expire. This value need to be between 2-999 days.
- Keep failed runs for x days. If specified (default 0), ECS automatically forces complete failed schedule runs when the specified number of days expire. This value must be between 2-999 days. Specify zero (0), if failed runs are to be kept indefinitely or until the system value configured in the ECS system defaults.
- Keep last x completed runs. This configuration is useful to automatically keep only the latest successful runs and 'roll off' the others. The value may be from 1-999, 0 indicates all completed runs are kept until the retention period is met.
- Auto delete when complete. Any successful Schedule run automatically deletes when it completes. This is useful for high volume situations where there is no need to keep an audit trail of the Schedule runs.
- Only keep most current run. Only one run, the most recent is kept. If the **Release next level when previous is ok** option is in effect, other runs are only deleted when the run completes successfully. If this option is not configured, all other runs are deleted irrespective of whether the run failed or completes successfully.

Schedule Level

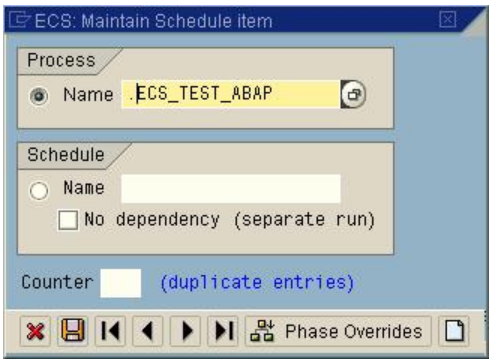
An internally generated number starting from 0001 uniquely identifies each level. You may enter a description to describe the purpose of the level. The level indicates the 'tier' of Schedule and enables multi-level control. All Processes and nested Schedules on the same level are released for execution concurrently. Every Schedule definition must have at least one level.

Only Release when all Preceding Items have Completed

This setting indicates that all Process and nested schedule items must complete successfully before this level of the Schedule is released for processing. If you specify this option at the Schedule definition level, the setting is defaulted automatically and may not change.

Schedule Item

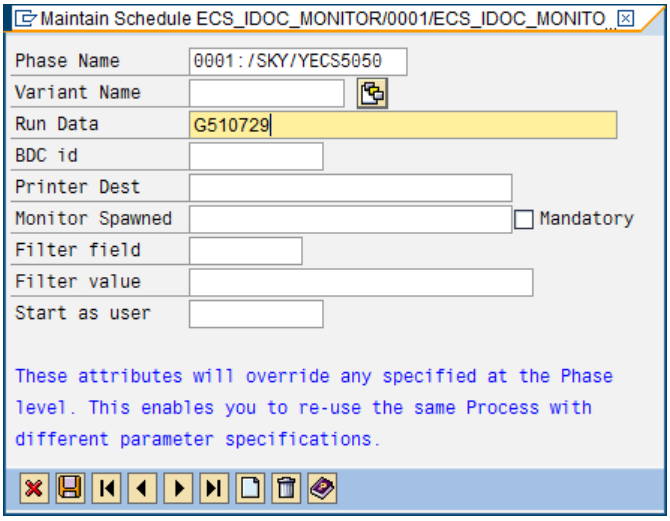
The item model screen prompts for either: a unique ECS Process, or nested Schedule that is to start when the Schedule level is released. You may specify multiple instances of the same Process or Schedule through a unique counter.



As mentioned previously, you may specify either a Process or another Schedule as a Schedule item. You may overcome duplicate Process and Schedule names by using the counter field, for example,1, and 2. You may override various attributes of the processes phases by using the phase override option, that is, position your cursor on the processes phase and click create.

Phase Overrides

Phase overrides allow you to specify alternate values for the processes phase to execute. This can be useful to use a different variant, and user to that of the process. In this way, you may also use template processes repeatedly in different schedules.



6.15.3.4 Scheduling Option Codes

On the Schedule definition screen, prompts for the following fields appear:

- Schedule option
- Effective from/ to dates times
- SAP Calendar
- **Substitute effective from/to date when Schedule is activated** check box
- **Set effective from time to current time when schedule activated** check box

You use these fields to define the scheduling interval. Note the use of &DAY, &NUMBER, &TIME, &DATE, &VARIABLE and &SCHEDULE variables. These variables are substituted either automatically, or through pop-up boxes when required. This method provides an easy mechanism to specify what type of schedule is required, without having to use complicated prompts. Therefore, it is easy to identify the schedule option **at a glance** from the main screen or definition screen.

The following Schedule options are available:

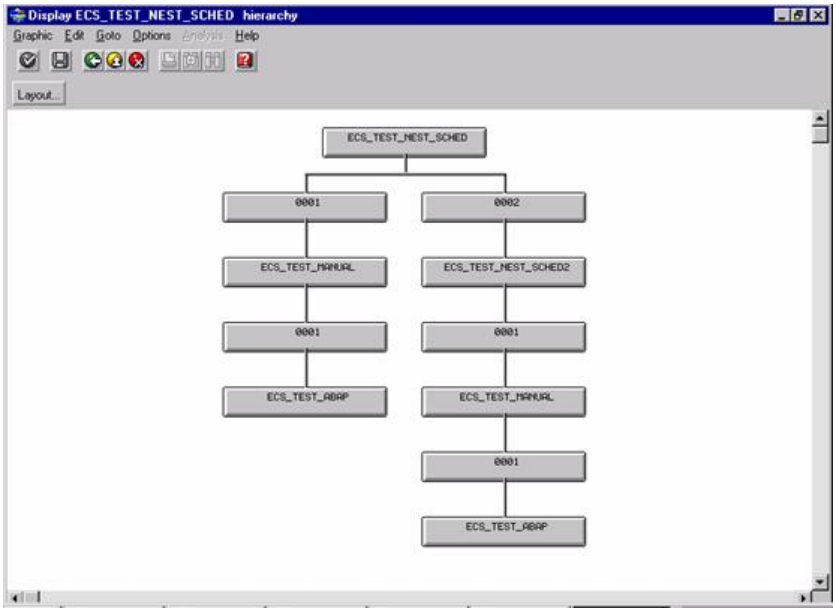
Schedule Option	Description
EVERY &DAY FROM &DATE AT &TIME	On a specific day starting from the specified date and time
EVERY &NUMBER DAYS FROM &DATE AT &TIME	Every x days interval starting from the specified date and time
EVERY &NUMBER MONTHS FROM &DATE AT &TIME	Every x monthly intervals starting from the specified date and time
EVERY &NUMBER WEEKS FROM &DATE AT &TIME	Every x weekly interval starting from the specified date and time
EVERY DAY AT &TIME	Every week day at a specific time

Schedule Option	Description
EVERY DAY, EVERY &NUMBER HOURS	Every week day, every x hourly intervals
EVERY DAY, EVERY &NUMBER MINUTES	Every week day, every x minute intervals
EVERY WORK DAY AT &TIME	Every working day at a specific time
EVERY WORK DAY, EVERY &NUMBER HOURS	Every working day, every x hourly intervals
EVERY WORK DAY, EVERY &NUMBER MINUTES	Every working day, every x minute intervals
ON &DATE AT &TIME	At a specific date and time
FIRST DAY OF EVERY MONTH	On the first day on the month
FIRST WORKING DAY OF EVERY MONTH	On the first working day of the month
LAST DAY OF EVERY MONTH	On the last day of the month
LAST WORKING DAY OF THE MONTH	On the last working day of the month
MANUAL	You can only start the schedule manually
NESTED SCHEDULE	You may execute the schedule definition only as a nested schedule
START AFTER SCHEDULE &SCHEDULE	Only start the Schedule when the named schedule completes

Schedule Option	Description
WHEN &EVENT IS TRIGGERED	Start the schedule when the named SAP background event is triggered.
WHEN &EVENT IS TRIGGERED WITH &VARIABLE	Start the schedule when the named SAP background event is triggered, qualified by a event parameter (&variable). ***Note:*** The current client may be dynamically substituted by specifying #CLIENT in the event name.
WORKING DAY &NUMBER BEFORE THE END OF EACH MONTH	On working day offset nnn before the end of the month, for example, 03 is the third working day before the end of the month.
WORKING DAY &NUMBER FROM THE BEGINNING OF EACH MONTH	On working day offset from the start of the month, for example, 03 is the 3rd working day from the beginning of the month.

6.15.3.5 Scheduling Graphical Hierarchy View

For complex schedule definitions (many levels, items and/or nested schedules), the graphical hierarchy view clarifies the Schedule definition and its relationships. Position your cursor on a Schedule node in the Schedule manager, and click the hierarchy graph icon. The following view appears:

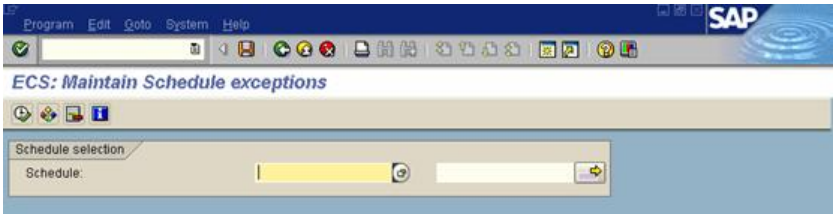


This view displays all the levels of the Schedule definition and its dependencies. Any nested schedules are expanded to their respective levels. You may use the **Layout** button or **Options** menu to dynamically alter the graphs display.

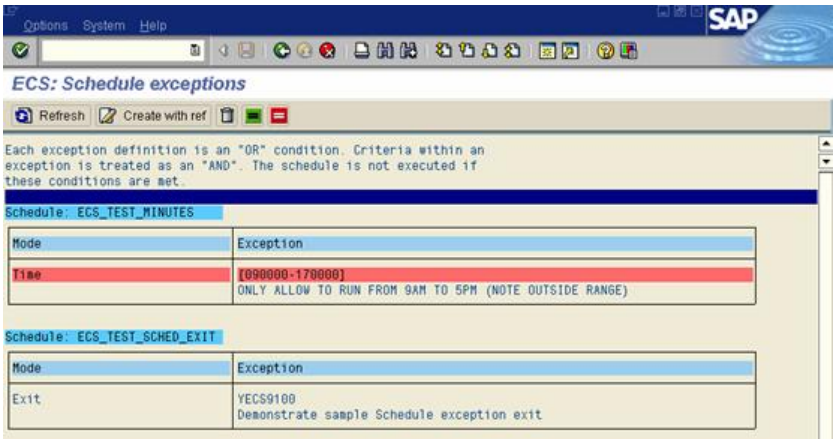
Schedule Exceptions

Apart from Scheduling options (Periodic, and Event) that you define on the Schedule definition, you may also define exceptions that may ***prevent*** the Schedule from started. For example: If you want a Process to execute every hour on every workday, you specify this as a Schedule definition scheduling option. However, if you only want the schedule to execute in business hours (09:00:00 - 18:00:00), this is an exception.

Exceptions are specified by either: clicking the  icon on the Schedule manager application toolbar or Schedule definition maintenance screen, or selecting the Maintain schedule exceptions from the edit menu. Once Schedule exceptions are defined against a Schedule, this icon also appears against the Schedule name on the Schedule manager list. This provides a 'fast path' selection to view or maintain the associated exceptions. Once selected, the following screen appears:



You may select all, a range or a specific Schedule. Click the **Execute** icon to display the exceptions.



The Schedule exception list is in Schedule order. You may create, change or delete entries by placing your cursor on the relevant area of the screen and selecting the **maintenance** option. You may specify as many exceptions as you require. Each exception may contain multiple criteria, such as date and time ranges. When the Schedule manager evaluates the exception definitions, each exception entry is treated as an OR condition, whereas the criteria within an exception is treated as an AND condition.

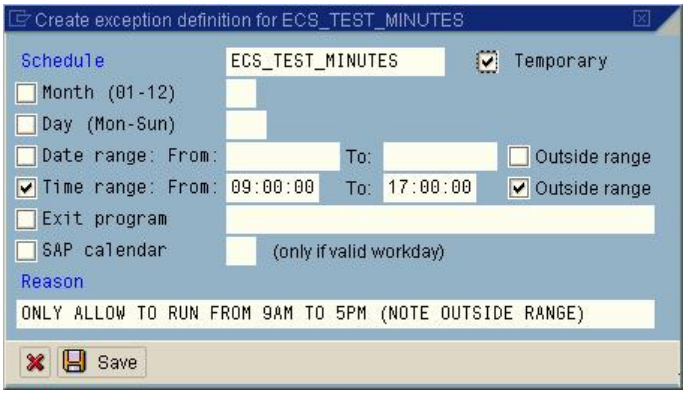
Enable/ Disable/ Temporary Exceptions

A exception may be disabled from use through the disable icon. Temporary exceptions are automatically disabled by the Schedule manager, the first time they are considered. Temporary exceptions are highlighted with a yellow background, disabled exceptions in red.

Schedule Exception Maintenance Screen

This screen lists the exception options that are possible. A single exception entry may contain multiple checks that are treated as an AND condition by the Schedule manager.

> *Important:* These options are exceptions, that is, do not execute the Schedule if the condition is met



Schedule Screen Fields and Icons

6.15.3.6 Temporary

The exception is automatically disabled by the Schedule manager after the first time it is considered. This feature is useful to implement 'once off' exceptions.

> ***Note:*** The temporary exception is disabled the first time it is considered, irrespective of whether it was actually triggered or not.

6.15.3.7 Month

A specific month number (01-12) when the Schedule is not required to execute.

6.15.3.8 Day

A specific day of the week (MON-SUN) when the Schedule is not required to execute.

6.15.3.9 Date Range

Do not execute the Schedule between these dates. The **outside range** check box is useful to enforce the opposite, that is, only execute the Schedule between these dates. For example, if you want to prevent a schedule from executing within a certain date range, enter From: yyyy/mm/dd To yyyy/mm/dd. If you only want to execute the schedule within a specific date range, enter the date range and select the **outside range** check box.

6.15.3.10 Time Range

Do not execute the Schedule between these time. The **outside range** check box is useful to enforce the opposite, that is, only execute the Schedule between these times. For example, if you want to prevent a schedule from executing within a certain time range, enter From: hh:mm:ss To: hh:mm:ss. If you only wish to execute the schedule within a specific time range, enter the time range and select the **outside range** check box.

6.15.3.11 Exit Program

Instructs the Schedule manager to call a user defined exit program passing it details of the Schedule run. You may then write your own programming code to either accept or reject the Schedule run.

> ***Note:*** Refer the following Schedule exception exit section for more details.

6.15.3.12 SAP Calendar

Specify a SAP calendar definition in order to execute the Schedule on specific days. The exception is triggered, if the current date is not a work day according to the calendar definition.

6.15.3.13 Reason

A free format comment to help further explain the exception definition.

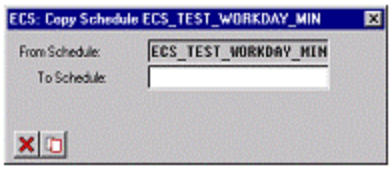
Schedule Exception Programming Exit

An external ABAP program may be executed to perform exception checking logic. The sample Schedule, ECS_TEST_SCHED_EXIT, demonstrates how this may be done, using a sample program YECS9100. ECS exports the details of the exception, held in structure YECS_SHEXP, to memory ID ECS_SCHEDULE_EXCEPTION_EXIT.

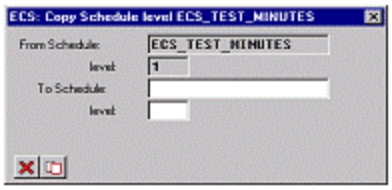
Example ABAP program (YECS9100)

```
* This program demonstrates how a schedule exception exit accepts
* parameters from ECS. RETURNCODE may be set to any non-zero value to
* abort starting the Schedule. Any other changes are ignored

REPORT  YECS9100.

DATA: V_PARM LIKE YECS_SHEXP.

* Receive parameter structure from ECS.
  IMPORT PARAMETER TO V_PARM
    FROM MEMORY ID 'ECS_SCHEDULE_EXCEPTION_EXIT'.

* Exception checking logic
  V_PARM-RETURNCODE = 1. " Prevent schedule run!

* Return parameter structure to ECS.
  EXPORT PARAMETER FROM V_PARM
      TO MEMORY ID 'ECS_SCHEDULE_EXCEPTION_EXIT'.
```

6.15.3.14 Copying Schedule Definitions Levels and Items

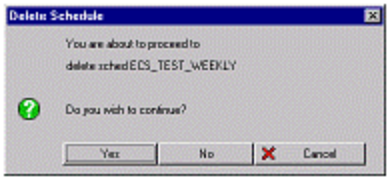


Type in the name of the Schedule definition to copy to and click on the copy icon.

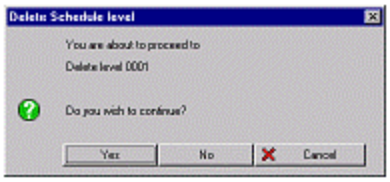


Type in the name of the Schedule and level to copy to and click on the copy icon. Note that the target Schedule must already exist.

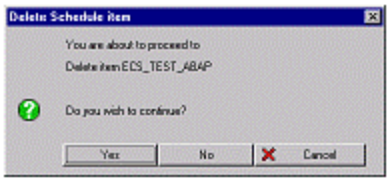
6.15.3.15 Deleting Schedule Definitions Levels and Items



This prompt confirms the deletion of a Schedule definition. Note that all the associated levels and items will be deleted as well.



This prompt confirms the deletion of a Schedule level. Note that all the associated items will be deleted as well.

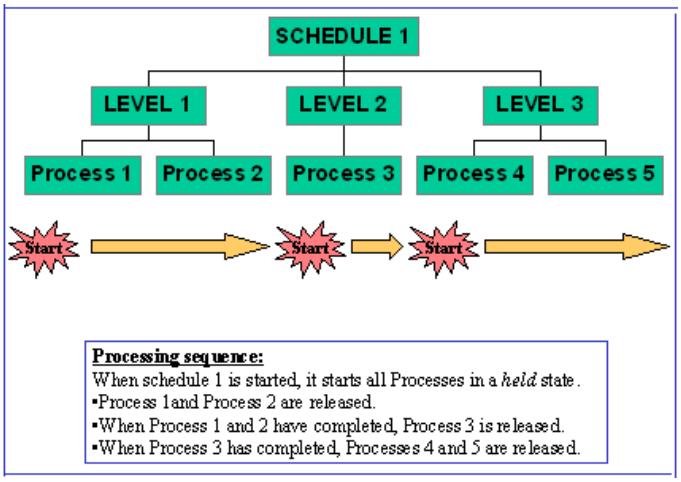


This prompt confirms the deletion of a Schedule item.

Once the delete has been executed, there is no way to restore the definition. Please be very careful with these options.

6.15.3.16 Multi Level Scheduling Controls

Schedule levels control the order in which Scheduled Processes are executed. Each level may contain one or more Processes or nested schedules that are executed concurrently. The next level is only released, once the previous level is executed. You may also make schedules dependent on other schedules through nesting them as schedule items or using a **schedule to schedule** option.



Schedule Levels

All Process and nested Schedule items defined for a level are released concurrently. You may configure the items for the next level to release only when Items for the previous level are completed successfully, that is, if any Processes fail, the next level is not released. You configure this by selecting the **only release when all preceding items have completed** check box. If you select this check box at the schedule definition level, the option is mandatory for all levels for the schedule.

Schedule Items

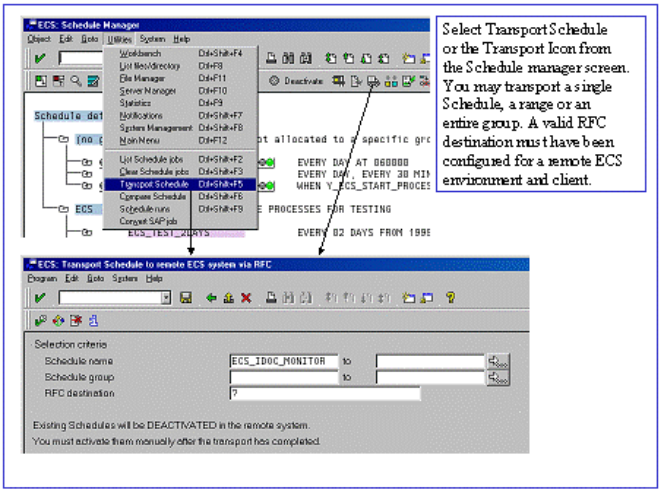
Each item identifies a Process or nested Schedule to execute. You may use the normal Process Phase enqueue and event controls to manage the sequence of execution of Phases within a Process. If schedule items are to execute sequentially, configure them as separate schedule levels.

Schedule to Schedule Dependencies

You may configure schedules either to execute when another Schedule is completed or may nest within another Schedule. You may use the **START AFTER SCHEDULE &SCHEDULE** option to define **Schedule to Schedule** dependencies, or you may nest a schedule within another schedule through the Schedule item.

Transporting Scheduling Definitions

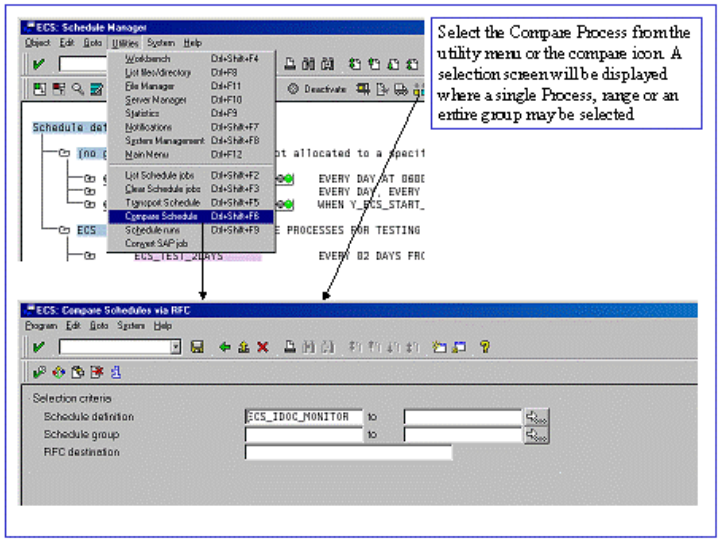
You can transport ECS Schedule definitions to another ECS environment on a remote SAP R/3 system. This is performed realtime using SAP R/3 -> SAP R/3 RFC connection. You need to configure all the RFC connections available using transaction SM59. You may transport a single Schedule, a range, or an entire group. Any existing Schedule definitions are completely replaced.

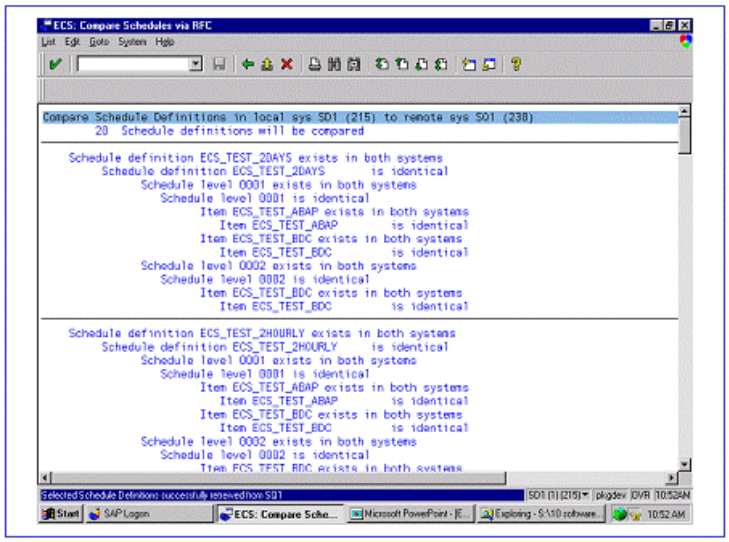


> ***Note:*** The ECS Schedule Transport facility does not use the SAP R/3 correction and transport system. This is because logic, as well as data definitions, is required when you move Schedule definitions between ECS environments.

6.15.3.17 Comparing Scheduling Definitions

You can compare ECS Schedule definitions between ECS environments on remote SAP R/3 systems. This is performed realtime using SAP R/3 > SAP R/3 RFC connection. You need to configure all the RFC connections available using transaction SM59. You can compare a single Schedule, a range or an entire group. A report that details the differences appears:



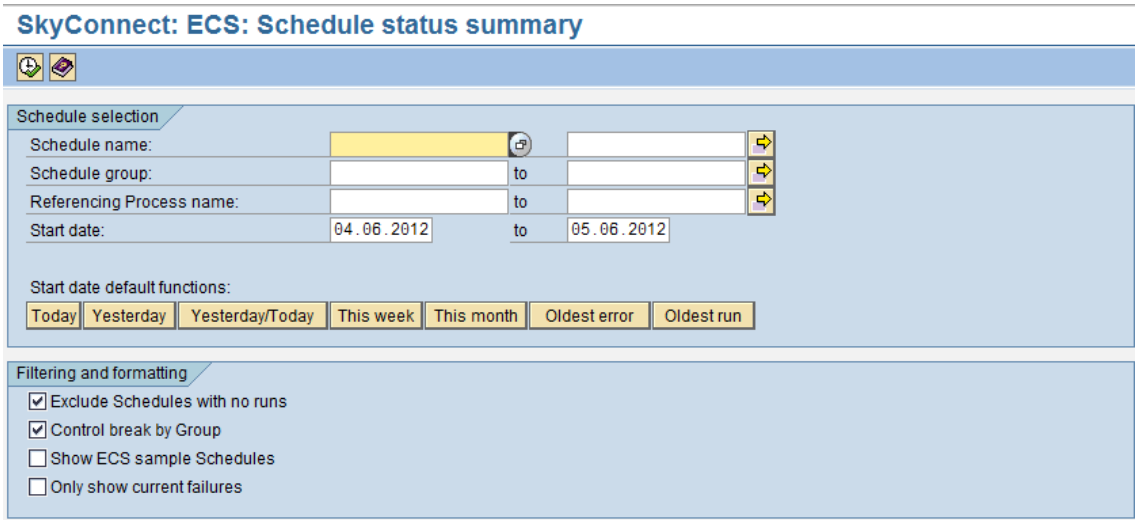


6.15.4 Monitoring Schedules

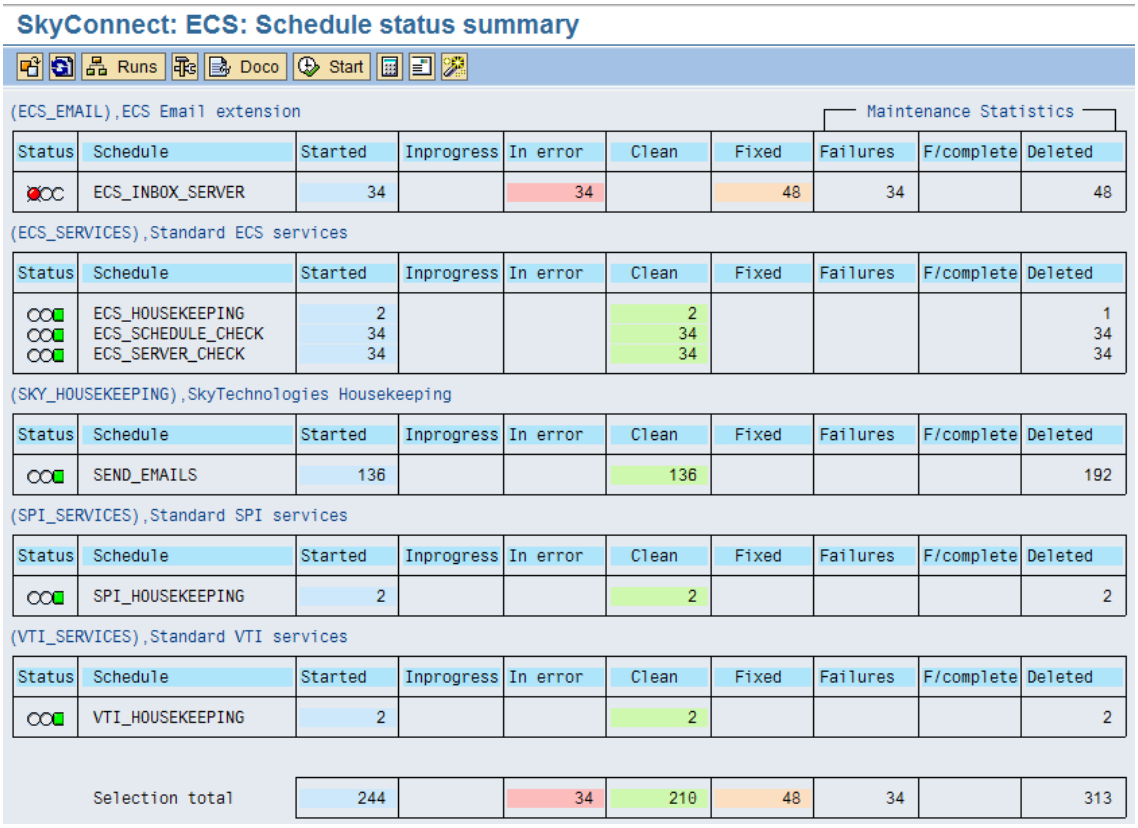
Apart from viewing the current status of schedules from within the workbench, ECS provides a high level schedule status summary dashboard from which to view all scheduling activity and drill down on specific areas of interest. You can display the details of each unique schedule run using the schedule run display utility. It is important to remember that ECS schedule definitions execute ECS processes at the end of the day and you can also monitor the schedule definitions using the standard ECS process monitoring tools.

6.15.4.1 Schedule Status Summary

You may start the schedule status summary from either the ECS menu or through transaction YECA. Just like the process status summary, it has a initial selection screen, you can restrict the data and some display options.



The summarized list provides a high level view of all the schedule executions and their current status.



SkyConnect: ECS: Schedule status summary

Runs | Doco | Start

(ECS_EMAIL),ECS Email extension — Maintenance Statistics

Status	Schedule	Started	Inprogress	In error	Clean	Fixed	Failures	F/complete	Deleted
	ECS_INBOX_SERVER	34		34		48	34		48

(ECS_SERVICES),Standard ECS services

Status	Schedule	Started	Inprogress	In error	Clean	Fixed	Failures	F/complete	Deleted
	ECS_HOUSEKEEPING	2			2				1
	ECS_SCHEDULE_CHECK	34			34				34
	ECS_SERVER_CHECK	34			34				34

(SKY_HOUSEKEEPING),SkyTechnologies Housekeeping

Status	Schedule	Started	Inprogress	In error	Clean	Fixed	Failures	F/complete	Deleted
	SEND_EMAILS	136			136				192

(SPI_SERVICES),Standard SPI services

Status	Schedule	Started	Inprogress	In error	Clean	Fixed	Failures	F/complete	Deleted
	SPI_HOUSEKEEPING	2			2				2

(VTI_SERVICES),Standard VTI services

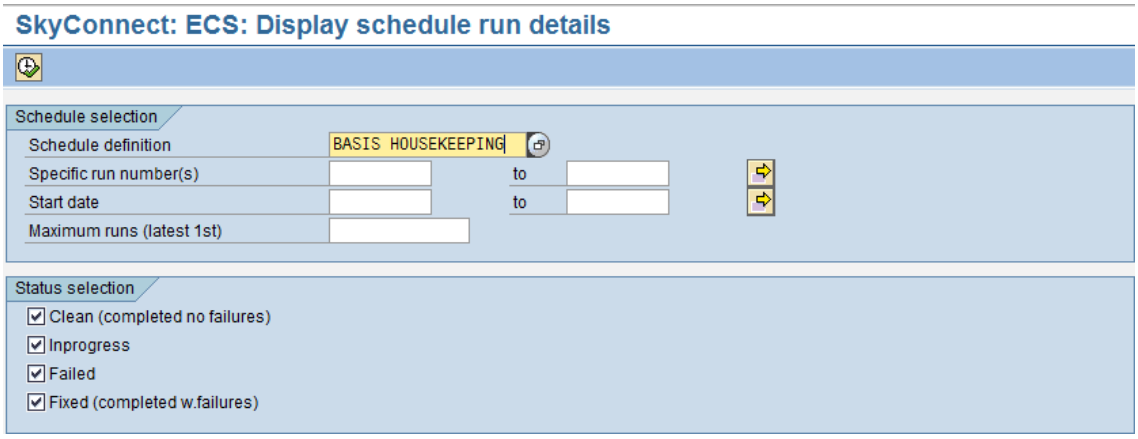
Status	Schedule	Started	Inprogress	In error	Clean	Fixed	Failures	F/complete	Deleted
	VTI_HOUSEKEEPING	2			2				2

Selection total	244		34	210	48	34		313

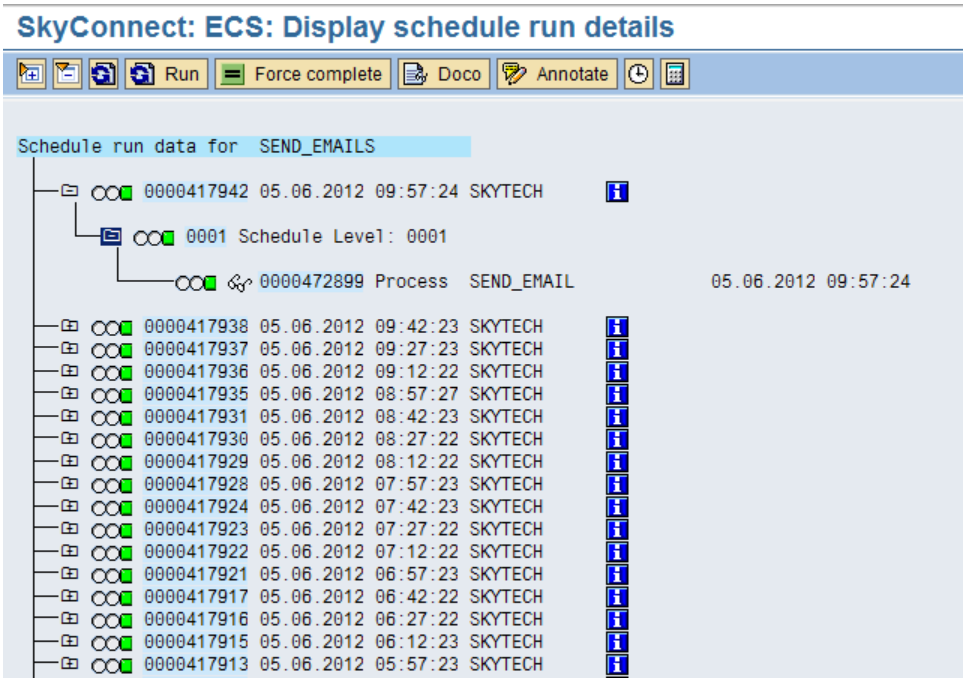
You may double-click a line or a hotspot to drill-down a particular area. The schedule run display appears.

6.15.4.2 Schedule Run Display

The schedule run display is a hierarchical view of all the levels and processes associated with a schedule execution. You may invoke the schedule run from either the workbench, the schedule status summary or directly through transaction YECU. If you invoke the run display directly, an initial selection screen appears.



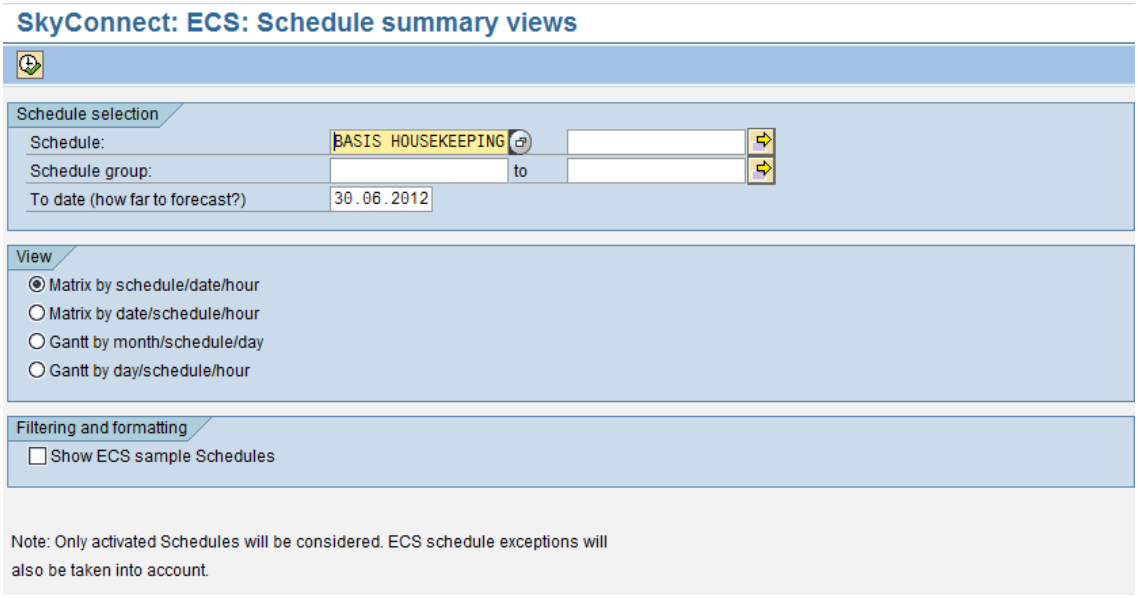
Each schedule run is given a unique number and you can drill-down the schedule levels and associated process runs.



From this screen, you can annotate, cancel, force complete, and delete runs. The log displays detailed internal information about the schedule execution.

6.15.4.3 Schedule Planning Report

This report provides multiple views of scheduling to enable users with consolidated information in-order to ascertain when schedules run and how often. The report is invoked from the schedule summary, workbench or run display through the summary view icon on the application tool-bar . A selection screen appears to filter the data and list format.



Depending on the view, a list is appears summarizing when the selected schedules run.

SkyConnect: ECS: Schedule summary views

05.06.2012 TUE

Schedule/Hour	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ECS_INBOX_SERVER													01	01	01	01	01	01	01	01	01	01	01	01
ECS_SCHEDULE_CHECK												01	01	01	01	01	01	01	01	01	01	01	01	01
ECS_SERVER_CHECK												01	01	01	01	01	01	01	01	01	01	01	01	01
SEND_EMAILS												04	04	04	04	04	04	04	04	04	04	04	04	04

06.06.2012 WED

Schedule/Hour	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ECS_INBOX_SERVER	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
ECS_SCHEDULE_CHECK	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
ECS_SERVER_CHECK	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
SEND_EMAILS	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04

07.06.2012 THU

Schedule/Hour	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ECS_INBOX_SERVER	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
ECS_SCHEDULE_CHECK	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
ECS_SERVER_CHECK	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
SEND_EMAILS	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04
SPI_HOUSEKEEPING					01																			
VTI_HOUSEKEEPING						01																		

6.16 Data Transformation

ECS provides a data transformation tool. It is designed to quickly and easily map data between a variety of formats including files, messages, XML, TSX, IDOCs and BAPIs, overcoming the need to develop custom ABAP code to format and map data for inbound and outbound interfaces.

6.16.1 Key Topics

Overview
Architecture
Dictionary Workbench
Translation Workbench
Examples

6.16.2 Cupid Overview

The Cupid data mapping and translation tool for SAP is supplied as a standard component of SkyConnect. It allows quick and easy mapping of data between:

- Files
- XML
- TSX messages
- IDOCs
- BAPIs
- SAP functions

Cupid is developed on request from Sky customers to provide a low cost, flexible data mapping and formatting tool specifically for SAP. This is primarily to avoid the need for expensive and complex middleware translation tools and secondarily to reduce the amount of programming involved in interface development. ECS interface management already has functionality to automatically detect and process files and provides an effective framework to process business transactions. Cupid addresses the gap, where previously developers had to develop the ABAP code to format and map the data for both inbound and outbound interfaces. Whilst not designed to be a fully comprehensive translation engine, you may use Cupid in the majority of cases and you may address the more complex requirements (if any) by using custom ABAP code.

6.16.2.1 Why Use Cupid?

- Eliminates the need for middleware translator products (runs inside SAP)
    - Reduced cost
    - Less complex infrastructure (eliminates another point of failure and administration)
    - Everything is configured and managed centrally in SAP

- Eliminates or reduces the need for programming, that is, reduced development cost
    - GUI workbench in SAP to configure data sources, conditions and field assignments
    - ABAP code is automatically generated for you
    - Fully integrated with the SAP data dictionary and ABAP workbench
- Provide comprehensive end-to-end transaction management, that is, reduced support cost
    - Monitor all interface processing from a central point
    - Schedule and sequencing processing correctly
    - Restart / recover failed transactions
    - Notify support staff of any failures

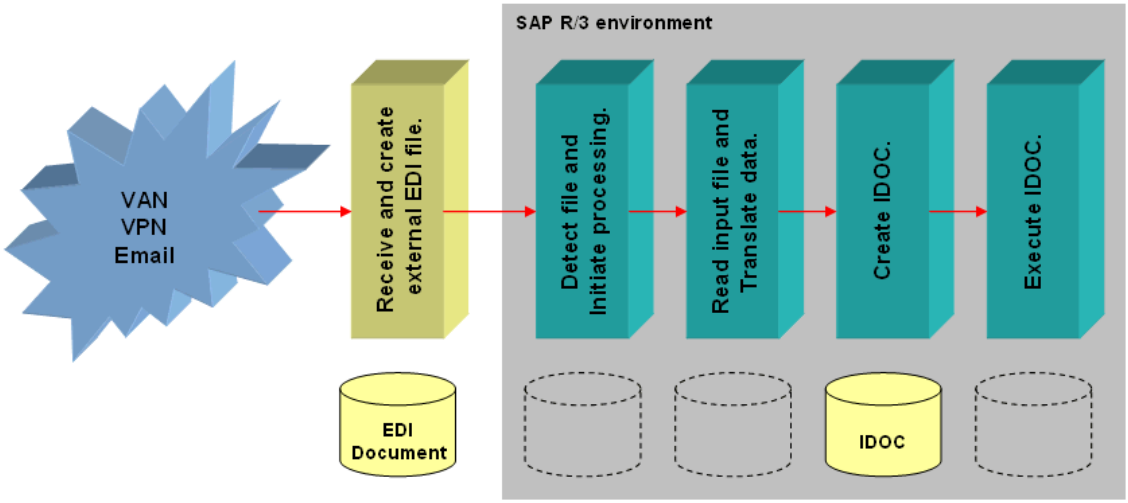
6.16.2.2 What can Cupid be used for?

You can use Cupid in any number of scenarios where interfacing in &/or out of SAP is required. Common situations in which Cupid is used include:

- EDI processing
- Inbound / outbound XML interfacing
- Translating interfaces with machinery (for example, Barcode labellers, PLC devices)
- Warehouse management interfaces

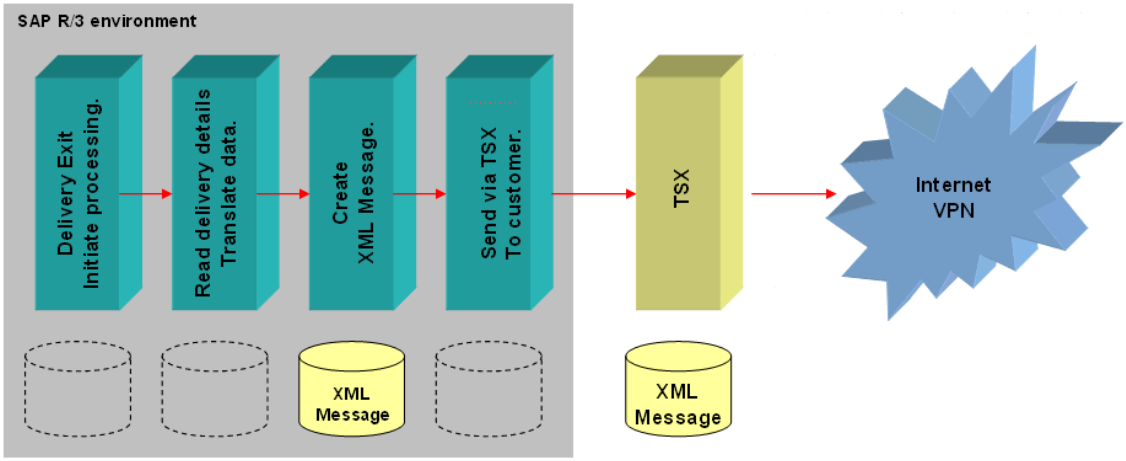
6.16.2.3 EDI Example

In this scenario, a EDI file containing data (EDIFACT or ANSI X12 standard) is received and created on the SAP host. This can be either from a VAN (value added network), an internet VPN connection (Virtual private network), an Email attachment, or through FTP (file transfer protocol). ECS file polling picks up the file and initiates an ECS process, the first stage of which is the Cupid runtime object that reads the input EDI file and creates an IDOC from the contents. The last stage submits the IDOC for execution and monitors its progress.



6.16.2.4 XML Example

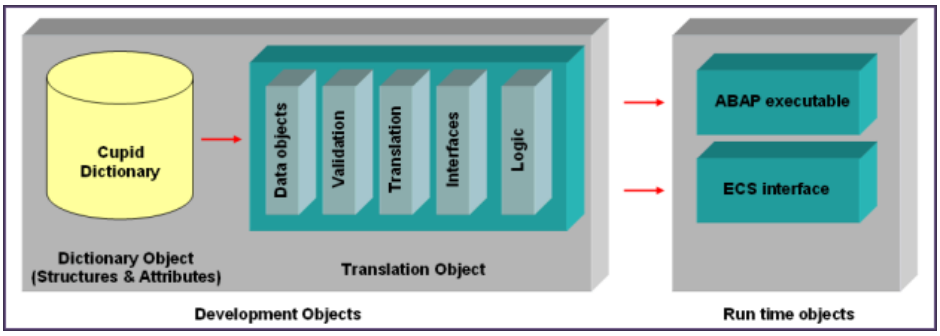
In this scenario, a delivery document exit is triggered in SAP. This initiates an ECS process, the first stage of which is the Cupid runtime object that converts the delivery details into an XML message format and passes the message to TSX to send to the customer. The process execution and monitoring is performed by ECS.



6.16.3 Cupid Architecture

6.16.3.1 Cupid Components

Broadly speaking, Cupid has two types of objects: development and run time. You can break down development objects further down into dictionary objects and translation objects. The diagram below provides an overview of the various components.



The translation object contains all the various definitions required to perform a translation. This includes:

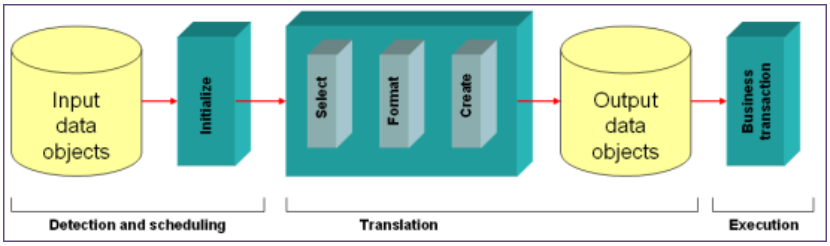
- input data sources
- output data targets
- translation rules
- validation rules
- any SAP interfaces required (for example, number range calls, other function modules, tsx)
- processing events and actions (logic)

The Cupid dictionary objects consist of the structure and attribute definitions of the various input and output data objects. Cupid translation objects are generated in to ABAP programs (runtime object) that you may run stand alone. However, the execution is designed to work with ECS interface management.

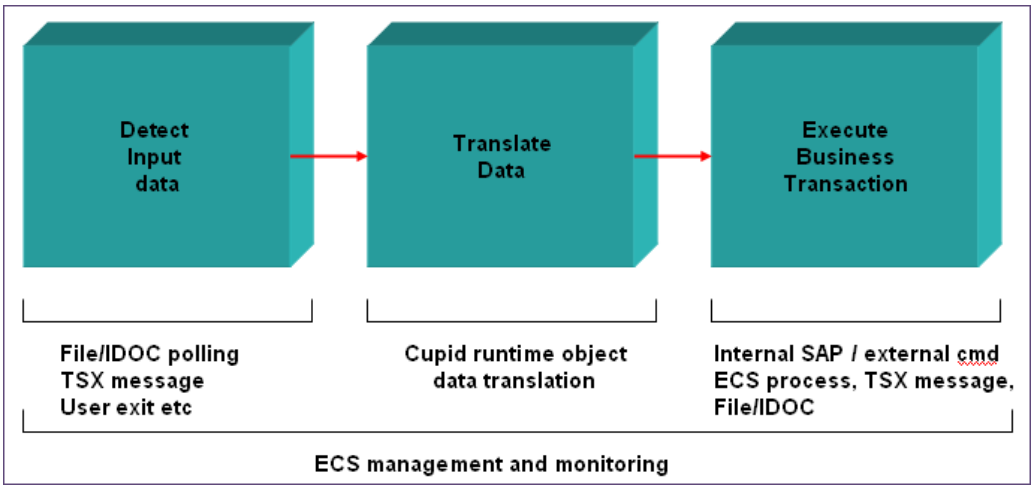
6.16.3.2 Execution Process

The Cupid execution process is relatively simple in functionality:

1. A primary input data source is detected and is scheduled for processing
2. The translation process is initiated
    - All or a selection of the data is read
    - Validation of the input data elements is performed
    - Any defined secondary input data object(s) is read
    - Conditional and formatting rules are applied
    - The new output data target is created
    - A nominated business transaction is initiated (optional)



In the above architecture diagram, ECS provides the overall framework to execute the entire process from initiation through to final completion. The Cupid data translator re-formats the data, ready for execution. You may summarize this as follows:



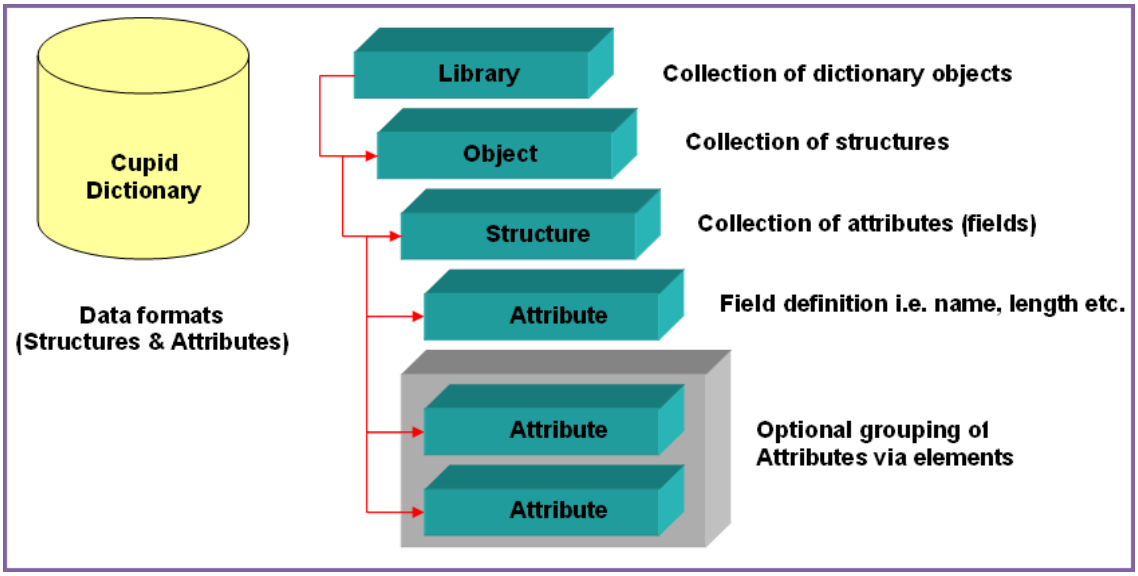
6.16.3.3 Cupid Concepts and Terminology

Cupid separates the data dictionary objects and related structures from the translation object definition. This is to allow the same dictionary object to use in multiple translation objects. There is a one to one relationship between a translation data object and a dictionary object, however you may define as many translator data objects as you like, even if you do not use the translator objects explicitly in any input / output operation.

> ***Important:*** Before you can define a translator data object, you must have defined the corresponding data dictionary object to use.

Dictionary Object

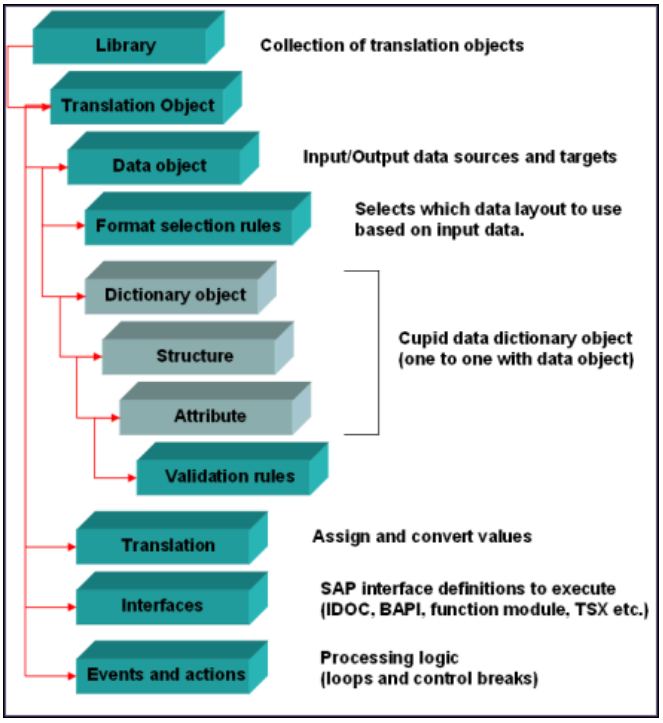
The Cupid data dictionary contains all the definitions that are used to define input and output data elements. You need to define definitions before you can create a translation object. It is structured this way, so that the you can share the same data definitions amongst many translation objects.



Translation Object

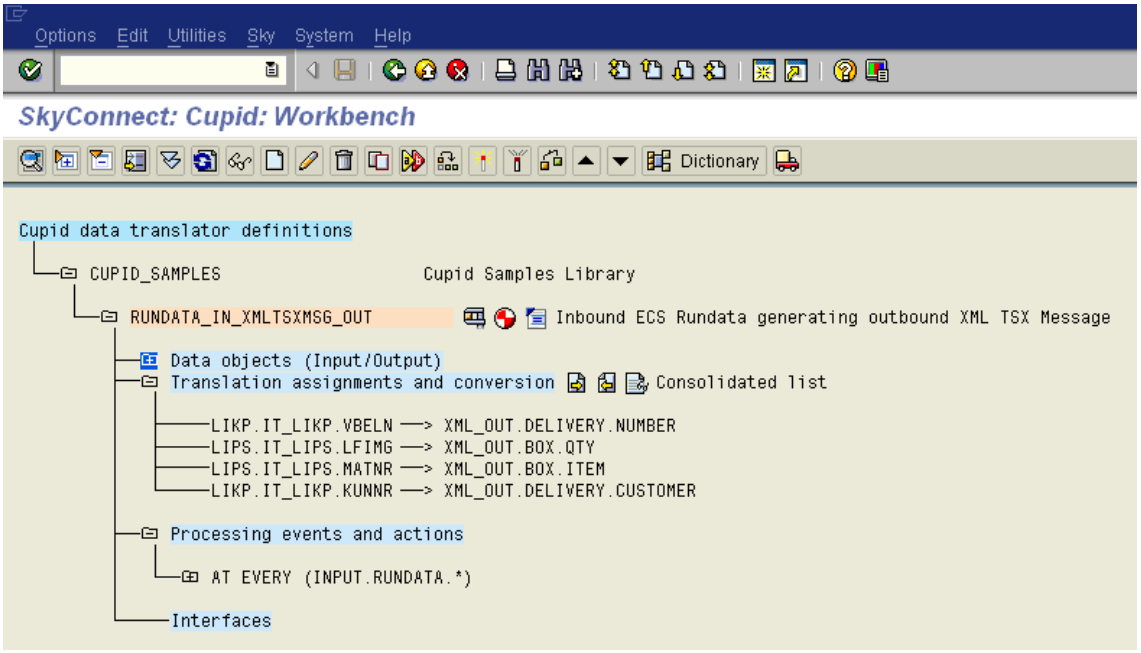
Before a translation object can be defined, you must first define all the data structures and attributes in the Cupid data dictionary. The translation object refers directly to a Cupid dictionary object for every input or output data object, thus all the structures that are going to be used must be declared and logically grouped into the same data dictionary object.

The translation object contains the entire configuration for the input/output operations, data validation and translation, SAP interfaces to invoke and the logic and actions to perform at various stages of processing; loops and control breaks.



Translation Definitions

A translation definition is a rule to map data from a source data object attribute to a target attribute. As part of the mapping, you may format the target attribute after the mapping occurred. Translation definitions exist as part of a translation object and get created in the Cupid translation workbench.



Interfaces

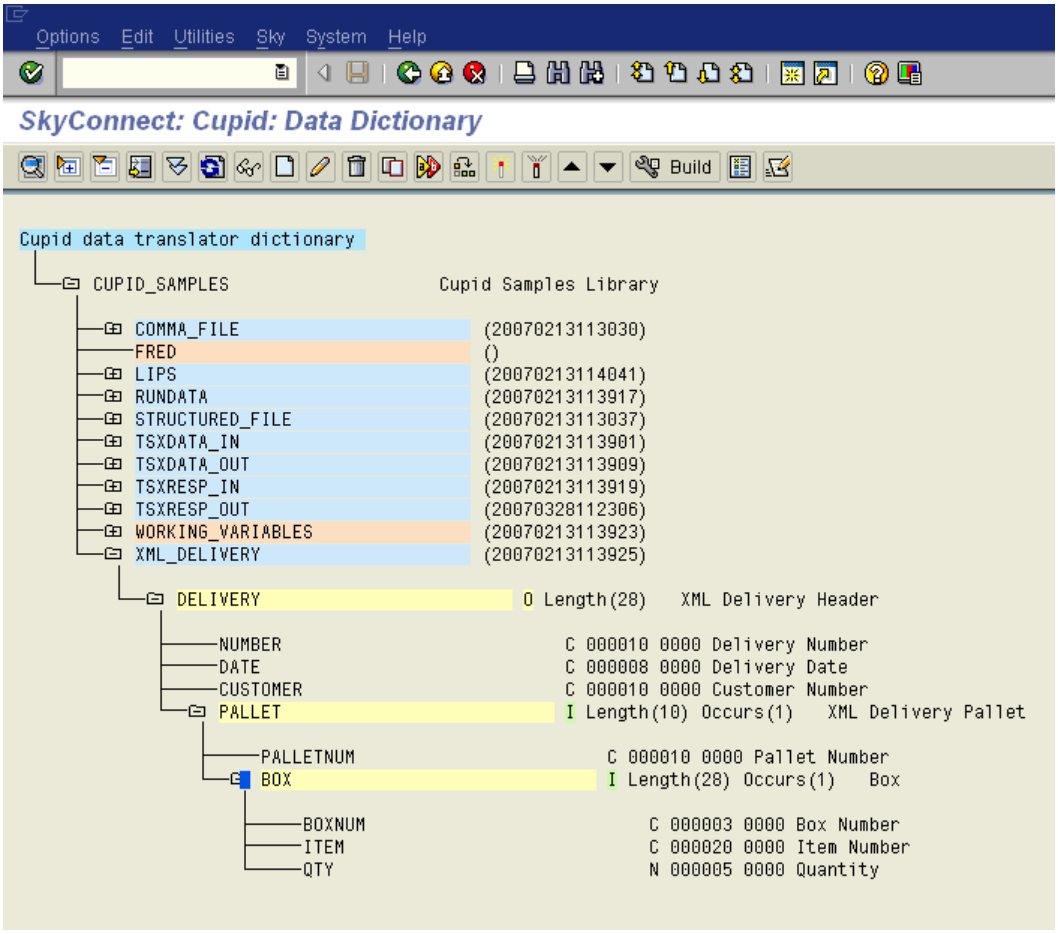
A translation object can make a call to a range of other objects. You can use the calls to an interface both to initiate a separate branch of processing and to retrieve information to include in the current processing of the Cupid object.
The types of interfaces supported by Cupid include;

- Function modules
- BAPIs
- ECS processes
- TSX message / file transfers
- Number range calls

Interface definitions exist as part of a translation object and get created in the Cupid translation workbench.

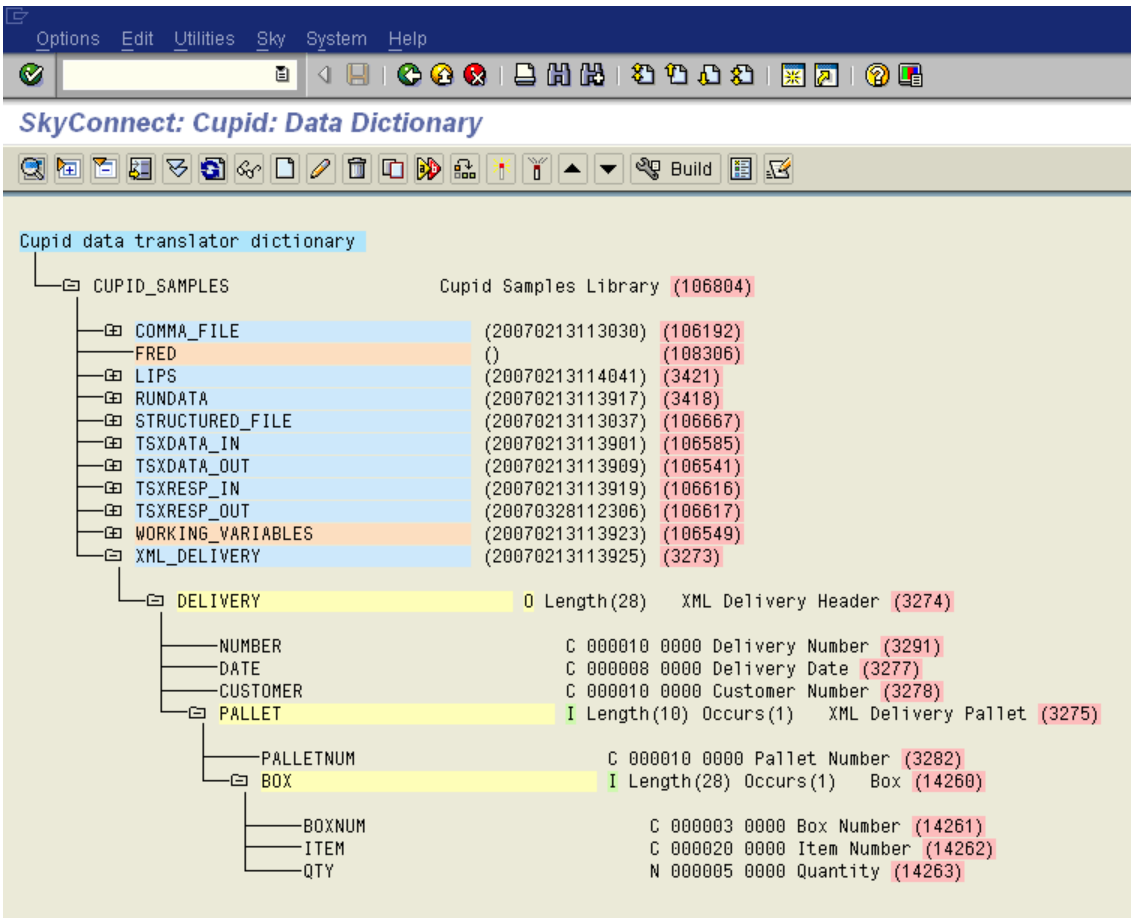
Object Numbers

Each Cupid component (library, objects, structures, and attributes) has an internal number associated with them. By default, these numbers do not appear in the workbenches.



You can display the internal numbers by choosing the menu option **Utilities** > **Show internal number** from either Cupid workbench.

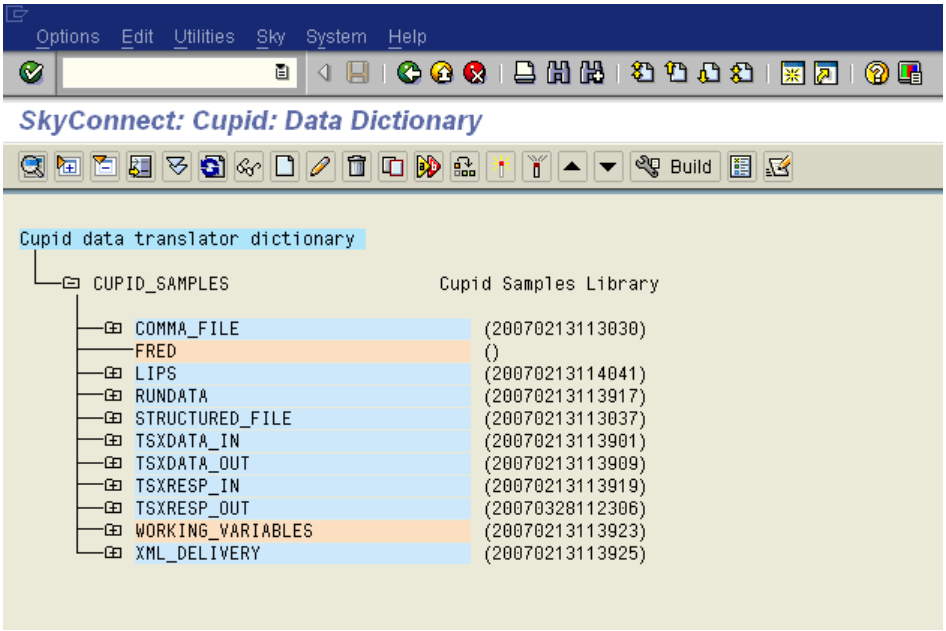
Once you enable the option, the internal numbers appear against each component.



Activation and Deactivation

All Cupid objects (translation and dictionary) have a status, either activated or deactivated. Once you activate an object, you can make no further changes. To make changes to an active object, you must deactivate it.

The status of an object appears shaded in the workbench. Inactive objects are shaded orange, active objects are shaded blue.



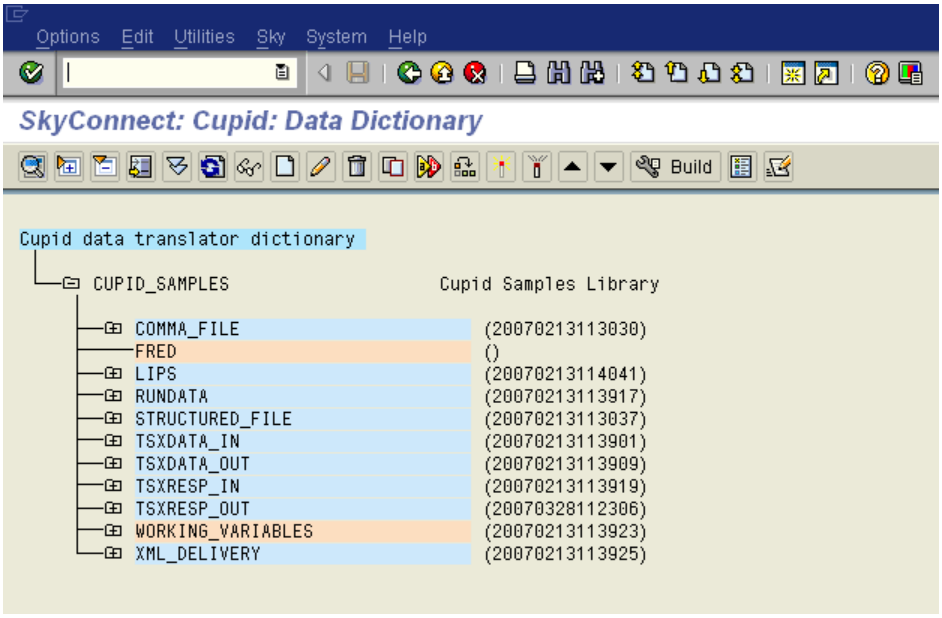
To activate or deactivate an object, select the object and click the **Activate/Deactivate** icon  .

> ***Important:*** When you use an inactive dictionary object as part of a translation object you receive a warning message.

Object Timestamps

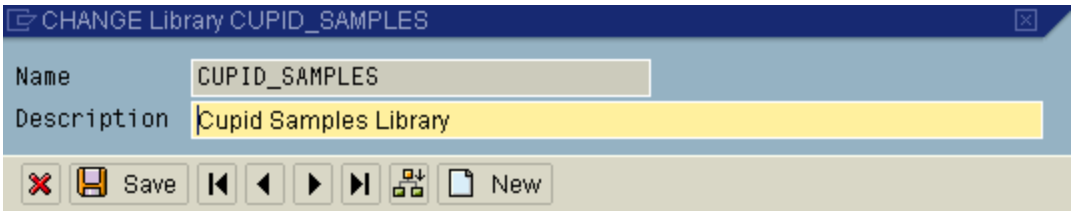
All Cupid objects (dictionary or translation) have a timestamp. Each time, an object is activated its timestamp is updated, allowing a developer to identify when an object was last modified.

Newly created objects, such as the dictionary object FRED in the screen grab below, has a blank timestamp until activated for the first time.



Libraries

A library provides a simple mechanism to logically group objects together and enables duplicate objects to co-exist. A typical use of libraries is to manage different versions of objects through an iterative development cycle. A new library is created by positioning your cursor on the top level node and clicking the create icon. You can change the library by positioning the cursor on the appropriate library name and clicking the change icon.



Name	A unique library name.
Description	Free format library description

The **Create new** icon creates another library entry and the 'next level' icon is useful to drill-down to define or maintain related objects without having to return back to the hierarchy. You may use the next and previous "arrows" to navigate to another library without returning to the hierarchy.

> ***Note:*** You may use the same library definition to group both data dictionary and translation objects.

Events and Actions

You define the processing logic within translation objects as events and actions.
Each translation object can have one or more events. Each event can have one or more actions.

For example, you may want to perform two specific steps each time a record is read from an input file:

- If field3 of the record contains the string FRED map it to the output file structure
- Create an output record.

You define this logic in Cupid as:

- Event: ON EVERY input data object
- Actions.
    a. If field3 contains FRED perform translation
    b. If field3 contains FRED create output record.

Interface definitions exist as part of a translation object and get created in the Cupid translation workbench.

Standard SAP Icons

As with standard SAP applications, you use the following basic icons in the Cupid workbenches.

Icon	Description
	Execute screen
	Back (previous screen)
	Save
	Cancel
	Page up, page down
	Refresh
	Choose a specific definition for selection
	Permanently expand one or all levels of a hierarchy
	Permanently collapse a section or level of a hierarchy
	Display, create, change, delete, copy, rename, reassign.
	Check / validate
	Move up, down, first, next, previous, last
	Activate / deactivate definition
	Generate definition

Icon	Description
	Test function
	View generated ABAP source code
	Next level or ECS run

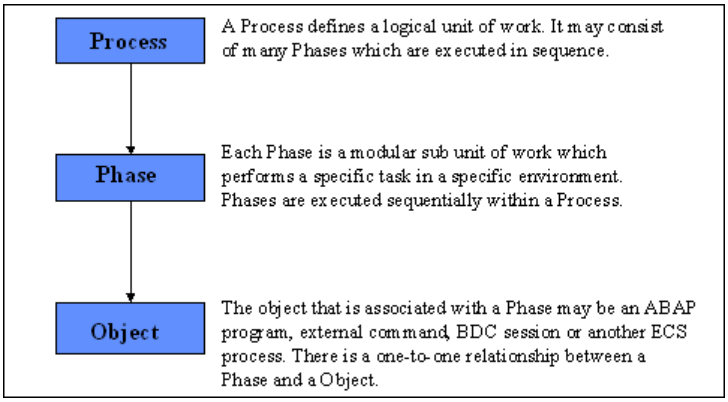
Specific Cupid Icons

Icon	Description
	Build definition from reference
	Attribute editor
	Cupid data dictionary
	Global attributes
	Hints

6.16.3.4 Concepts and Terminology

Processes, Phases and Objects

A process is a logical grouping of executions (Phases) that together make up a single logical unit of work (run). The executions can be disparate (that is, internal to SAP R/3, external to SAP R/3), utilize different techniques (that is, custom ABAP program, and SAP R/3 standard programs) that collectively constitute a business process. ECS provides all the controls necessary to effectively execute, monitor and restart interfaces and periodic processing. You use the ECS Workbench is to define Process and Phase definitions.



Execution of a Process

You may initiate an ECS Process in many ways, explicitly from within a program, manually using the **Start Process** option in ECS, through SAP R/3 event, and a file arriving in a directory. Execution always starts with the 1st Phase in the sequence and then subsequent phases. You may assign run time attributes to the Process definition that govern how it is executed and how you monitor and control the Phases.

Execution of Phases within a Process

Phases are executed in a sequential fashion. You may explicitly start phases 'programmatically' or automatically by ECS when the previous Phase is completed. In the ideal case, Phases are started

and ended explicitly by calls to ECS within the executable program or script. This gives the programmer full control over start and end phases with specific parameter, enqueue and interval options. You may also configure phases to kick off automatically once the previous phase successfully completes. For example, once phase-1 generates the BDC session, start phase-2 to process it.

Groups

Group definitions are optional and you use to categorize ECS Processes and Schedules for easy identification. Groups are useful to restrict reporting criteria. You may also use groups to secure areas, for example, Basis, Accounts Payable, and Accounts Receivable. You may only assign a Process or Schedule to one group.

Filter fields and values

It is possible to further categorize Process "Phase" runs at a low level (for example, plant). You may assign filter fields and populate at run time. These are useful to segregate Process based on the transaction data, for example, display all the warehouse management transactions for plant 1234. They are also useful to direct notifications to specific users or groups of users, for example, if material movement fails for plant 1234, notify John Smith.

Queuing Controls

ECS provides many controls to effectively manage concurrent processing and avoid failures due to the exclusive control of SAP resources. You may effectively serialize processing with logical locking and configure automatic checks to prevent accidental executions.

Serialization

The ECS enqueue manager provides locking mechanisms that 'single thread', the execution of ECS Processes and Phases. This is necessary to prevent the failure of conflicting programs over the same resource. For example:

- Only one task may update a materials stock within the same plant at a time, therefore we need to check whether the plant / material is updated. If it is, you suspend the execution of the Phase until the update is completed. In-order to detect that the plant and material is currently enqueued (locked), ECS publishes the enqueue data provided by the program and checks to see if it is

currently 'owned' by another task. Thus an enqueue is effective across all active Phases. In contrast to enqueue, serialization only affects a specific Process definition. It may be necessary to 'single thread' an entire Process. that is, all other Process runs are suspended while there is one active. ECS release subsequent runs, once active runs complete.

- In order to avoid SAP 'resource locking' failures, an interface to upload bill of material and routing data is serialized in ECS so that only one Process may execute at a time.

Single Instance and Frequency Controls

It is often required to ignore the execution of a process if one already runs, or abort the execution if a run already occurred. For example:

- We do not want to start another invoicing run if one already executes, because the data is picked up and processed by the current run. You want to ignore the submission of another task.
- Do not allow the monthly material roll to proceed if one already run today.

ECS enables these controls to configure for Process definitions. You may need the controls to prevent duplication and accidental submission of interfaces and / or periodic processing.

Click here to go back to the top.

6.16.4 Cupid Dictionary Workbench

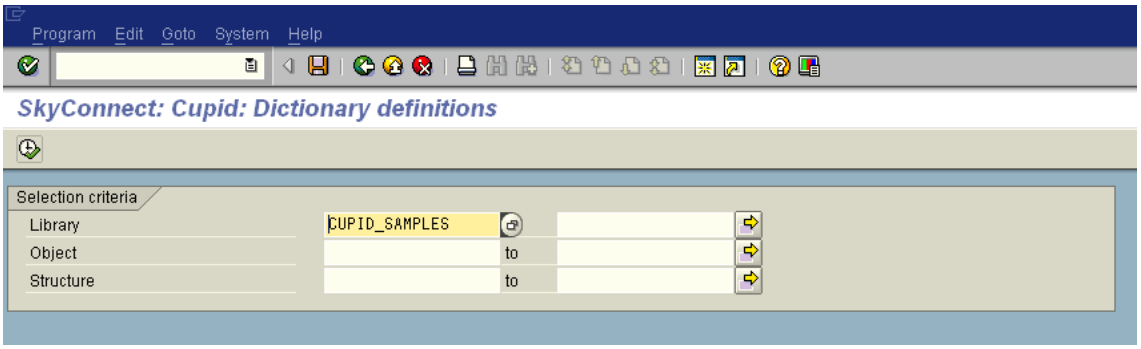
The Cupid data dictionary workbench provides a central repository for managing data structures and attributes. You may reference these definitions by one or more translation objects.

> ***Note:*** Although, you may build Cupid data dictionary definitions based upon SAP dictionary objects, they are completely separate from the SAP data dictionary definitions.

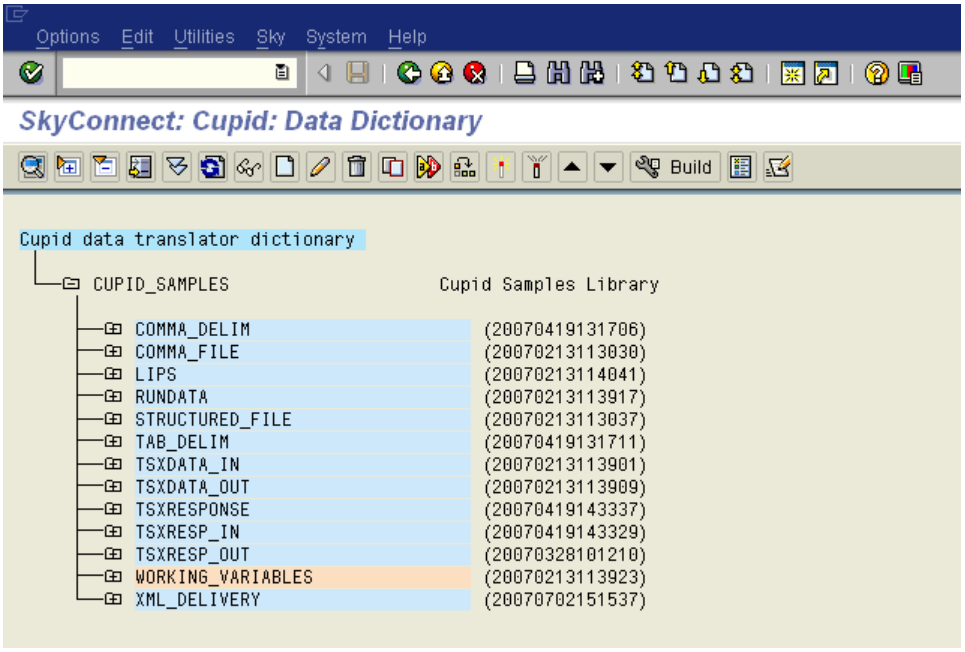
You can access Cupid data dictionary workbench in a couple of ways:

- Execute transaction YCPDD
- From the Cupid translation workbench, click the dictionary icon Dictionary on the application toolbar
- From the Cupid translation workbench, click the dictionary icon against a data object.

The following selection screen appears:



This selection screen is used to filter the definitions selected for display. Specify selection criteria, or leave blank, and click the Execute icon. The main dictionary maintenance screen appears.



A hierarchy of the definitions selected appears. From this screen, you define dictionary objects, structures and attributes. You create and maintain definitions by clicking the appropriate level and then clicking the relevant icon on the application toolbar.

Once you create a library, you may now create dictionary objects, structures and attributes.

6.16.4.1 Creating Cupid Dictionary Definitions

You can create Cupid dictionary objects in two ways:

- Manually – user defined structures and attributes.
- Automatically based on an SAP reference (table, FM, and IDOC) or XML reference using an XSD file.

6.16.4.2 Automatically Building Dictionary Objects

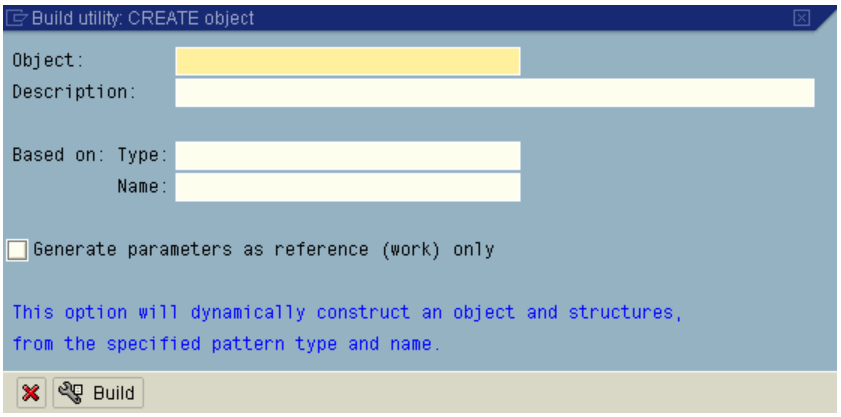
To save keystrokes and time, you can automatically create a dictionary object based on an existing SAP dictionary object.

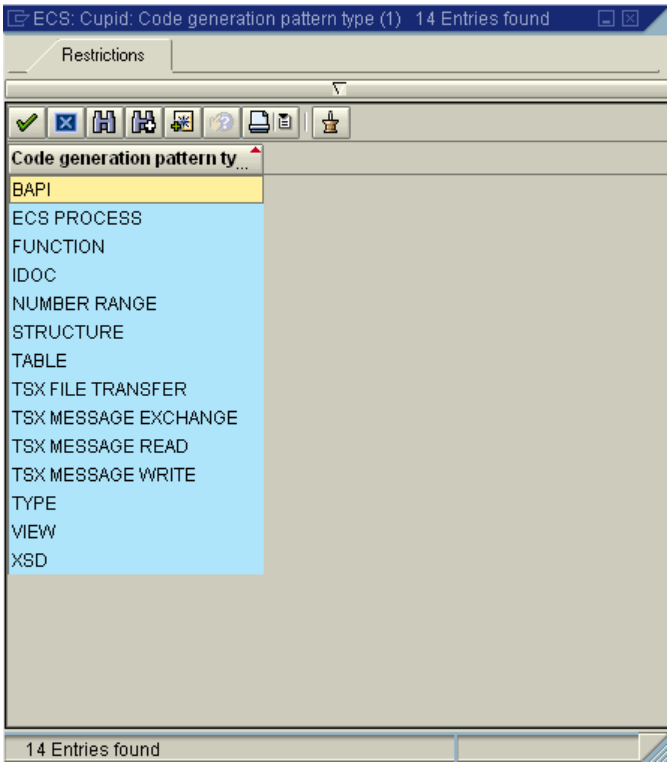
To create a dictionary object based on an existing SAP dictionary object, follow these steps:

1. Click the **Library** node > **Build** icon 

on the application toolbar.

   The following pop-up appears:





2. Enter the following:

Field	Description
Object	The name for the dictionary object. This must be unique within the library.
Description	A free format description of the dictionary object.
Based on Type	The type of object from which to construct the dictionary object, for example, IDOC, BAPI, Function, Table, View, Type, Structure or XSD.
Based on Name	Name of the object that you use as the basis of the definition.

Field	Description
Generate parameters as reference only	Any data definitions or parameter definitions are generated as 'reference' that means they are only used in conversion and / or execution only, and not in data input or output operations.

3. Once you complete all the entries, click the **Build** icon and the definition is constructed. If a definition already exists, you are prompted to replace it.

> ***Note:*** When you generate a Cupid object using type XSD, the name field is not used. Instead, you are prompted for the path to the XSD file on your computer.

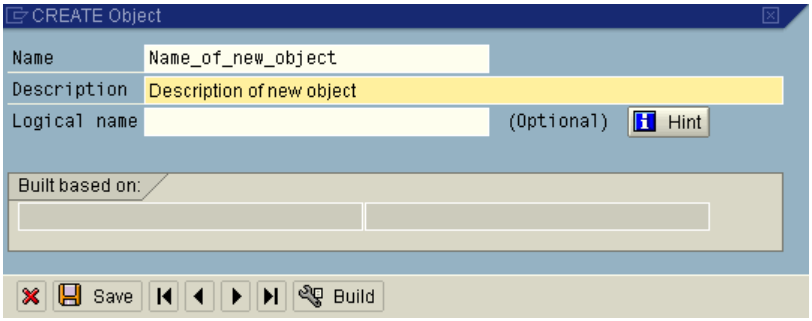
6.16.4.3 Manually Building Dictionary Objects

When you manually build Cupid dictionary objects, you need to define the following dictionary components:

- Objects
- Structures
- Attributes
- Elements (optional)

Building a Dictionary Object

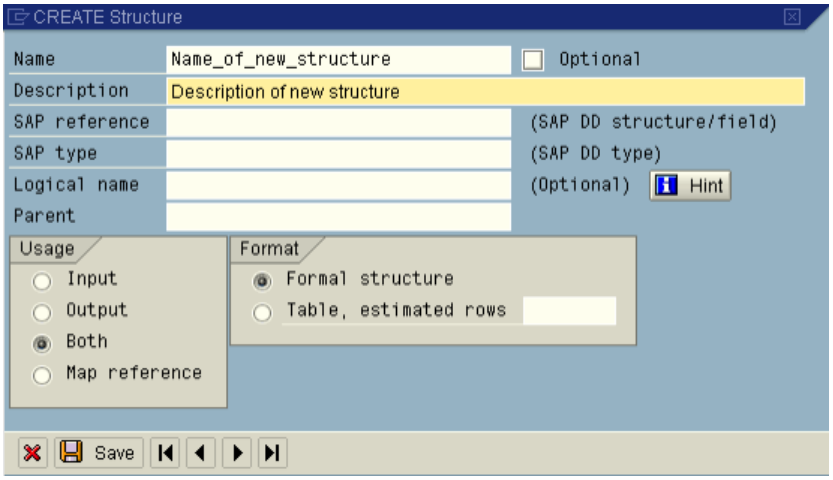
To create a new dictionary object within a library, click the library level and then the create icon on the application toolbar. To change an existing object, click the object name and then click the **Change** icon on the application toolbar.



Field	Description
Name	The name for the dictionary object. This must be unique within the library.
Description	A free format description of the dictionary object.
Logical Name	Optional: Used for XML objects where the same tag name may occur multiple times at different points in a document having different underlying structures.

Dictionary Structure

To create a new dictionary structure within an object, click the object level and then the **Create** icon on the application toolbar. To change an existing structure, click the structure name and then the **Change** icon on the application toolbar. The following dialog appears:



Field	Description
Name	The name for the dictionary structure. This must be unique within the library.
Description	A free format description of the dictionary object.
SAP reference/type	Not applicable for manual build.
Logical Name	Optional: Used for XML objects where the same tag name may occur multiple times at different points in a document having different underlying attributes.
Parent	Optional: Used for XML objects to allow structures to nest within other structures for complex XML hierarchies
Usage	The default usage for the structure, that is, whether it is input, output, both or for reference only.

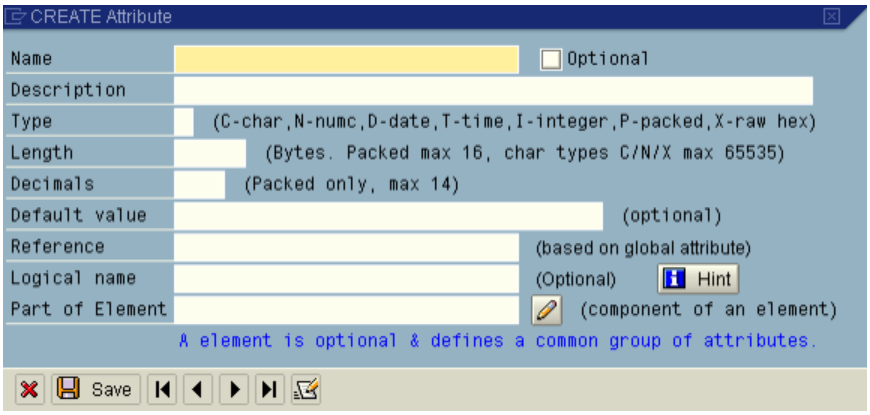
Field	Description
Format	The default format of the structure, that is, whether it is a fixed structure or an array.
Format	The default format of the structure, that is, whether it is a fixed structure or an array.

Click the **Hint** icon 

. The additional information appears that can help with this screen.

Dictionary Attribute

Attributes describe fields within a structure. You may maintain the order of attributes within a structure using the up / down arrows on the application toolbar on the main dictionary screen.



Field	Description
Name	The name of the attribute. This must be unique within the structure.
Description	A free format description of the dictionary attribute.
Optional	A default indicator that the attribute is optional.
Type	The data type, that is, C-Character, N-Numeric character, D-Date, T-Time, I-Integer, P-Packed, X-Hex.
Length	The length of the data type in bytes.
Decimals	The number of decimal places (Packed data types only).
Default value	An optional default value to assign to the attribute.
Reference	A global attribute reference (see the section on global attributes)

Field	Description
Logical Name	Optional: Used for XML objects where the same tag name may occur multiple times at different points in a document having different underlying attributes
Part of element	Nominates that the attribute is part of an element group. You may use the change icon to the right of this field to maintain the element groups.

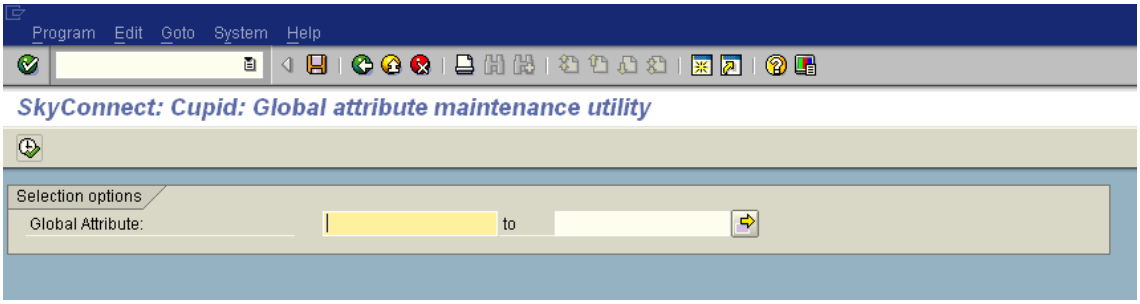
Click the **Hint** icon 

to present the additional information that can help with this screen.

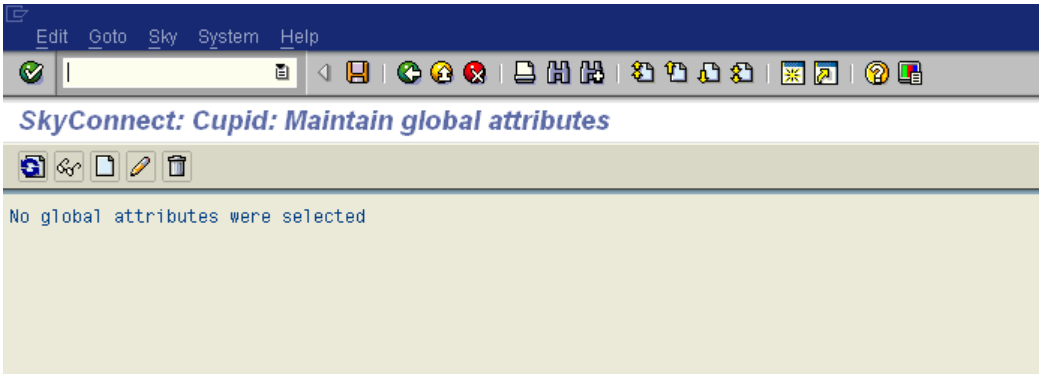
Global Attributes

Global attributes are generic definitions that you may incorporate into one or more attribute definitions by reference, that is, the attribute inherits type, length from the global attribute. In this way, you encourage the standard definitions and may maintain centrally, for example, if you increase the length of the global attribute, all related attributes automatically change.

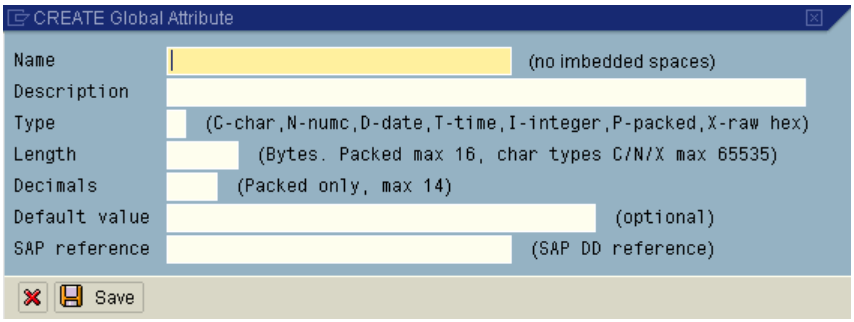
You invoke the global attribute maintenance facility using the  icon on the application toolbar. A selection screen then appears to limit the display and then the main global attribute maintenance screen appears.



Click the **Execute** icon to display the list screen.



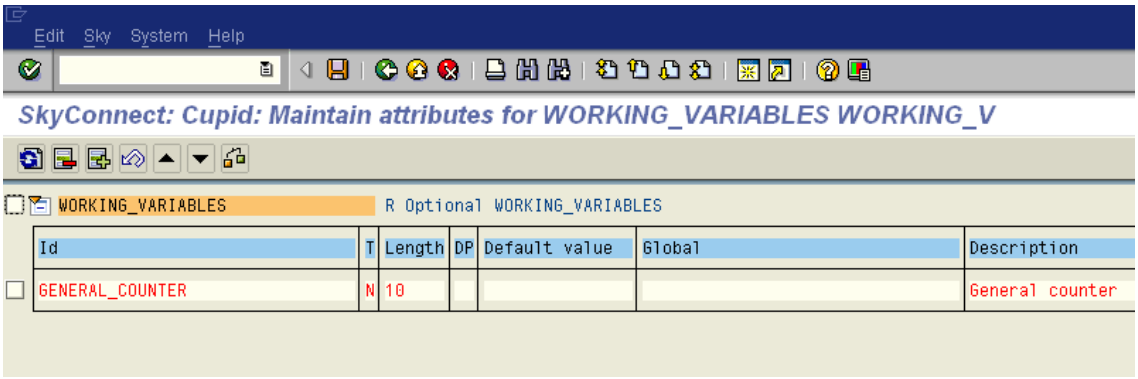
The **Create**, **Change** and **Delete** icons on the application toolbar provide access to maintain the global attribute definitions.



Full Screen Attribute Editor

Maintaining a large number of attributes from the main dictionary hierarchy can be tedious. To make maintenance easier, a full screen attribute editor is provided. You invoke this editor from the main dictionary screen using the  icon on the application toolbar.

If you do this from the main dictionary screen, you must position your cursor on an object or structure level. The following screen appears:



A list of structures and their attributes appear. You can use the expand and collapse icons to more easily manage large numbers of attributes. To change, simply over type the value. To insert and delete rows, select the check box and then the appropriate icon on the application toolbar.

When you complete the maintenance, click the **Save** icon to post the new definitions.

6.16.5 Managing Cupid Dictionary Definitions

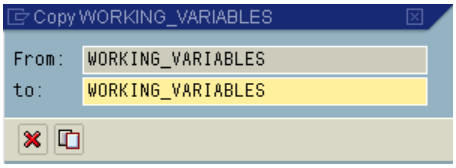
The following sections explain how to use the Cupid Dictionary Workbench to:

- Copy objects, structures attributes
- Modify objects, structures attributes
- Rename objects, structures attributes
- Reassign objects, structures attributes.

6.16.5.1 How to Copy a Dictionary Definition

Once you create a dictionary definition (library, object, structure or attribute), you can copy it to create another definition of the same type.

To copy a dictionary definition, select the definition, then click the **Copy** icon in the menu bar. A dialog appears prompting for the new definition name.



> ***Important:*** Copying of active definitions is not allowed. You need to first deactivate the definition.

6.16.5.2 How to Modify a Dictionary Definition

You can change the existing and inactive dictionary definitions. To change a definition you can either:

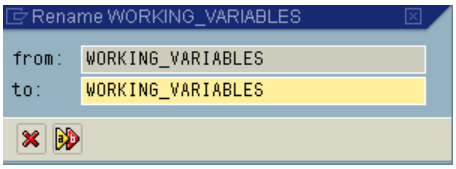
- Select the definition then click the **Change** icon  in the menu bar
- Double-click the definition.

Depending on the type of definition changed, different fields are available.

> ***Note:*** To change the name of a definition, you need to use the Rename function.

6.16.5.3 How to Rename a Dictionary Definition

You can rename dictionary definitions by selecting the definition, then clicking the **Rename** icon on the menu bar. You are prompted for the new name.



> ***Important:*** Renaming of active definitions is not allowed. You need to first deactivate the definition.

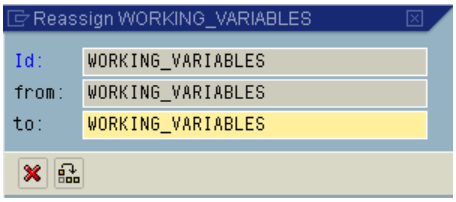
6.16.5.4 How to Reassign a Dictionary Definition

You can reassign dictionary definitions from one parent to another. This allows the following types of reassignment:

- Moving an object from one library to another
- Moving a structure from one object to another within the same library
- Moving an attribute from one structure to another within the same object

To reassign a definition, select the definition then click the **Reassign** icon On the menu bar.

You are prompted for the new parent definition to assign the item to.



> ***Important:*** Reassigning of active definitions is not allowed. You need to first deactivate the definition.

6.16.6 Cupid Translation Workbench

The Cupid translation workbench executes in SAP. You use this workbench to configure all translation object definitions. The Cupid dictionary object workbench that provides the facility to define and edit Cupid data dictionary structures and attributes, is easily accessible from here.

6.16.6.1 Naming Conventions

Cupid automatically checks (and corrects) the names that you enter, Only alphanumeric (A-Z, 0 - 9) and (_) are allowed. All other special characters are automatically substituted with '_'. The name may not contain embedded spaces, these are automatically compressed. Only upper case names are supported.

6.16.6.2 Reserved Words

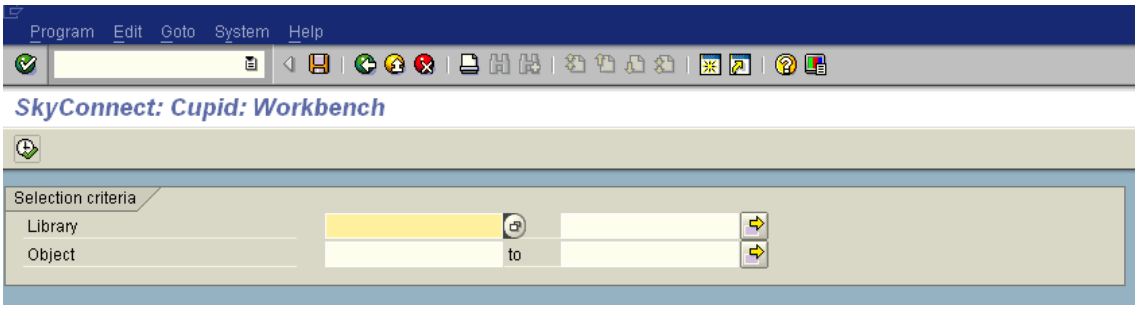
Cupid supports a reserved word list that is maintained through table /SKY/YECS_TRESERVED. The purpose of the reserved word list is to automatically translate any reserved word with an alternate value. This applies mainly to Attributes and Parameters, for example, the 'return' parameter is required by BAPIs to monitor errors and messages.

6.16.6.3 Object Activation and Time-Stamps

Each Object has an explicit status and is date/time stamped whenever it is activated. You must deactivate an object before you may change and then reactivate again in-order to execute the run time component. This mechanism is necessary in order to provide integrity between the generated code and the workbench configuration.

6.16.6.4 Starting the Cupid Workbench

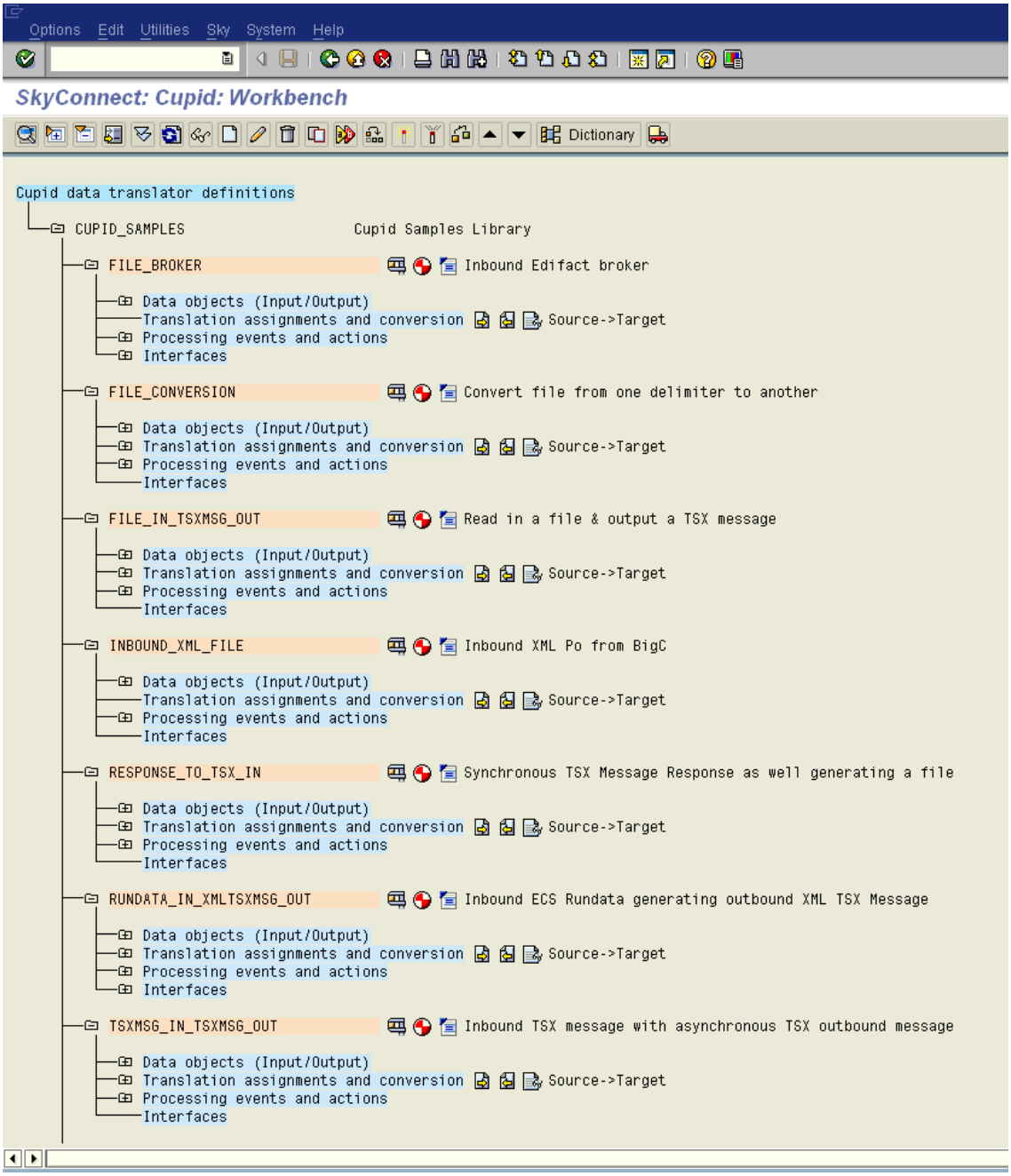
Log onto SAP and execute the YCPD transaction. The following selection screen appears:



You use this selection screen to filter the information that appears on the main workbench screen. Enter any specific selection criteria, or leave it blank, and click the **Execute** icon . The main cupid workbench screen appears.

6.16.6.5 Main Cupid Translation Workbench Screen

You use the main workbench screen to create translation objects. The definitions appear in a hierarchy. You create and maintain definitions by placing your cursor on the appropriate level and clicking the relevant option on the application toolbar.



The main workbench screen is used to:

- Create and maintain translation object definitions
- Generate the ABAP and ECS run time objects
- Go to the Cupid data dictionary
- Check the Cupid installation (auto configuration)
- Export and import definitions between remote SAP systems

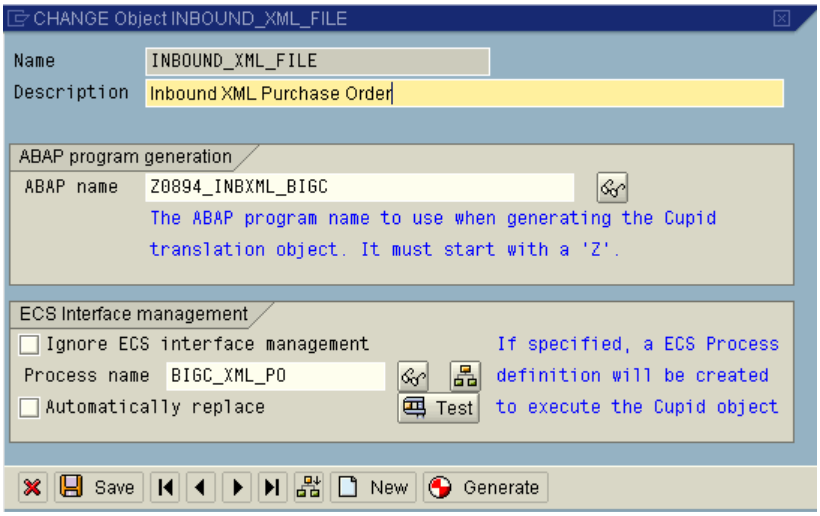
6.16.6.6 Creating a Translation Object

A number of steps are involved when creating a new translation object:

- Create the new translation object definition
- Adding the required data objects
- Adding translation definitions
- Adding required interfaces
- Adding processing logic.

Creating a Translation Object Definition

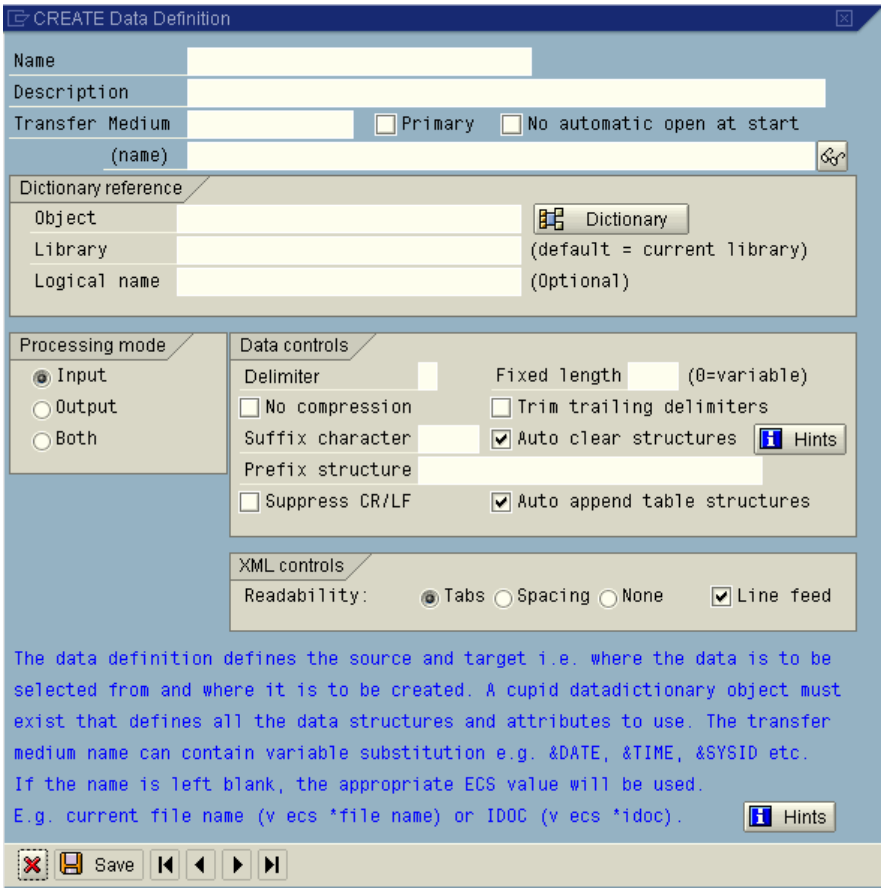
You create a new object by positioning the cursor on a library node and clicking the **Create** icon.



Field	Description
Name	The Object name must be unique within the library, but may be duplicated across different libraries.
Description	Free format object description.
ABAP program generation	This is the name of the ABAP program that Cupid generates for the translation object definition. You can use the display button to invoke the ABAP editor to view and / or maintain the generated ABAP code. You may re-generate the ABAP program at any time.
ECS interface management	Cupid generates code for ECS interface management, that is, imbeds ECS commands to retrieve parameters. This option enables you to automatically generate a ECS Process definition to execute the generated Cupid ABAP program. The **Display** button invokes the ECS Process workbench to view or maintain the generated definition.

Adding Data Objects

A data object defines input, output and reference data definitions and how they are to be processed by Cupid. There is a one-to-one relationship between the translator data object and a dictionary object, thus the dictionary object must contain all the structures used to define the data attributes. Selection criteria (conditions) are then configured to determine what structure is to be used when. Validation rules may also be configured against each attribute (input only). Various aspects of a structure and attribute may also be overridden. Thus a data object encapsulates the source or target of data and how it is to be accessed.



Field	Description
Name	The Data Object name must be unique within the Object.

Field	Description
Description	Free format data object description.
Transfer medium	The transfer medium identifies the access method used to retrieve or create data, for example, File, IDOC, ECS parameter, SAP table, TSX message, TSX response, XML file, XML TSX message.
Transfer medium name	The transfer medium name identifies how to derive the name of the file, IDOC, and table. If left blank, Cupid attempts to get this from ECS. You may imbed dynamic substitution variables in the name to make it unique (click the **Hints** button to get a list of valid substitution values).
Transfer medium primary indicator	Indicates whether the input data object is the primary data source (that is, used to drive the Cupid translation process). You need to allocate the **Primary** check box to one (and only one) input data object.
No automatic open indicator	By default, data objects automatically open when the translation starts. If you enable this option, the data object does not automatically open and a you need to trigger a manual OPEN action prior to reading or writing to the data object.
Processing mode	Indicates whether the processing is input, output or reference. Reference indicates that no actual I / O is performed, but the data object structures and attributes are used only for reference, that is, work areas.
Delimiter	The delimiter option indicates whether the data format is fixed or delimited by a special value, for example, ','. You may not specify '~' as a data delimiter.
Fixed length	This option causes output to generate as fixed length records. Any data over this length is written out as new fixed length records until the end is reached.

Field	Description
No compression	***Note:*** To use with the **Delimiter** option only. The **No compression** option is for output only and causes delimited output to process using the full length of the fields, that is, include spaces. By default, Cupid compresses out training spaces.
Trim trailing delimiters	This option causes trailing delimiters with no intermediate value to trimm back to the first value from the end of the string.
Suffix character	The optional suffix character causes all output file data to suffix with the specified character.
Auto clear structure	When you enable, the structure applied to the data object is cleared for each record read in. If you disable, values not modified between records are retained.
Prefix structure	The name of the structure (from the Cupid data dictionary) to apply as a prefix. This option is useful for situations like generating an IDOC where data records are prefixed with a common set of fields.
Suppress CR/LF	When you enable, forces Cupid to behave in binary mode. This is useful when records are delimited by a record other than CR/LD (for example, EDI where records are delimited by apostrophes).
Auto append table structures	When you enable, input data objects with an occurs structure automatically appends matching records.
XML Controls	For output XML data objects indicates whether XML should be indented for readability using tabs, spaces or not at all. Also whether you should use a linefeed (CRLF) at the end of each record.

Click the **Hint** icon 

to present additional information that can help with this screen.

The Primary Input Data Object and Multiple Inputs

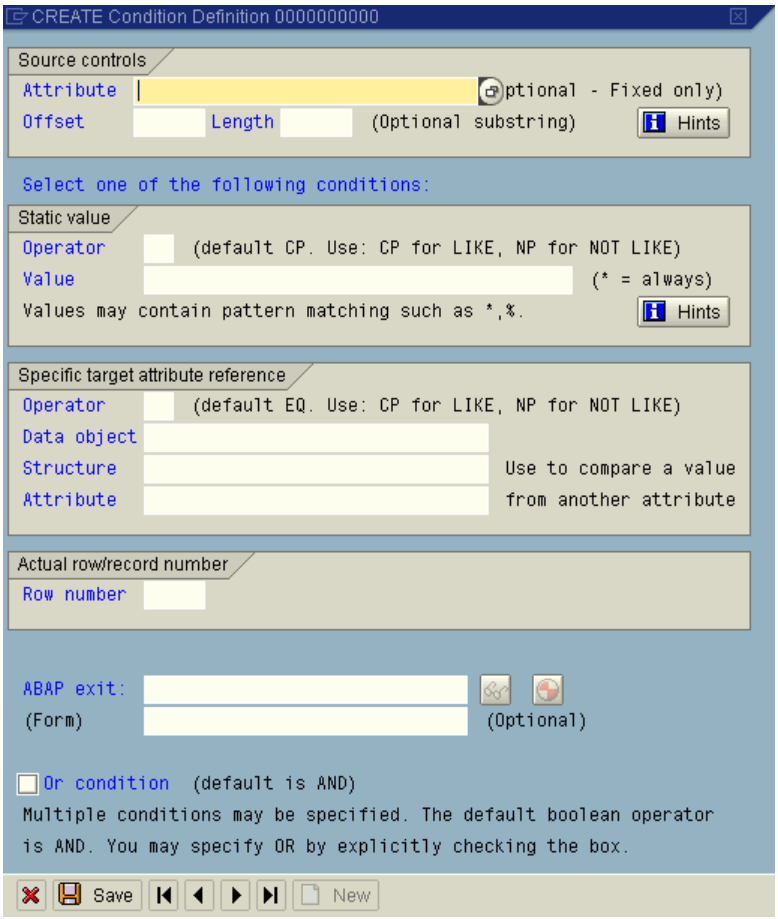
You must have at least one primary input data object. This forms the "main loop" of the processing. You may selectively read other input data objects at controlled points using the SELECT event action.

Using Dummy Data Objects to Include Additional Work Structures

Just because you specify a data object, it does not mean that you need to use it. This is a convenient way to include other data dictionary objects, thus working around the cupid limitation of one data dictionary object per data object. Configure the dummy data object as output and the related structures are automatically available for processing.

Selection Criteria - Condition

You define the selection criteria (conditions) at the data object structure level and may only specify for input data objects. They basically determine which data structure to use based on the input data. You need to specify at least one for input File processing.



Field	Description
Offset (from 0)	Map the structure based on a value at a specific offset and length within the input data record. You may use an offset of zero (0) to indicate the start of the record.
Static Value	Compare a static value against data in the record. If the condition evaluates to true, the associated structure is mapped. You may use standard pattern matching that is, *, % and underscore. A value of * indicates every input record.

Field	Description
Attribute Comparison	Compare a value from another Cupid attribute against data in the record. If the condition evaluates to true, the associated structure is mapped.
Row number	Enables you to apply a structure to a specific row number, for example, the first record is always a header.
ABAP exit	You may call an optional ABAP exit to help determine more complex selection criteria. See the section on ABAP exit processing for more details.
Or condition	You may specify multiple selection criteria definitions. By default, they are AND. Select this check box to indicate an OR condition instead.

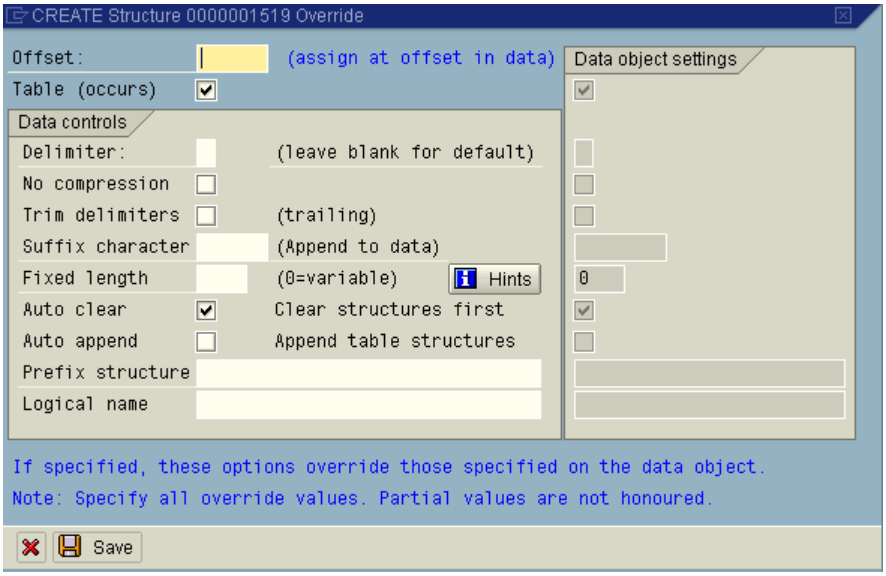
Click the **Hint** icon 

to present additional information that can help with this screen.

Structure Overrides - Offset and Delimiter

Cupid assigns a structure from offset (0), by default; that is, the start of the input data. However, in many cases, there is a requirement to assign (map) a structure at an offset in the data. You may specify this offset, by double-clicking (or selecting the change icon) on the structure from the main hierarchy. The following dialog appears. You may also override the default delimiter processing specified at the data object level.

The screen displays the existing data object values in the right hand side for comparison.



Field	Description
Offset	The offset from (0) in bytes that the structure is to map to. Example, an offset of 10 assigns the structure starting from the 11th byte of the input data.
Table (occurs)	If you select the check box, the structure is declared as a internal table (array) definition. This may be used to override the dictionary default.
Delimiter	The **Delimiter** option indicates whether the data format is fixed or delimited by a special value, for example, ','. You may not specify '~' as a data delimiter.

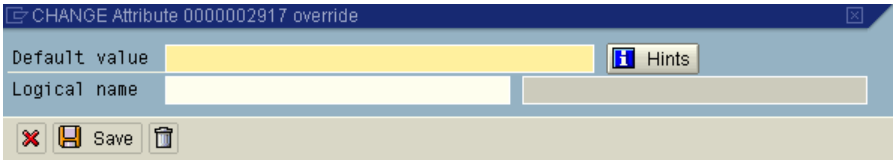
Field	Description
No compression	***Note:*** To use with the **Delimiter** option only. The **No compress** option is for output only and causes delimited output to process using the full length of the fields, that is, include spaces. By default, Cupid compresses out training spaces.
Trim delimiters	This option causes trailing delimiters with no intermediate value to trim back to the first value from the end of the string.
Suffix character	The optional suffix character causes all output file data to suffix with the specified character.
Fixed length	This option causes output to generate as fixed length records. Any data over this length is written out as new fixed length records until the end is reached.
Auto clear	When you enable, the structure applied to the data object is cleared for each record read in. If you disable, values not modified between records are retained.
Auto append table structures	When you enable, input data objects with an occurs structure automatically appends matching records.
Prefix structure	The name of the structure (from the Cupid data dictionary) to apply as a prefix. This option is useful for situations like generating an IDOC where data records are prefixed with a common set of fields.
Logical name	The logical name to apply to the structure. This is used for XML data objects to match the structure name to an alternate XML tag.

Click the **Hint** icon 

to present additional information that can help with this screen.

Attribute Default Override

You may override the default dictionary default by double-clicking (or selecting the **Change** icon) on the attribute from the main hierarchy. The following dialog appears:



As well as static values, you may use dynamic substitution variables to automatically assign the following defaults at run time:

Variable	Description
&DATE	Current SAP system date. You may also use date variations of &YYMMDD, &YYMM, &YY, &YYYY, &MMDD, &MM, &DD.
&TIME	You may also use Current SAP system time, &HHMM.
&LOCALDATE	Current SAP local date
&LOCALTIME	Current SAP local time
&TIMEZONE	Current SAP local time zone
&CLIENT	SAP client number
&HOST	SAP host name
&SYSID	SAP system identifier
&USERID	SAP user ID
&RNUMBER	ECS run number of process
&RDATA	ECS run data of process

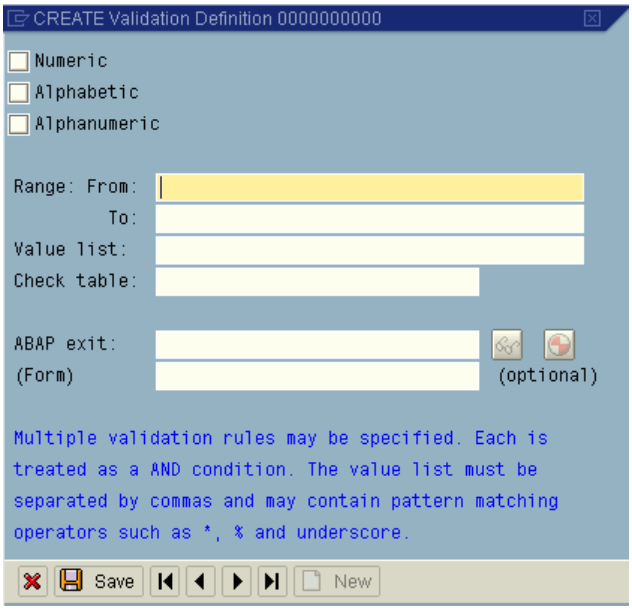
Variable	Description
&UVAR1/2/3	ECS User variables 1/2/3

Note: You can use other Cupid attribute values by specifying them in the format ({structure}-{attribute}).

Click the **Hint** icon  to present additional information that can help with this screen.

Input Validation Rule

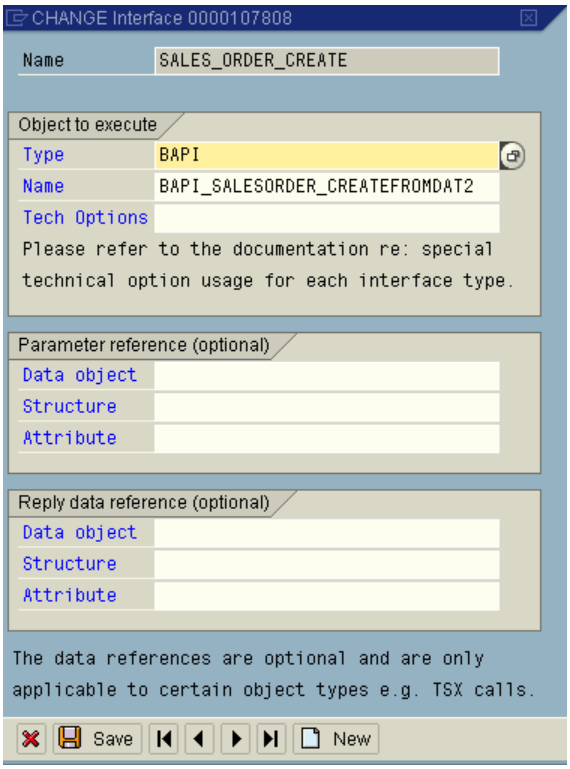
Validation rules are optional and you may only apply to attributes of input data objects. You may specify multiple validation options. If the validation check fails, Cupid processing immediately halts.



Variable	Description
Content checks	Select any one of the numeric, alphabetic or alphanumeric check boxes.
Range	This is a from / to value range for the data.
Value list	A list of values separated by commas.
Check table	An existing SAP table of values. Cupid uses the first field of the primary key to perform the check.
ABAP exit	An optional ABAP exit may be called to help perform more complex validation. Refer ABAP exit processing for more details.

Adding Interfaces

Interface definitions are optional. You use interface definitions to define how Cupid is to invoke an interface when an execute action is defined for an event.

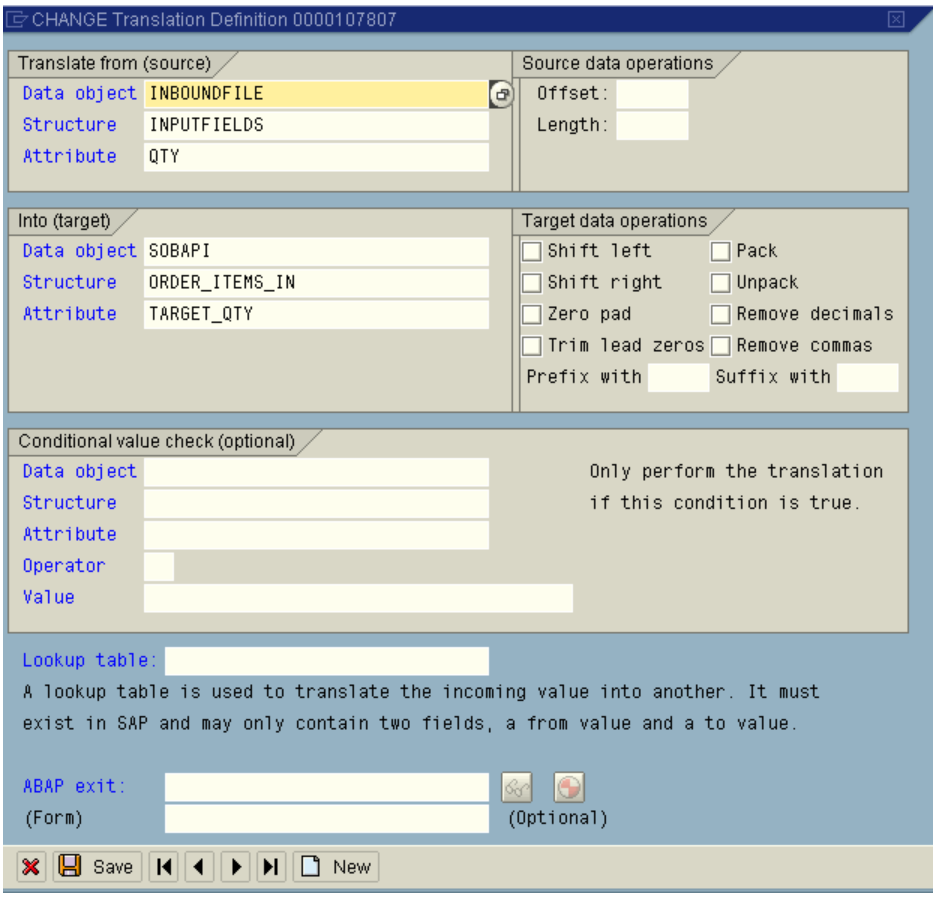


Variable	Description
Name	A unique name used to identify the interface definition.
Object	Defines the Object to execute, for example, SAP IDOC, BAPI, FM, ECS process, TSX message, and Number range. The object is qualified by a type (for example, BAPI) and a name.
Tech Options	Optional reference value for the interface object. Currently you use this only for object type number range - where the tech options is used to specify the number range interval.

Variable	Description
Parameter Reference	Nominates a Cupid data object, structure or attribute to pass to the object.
Reply data	Nominates a Cupid data object, structure or attribute that receives data returned from the object.

Adding Translation Definitions

Translation definitions are basically used to map values from one structure to another. There are options to perform an automatic substring of the source data and basic formatting of the target data after the move. The translation can be made conditional i.e. based on the value of another attribute. A translate table may also be used to look up the input data value and automatically substitute another. As many translation definitions as required may be created using the same data references.



Variable	Description
Translate from	The data object, structure and attribute from which to source the data. You may specify an offset and length to substring a value.

Variable	Description
Translate into	The data object, structure and attribute to assign the data to.
Formatting	You may automatically format the data moved into the target attribute after the assignment using the shift left / right, zero pad, trim leading zeros or pack / unpack, prefixed / suffixed.
Condition	You may make translations optional, based on the comparison of an attribute against a static value. The translations is only performed if the comparison evaluates to true.
Translation table	You may specify a 'look up' table. This table must exist in SAP and is used to automatically translate values using a cross reference table. If an entry does not exist, Cupid processing halts immediately. Cupid takes the first primary key field for the input selection and the first non-key field for the output assignment.
ABAP exit	You may call an optional ABAP exit to help perform more complex translations. Refer ABAP exit processing for more details.

Adding Processing Logic

Cupid events control the processing logic and what actions to take. It is important to first understand the types of event that are possible and the sequence in which they are processed.

Typical Processing Loop

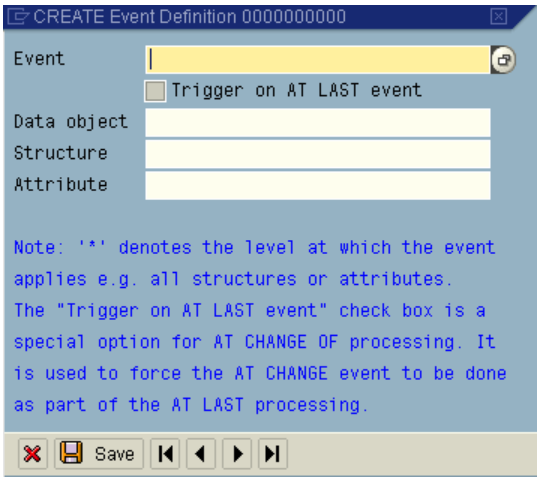
Initialization
Loop through primary input data

- Map data structure depending on data values
- At first occurrence of .....
- At change of value for .....
- Translate values depending on the structure
- At every occurrence of .....
  End of loop
  At last occurrence of data object .....
  Termination

> ***Note:*** Translation did not occur yet for the 'at first' and 'at change of' events. If you require this, you may trigger a TRANSLATE action explicitly in these events.

Adding an Event

To create a new event, click the **Processing events and actions** node and then the **Create** icon on the application toolbar. To maintain an existing event, double-click the entry, or position your cursor on it and then click the **Change** icon on the application toolbar. The following event dialog then appears:



Field	Description
Event	The event code, that is, INITIALISE, AT FIRST. Use the drop-down for allowed values.
Data object, structure and attribute	The level (control break) at which you want to apply the event. This varies per event type, that is, some levels are mandatory, optional or not allowed. Refer the following table. A '*' that indicates all occurrences are affected by the event.

6.16.6.7 Types of Event and Sequence

Event/Sequence	Description
INITIALISATION	Performed when the Cupid translation object starts and before any input is processed.

Event/Sequence	Description
AT FIRST	At first occurrence of a data object or structure
AT CHANGE OF	At change of value of a data object or structure
AT EVERY	At every occurrence of a data object or structure
AT LAST	At the last input value of a data object
TERMINATION	Post translation processing
<NAME>	Custom event of name <NAME>. You can call this event directly by an action. This allows common action logic to define once and call repeatedly as required.

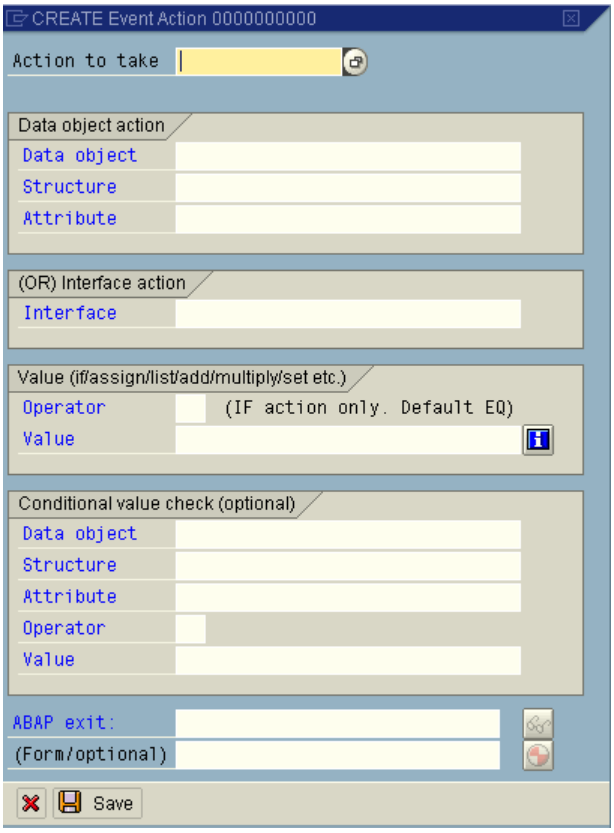
6.16.6.8 Control break levels allowed by event

	Data Object	Structure	Attribute
AT CHANGE OF	M	O	N
AT EVERY	M	O	N
AT FIRST	M	O	N
AT LAST	M	N	N
INITIALISATION	N	N	N
TERMINATION	N	N	N

In the above cross reference table, M = Mandatory, O = Optional, N = Not allowed. The first column is data object, second is structure and last is attribute level, for example, you may specify no control break at all for the initialization and termination events (N).

Adding an Action

Once you define an event, you may associate processing actions with it, that is, what is going to happen when the event is triggered. To create a new action, click the event node and then the create icon on the application toolbar. To maintain an existing action, double-click the entry, or position your cursor on it and then click the change icon on the application toolbar. The following action dialog then appears:



Variable	Description
Action to take	The action code, that is, APPEND, and CREATE. Use the drop-down to display all the allowed values.
Data object action	The data object, structure or attribute on which to take action. The level at which an action can perform can vary (see the following cross reference table).

Variable	Description
Interface	EXECUTE action only. This is the interface definition to invoke.
Value	ASSIGN / IF / ADD / SUBTRACT / DIVIDE / MULTIPLY / LIST actions. For assign, you may assign either a static value or a dynamically substituted value, for example, &DATE, &TIME. See the following table for a valid list of substitution variables.
Condition	You may make an action optional, based on the comparison against the static value.
ABAP exit	You may call an optional ABAP exit to help perform more complex translations. Refer ABAP exit processing for more details.

6.16.6.9 Type of Action that may be Performed

Action	Description
ABORT	Abort Cupid processing
ADD	Add a static value to an attribute
APPEND	Append a row of an internal table (occurring structure).
APPEND XML	Append a structure to an XML data object.
ASSIGN	Assign a fixed value to a structure or attribute
CALL USER EXIT	Call a Cupid user exit
CLASSIFY	Perform an ECS classification using an attribute
CLEAR	Initialize a structure or attribute. If you perform this at the data object or structure level, the internal Cupid row counter is also reset.

Action	Description
CLEAR TABLE	Initialize a table.
CLEAR XML	Initialize an XML structure.
CLOSE	Close a data object and release associated resources.
CREATE	Creates a output data object definition, for example, output file record
CREATE XML	Creates an output XML data object definition. You can perform at the data object or structure level.
DIVIDE	Divide an attribute by a static value
ENDIF	Terminate a previous IF statement
ENDLOOP	Terminate a previous LOOP statement
ENDWHILE	Terminate a previous WHILE statement
EXECUTE	Execute a SAP interface (as per Cupid interface definition).
FAIL	Fail the ECS process stopping Cupid processing
IF	IF operator. Allows a block of Cupid actions to be conditional as opposed to putting a condition on a single action.
LEAVE	Leave a loop / while process
LIST	Log a static value &/or attribute value
LOOP AT	Loop through an internal table
MULTIPLY	Multiply an attribute by a static value

Action	Description
OPEN	Open a data object
SELECT	Select matching records from the input data object
SELECT ALL	Select all matching records from the input data object into the applied table
SET ECS ENQUEUE	Configure ECS enqueue value from the specified attribute
SET ECS FILE	Configure ECS file variable from the specified data object
SET ECS RDATA	Configure ECS run data from the specified attribute
SET ECS RTEXT	Configure ECS run text from the specified attribute
SET OBJECT NAME	Configure the data object name
STOP	Stop processing immediately
SUBTRACT	Subtract a static value from an attribute
TRANSLATE	Force Cupid to perform a translation. You may use this n AT FIRST / AT CHANGE OF EVENTS to force a translation to occur before the internal translate event is triggered.
TRIGGER EVENT	Trigger custom event <NAME>
WHILE	Perform while loop while specified attribute equals the static value

6.16.6.10 Assignment Substitution Variables

Variable	Description
&DATE	Current SAP system date. You may also use date variations of &YYMMDD, &YYMM, &YY, &YYYY, &MMDD, &MM, &DD.
&TIME	Current SAP system time
&LOCALDATE	Current SAP local date
&LOCALTIME	Current SAP local time
&TIMEZONE	Current SAP local time zone
&CLIENT	SAP client number
&HOST	SAP host name
&SYSID	SAP system identifier

6.16.6.11 Control Break Levels Allowed by Action

	Data Object	Structure	Attribute	Interface	Value
APPEND	M	M	N	N	N
ASSIGN	O	O	O	N	O
CLEAR	O	O	O	N	N
CREATE	O	O	N	N	N
EXECUTE	N	N	N	M	N
TRANSLATE	M	M	N	N	N

In the above cross reference table, M = Mandatory, O = Optional, N = Not allowed. The first column is data object, second is structure, third is attribute, fourth is interface and the last is value level.

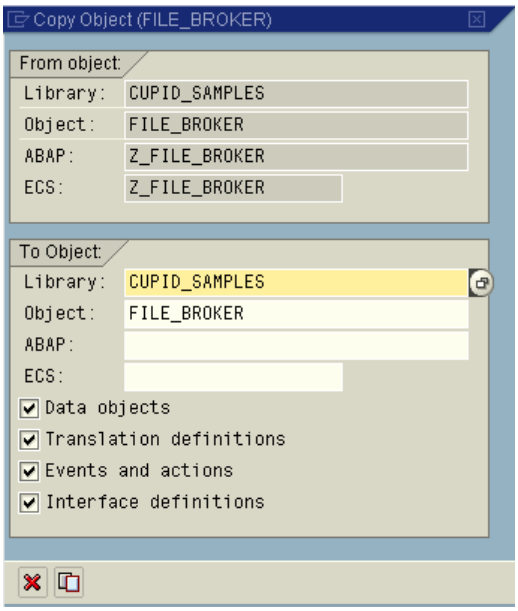
6.16.6.12 Managing Cupid Translation Objects

The following sections explain how to use the Cupid Translation Workbench to:

- Copy translation objects
- Rename translation objects
- Reassign translation objects
- Generating run time objects.

How to Copy a Translation Object

Once you create a translation object, you can copy it to create another definition. To copy a translation object, select the object then click the **Copy** icon in the menu bar. A dialog appears:



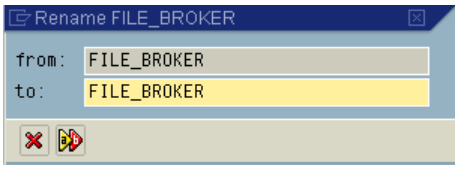
Enter the following:

Field	Description
Library	The target library for the new object
Object	New translation object name
ABAP	New object ABAP report (that is, run-time object)
ECS	New object ECS process name
Data Objects	Include data object definitions from source object
Translation definitions	Include translation definitions from source object

Field	Description
Events & actions	Include event and action definitions from source object
Interface definitions	Include interface definitions from source object

How to Rename a Translation Object

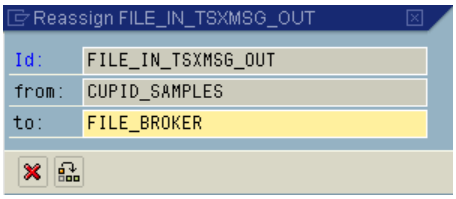
You can rename translation objects by selecting the definition, then clicking the **Rename** icon on the menu bar. You are prompted for the new name.



> ***Important:*** Renaming of active objects is not allowed. You need to first deactivate the object.

How to Reassign a Translation Object

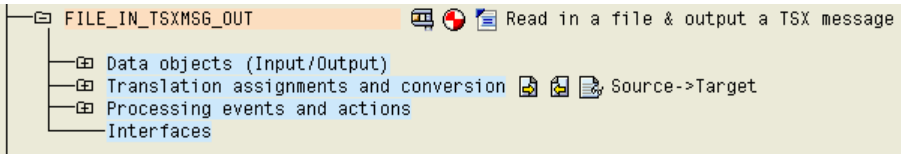
You can reassign a translation object from one library to another. To reassign an object, select the object then click the **Reassign** icon on the menu bar. You are prompted for the new parent definition to assign the item to.



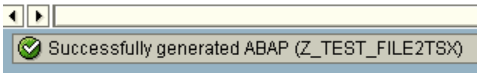
> ***Important:*** Reassigning of active objects is not allowed. You must first deactivate the object.

Generating a Translation Object

You need to generate a translation object to create the run-time ABAP program. To generate an object, select the object, then click the generate icon at the right of the object title bar.

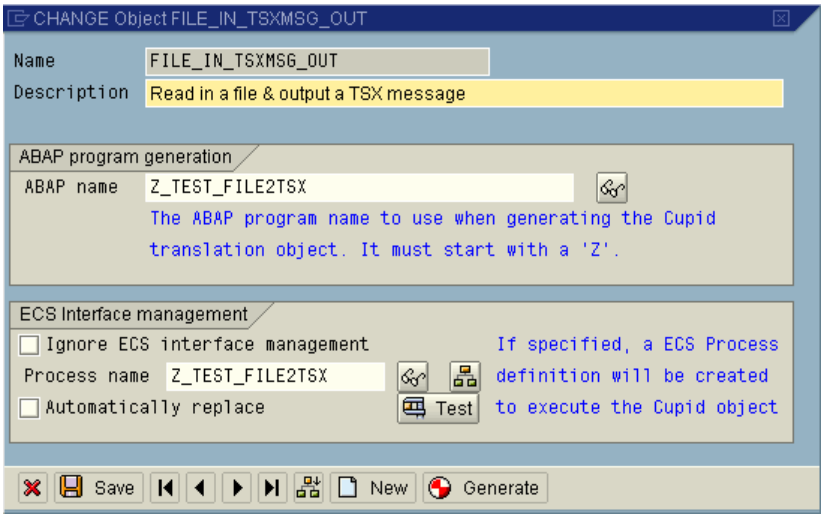


If the run-time ABAP is successfully generated, a message appears in the status bar at the bottom of the SAP Gui window.



> ***Note:*** By default, each time you generate the run-time object, the ECS definition is also regenerated.

To prevent this from happening, clear the **Automatically replace** check box in the translation object definition.



Transporting Cupid Objects

The transporting of Cupid objects between SAP environments involves two steps:

- Transport of Cupid run-time object(s)
- Transport of Cupid definitions (optional).

Transport of Cupid Run Time Objects

The transport of Cupid run time objects uses the SAP standard transport mechanism (that is, TMS / tp). The steps involved to transport a Cupid run-time object are:

- Create a transport request
- Link the Cupid run time ABAP to the transport request
- Release the transport
- Import into target system.

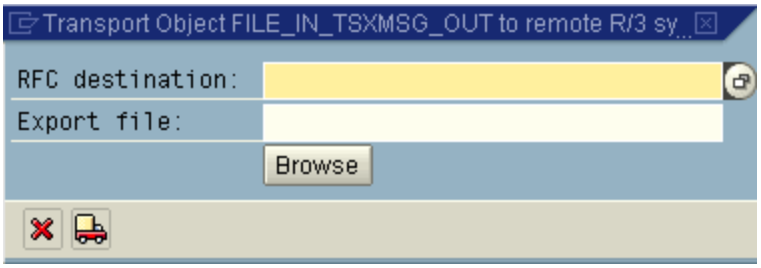
> ***Note:*** Remember to include any Cupid exits associated with the run-time object.

Transport of Cupid Definitions

You can transport cupid definitions themselves (translation object and associated dictionary objects) from one SAP environment to another.

To do this, select either the individual translation object or library of objects to transport and then click the Transport icon  in the menu bar.

You are prompted to provide either the local file path for the transport file or an RFC destination for the target SAP system.



If you select the RFC transport option, the Cupid definition(s) are created on the target system. If the file transport option is used, then you need to import the generated file on the target system.

To import a Cupid transport file, from the Cupid translation workbench, select **Utilities** > **Import transport file**.
You are prompted for the path to the transport file.

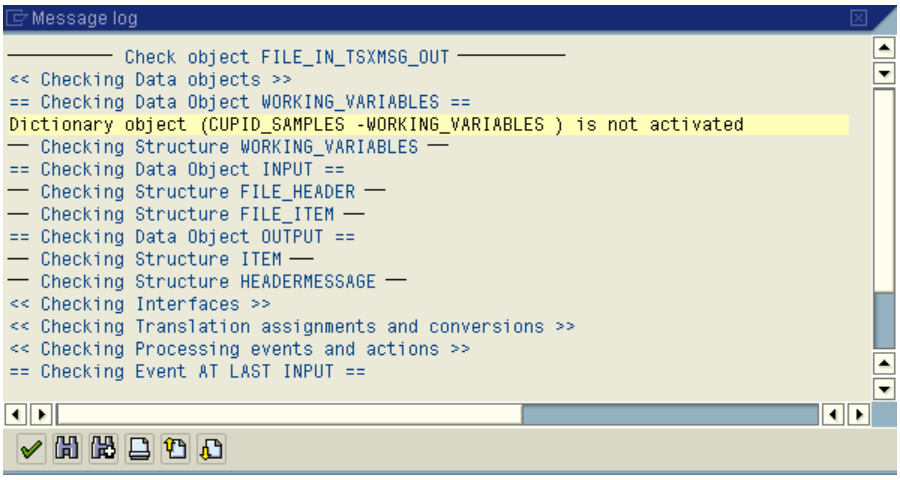
> ***Note:*** This step is optional. The Cupid object definitions are not required for the run time objects to operate.

Checking a Translation Object

The Cupid translation workbench provides a check function that validates the consistency of a translation object.

To check an object, select the object then click the **Check** icon in the menu bar. The results of the consistency check appears in a separate window. Any warning or errors identified are indicated.

In the following example, a warning is shown because one of the included data objects is inactive.



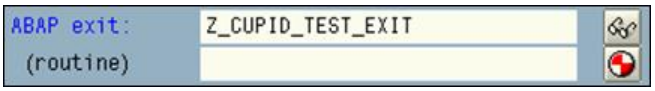
6.16.7 ABAP Exit Processing

In many cases, where you require more complex processing than Cupid can typically provide, you may use an ABAP user exit. Cupid can automatically generate user exit code for you, along with all the parameter passing, and variable declarations.

You may invoke exits for the following definitions:

- Validation
- Selection criteria (condition)
- Translation
- Event action

You may also configure an optional routine identifier; this is not used by Cupid, but you may reference in the ABAP exit to determine what area is processed. This is especially useful if the same ABAP exit is used for multiple cases. You can then use the free format routine identifier to identify the type of processing required.



To automatically generate the exit code, click the generate icon . You may then use the display button to invoke the ABAP editor to view or maintain the generated source code.

6.16.7.1 Custom Code

When adding custom code to the exit program, avoid placing code between the Cupid code markers. These are clearly identified by the "* <ECS_CUPID_.....START>" and "* <ECS_CUPID....._END>" ABAP comment statements. If the ABAP exit is re-generated, only the code between these markers is regenerated.

6.16.7.2 Data Processing

When an exit is invoked, it is passed all the structures and some control variables through ABAP memory. You may manipulate any structure value and return back to cupid to continue processing. The name of any routine identifier is passed through the v_current_exit_routine variable.

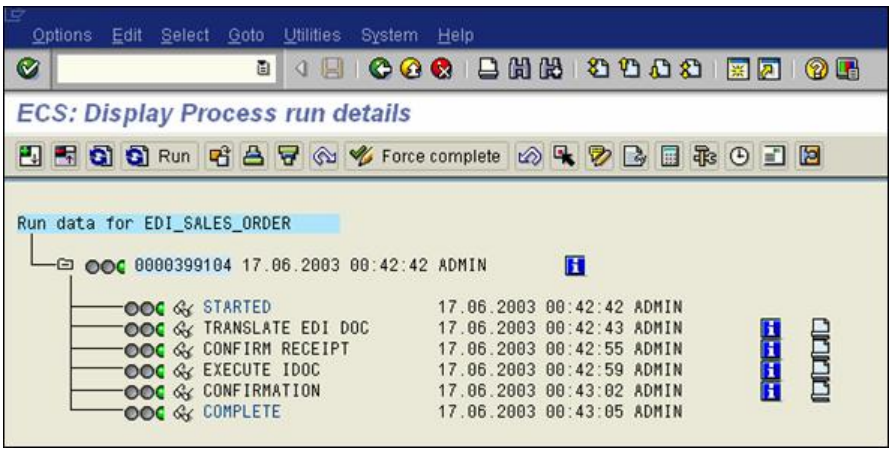
6.16.7.3 Error Processing

If a processing error is encountered by the exit and it wishes to halt Cupid processing, you need to set the v_exit_rc and v_exit_message variables.

6.16.8 Supporting Functionality

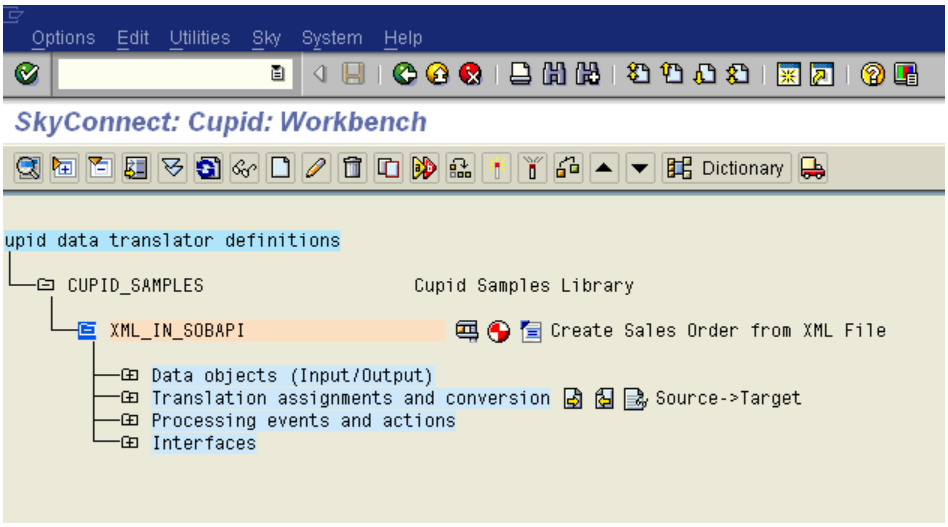
6.16.8.1 ECS Processing

Once the Cupid run time object is generated and activated, you may use standard ECS processing to execute it. As mentioned previously, the Cupid data translator automatically generates all the ABAP code necessary to execute the translation. Standard ECS processing is also used to monitor this execution, notify support of any failures and restart any failures.



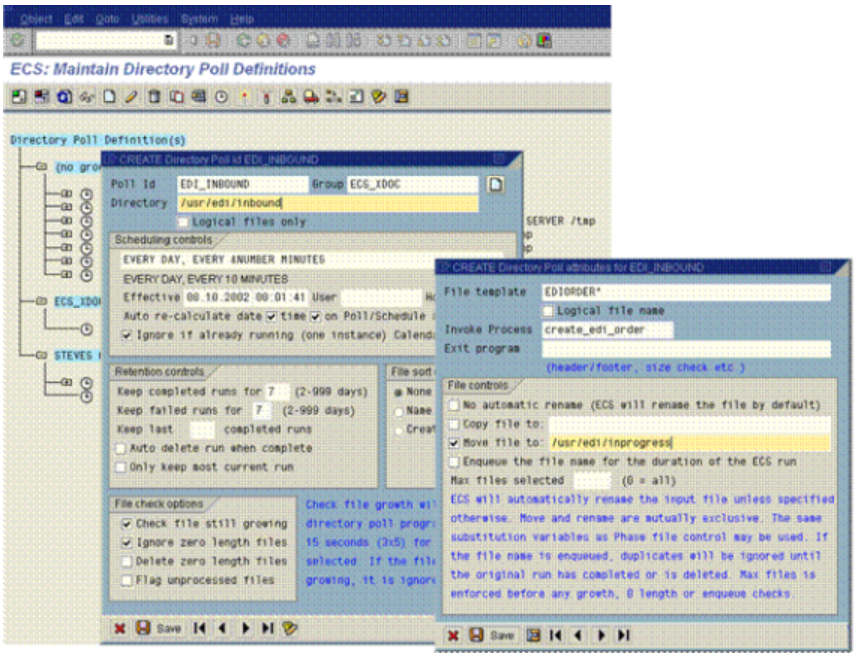
You may monitor all processing centrally in SAP in one central location. The ECS interface management framework enables real time monitoring, notification and reprocessing capability. It also manages the proper sequencing of execution to prevent failures due to resource conflicts.

You can trigger ECS processing for testing purposes directly from the Cupid translation workbench through the test icon.



6.16.8.2 File Polling

File polling capability is a standard component of ECS. This facility enables you to periodically check a nominated directory for the existence of certain files. It also contains standard controls to automatically rename and / or move the file to another directory and make sure the file is complete, that is, is not still growing. Once a file is detected, an ECS Process is automatically started, passing the name of the file to process. You may simply perform the configuration of a file poll definition from the ECS Process workbench. Simple screens prompt for the directory to poll and the name of the file to check for.



6.16.8.3 TSX

The TSX component is also included with ECS. TSX provides the means to send and receive files, messages and XML from remote systems (both SAP and non-SAP). TSX is integrated with Cupid providing the means for Cupid to either process an inbound file / message or pass to TSX for delivery a generated file / message.

6.16.9 Cupid Examples

In the following section a number of examples of Cupid translations are provided.
For further examples, refer to:

- Cupid samples library - contains a range of sample Cupid objects
- Cupid teach yourself guide.

6.16.9.1 Outbound XML Messaging using TSX

In this example, Cupid generates a XML message and pass it to TSX.
The primary input data object is ECS run data. Secondary inputs are SAP tables.

The role of the Cupid object in this process is to:

- Take the delivery document number passed in
- Extract delivery details (items, pallets, and boxes)
- Generate the required delivery XML document
- Call TSX passing the XML message

The required XML format is as follows:

XML Format

```
<DELIVERY>

     <NUMBER>80000002</NUMBER>

     <DATE>20080527</DATE>

     <CUSTOMER>1</CUSTOMER>

     <PALLET>

<PALLETNUM>1</PALLETNUM>

<BOX>

     <BOXNUM>1</BOXNUM>

     <ITEM>A100</ITEM>
```

```
      <QTY>100</QTY>

</BOX>

<BOX>

      <BOXNUM>2</BOXNUM>

      <ITEM>A100</ITEM>

      <QTY>100</QTY>

</BOX>

</PALLET>

</DELIVERY>
```

TSX Prerequisites

This example requires the following prerequisites:

- a TSX java server is configured and running
- a TSX logical destination and message called XML_DELIVERY exists
- you have a delivery document with line item (s) available to test with

Refer to the TSX documentation for details on configuring a TSX java server, logical destinations and messages.

Cupid Definition

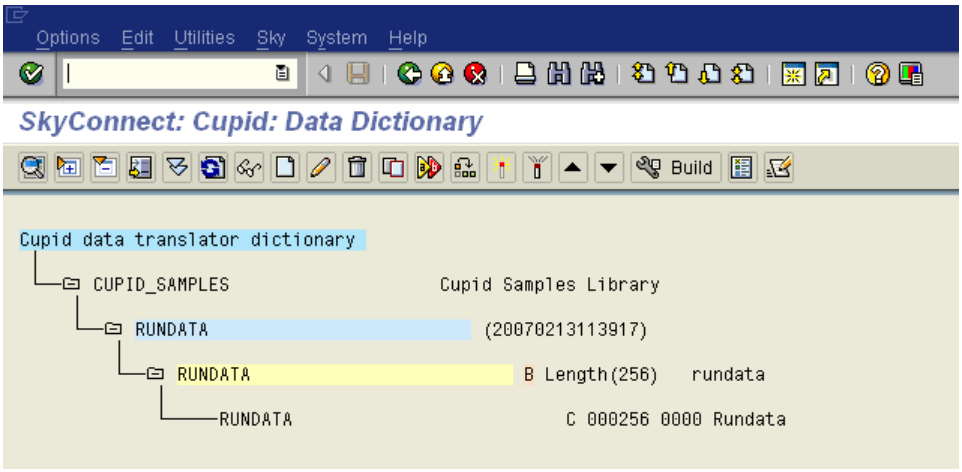
Dictionary Objects

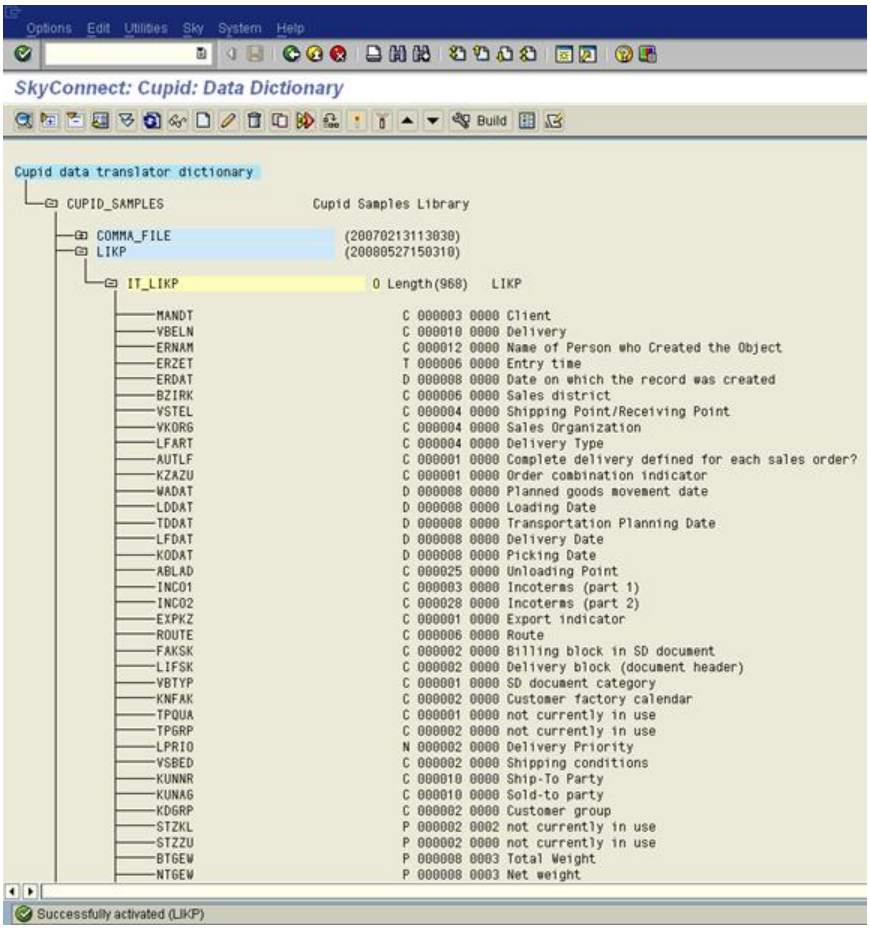
For this example, you require four Cupid dictionary objects:

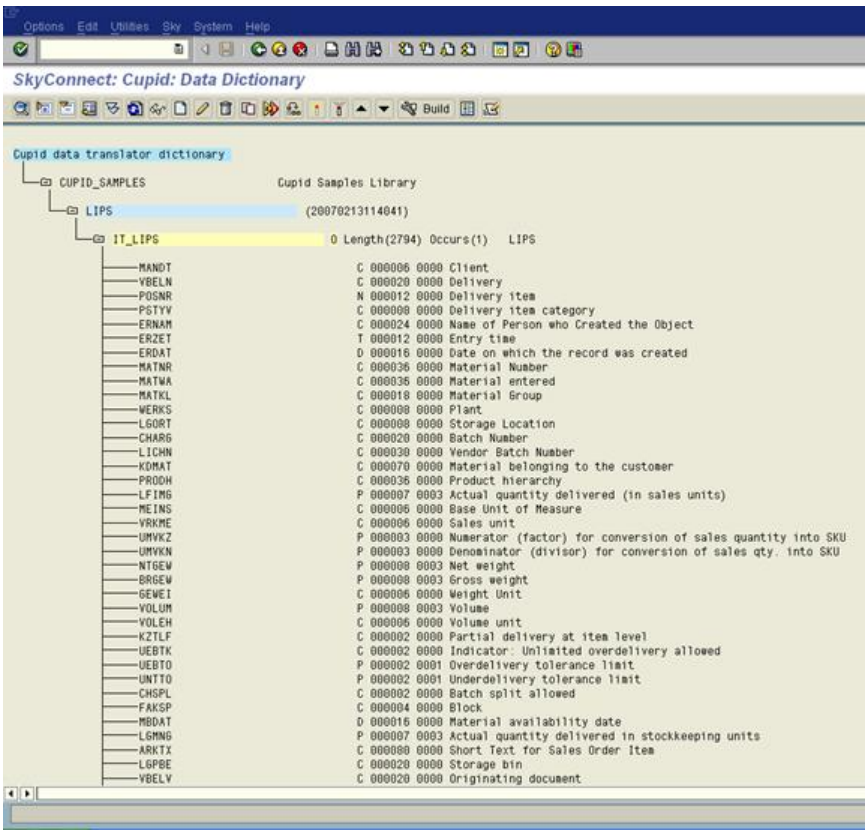
- Object for delivery document number from user exit through ECS run data input
- Object for SAP table LIKP to extract delivery header details
- Object for SAP table LIPS to extract delivery item details
- Object for delivery XML message output.

You can quickly build the two table data dictionary objects (LIKP and LIPS) that referencethe SAP dictionary items.

You need to build the remaining two dictionary objects manually.



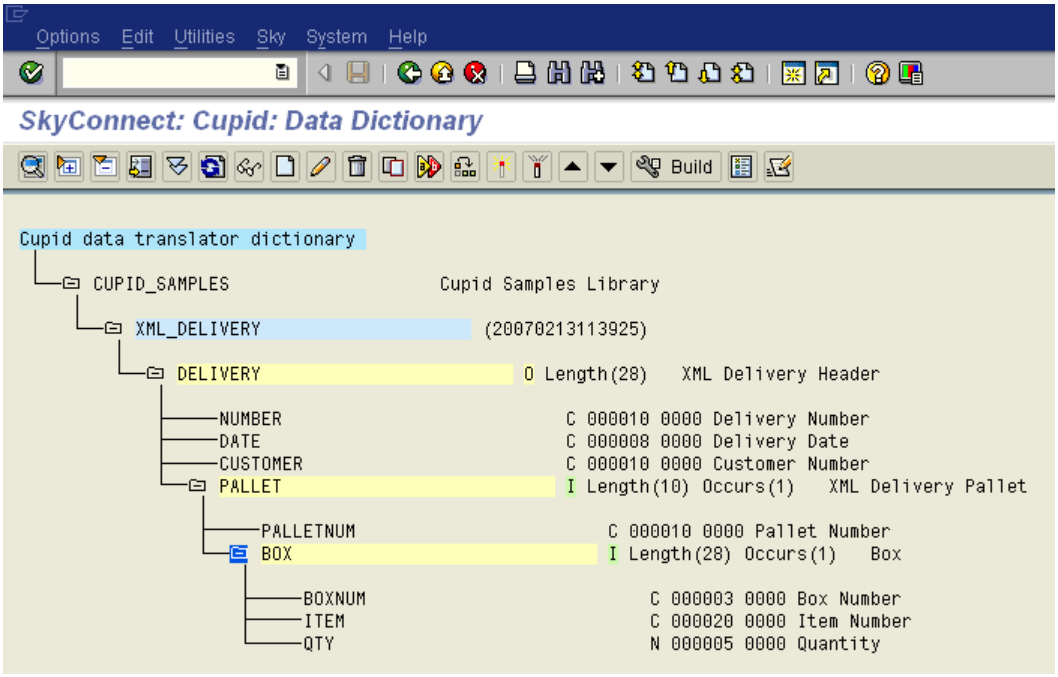


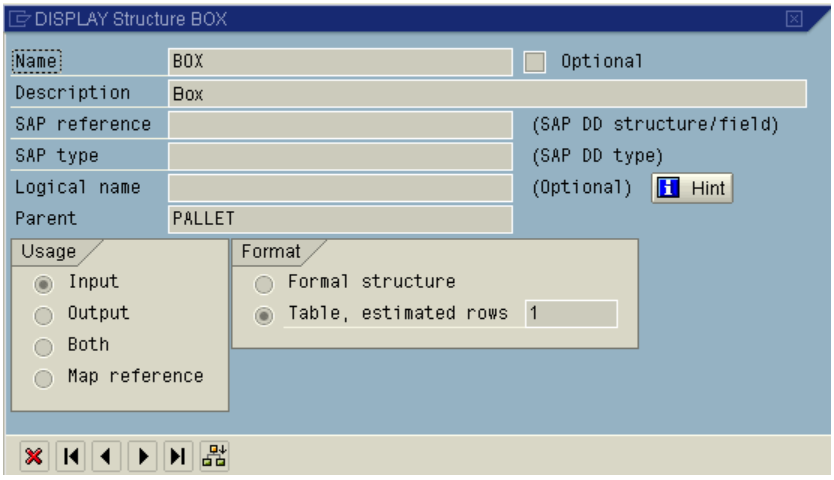


Note: With the XML dictionary object, there are actually three structures:

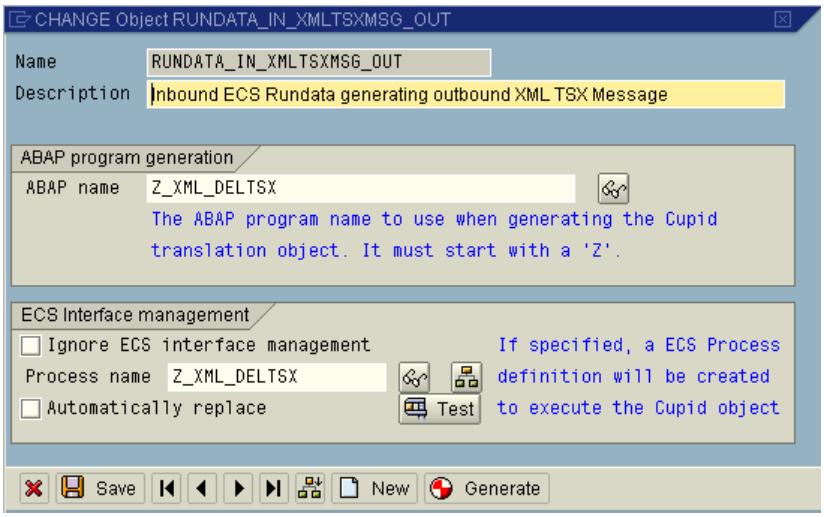
- Box
- Pallet
- Delivery

For Box and Pallet, they have a parent structure assigned to create the nested hierarchy. Also the names of the structures and attributes need to match the XML tags required.





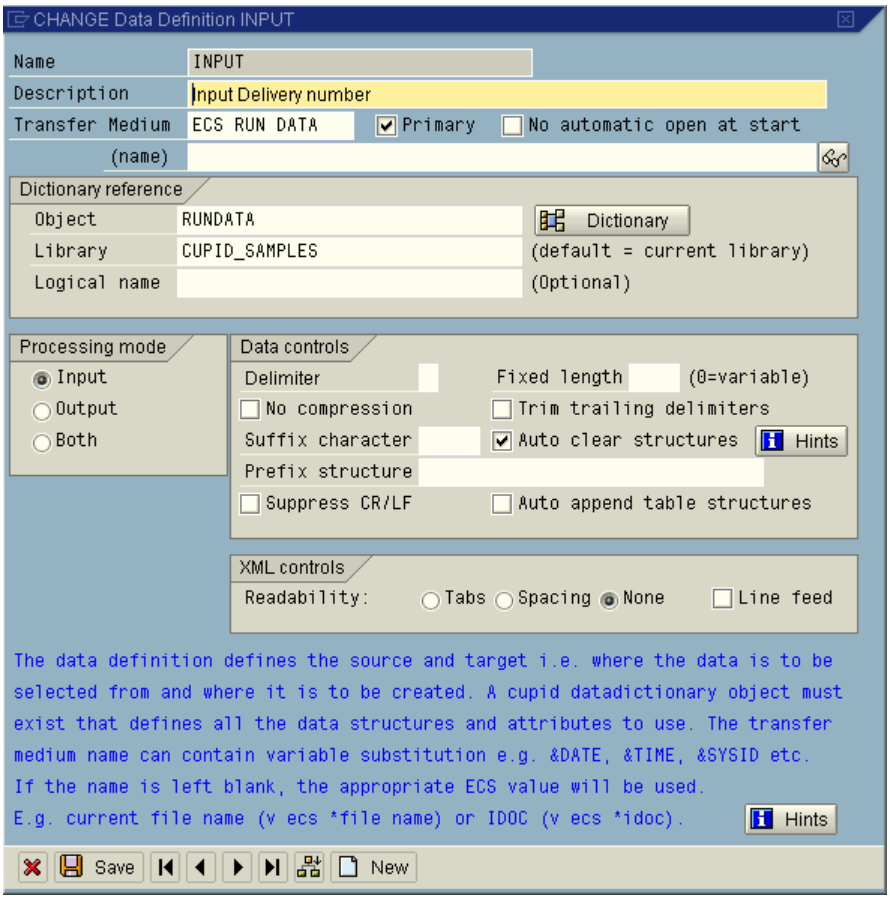
Translation Object



Data Objects

Primary Input Data Object

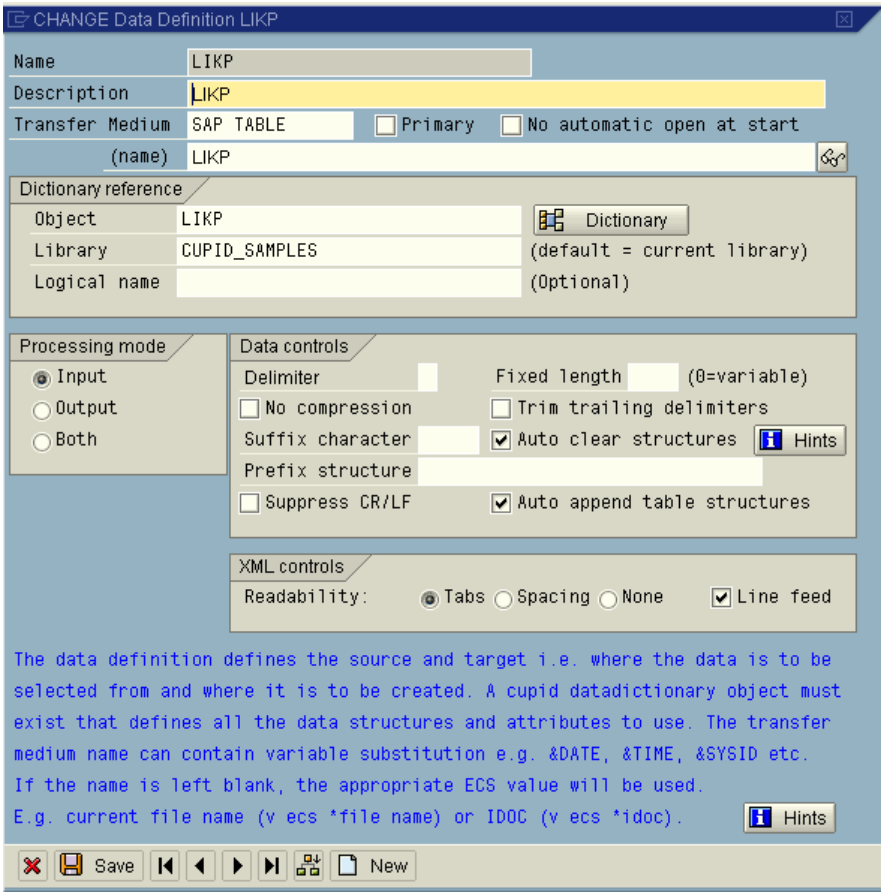
The primary data input in this example is the ECS rundata that is passed by the user exit. The rundata includes the delivery document number.

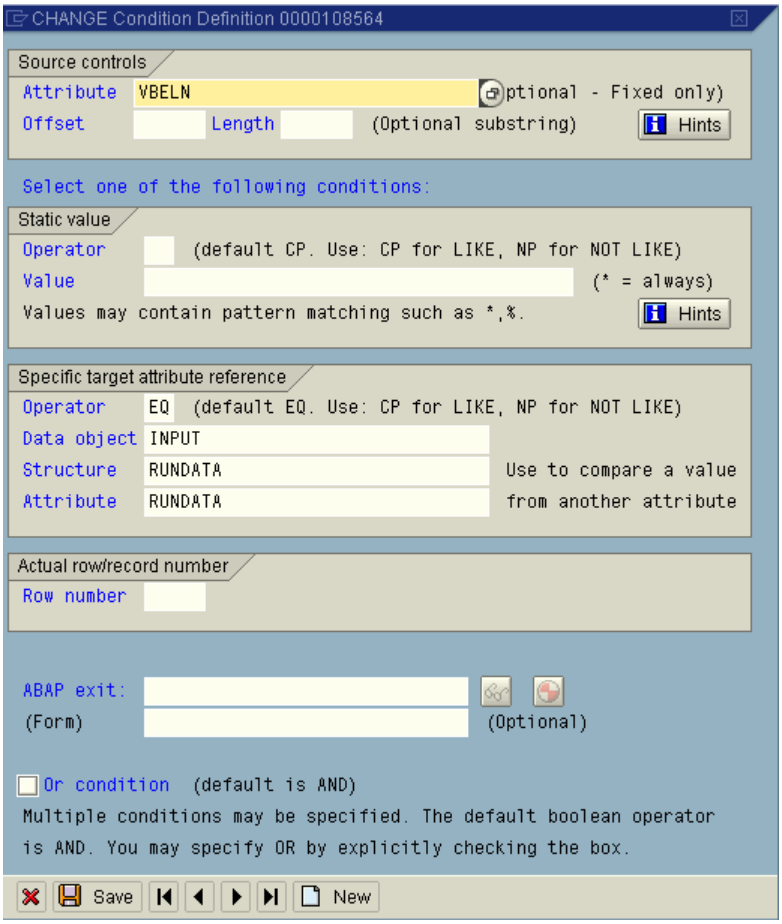


LIKP Input Data Object

The delivery document number received from rundata is used to select the required delivery header details from the SAP table LIKP.

> ***Note:*** The selection condition is applied to this data object to restrict the results selected from LIKP.

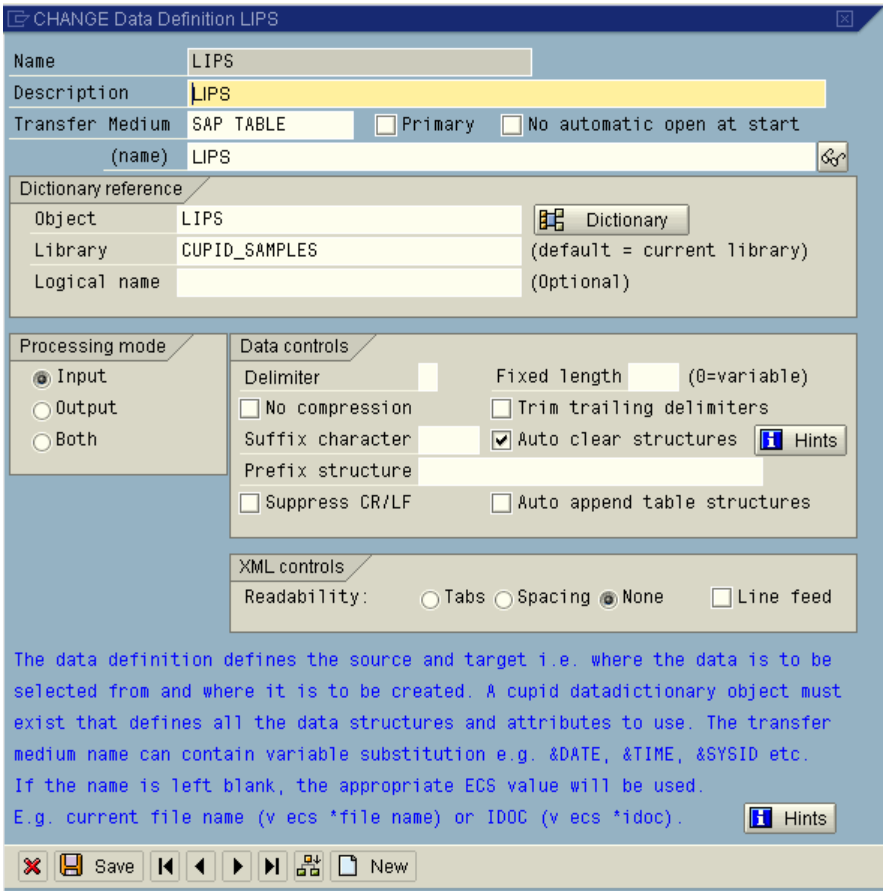


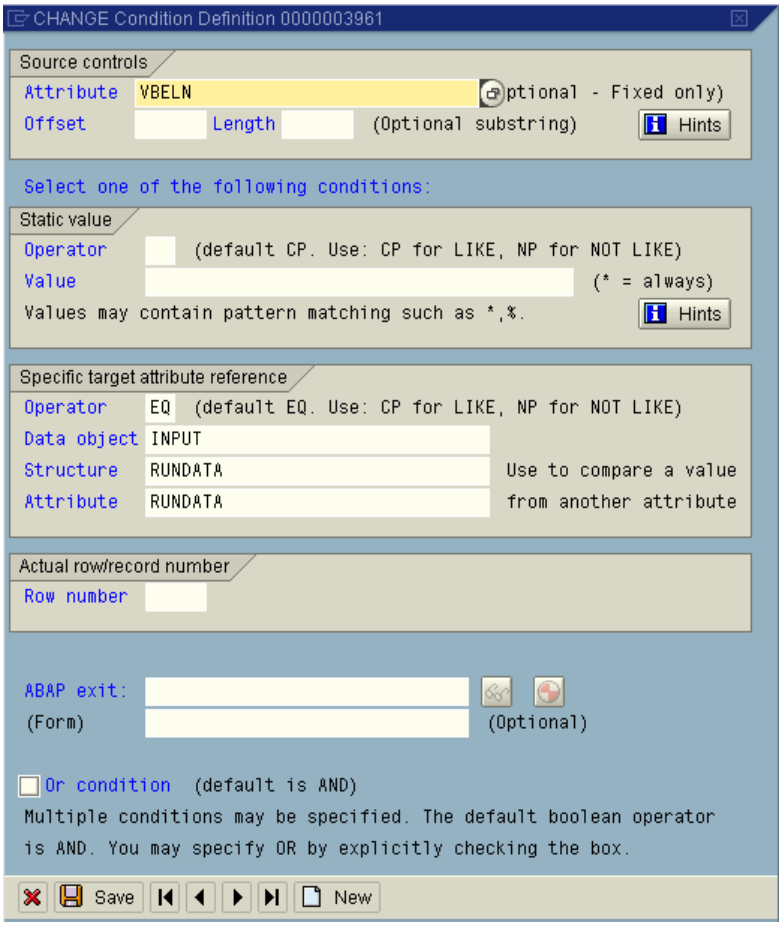


LIPS input data object

The delivery document number received from rundata is used to select the required delivery details from the SAP table LIPS.

> ***Note:*** The selection condition is applied to this data object to restrict the results selected from LIPS.

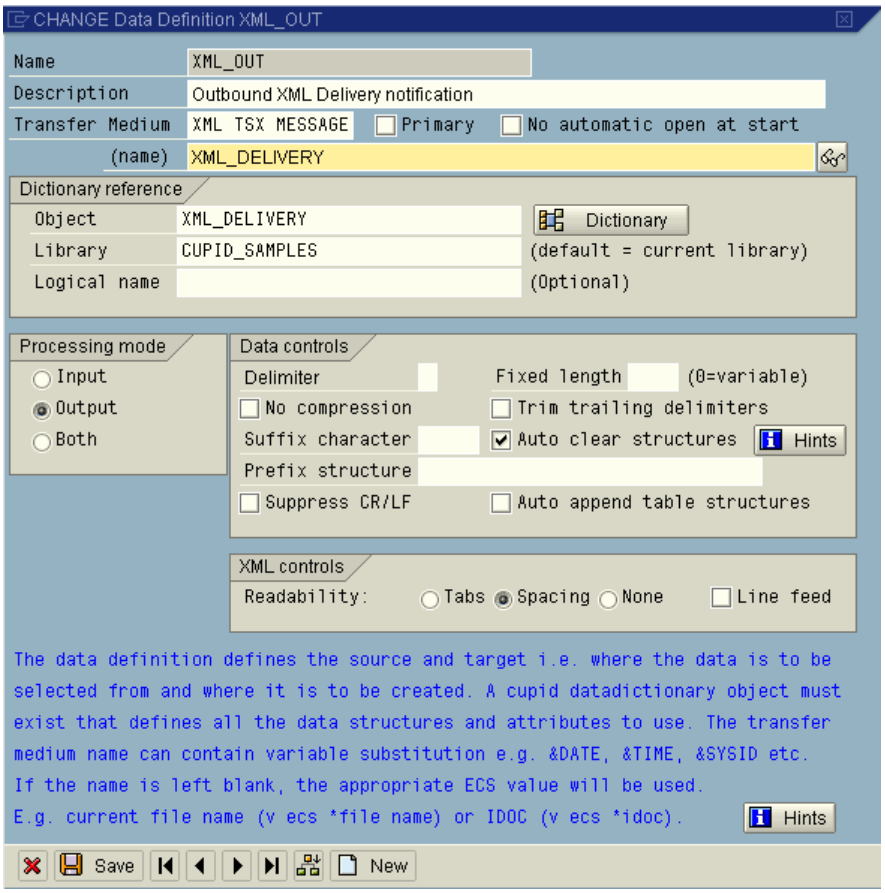




Output Data Object

The output data object uses transfer medium XML TSX MESSAGE.

> ***Note:*** The transfer name is XML_DELIVERY. In the case of TSX transfer mediums (XML, TSX MESSAGE or FILE) the name field corresponds to the TSX logical destination that is used.



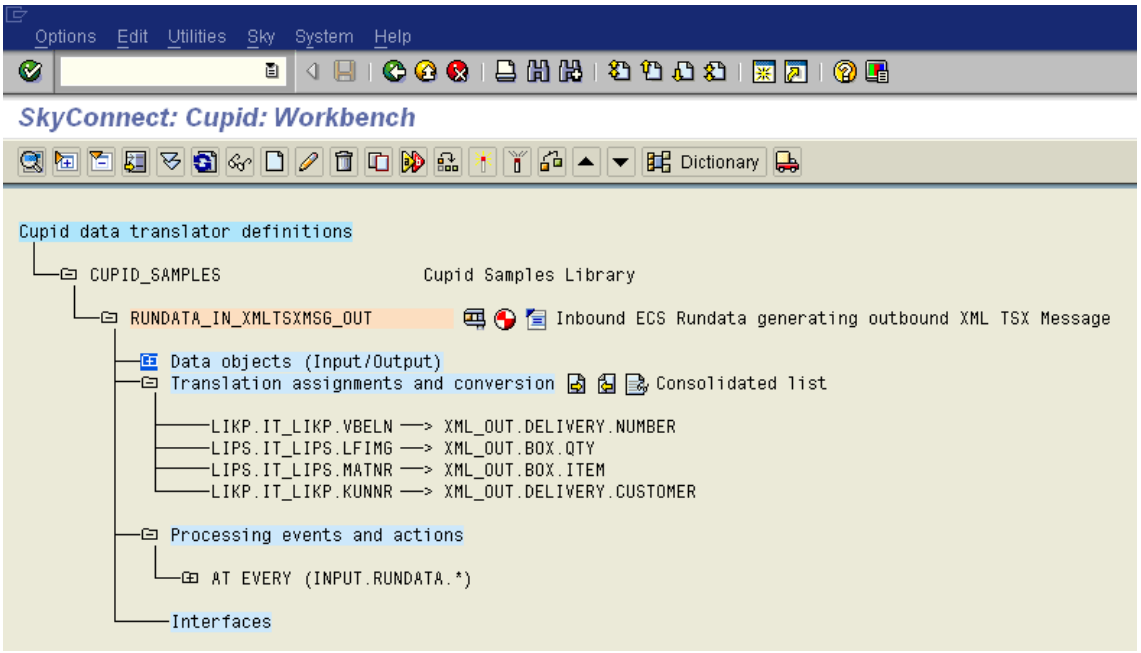
Interfaces

For this example, you do not require interfaces. The call to TSX is performed through a CREATE action in this case.

Translations

In this basic example, you need to map certain delivery details from the LIPS table into the corresponding XML tags in the output data object. You need not format as part of the translation process:

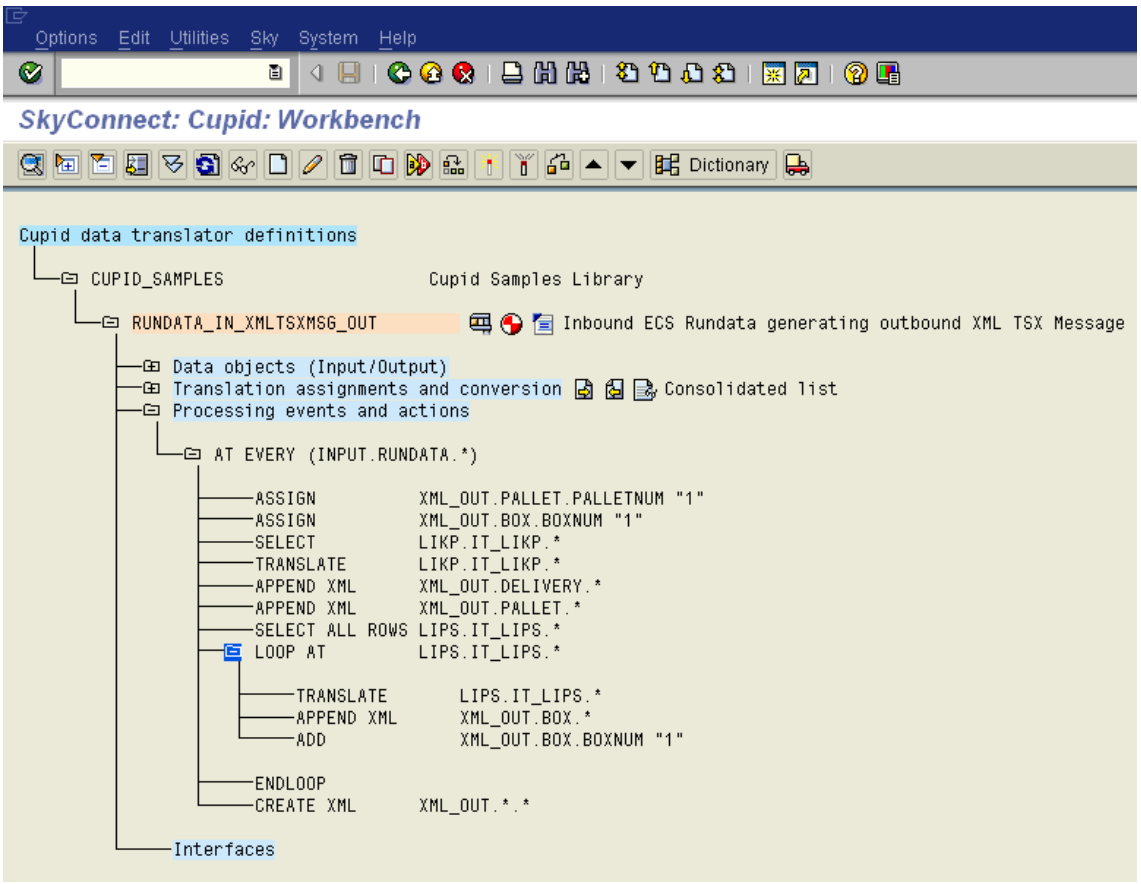
- Delivery document number (LIKP-VBELN)
- Delivery customer (LIKP-KUNNR)
- Item quantity (LIPS-LFIMG)
- Item material number (LIPS-MATNR).



Logic

The logic for this scenario is as follows:

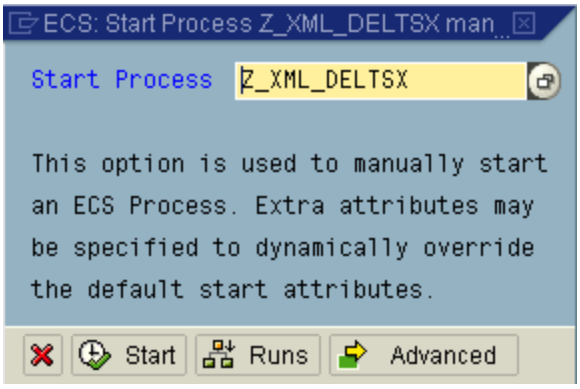
- Happen whenever a delivery number is passed in
- Configure the pallet number to 1 (in a real world scenario, this is sourced)
- Configure the box number to 1 (in a real world scenario, this is sourced)
- Select matching record from LIKP
- Map info from LIKP into XML delivery tags
- Create XML delivery header tags
- Select matching record (s) from LIPS
- Create XML tags for each line item
- Send delivery message.



Testing the Scenario

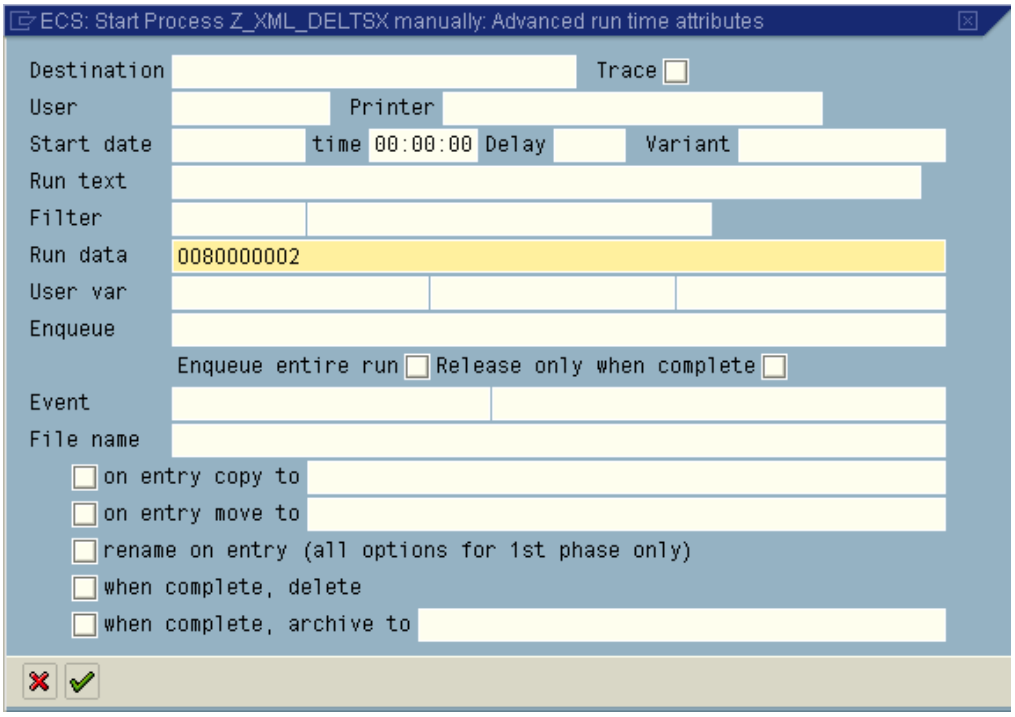
To perform a unit test of this scenario, follow these steps:

1. Compile the run time object.
2. Click the test button against the object.



3. Click the **Advanced** button.

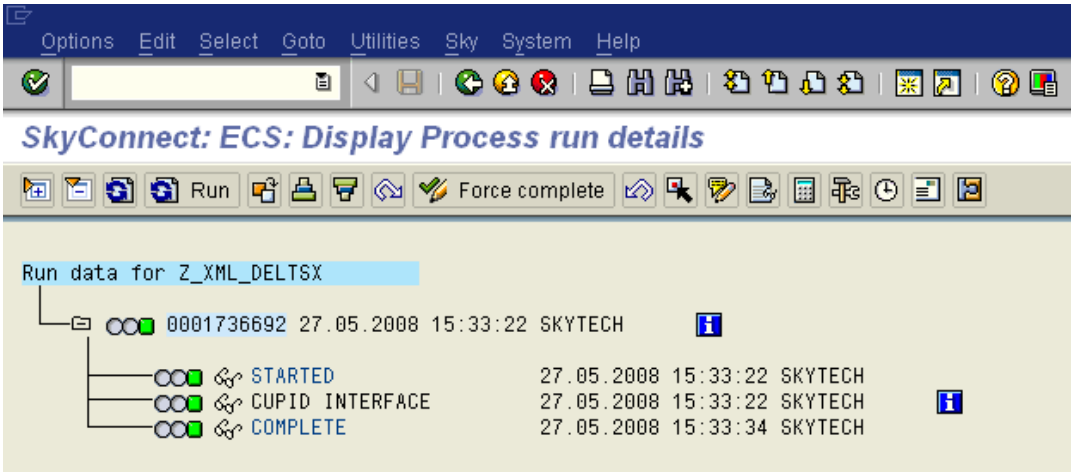
4. In the run data field, enter a delivery document number to test with.



5. Click **OK**.

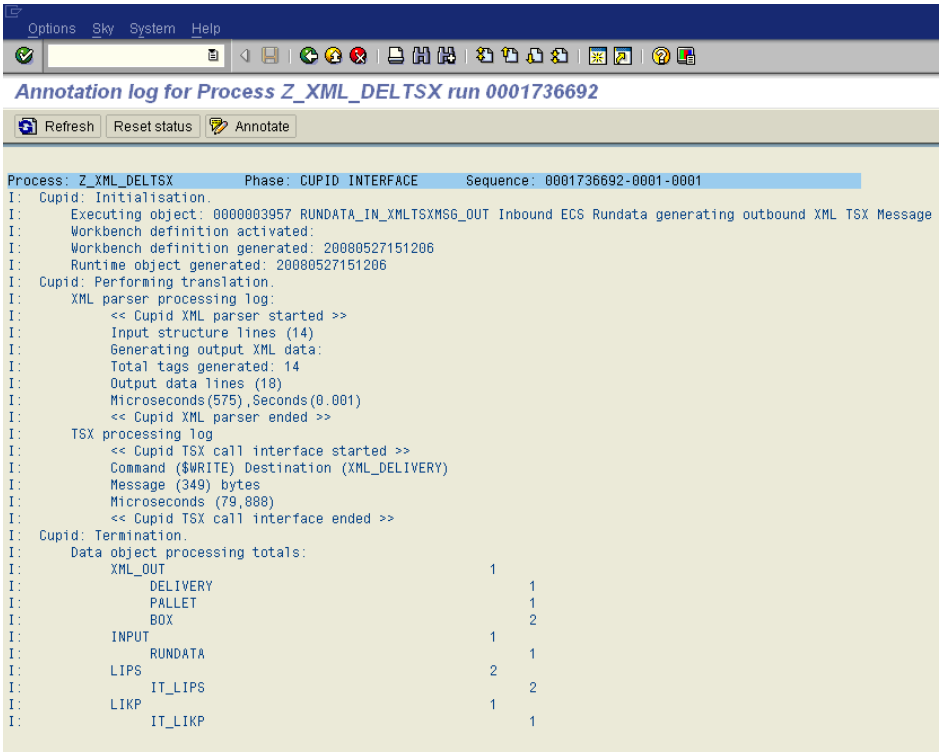
6. Click **Start** to initiate the process.

   The process now runs. After completion, it should look like the following screen grab.



The ECS annotation log shows the number of records processed for each data object.

Notice the record counts for LIKP and LIPS.



Finally, check the TSX java server log to confirm the message is received.

6.16.9.2 TSX Java Server Log

```
27/05/2008 15:33:34 INFO: [TsxServerRequestHandler1] Opened connection
to 192.168.2.64, port 5095.
```

```
27/05/2008 15:33:34 INFO: [TsxServerRequestHandler1] Wrote this
message to 192.168.2.64, port 5095: <DELIVERY>
<NUMBER>80000002</NUMBER> <DATE>20080527</DATE> <CUSTOMER>1</CUSTOMER>
<PALLET> <PALLETNUM>1</PALLETNUM> <BOX> <BOXNUM>1</BOXNUM>
<ITEM>A100</ITEM> <QTY>100</QTY> </BOX> <BOX> <BOXNUM>2</BOXNUM>
<ITEM>A100</ITEM> <QTY>100</QTY> </BOX> </PALLET></DELIVERY>
```

```
27/05/2008 15:33:34 INFO: [TsxServerRequestHandler1] Closed connection
to 192.168.2.64, port 5095.
```

```
27/05/2008 15:33:34 INFO: [TsxXmlMessagePort5095] Opened connection
from skytech64.skytechnologies.local on local XML message port 5095.
```

```
27/05/2008 15:33:34 INFO: [TsxXmlMessagePort5095Connection7] Received
this message: <DELIVERY> <NUMBER>80000002</NUMBER>
<DATE>20080527</DATE> <CUSTOMER>1</CUSTOMER> <PALLET>
<PALLETNUM>1</PALLETNUM> <BOX> <BOXNUM>1</BOXNUM> <ITEM>A100</ITEM>
<QTY>100</QTY> </BOX> <BOX> <BOXNUM>2</BOXNUM> <ITEM>A100</ITEM>
<QTY>100</QTY> </BOX> </PALLET></DELIVERY>
```

```
27/05/2008 15:33:34 INFO: [TsxXmlMessagePort5095Connection7] Closed
connection from skytech64.skytechnologies.local on local XML message
port 5095.
```

6.16.9.3 Using Cupid Elements

Cupid elements provide the means to have an extra layer between structure and attributes.This is useful in situations where fields in records have sub components.

For example, if you have records structured as follows:

Example

```
field1=abc,field2=def,field3-ghi
```

In this case the field separator is a comma.

But the fields themselves are broken into a key and value separated by another delimiter - in this case an equals sign.

A common scenario where this occurs is with EDIFACT EDI documents.

In this example, an inbound TSX message sends in a message that Cupid parses and then log the details.

Just to be different, the field separators are an ampersand. The key / value pairs are delimited by a plus sign.

Example

```
record1+abc&record2+def&record3+ghi
```

Elements Prerequisites

This example requires the following prerequisites:

- a TSX java server is configured with a message port and host connection to SAP
- the TSX SAP add-in listener is configured to start this Cupid process when a message is received.

Refer to the TSX documentation for details on configuring a TSX java server and the TSX listener.

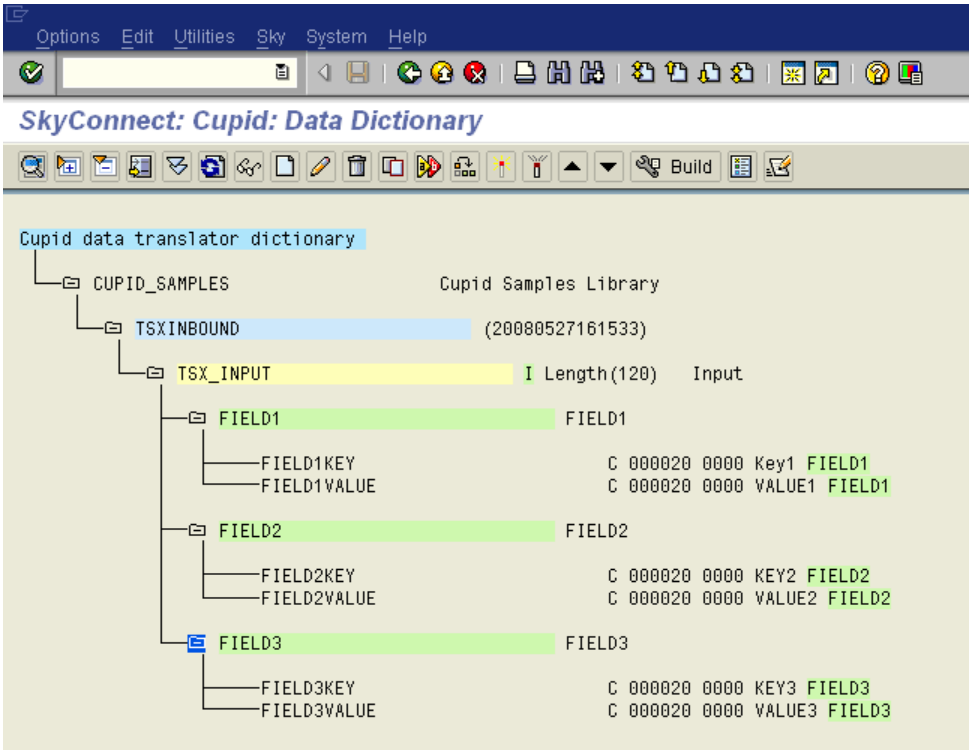
Cupid Elements Definition

Dictionary Objects

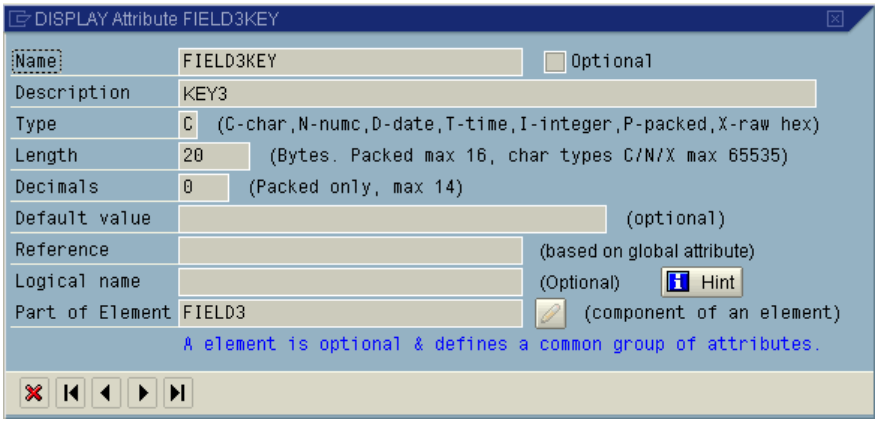
For this example, you require one Cupid dictionary object:

- object for inbound TSX message

You need to build this object manually.



Notice that each attribute is part of an element.



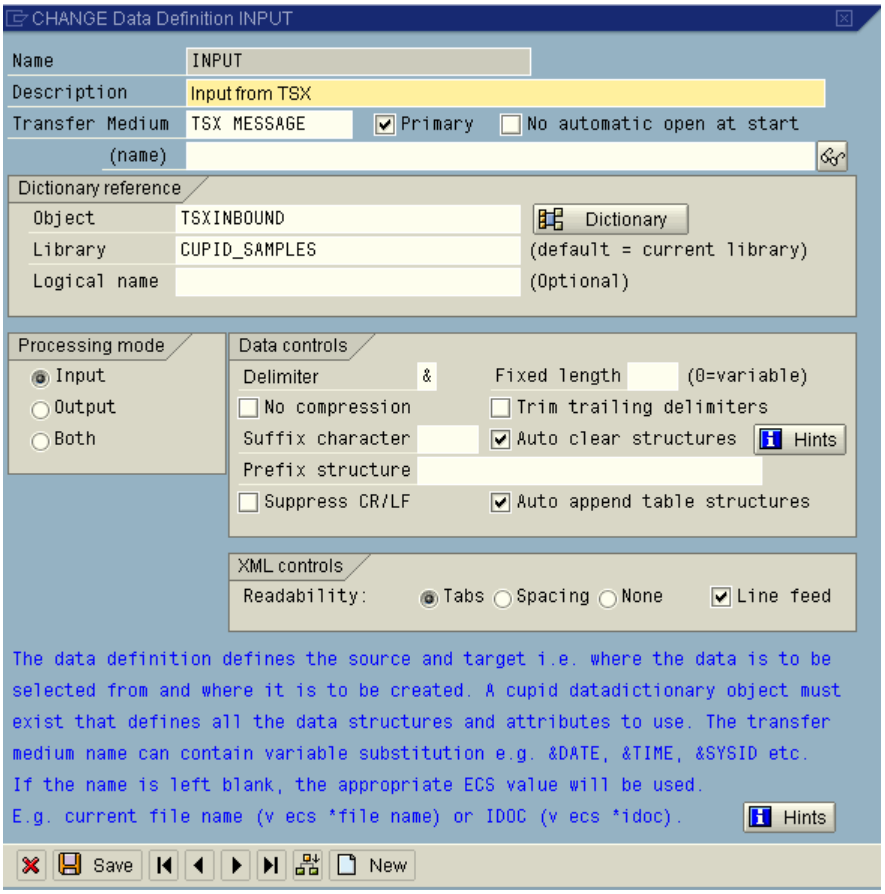
Translation Object

Nothing specific.

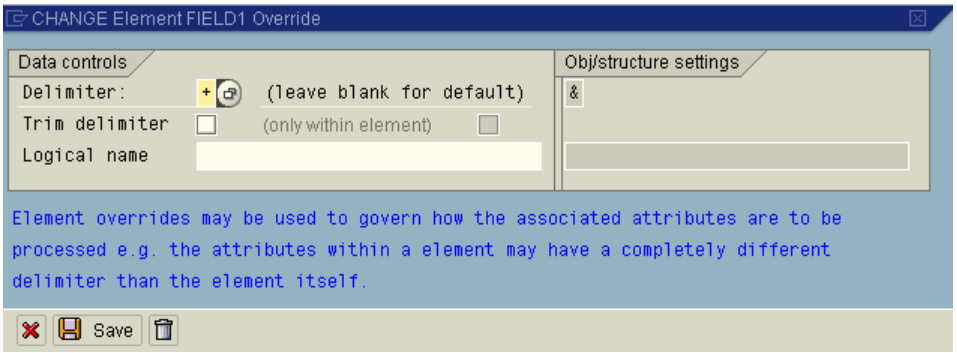
Data Objects

Primary input data object

The primary data input in this example is a TSX message. Notice that the field delimiter is set to an ampersand.



For each of the elements (FIELD1/2/3), you need to define a structure override to configure the element delimiter.



Interfaces

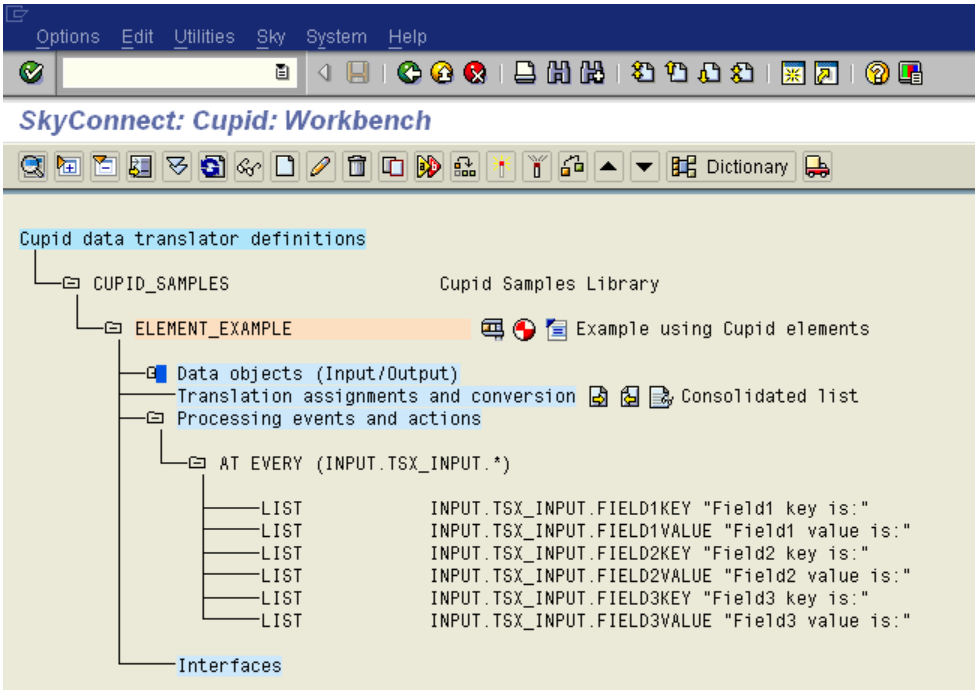
For this example, you require no interfaces.

Translations

For this example, you require no translations.

Logic

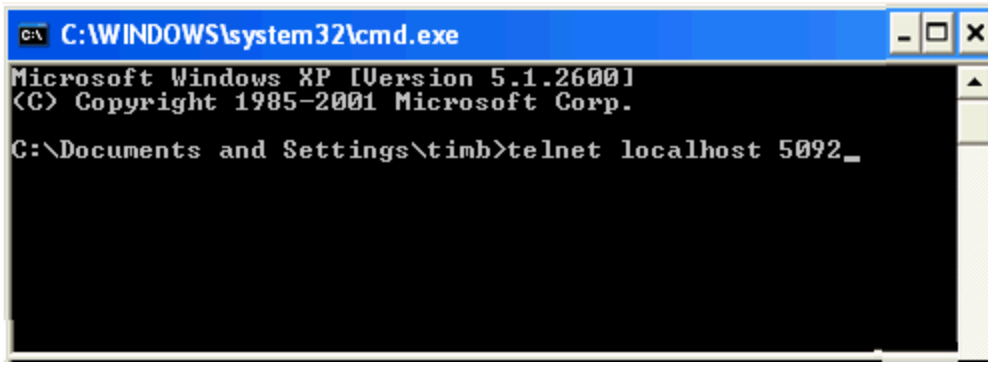
In this example, the Cupid object logs each key and value received.



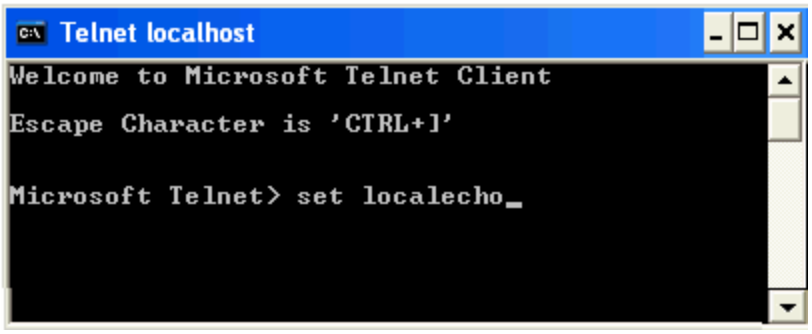
Testing the Elements Scenario

To perform a unit test of the Elements scenario, follow these steps:

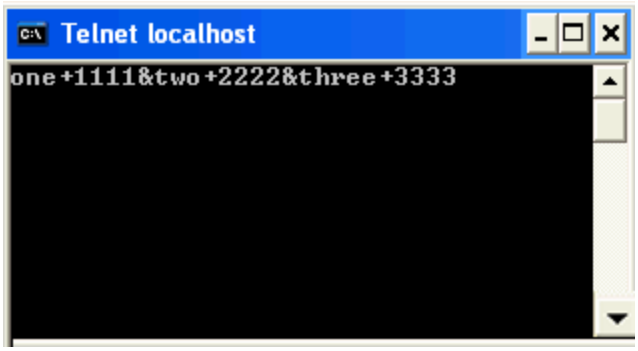
1. Compile the run time object.
2. Ensure the TSX listener is set to start the Cupid process when a message is received on the designated port.
3. Ensure your TSX java server message port is set to NotifyHost = true.
4. Start a command shell.
5. Telnet to your TSX java server on the port for the message port.



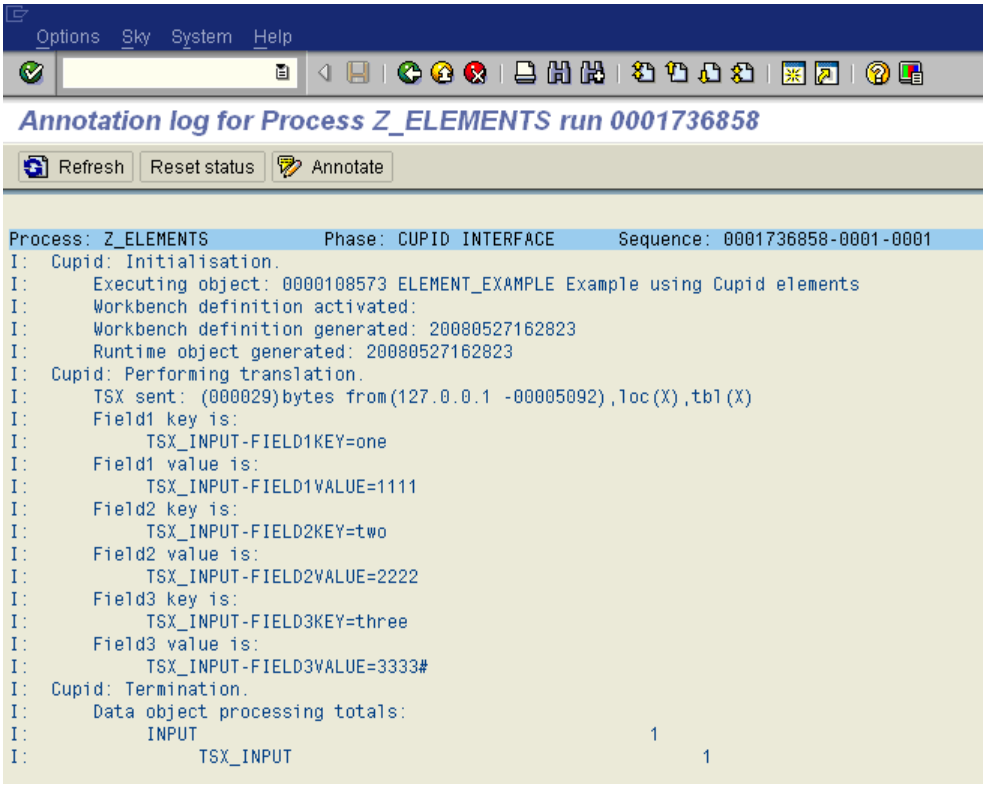
6. Ensure local echo is enabled, so you can view the input (NB on WinXP use CTRL] to enable the prompt).



7. Send in the message (NB do not forget to press **Enter** at the end)



8. You should see an ECS run for your process and the annotation log should list the three field key / value pairs.



6.16.10 TSX Glossary of Terms and Abbreviations

Term	Description
ABAP	SAP own internal programming language. ABAP allows SAP and its customers to develop and deploy SAP applications. ABAP is also portable to any computer platform that SAP supports, without change.
API	Application programming interface. A standard documented function call, object method or program that you may invoke programmatically to perform a specific function. Most software vendors provide one or more APIs in the form of a software development kit (SDK).
ECS	The Sky ECS (Event Control System), used to manage interfaces and background processing in SAP.
EDI	Electronic Data Interchange / Interface. A mechanism and standard to transmit business documents between customers and suppliers. There are two primary standards used, EDIFACT and ANSI 12.
FTP	File Transfer Protocol. A means to transmit files from one system to another.
GUI	Graphic user interface, by which the user controls navigation and input using a mouse or touch screen, for example, Microsoft windows.
SAP	An integrated, packaged ERP software solution for big businesses, incorporating Financials, Logistics and HR modules. SAP is popular amongst the tier 1/top 1000 companies. There are approximately 80,000 implementations worldwide at time of writing that equates to roughly 70% of the ERP software market.
SAPGUI	The SAP graphical user interface, specifically used to execute SAP applications.
SDK	Software development kit. These are a standard suite of common functions, provided by the software vendor, that help programmers to develop programs to communicate and interact with their product.

Term	Description
SOAP	Simple Object Access Protocol. A means to transfer business documents between systems using an XML standard for messaging.
TSX	The Sky TSX (TCPIP Sockets Exchange) component of ECS provides a means to send files and messages between SAP systems and external systems.
VAN	Value added network. A service provided usually by telecommunications companies to transmit data between customer and suppliers computer systems.
VPN	Virtual Private Network. VPN enables secure connections between two computer systems through the internet. This is a very cost effective way to transmit data.
XML	Extensible Mark up Language. XML is a language used to describe data. It is not a standard. XML is fast becoming the mechanism to describe business transactions, super ceding the more cumbersome "fixed" data records that were previously prevalent in EDI.
XSD	XML Schema Definition. XSD provides a means of defining a valid structure for a particular XML document.

Term	Description
ABAP	SAP own internal programming language. ABAP allows SAP and its customers to develop and deploy R/3 applications. ABAP is also portable to any computer platform that R/3 supports, without change.
API	Application programming interface. A standard documented function, object method or program that you may invoke programmatically to perform a specific task. Most software vendors provide one or more APIs in the form of a software development kit (SDK).
ECS	The Sky ECS (Event Control System), used to manage interfaces and background processing in SAP.

Term	Description
HTML	The basis of the World Wide Web. HTML is a Document Type Definition (DTD), or subset, of the Standard Generalized Mark-up Language (SGML). It is a common language that allows user to define graphic pages for display on Web browsers. SGML is an open document definition language much in use in the publishing industry.
HTTP	HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system. It is a generic stateless object-oriented protocol that you may use for many similar tasks such as name servers, and distributed object-oriented systems, by extending the commands, or "methods", used. A feature if HTTP is the negotiation of data representation, allowing systems to build independently of the development of new advanced representations.
Java	A highly portable programming language designed by Sun. Java is widely used on all major computing platforms including Windows, Unix, AS/400 and IBM mainframe systems. Java allows programmers to seamlessly execute their applications, irrespective of the operating system or hardware. Refer http://www.java.sun.com for more details.
JDBC	An open database connectivity standard for Java applications, similar in concept to Microsoft ODBC. Vendors provide JDBC drivers that support various levels of compliance to the JDBC standard. These JDBC drivers then handle the communication with SQL relational databases, therefore abstracting any specific knowledge of the type of database away from the application.
JVM	A Java Virtual Machine is a Java run-time environment for a specific operating system and processor. The Java application runs inside of the JVM that abstracts it from having to know specifics about the environment, it is running in. JVMs are available for all major devices and operating systems. A comprehensive list is published by Sun, the owners of the Java standard. Refer http://www.java.sun.com for details.

Term	Description
SAP	An integrated, packaged ERP software solution for big businesses, incorporating Financials, Logistics and HR modules. SAP R/3 is popular amongst the tier 1/top 1000 companies. There are approximately 36,000 implementations worldwide at time of writing that equates to roughly 70% of the ERP software market. Refer http://www.sap.com for more details.
SAPGUI	The SAP graphical user interface, specifically used to execute R/3 and R/2 applications.
SDK	Software development kit. These are a standard suite of common functions, provided by the software vendor, that help programmers to develop programs to communicate and interact with their product.
TCP/IP	Transmission Control Protocol (TCP) provides a reliable, session-based service for the delivery of sequenced data packets across an internet. The Internet Protocol (IP) provides the network layer of the TCP/IP stack. It provides the basic mechanism to route packets of data packets on the internet.

Term	Description
TCP/IP sockets and ports	Every computer and 'network' aware device has a unique IP address on the network to identify itself. Within its unique address, each computer / device may have one or more 'network aware' services that actively transmit, receive or passively listen for information on the network. A unique port number is used to discriminate between the different network services within the same computer / device. A port number (typically 16 bit) is anything from 1 - 32K where 1 - 1,024 is normally reserved by the operating system. Standard ports include 22-FTP and 23-Telnet. Most operating systems support a services file where you may document customer IP address and Port configurations. TSX IP and Port configurations are totally configurable. Refer Configuration options for more details on selecting and specifying ports. TSX transfers all messages and files using the universal TCP/IP sockets protocol. Sockets allow for the bi-directional transfer of information (messages) between two unique points on the network (similar to a telephone conversation). The sender and receiver are uniquely identified by an IP address and port, for example, 192.168.2.70 - 5082 (similar to a telephone number). You need to follow a conversation protocol between the sender and receiver to transfer data, for example, open a connection, write a message, and wait for response (similar to a telephone conversation).
TSX	The Sky Technologies TCP/IP Sockets Exchange software enables applications to send messages, files and interface with devices over a TCP/IP network. It is essentially a messaging layer that can enable direct TCP/IP calls from SAP R/3.

6.17 Integration Utilities

ECS provides some light weight integration utilities to help interface with external sub-systems.

6.17.1 Email Extension

The ECS Email Extension enables you to send files, spooled output and even logs from within SAP to an external recipient through an ECS process / phase. This extension also supports the receipt of data through email into SAP. It uses SAP SAP Office API, to send and retrieve Email and any other attachments, from the External Email Server, such as Exchange. This technical guide describes the ECS Email extension architecture and how to configure it. It is intended for IT staffs who wish to implement or support outbound and inbound email interfaces. It is assumed that the reader already has knowledge of ECS.

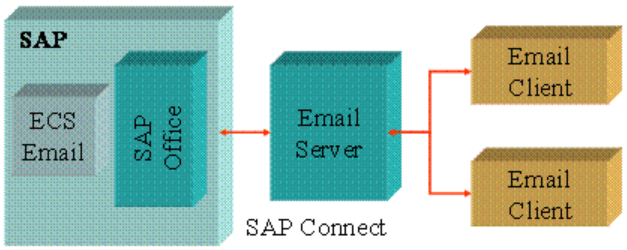
6.17.1.1 Email Extension Overview

The ECS Email enables a user to configure a destination to which you can send data from within SAP to an external email recipient by an ECS process / phase. This data might include file, spooled output or an ECS log. The recipients may be internal company users or alternatively, external users to the main SAP system (for example, Customers and suppliers). The functionality also supports the receiving of data through email into SAP, that is, attachments. The ECS Email extension utilizes the services of SAP's SAP Office to deliver email to its intended recipients.

The Email extension is capable of taking file and spool output from other phases within an ECS process, performing the required format conversions and then passing the results to SAP office to perform the delivery automatically. You can automatically distribute Standard SAP or customized reports, processed files, and error reports from ECS to staff and customers through Email as attachments, without any manual intervention required. If all intended recipients are internal SAP Office users, then no additional configuration beyond a standard SAP install is required. However, if some or all of the intended email recipients are accessible through an external email system, such as MS Exchange, Lotus Notes, or the Internet, then you must set up the SAP Connect email configuration. See your SAP Basis administrator if this functionality is required.

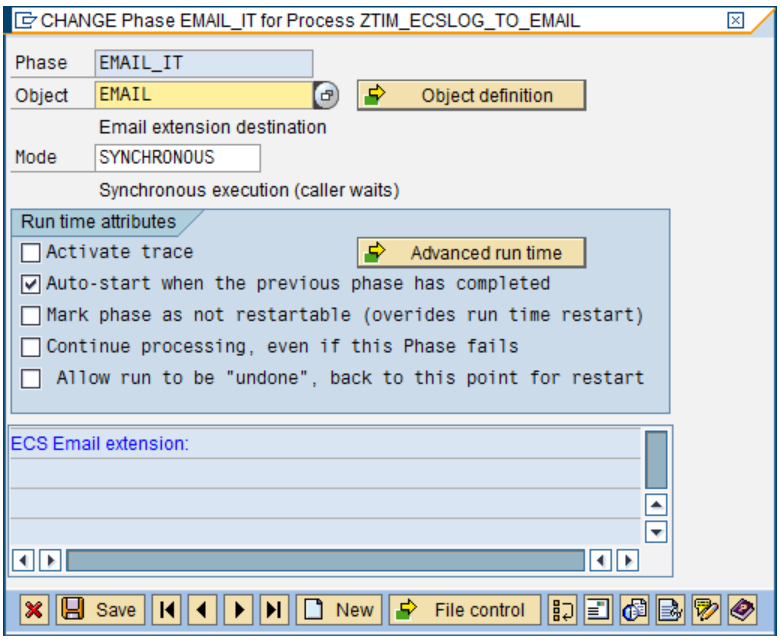
Architecture

The ECS Email extension uses SAP standard APIs to send and retrieve email and any attachments. You use ECS interface management to schedule Email processing.



6.17.1.2 Outbound Email Configuration

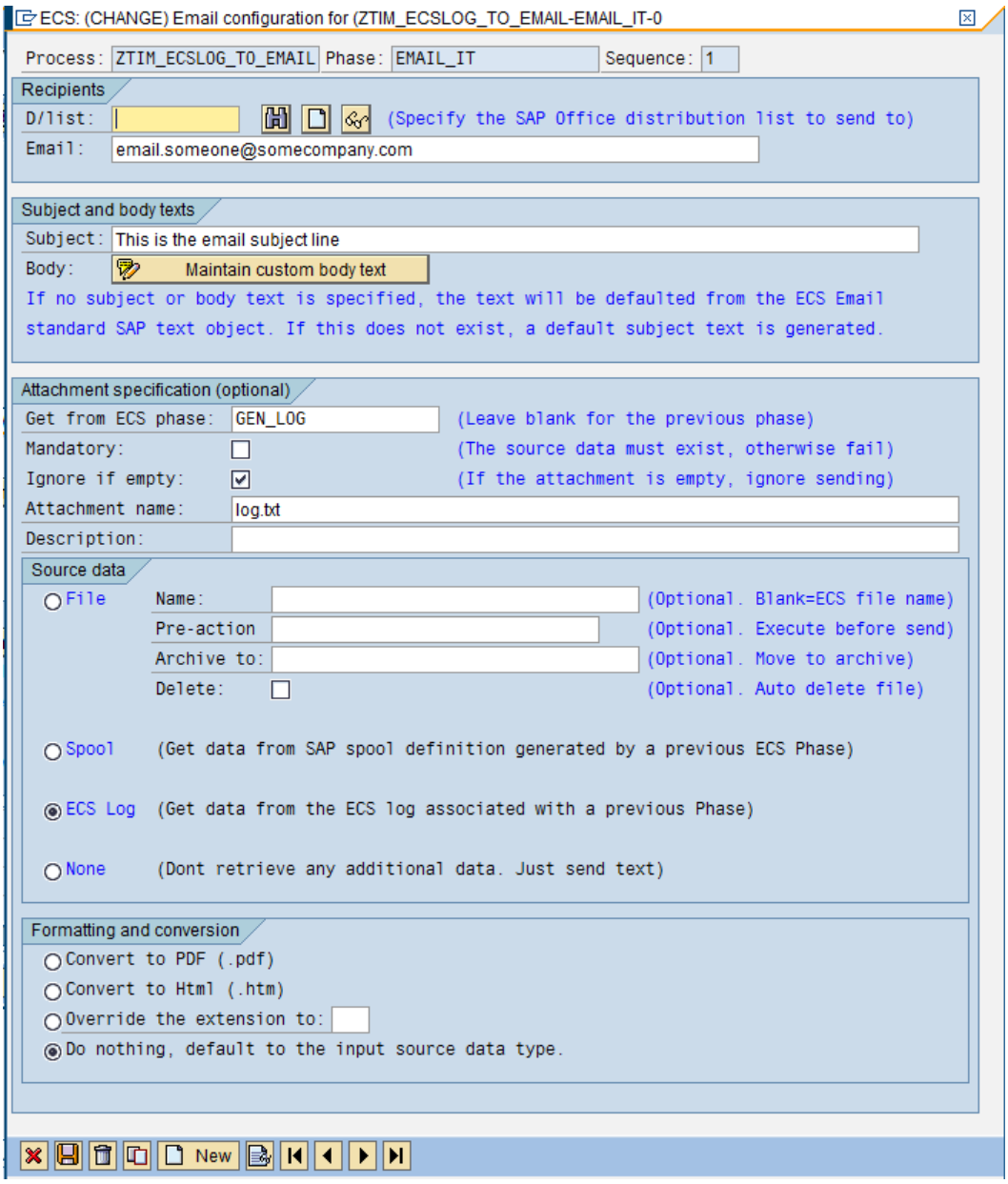
The ECS Email extension is embedded into the ECS workbench framework, that is, you create a phase definition that emails data created from either a previous phase (most common), or an existing file created by a totally separate process. As well as this definition, you may override various options at run time through a previous phase, for example, subject, body, list of email addresses.



Specify an ECS object type of EMAIL. You may specify any of the execution modes. If the EMAIL object type is not available, then the ECS Email extension is not properly installed. Contact your support or refer to the Sky Product Installation guide for more details.

Click the **Object Definition** button to configure the Email attributes. The following screen appears.

> ***Note:*** You can configure multiple email definitions for a single phase. You scroll between them using the left and right arrows.



The outbound Email configuration prompt allows you to specify:

- To whom you need to send the email. This is done using the standard SAP office distribution lists. You can find, create or display SAP office distribution lists from here.
- Specify the data source, that is, any attachment to send with the Email and the special processing attributes and format attributes of the data.

- Specify the Email subject and body texts. If nothing is configured here, text is defaulted from a standard SAP text object called "YECS_EMAIL_STANDARD_TEXT". If this does not exist, a default subject, "The ECS Email extension automatically generated this Email" appears.

Note: You may override many of these options at run time through a previous ECS phase.

Field	Description
D/list	SAP Office distribution list to send email to. Refer using SAP office distribution lists for more information.
Email	A specific email address to send the email to
Subject	Email subject to use
Body	Email body content
Get from ECS phase	If you have to send an attachment with the email the phase from the current ECS process to derive the attachment from
Mandatory	The attachment must exist. If it does not the ECS process fails
Ignore if empty	If the attachment is empty, then don't send the email
Attachment name	The name to apply to the attachment. This is only relevant for spool and ECS log attachments. If the attachment source is a file then its filename is used.
Description	A description field.

Field	Description
Source	• File - get the attachment from the ECS file attachment variable • Spool - generate an attachment from SAP spool • ECS Log - generate an attachment from an ECS annotation log • None - no attachment
Formatting and Conversion	These options currently only apply to attachments derived from spool

Using SAP Distribution Lists

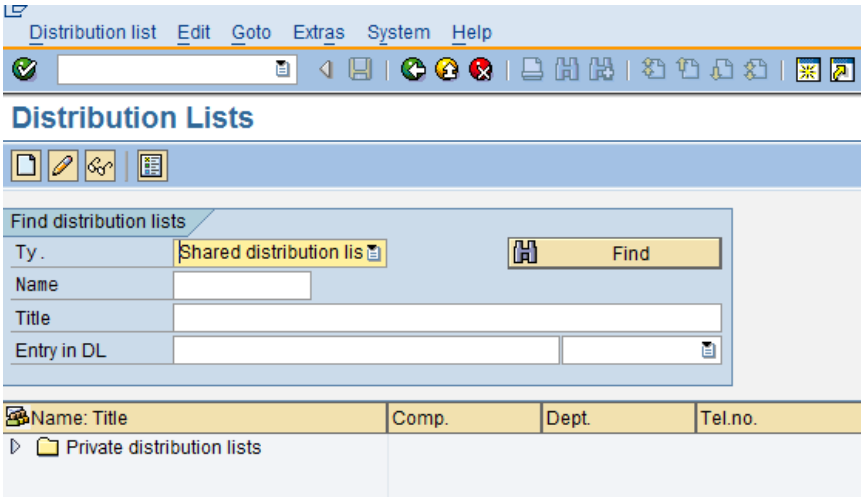
The distribution list contains those people for whom you need to send data during the execution of the given process / phase. A distribution list can contain internal users, external addresses, distribution lists and addresses and internal users. The Display DL facility is available from the Create, Change and Display destination details screens, the Search DL and Create DL screens are available from the Change Destination Detail screen.

- Searching for distribution lists
- Creating a distribution list
- Displaying distribution list members

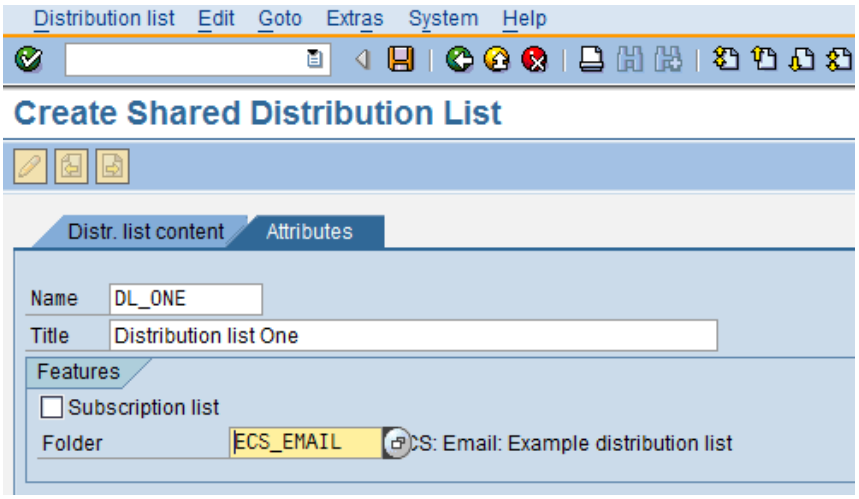
> ***Note:*** The distribution lists used by ECS Email are standard SAP Office public distribution lists. To be able to use existing SAP distribution lists, simply check they are designated public and then you can use the lists immediately.

Creating a Distribution List

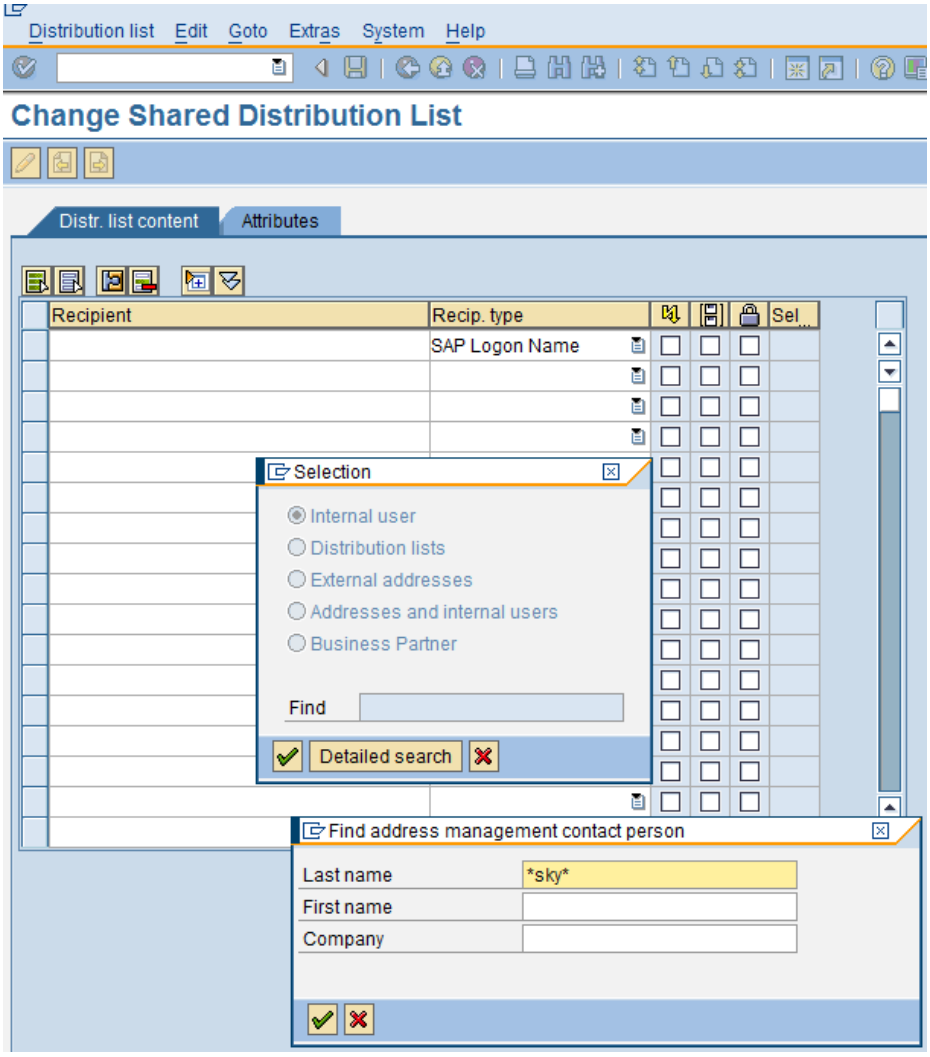
1. Click the **Create DL** button.
   The following screen appears:



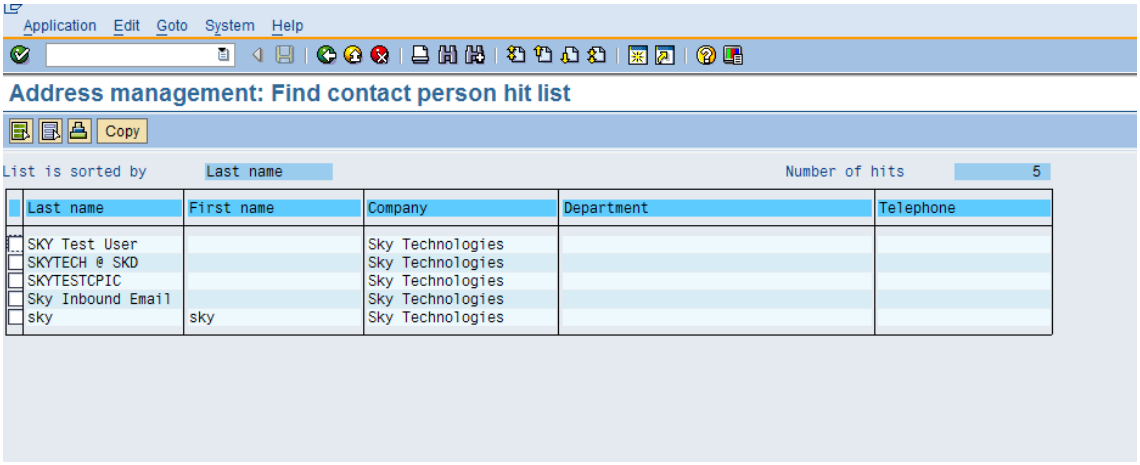
2. Enter the required details and click the **Create** icon from the application toolbar.



3. Enter the details as required. Click **Save**.
   The following screen appears allowing to add members to the newly created Distribution list:



4. Select the drop-down option for the member entry pops up the selection box shown below. To refine the search selection criteria, select the type of member entry required. Further refinement of the search selection criteria is available from the **Find address management contact person** dialog. Wild cards are valid; no entry in these fields displays all available members. Results from the search appear for selection.

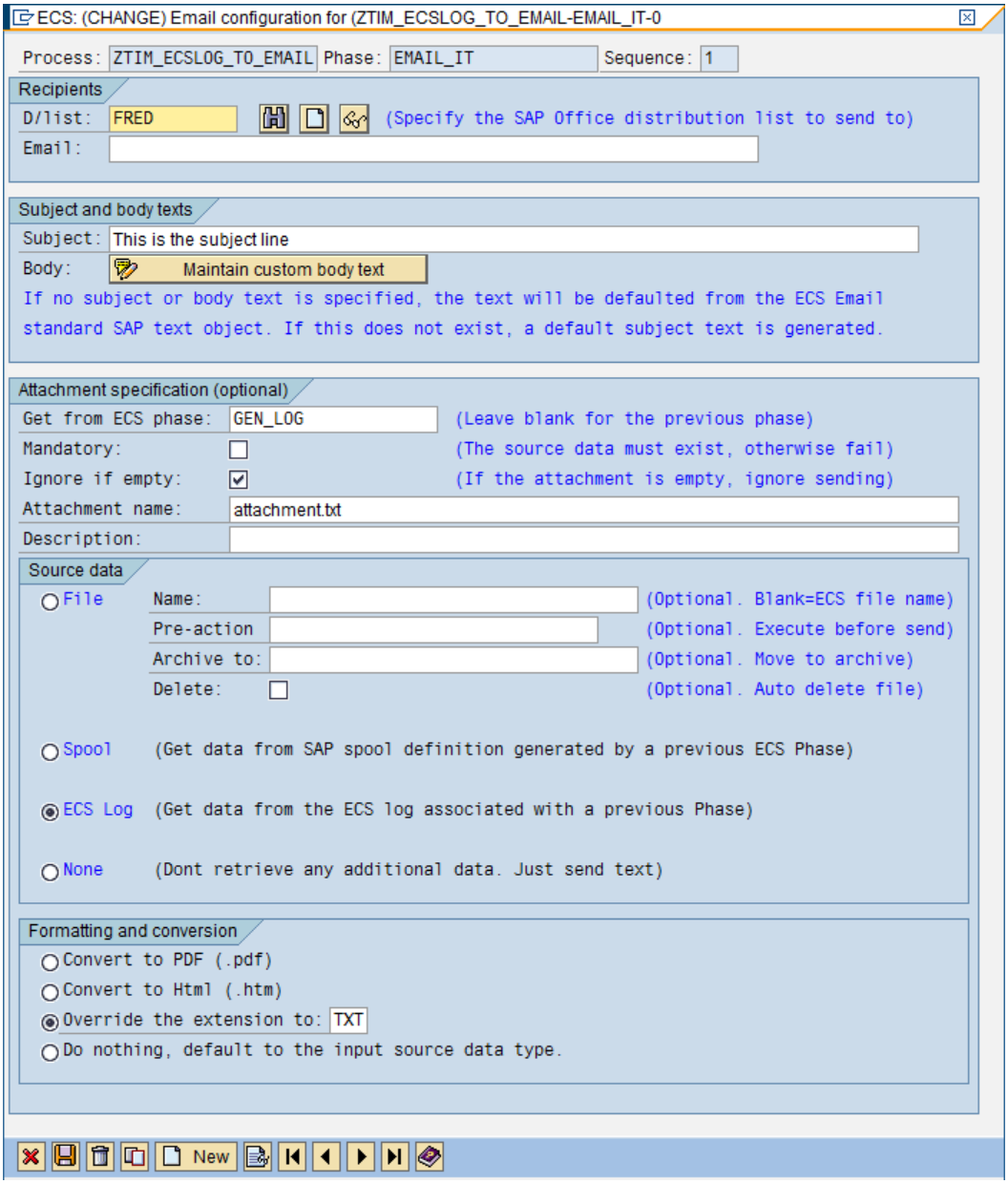


5. Select those members required for the distribution list by selecting the check box and then select **Copy** from the application toolbar.

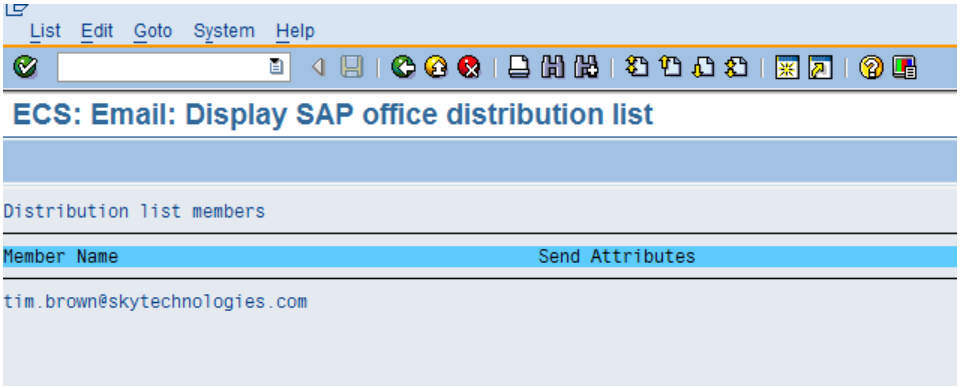
Click here to go back to the top.

Displaying Distribution List Members

To display the members of the distribution list, ensure the distribution list field has a valid value, then click the **Display DL** button.



The members of the distribution list that are entered into the Distribution list field appear.



The names of the members together with the send attributes appropriate for each member are listed.

Click here to go back to the top.

Outbound Email Overrides

You may program overrides to the email attributes in the phase prior to the email phase for the subject, email list, body text, and distribution list.

ECS Run Data

You may configure the following comma delimited overrides in the ECS run data (ecs_set_run_data) using the following options:

- DLIST = New distribution list
- EMAIL= New internet email address
- SUBJ = New subject
- ANAME= New attachment name
- ATYPE= New attachment type.

ECS Export Data

You may export the following tables and variables (ecs_export_data):

- IT_EMAIL_ADDRESS (char 128 internal table) - list of email addresses
- V_EMAIL_SUBJECT (char 80) - subject text
- IT_EMAIL_TEXT (char 128 internal table) - body text

> ***Note:*** These overrides are passed as ECS export parameters and need to be the same name and data type as specified.

6.17.1.3 Inbound Email Configuration

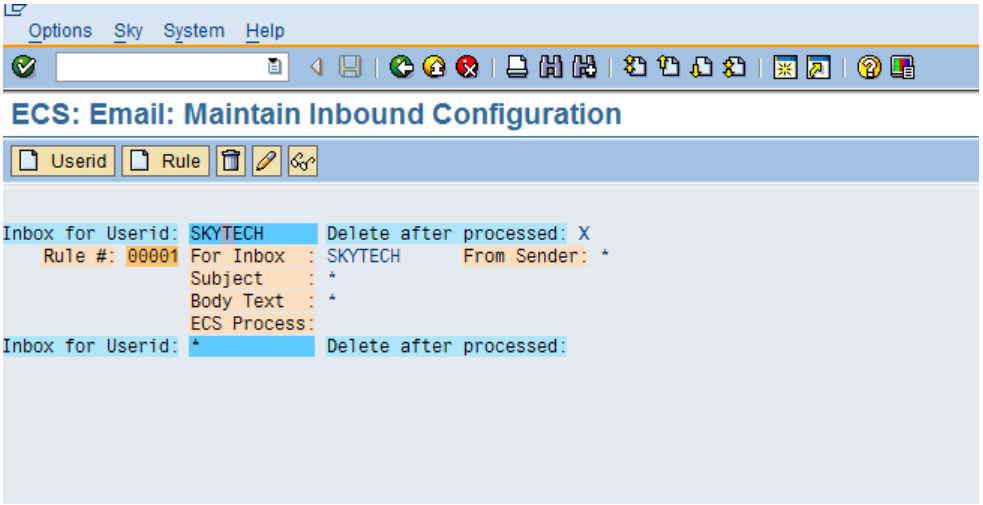
- SAP Office Inbox Monitor
- Configuring Inbox Monitoring Rules
- Custom Interface for Email Processing

SAP Office Inbox Monitor

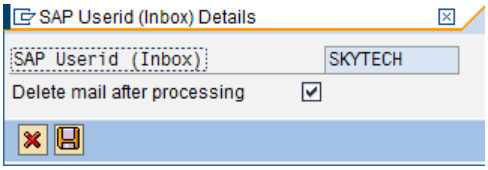
The inbound email functionality monitors a standard SAP inbox for email definitions and then starts the configured ECS Process, passing any attachments as files. The ECS_INBOX Schedule and Process definitions are automatically created by the ECS Email extension check installation procedure as part of the installation process. You must activate the ECS_INBOX schedule definition to start the monitoring process. You may customize the schedule as required.

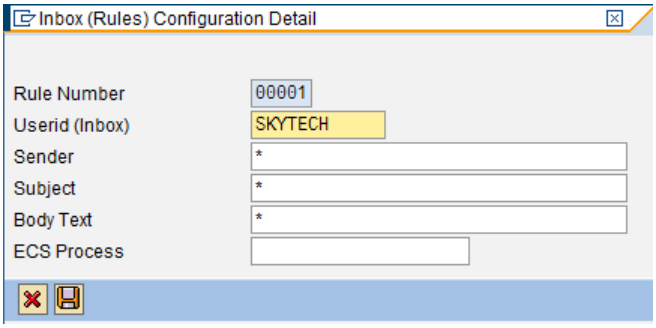
Configuring Inbox Monitoring Rules

You maintain the inbound email configuration through a supplied configuration program. Execute this program through transaction /n/SKY/YEML or the shortcut YEML (if configured) and the following screens appear:



You maintain the inbox SAP office User IDs and rules by clicking the create, change and delete icons on the application toolbar. Existing definitions are expanded by double-clicking the user id level, that is, to show the configured rules.





Custom Interface for Email Processing

Each rule defines a wild card selection criterion that is to be applied to the sender, subject and body text. This allows the inbound emails to be effectively filtered. If the criterion matches, then a ECS Process to start is specified. The ECS process is passed the email data through files, a list of attachments and their associated files is passed in internal table IT_ATTACHMENTS. The email body text is passed as attachment "EMAIL_TEXT" in both the "it_attachment" internal table and as the current file. A sample receiving ABAP program, YECS0054, is provided as part of the installation that explains how to receive and process the data.

Example Code

This example ABAP program is executed by a ECS Process that is initiated by an inbound email definition. It basically accepts the attachment data and lists the contents to the ECS log.

```
* ---------------------------------------------------------------------
----
* This sample program demonstrates how to receive inbound email data
* from the ECS Email Inbound server.
* ---------------------------------------------------------------------
----
REPORT  YECS9200.
```

```
* Data declarations.
INCLUDE YECSSDK.
DATA: IT_ATTACHMENT       LIKE YECS_EMAIL_ATTACHMENT
                          OCCURS 0 WITH HEADER LINE,
      V_INPUT_RECORD(255) TYPE C,
      V_LINES             TYPE I.

* ---------------------------------------------------------------------
----
* Main
* ---------------------------------------------------------------------
----
* Get ECS parameters
  ECS_IMPORT_PARAMETERS.

* Import attachment cross reference.
  ECS_IMPORT_DATA IT_ATTACHMENT.
  ECS_APPEND_LOG_ULINE.
  ECS_APPEND_LOG 'I' '1' 'Sample Inbound Email program'.

* Annotate message text to the ECS log.
  ECS_APPEND_LOG_ULINE.
  ECS_APPEND_LOG 'I' '2' 'Message text (1st 50 lines)'.

  ECS_OPEN_FILE_FOR_INPUT.
  ECS_READ_FILE V_INPUT_RECORD.
  WHILE V_ECS_*RC = 0.
        IF SY-INDEX > 50.
           ECS_APPEND_LOG 'W' '3' 'Truncated at 50 lines ...'.
           EXIT.
        ENDIF.
        ECS_APPEND_LOG 'I' '3' V_INPUT_RECORD.
        ECS_READ_FILE V_INPUT_RECORD.
```

```
  ENDWHILE.

* Annotate attachment data to the ECS log.
  ECS_APPEND_LOG_ULINE.
  ECS_APPEND_LOG 'I' '2' 'Attachment data (1st 50 lines of each)'.

  DESCRIBE TABLE IT_ATTACHMENT LINES V_LINES.
  IF V_LINES = 0.
     ECS_APPEND_LOG 'I' '3' 'There are no attachments'.
  ENDIF.

  LOOP AT IT_ATTACHMENT.
       ECS_APPEND_LOG_ULINE.
       ECS_SET_FILE_NAME IT_ATTACHMENT-ATTACHMENT_FILE.
       ECS_APPEND_LOG 'I' '3' IT_ATTACHMENT-ATTACHMENT_NAME.

       ECS_OPEN_FILE_FOR_INPUT.
       ECS_READ_FILE V_INPUT_RECORD.
       WHILE V_ECS_*RC = 0.
             IF SY-INDEX > 50.
                ECS_APPEND_LOG 'W' '4' 'Truncated at 50 lines ...'.
                EXIT.
             ENDIF.
             ECS_APPEND_LOG 'I' '4' V_INPUT_RECORD.
             ECS_READ_FILE V_INPUT_RECORD.
       ENDWHILE.
  ENDLOOP.
  ECS_APPEND_LOG_ULINE.

  ECS_COMMIT_LOG.
```

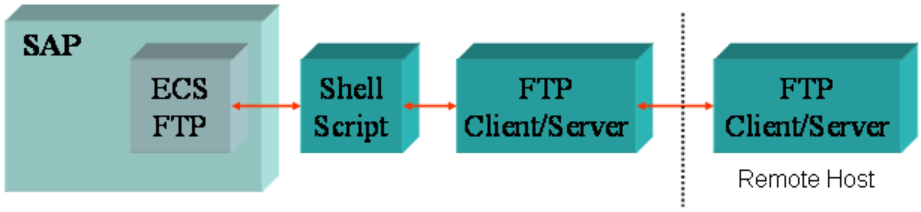
6.17.2 FTP Extension

The SkyConnect FTP Extension provides outbound and inbound file transfer directives using the file transfer protocol, between SAP and external systems. The FTP Extension provides end to end transaction management, high visibility processing, and is completely integrated with the ECS. Key features include:

- Bi-directional file transfer interface between SAP and external recipients
- Ability to easily distribute files to multiple destinations using the standard FTP protocol
- Seamlessly integrates with SkyConnect interface management.

6.17.2.1 FTP Extension Architecture

The FTP processing is invoked as an ECS phase. Configuration determines the destination to either PUT or GET files from. The ECS FTP extension dynamically constructs a FTP command script that is then piped to an external command shell that executes it. The ECS FTP extension supplies all the components required for either a Unix or Windows system.



You use the ECS FTP Server Extension primarily for the transfer of files in background processes. FTP is a good method of transferring files in the background (batch), as other methods of file transfer, such as local PC upload and download (WS_UPLOAD and WS_DOWNLOAD) are not available to batch jobs. In order to use the FTP server, both the local and remote systems must support the FTP protocol to allow communications. In particular, the SAP (local) host system must have an FTP Client program available, and the remote host must have an FTP Server running. It is also possible to transfer files to and from a third host, if the directory / file system of the third host is mounted or otherwise available to the FTP server of the second host. For example, to send a series of files created under ECS to a remote system supporting FTP configure the destination details and enable an FTP Phase. ECS can then manage the transfer of the files to or from the remote system for you. If it was required to have files sent to a drive on a LAN, the LAN administrator may be able to allow SAP to FTP the files to the server by installing an FTP server on the destination host, or allowing access to the LAN directories from another FTP server. The benefits of using the ECS FTP server extension are the same as those benefits of using other ECS processing techniques, for example, audit trails, restart/recovery, notifications.

6.17.2.2 FTP Extension Configuration

The SkyConnect FTP extension is supplied with all the software required to make use of the FTP process including:

- The ECS FTP SAP add-in component
- ECS FTP Host script samples

Configuration Steps

The configuration steps required are as follows:

- Create a Logical Command
- Create a Command Script
- Define FTP Destinations
- Configure an ECS Phase to use FTP

Creating the SAP FTP Logical Command

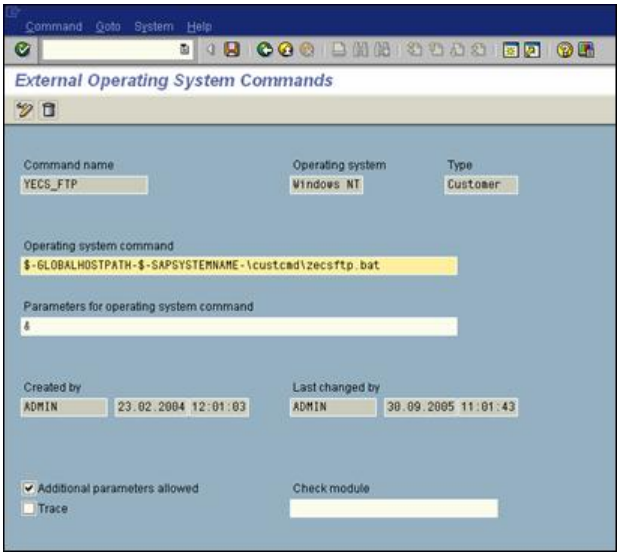
The FTP program operates by calling a logical command that relates to a host operating system command or script that performs the FTP process. This logical command must call a simple shell file for technical reasons. The FTP external command is defined in SAP using transaction SM69.

The command YECS_FTP is automatically created by the FTP installation procedure, but may need to modify according to your environment.

> ***Note:*** You need to implement a "bat" command file in a Windows environment. See the appendix for details. Sky can also provide this for you if required.

Name	OS	Command	Parameters
YECS_ FTP	Unix	ftp	?
YECS_ FTP	Windows	$-GLOBALHOSTPATH-$-SAPSYSTEMNAME- \custcmd\zecsftp.bat	?

Example Windows Command



The SAP FTP Command Script

The FTP transmission is driven by the use of an FTP script that is stored as a Sapscript standard text. You need to create this script as follows:

To access Sapscript maintenance, use the menu path **Tools** > **Word Processing** > **Standard Text**, or transaction SO10.

1. Create a standard text YECS_FTP_COMMAND_FILE, text id ST
2. Copy and paste the text from below into the editor
3. Save your changes.

> ***Note:*** Consider the following:
>
> 1. Sapscripts are client dependent and you need to maintain for each client that FTP is to use with.
> 2. Use the text editor option within SO10 when editing the FTP script. To change to the text editor use the menu path **Goto** > **Change Editor**. If the editor looks WYSIWYG, then it is the wrong editor.
> 3. Case may be important for some operating systems (for example, UNIX), both locally and on the destination machine.
> 4. Use lower case for parameters.

Example YECS_FTP_COMMAND_FILE standard text

```
/* This script is to be piped to FTP, ftp -v -n < thisfile
/* Note the -n switch to turn off automatic logon
/* REMEMBER, case is important in some OS
/* ALSO ALWAYS use lower case for the parameters!
/* ALSO any other text on the &_complete line will not be
/* passed to the FTP program, it is interpreted by ABAP
```

```
/* ALSO Use the text editor (not the SapScript editor) to change
this
/* script, the SapScript editor puts extra controls in <(> which
make
/* the text indecipherable!
* prompt "Toggle Interactive
* open &rem_mach "Open the machine
* user &rem_user &rem_pass "Log On
* &tf_ttype "Set Transfer Mode
* &tf_fmode &loclfile &rem_dir&rem_file "Transfer
* bye "Log off remote system
* &_complete=226 Transfer "Set end of transfer string
* &_complete=226 File transfer "Set end of transfer string
```

Comments in the Script

Lines with the tag column (left column of sapscript editor) set to /* or any text inserted after quotation marks is regarded as comment lines and are not sent to the FTP program. There currently exists no way of placing quotation marks " in the script.

Parameters in the Script

Any field within the script beginning with & is interpreted as referencing a field from either the destination table (YECS_FTPDN) or the process table (YECS_FTPPR). To use the values of any of these table fields within the script, reference this value by inserting an ampersand followed by the variable name, for example, &rem_file. At run time, &rem_file contains the value of rem_file held within the corresponding table.

The special field &_complete is used to build a list of texts signifying that the transfer is successful. You can define multiple success types and they are cumulative, that is, if ANY are found, then success is assumed.

> ***Note:*** Any commands on the &_complete line are not actually passed to the FTP program.

Common Error Causes and Traps

1. The following are things to be wary of as they can cause unexpected problems: &_complete should always have more than just an FTP code. For example, Use "226 Transfer Complete" instead of just "226". It is possible that "226" turns up elsewhere such as in "226 bytes received".
2. Change directory (cd or the OS equivalent) can sometimes fail and in turn cause the file to deposit in the wrong directory.

> ***Note:*** In the above script, the command "put" qualifies the file. By performing cd and then an unqualified "put" (useful in the event of a failed cd, or when using commands that do not allow qualified file names), the file is still transmitted but may be placed into the incorrect directory.

Defining FTP Destinations

ECS includes functionality that allows to transmit the files to external systems through standard FTP (File Transfer Protocol). You can configure ECS so as to allow for the data exported from the originating system to validate prior to transmission, thus ensuring that any data imported into the destination production systems are valid. Essentially the configuration for these processes involves:

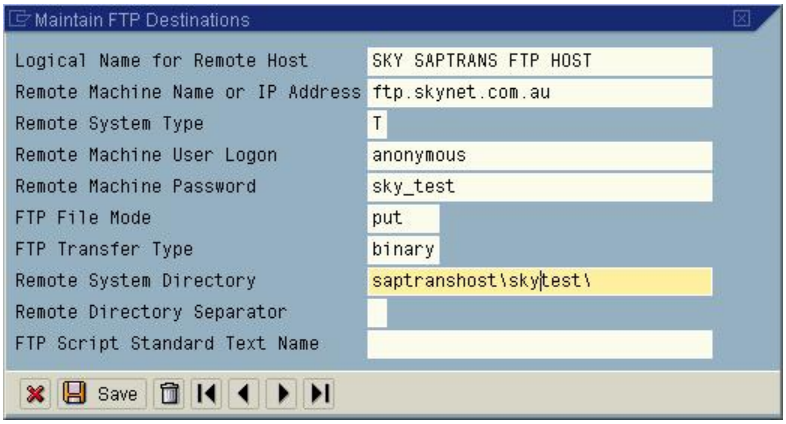
- The identification of the remote destination system together with relevant transmission information, login details, file destination details and an FTP script to use for the transmission.
- The identification of the process requirements relating to file to transfer, the transfer type and the FTP script to use for the transmission.
- The building of an FTP script that is to run at transmission execution

> ***Note:*** It is recommended that you use a desktop or other FTP Client, and manually attempt the FTP transmission, prior to configuring the Destination and Process records. This allows you to gather all the required information, and test various items, such as directory and file permissions.

The ECS FTP Server extension defines multiple destinations for each SAP Client. An FTP Destination describes an FTP Server with which ECS FTP communicates. It contains many items including the Server IP address (or DNS Names), purpose of the server, user ID and password used for FTP communications, and the directory separator used by that host. A destination can also contain the defaults used when communicating with this destination. These include transfer mode and type, the remote directory, and the FTP command script to use with this host.

> ***Note:*** Refer the previous configuration section regarding FTP Command Scripts. You may need to develop a separate command script for specific host types, such as an IBM AS/400 server or similar, depending upon the requirements of that server.

Run transaction /n/SKY/YFTP or shortcut YFTP to maintain FTP destinations. The following screen appears:



You need to enter the following fields:

1. Remote host logical name
2. Remote machine name or IP address
3. Remote system type.

Field	Usage	Possible Values
Name	Logical name for the remote system	Sales,Prod, Dev
Remote Machine	Remote machine name or IP address	
Remote Type	The purpose of the remote machine, that is, Production, Test, Development	
Remote User	The User Name for logging on to the remote machine	
Remote Password	Password for logging on relevant to the above user	

Field	Usage	Possible Values
Directory Separator	The directory separator used by the local FTP client when communicating with the destination FTP server (Some clients can automatically translate \ to / when required)	\, /, .
Transfer Mode [a]	Transfer mode to the machine	Put, append
Transfer Type [a]	Transfer type to this machine	Binary, ASCII
Remote Directory [a, b]	The remote directory on which to put the file	
FTP Script [a]	The standard text (SapScript) that is to use for the file transfer	

1. These fields *may be overridden* at the ECS process level.
2. These fields are file or directory fields, the standard variable substitution for logical or physical file names are applied.

This configuration is based on a default override principle with respect to the corresponding Process Configuration. The Transfer Mode, Transfer Type and Remote Directory fields of the Destination configuration are overridden by any values entered into the corresponding fields of the Process configuration.

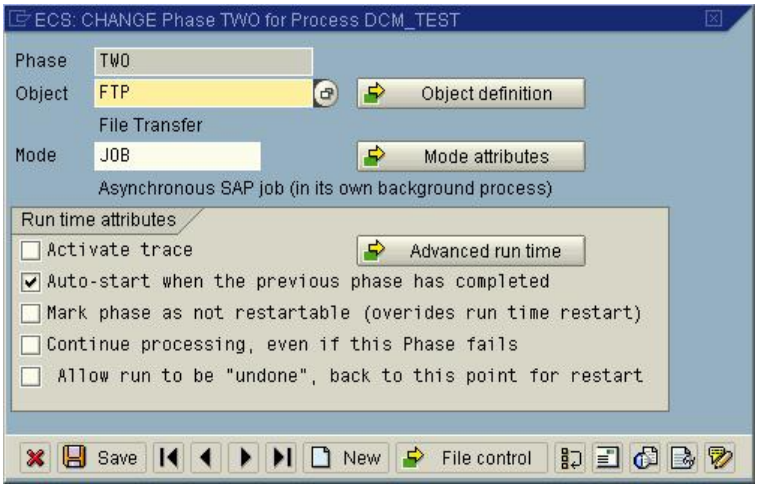
Defining ECS Processes to use FTP

ECS FTP process records are now defined as MPO attributes of a ECS Phase. Phase FTP records are stored in the /SKY/YECS_FTPPR table. Each ECS FTP phase may process multiple FTP transmissions. Each transmission request can have a separate destination; file name, directory, and command file. It is also possible to enter FTP records that are not tied to a specific ECS phase. These records are entered with a blank phase name. Such records are executed for all FTP phases within a process.

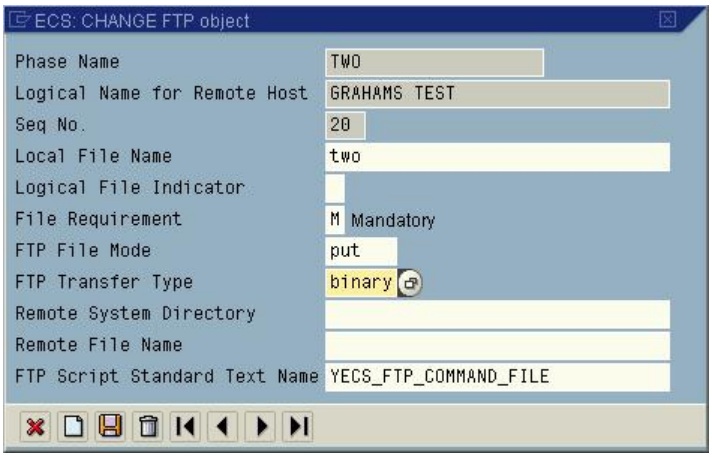
> *Important:* Whilst it is normal to transmit only one file per FTP transmission, it is possible to use the standard FTP MPUT and MGET commands with suitable modified FTP Command Scripts. Note that this inhibits some of ECS recovery capabilities, as it is possible for these multiple file commands to transmit only a subset of the requested files. Use these features with caution.

The concept is that a previous ECS Phase creates the file, or files, to transfer and a subsequent FTP phase is implemented to perform the actual transmission.

Create an ECS Phase definition, selecting the FTP object type and the ECS execution mode.



Click **Object definition** to define the FTP attributes. The following screen is displayed:



This screen is used to define the FTP attributes that the ECS FTP extension uses to generate a file transmission. You may use the following substitution variables in the local and remote directory / file names: `&ECS_FILE`, `&ECS_VAR1/2/3`.

Further Customizing and User Extensions

In the event that you need to store additional information and transmit in an FTP script, you can employ two methods:

1. "Hard" code these additional fields directly into a script using the script as a repository. The disadvantage is that the parameter may be particular to the process or machine and may need to change in many places thus requiring the copying of the script multiple times.
2. Create and append a structure to the tables YECS_FTPPR and/or YECS_FTPPN that contains the fields required.

Customer Appends

Any fields added to the standard tables are available as variables simply by addressing them in scripts as &variable_name (that is, In the same manner as standard fields). The hierarchy of tables is the Destination (YECS_FTPPR) then Process (YECS_FTPPN). If values for the same field name exist from both tables, the process table values override the destination table values.

> ***Important:*** Always ensure append is used as this enforces certain name space rules that avoids future ECS conflicts as well as ensuring the script is upgrade independent. If fields are added in any other manner, compatibility with ECS upgrades is not guaranteed.

6.17.2.3 FTP Extension Examples

Example Unix Script

The shell program must pipe a temporary file as source input to FTP, a sample follows:

```
#!/bin/ksh
# Usage: Pipe Command file to FTP Command, SAP will not let you
#        pipe natively as it is deemed a "Security Risk"
# Input: Fully qualified command file name eg //usr/sap/tmp/command.txt
# Output: StdOut will contain the log from FTP in verbose mode
# note that the netrc file is NOT used for the logon, this
# must be explicitly given in the command file
#
pn=$\{0##*/\}
USAGE="Qualified, Command file name"
# Check the number of parameters
if \[ $# -ne 1 \]
then
echo $\{USAGE\}
exit 1
fi
ftp -v -n < $\{1\}
```

> ***Note:*** The same could be written in any language, the only functional line of code is ftp -v -n < ${1}, call ftp, verbose mode, no auto logon, piping the file passed as the variable.

Example Windows Script

```
@echo off
REM  MS-DOS
REM  Input:   Fully qualified command file name
REM  Output:  StdOut will contain the log from FTP in verbose mode
REM           note that the netrc file is NOT used for the logon,
REM           this must be explicitly given in the command file
REM
REM Check the number of parameters = 1
if %1x==x goto noparam
if not %2x==x goto toomany
```

```
echo Program %0 called with parameter %1
REM Call the FTP Program
@echo on
ftp -v -n < %1
@echo off
goto exit
REM No parameters
:noparam
echo Usage: One parameter, qualified file name
goto exit
REM At least two parameters
:toomany
echo Usage: One parameter, qualified file name
goto exit
REM Finished
:exit
```

6.17.2.4 Troubleshooting

The ECS FTP extension is largely self explanatory in terms of errors. When an FTP Transmission fails, ECS appends the standard input / output from the FTP session to the ECS log. You can therefore examine the session and rapidly determine the cause of the error. To assist with problem determination, it is recommended that you carry out FTP sessions manually first, prior to configuring them, and also when errors are encountered. You can solve many slightly obscure problems readily with a manual session, such as permission errors.

Errors Before any Files are Sent

The following error conditions stop the phase prior to any transmission:

Error Message	Resolution
Directory Delimiter cannot be found	• The ECS system constant for directory delimiter is blank • Run the ECS installation check for further information on correcting the problem
ECS Constant {Name} cannot be found	• ECS is expecting a constant to be defined for the default transmission text • Run the ECS check installation program for more information on correcting the problem
Could not find the Host {Name}	• A process was found referring to a logical destination, but the Logical Remote Host did not exist. • Modify the configuration to restore this mapping
Mandatory file not found	• One of the files listed as mandatory in the process is not found on the local machine. • Check the configuration

Error Message	Resolution
Transmission from Non Productive to Productive System	• ECS does not allow FTP from a non-productive system (per table T000) to a system that is flagged as productive in the ECS configuration
The standard text {name} could not be found in the current system	• A standard text that is referenced in the FTP configuration could not be loaded. • Change the configuration or create the text
End of transfer text (&_complete) was not found in the text {textname}	• The end of transfer variable is mandatory • Modify the standard text to include the field

Errors During File Transmission

The following are errors that occur during file transmission, the program still attempts all other transmission steps, but the phase itself ends in an error as re-processable.

Error Message	Resolution
Cannot create temporary file {name}	• ECS must generate a temporary file to pass to the FTP command, security access may not allow the creation of the file or another operating system error has occurred. • Relevant problem details may be available in the ECS log
Cannot write to file {name}	• ECS successfully generated the temporary file, but a problem has occurred when writing data to the file. • Relevant problem details may be available in the ECS log
Cannot close file {name}	• ECS successfully generated and written the temporary file, but a problem occurred when closing the file. • Relevant problem details may be available in the ECS log

Error Message	Resolution
Logical Command {name} Failed	• ECS could not execute the logical command. • Relevant problem details may be available in the ECS log
FTP Failed	• The FTP success strings from the standard text were not returned by FTP. • The log contains all relevant information pertinent to resolving the problem, namely the output from FTP and the command strings that ECS was looking for
Command File {name} could not be deleted	• This error occurs after transmission and in the absence of any other errors the transmission was successful. However there are ECS temporary files left in the working directory that you should remove

Information and Warning Messages

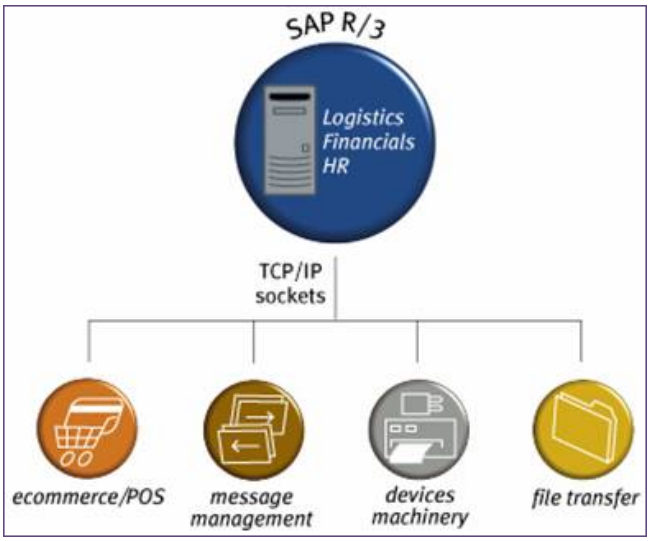
Error Message	Resolution
No Transmission Plan Found	The phase that is running has no transmission phases currently configured
Transmission from Productive System to Non Productive system	This message occurs in the event the configuration reflects the FTP of files from a productive system (per table T000) to a system that is flagged as non productive in the ECS configuration

6.17.3 TSX

TSX (TCP/IP Sockets Exchange) provides real time peer to peer message and file transfer between remote applications, over a TCP/IP network. This component is written in both ABAP and Java programming languages, thus rendering it highly portable to execute on those platforms that support SAP and the Java Virtual Machine.

6.17.3.1 TSX Overview

TSX provides real-time "peer-to-peer" messaging and file transfer between an unlimited number of SAP and non-SAP systems on a TCP/IP network. It is an effective way to connect "network aware" machines, devices and applications to SAP. TSX provides an easy to use "simple" protocol that eliminates the need for complex middleware, and FTP scripting. TSX effectively provides the "plumbing" within which to transfer data between remote applications. The TSX architecture consists of both a SAP add-in component written in ABAP and a Java component. This makes TSX highly portable and able to execute on any platform supported by SAP and the Java Virtual Machine specification.



Peer-to-Peer Messaging

Peer-to-peer messaging is the real-time transfer of single items of data between two applications. This occurs in a conversational 'synchronous mode' (similar to a telephone conversation). This mode of interface is best suited for program to program applications that transfer small items of data.

File / Directory Transfer

TSX supports the transfer of files or entire directories across the network. TSX file transfer uses an internal protocol to guarantee delivery. Options are also available to automatically launch applications when a file/directory is arrived. TSX supports any file type or size. TSX does not support the FTP protocol. Automatic directory polling is also supported.

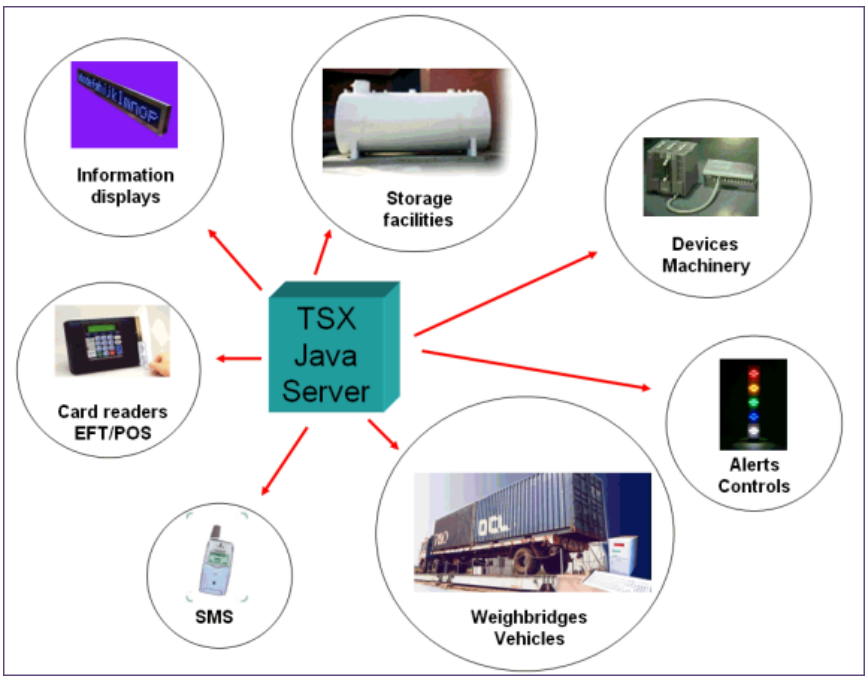
Serial Devices

TSX can communicate (send/receive) with any serially connected device, for example, PLC, and weighbridge. You use standard configuration options to define each serial connection and how the information is formatted and managed.

What can TSX be Used for?

You may use TSX to communicate inbound and outbound messages, files or entire directories to any application that supports the TCP/IP sockets protocol or serial device. Examples include:

- File based interface applications
- Program-to-program messaging applications with SAP
- Monitoring factory machines and PLC devices, for example, automatic goods receipt and machine efficiency
- Communicating with banking payment gateways, for example, Ingenico OCV
- Communicating with HR time 'clock in' devices
- Custom 'real time' interfaces to external specialist packages, for example, remote vehicle location tracking. Meter reading applications
- POS equipment and interfaces
- Weighbridges
- Overhead text displays
- SMS gateways.



Why use TSX?

Run it Anywhere

Because of its unique architecture, TSX can quite literally run anywhere from Windows laptops to Mainframe servers without change. It uses a very simple protocol that is based on the universal TCP/IP sockets protocol. TSX has the flexibility to transmit many forms of data, delimited in many different ways, for example, variable length messages with imbedded lengths, binary or ASCII files. Because TSX is highly portable, you may utilize existing hardware, eliminating the need for dedicated hardware.

Easy Installation and Management

You may monitor or manage TSX through Web browser or command line options. Its configuration is text file based and so you may maintain using any editor, for example, notepad, VI, and Word. The TSX components are downward compatible with previous releases, allowing for 'phased' implementations.

End-to-End Transaction Management

TSX provides comprehensive end-to-end transaction management. Most integration solutions with SAP 'wash their hands' of all responsibility once a business transaction is issued to SAP. TSX provides a standard framework that ensures data integrity, performance and the correct sequencing of business transactions from their inception to their final completion.

Eliminates the Need for Complex Middleware

TSX has a simple, easy-to-use robust architecture, that supports both "peer-to-peer", messaging and file / directory transfer between R/3 and non-R/3 applications. In many cases, you may use TSX to manage the entire messaging infrastructure between remote hosts, thus eliminating the need for multiple layers and expensive, over-complex middleware and EAI solutions.

Built-in Redundancy and Store and Forward Capability

TSX has built-in automatic mechanisms to re-establish lost network or SAP connections, time out controls to break 'hung' connections, restart itself if it is canceled or fails and automatically store and forward messages and file transfer requests if network or SAP connections are down.

Seamlessly Integrate Applications

As well as transfer messages and files, TSX can automatically launch SAP and external applications when data arrives. This enables the sending and receiving applications to be seamlessly integrated, eliminating the need for script files, scheduling, polling, and manual intervention.

High Performance Architecture

TSX provides high speed data transfer capability that is consistently sub-second. You may compress to optimize data transfer in environments with low bandwidth, for example, between remote servers and a local R/3. You may scale TSX to any platform in either distributed or centralized configurations. You may perform file transfer in a full or semi real-time mode, eliminating time wastage due to directory polling and time based assumptions in processing. TSX is fully multi-threaded, enabling it to automatically scale, depending on demand. You may configure maximum limits to limit TSX usage of system resources. TSX supports both multi-threaded RFC client and server connections. SAP load balancing is supported.

Portability and Flexibility

TSX is written in ABAP and Java. This means that it is highly portable to any SAP environment and runs on any Java Virtual Machine 1.1.7+. The Java Virtual Machine (JVM) specification is widely supported on by all major operating systems. Therefore, you may execute the same version of TSX, without change on Windows: XP/NT/2000/2003/98/95 and Unix: HP-UX, IBM-AIX, Sun or Linux.

6.17.3.2 TSX Architecture

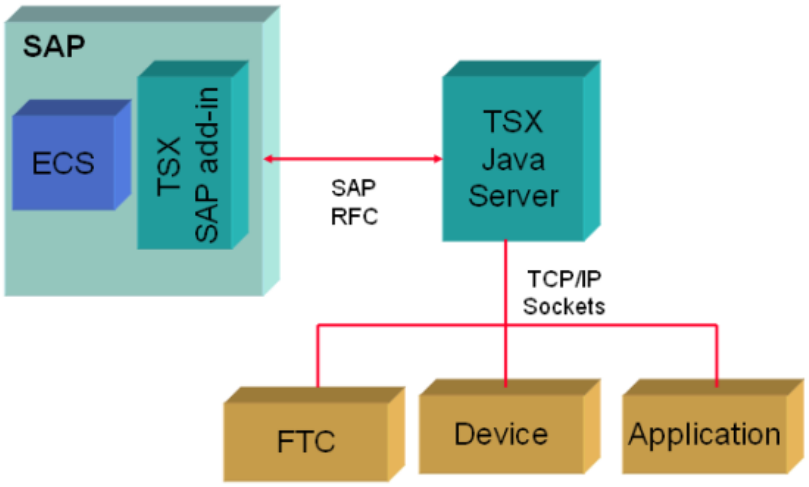
TSX Architecture consists of the following:

TSX Components
Sample Topologies
TSX Concepts and Terminology

TSX Components

TSX is made up of three components:

- TSX SAP add-in
- TSX Java server
- File Transfer Client (FTC)



TSX Java Server

The TSX Java server is responsible to control all data transmissions to and from its configured message ports, file ports and / or serial devices. It is a fully multi-threaded application, allowing it to perform multiple tasks at once. The TSX Java server is the "heart" of the TSX application, controlling all network traffic and connected devices.

TSX SAP Add-in

The TSX SAP add-in enables ABAP applications to communicate with the TSX Java server. It also provides a fully functional workbench used to configure how data is to be sent and received.

FTC (File Transfer Client) - Command Level File Transfer

FTC is called by external applications or through the command line to transmit files / directories to a TSX server on another remote system. Because FTC is a client, it is only active for the duration of the call, that is, does not run as a continuous service.

Sample Topologies

You can deploy TSX in a number of different arrangements depending on requirements. The following sections show different TSX deployment scenarios used to meet specific requirements.

Direct Program to Program Communication

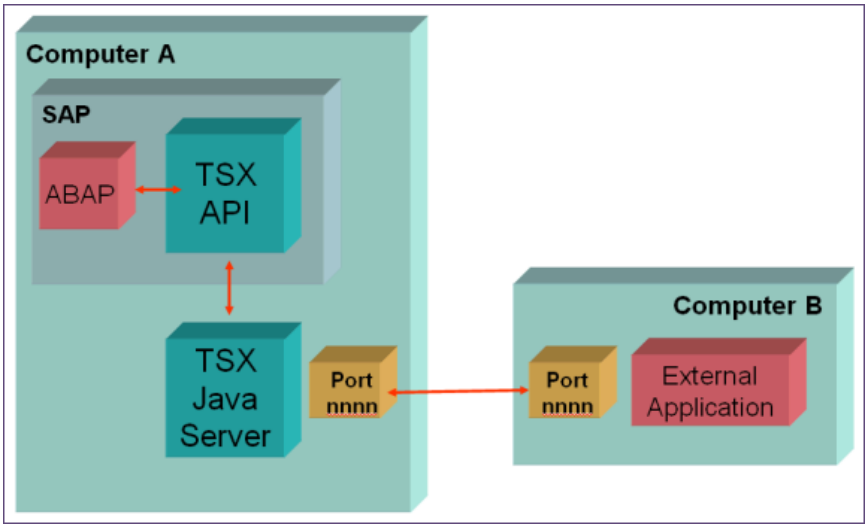
The scenario is commonly used in:

- Web services
- Communication between SAP and a third party system inside a network of an organization

In this case, a direct bi-directional conversation starts between SAP and an external application.

The TSX Java server may run locally on the SAP host, or remotely on another server. You may deploy multiple TSX Java servers to service specific applications on remote hosts.

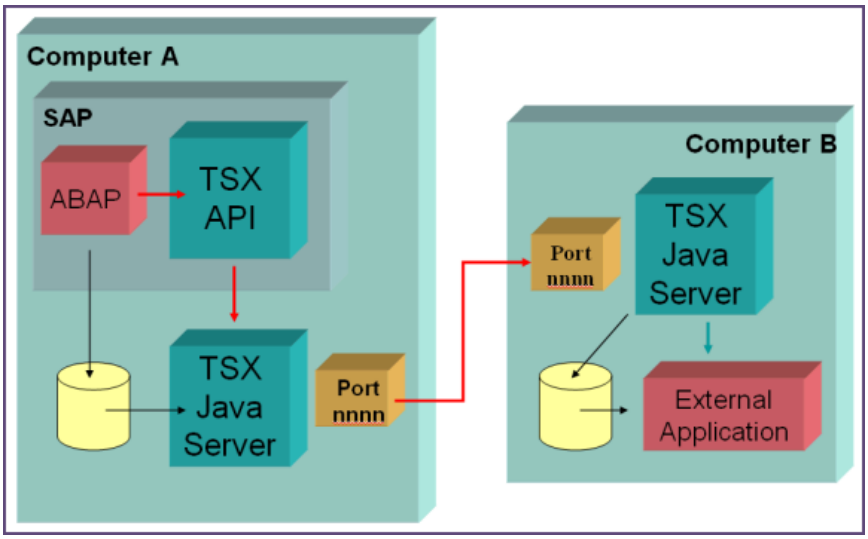
The SAP ABAP application starts the connection and communicates with the external sockets application using the TSX API (function calls). See the ABAP programming section for details. You must write the external application to use the TCP/IP sockets API.



Note: The TSX Java server may reside locally on the SAP host or remotely on another host. The external application communicates with the TSX Java server using the TCP/IP sockets protocol.

File Transfer - SAP to External

A SAP application may send files or complete directories to a remote application. The entire transfer operation happens in a semi-real time mode and you may entirely automate, that is, you may automatically automate the receiving application launched once the file/directory arrives.

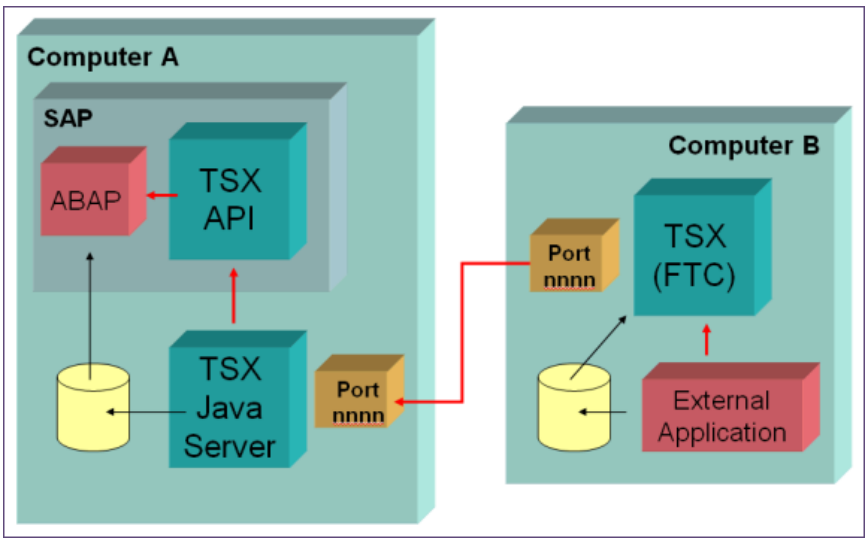


Note: In this example, a TSX Java server runs on the SAP host to receive the files. This is because you need to copy the files locally so that the receiving ABAP application can process them. TSX also supports the sending of file contents directly between SAP and a remote TSX Java server. In other words, you do not require a local TSX server.

Important: TSX can automatically transfer the file(s) and launch an external application if required.

File Transfer - External to SAP

The remote application invokes the FTC Java Client with the name of the file / directory to send. The TSX Java server that runs on the target system, receives the file data and launches a SAP application to process it. The entire transfer operation happens in a semi-real time mode and you may entirely automate from end-to-end. The file is passed to the ABAP program through a Sky ECS Process and you may use a TSX user exit may to determine which ABAP to execute.



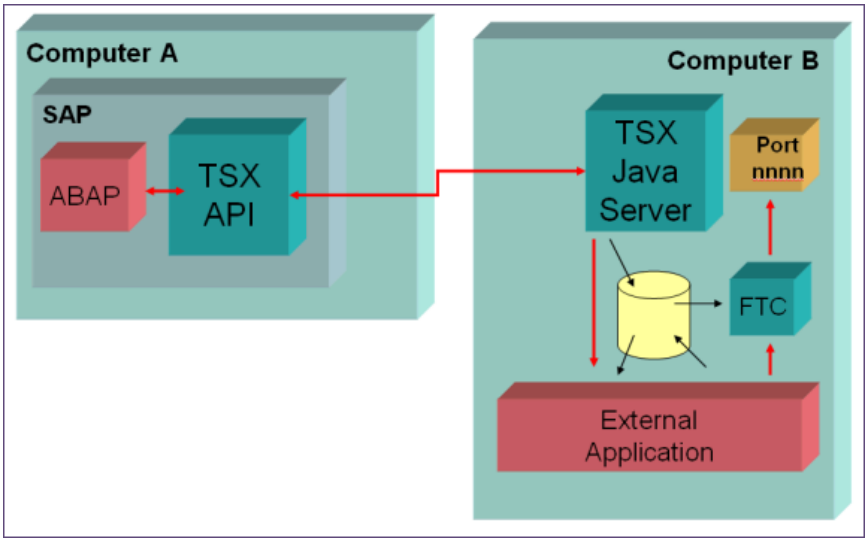
Note: In this example, a TSX Java server runs on the SAP host to receive the files. This is because you need to copy the files locally so that the receiving ABAP application can process them. TSX also supports the sending of file contents directly between SAP and a remote TSX Java server. In other words, a local TSX server is not required. See the following section "File transfer (direct to/from SAP)" for more details.

Important: TSX can automatically transfer the file(s) and launch the receiving APAB application through ECS if required.

File Transfer - Direct

TSX also supports performing file transfers directly between SAP and a remote TSX server, that is, no local TSX server is required on the SAP host.

This is useful to not only avoid having to install a TSX Java server, but to support cases where it is difficult to install a Java environment, for example, AS/400. In this scenario, the complete file is transmitted between SAP and the remote TSX server using RFC.



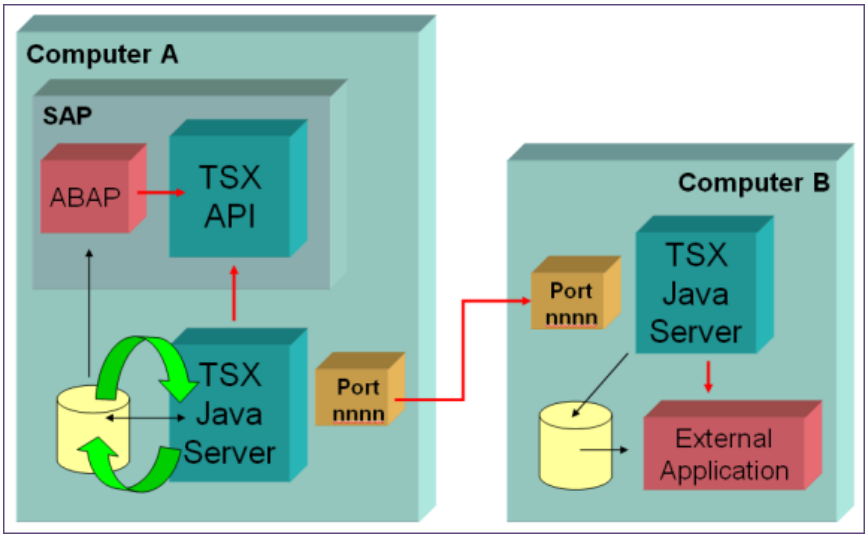
The only difference in this case is that SAP communicates with a remote TSX Java server, without the need to store the files locally. The TSX API in SAP receives all the file data and then automatically creates the file locally on the SAP host and then invokes the ABAP application, passing on the new file name.

Automatic Directory Polling

You may configure TSX to automatically poll a directory and transfer any files it finds to either an external system and / or SAP. This can be useful in cases where you use an inbox type scenario, for example, EDI.

The polling interface automatically moves files to working, archive or error directories as part of the processing.

You may implement Custom Java exits to perform more complex pre-processing.

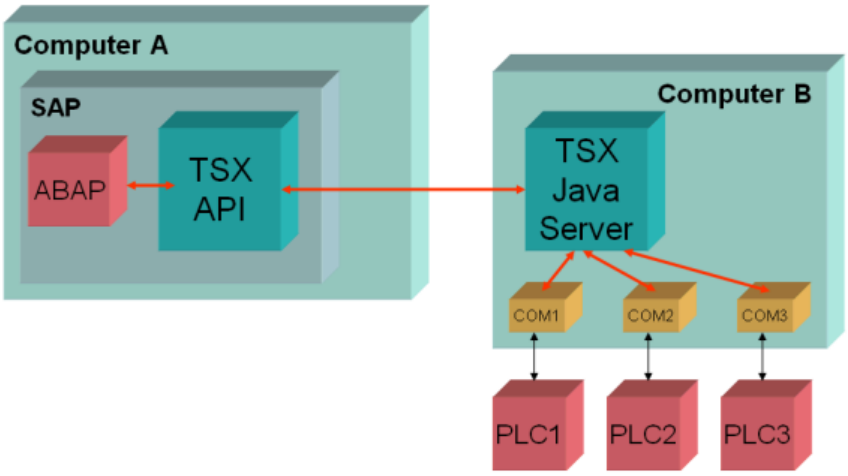


Serial Device Interfaces

SAP has no native serial device interface.

TSX has an in-built interface to communicate with serially connected devices. This enables SAP applications to communicate directly with devices such as:

- PLC (programmable logic controllers) devices
- Weighbridges
- Factory floor machinery
- Overhead text displays
- EFTPOS units
- Cashdrawers.



> ***Important:*** Serial connections are physical links to a computer that means that a TSX java server must run on the same computer.

TSX Concepts and Terminology

Message

A message is simply a string of data that is transmitted between two points. TSX needs to be told what the length or delimiter of the message is.

Logical Destinations and Message Types

In order to make programming in ABAP easier and avoid having to 'hard code' IP address and message format control parameters within programs, two 'side info' tables are provided that are accessed using logical names. The caller only has to provide these logical names when they call the TSX SAP add-in and the configuration is derived from the tables. See the section on the SAP TSX workbench for more details. You need to tell TSX about the format of the data, that is, what is the length of the message? What is it delimited by? All of this is configured in the message type definition.

TCP/IP Sockets

Every computer and 'network' aware device has a unique IP address on the network to identify itself. Within its unique address, each computer / device may have one or more 'network aware' services that actively transmit, receive or passively listen for information on the network. A unique Port number is used to discriminate between the different network services within the same computer / device. A port number (typically 16 bit) is anything from 1 – 32K where 1 – 1,024 is normally reserved by the operating system. Standard ports include 22-FTP and 23-Telnet. Most operating systems support a services file where you may document customer IP address and Port configurations. TSX IP and Port configurations are totally configurable. See the "Configuration" section for more details on selecting and specifying ports. TSX transfers all messages and files using the universal TCP/IP sockets protocol. Sockets allow for the bi-directional transfer of information (messages) between two unique points on the network (similar to a telephone conversation). The sender and receiver are uniquely identified by an IP address and port, for example, 192.168.2.70 – 5082 (similar to a telephone number). You need to follow a conversation protocol between the sender and receiver to transfer data, for example, opens a connection, write a message, and wait for response (similar to a telephone conversation).

Fail-Over Processing

TSX supports fail-over processing. If a call to the primary destination fails and an alternate destination is configured, TSX automatically switches over to use the alternate. You can make this switch permanent after a number of attempts using a 'force switch' threshold. You can also instruct TSX to retry the primary destination every 'nnn' number of calls using the 'force retry' threshold. You may daisy chain alternate destinations in this fashion to an unlimited depth, however TSX has a hard coded limit of 100 total attempts to prevent any loops. Refer TSX Workbench for more details.

6.17.3.3 Installation of TSX

To install the TSX Java server and the TSX SAP add-in components, download the Expert Release. Once you install the products, configure the TSX Java server and SAP add-in definitions.

TSX Java Server Configuration

Each TSX Java server has a configuration file **tsx.cfg**, in which all the connection and system options are defined. The file is a text file that contains distinct sections and options. The configuration file is global and is the same format irrespective of the platform. A sample configuration file (`tsx.sample.cfg`) is provided with the TSX java installation and you can use as a base to copy and create a new configuration file. When the TSX Java server starts, it reads in the configuration file located in its home directory into memory and processes the options. All initial processing is written to the TSX system log, including any warnings and errors. All configuration options are grouped logically into standard sections that are identified by [ ], for example, [General]. Each option is a simple assignment, for example, ServerId = NEPTUNE. Many of the options are optional and have predefined default values. The following is a summary of all the sections and associated options.

General

This section contains parameters of a general nature that affect the operation of TSX as a whole.

Option	Description	Default
ServerId	A unique identifier to distinguish this particular instance of TSX from any others that may run on the same machine and / or against the same SAP instance. You can use this identifier to perform restricted data downloads. If left blank or not configured, then the TSX server ID is configured to the name of the host machine on which the TSX server is runs.	As noted.
ServerGroup	This is an arbitrary identifier that is used to group together TSX servers that have common behavior (for example, all those in a certain geographic location) so that you can treat the servers in a similar fashion by SAP-side processing.	- (no group)
RestrictToHost	Name of the host on which TSX must run. If a different host name is detected, TSX shuts down immediately.	- (any host)

Option	Description	Default
RestrictToUser	The user ID that you use to start TSX. TSX may require specific authorizations, and thus a special user may need to start the TSX. If specified, and this user is not the current user, TSX shuts down immediately.	- (any user)
LicenseFile	The name of the license file.	license.dat
AutoRestart	A true / false flag that, when you enable, causes TSX to automatically restart itself if it is not shut down properly (for example, in the event of the process crashing or being killed).	False
DateFormat	The format in which you need to enter dates and display.	dd/mm/yyyy
TimeFormat	The format in which you need to enter times and display.	hh:mm:ss
TimeZone	The time zone in which the TSX server is runs (overrides the default configuration that the Java Virtual Machine supplied).	JVM default

Heartbeat

This section contains information about the way in which TSX regularly reports its status to SAP.

Option	Description	Default
HeartbeatInterval	The period of idle time (in seconds) between SAP status notifications when not connected.	15
HeartbeatConnectedInterval	The period of idle time (in seconds) between SAP status notifications when connected. If left blank or not specified, this defaults to the same value as the HeartbeatInterval.	As noted.

Option	Description	Default
HeartbeatHostName	An override for the name of the host reported to SAP during the heartbeat status notification. If left blank or not configured, then the name of the host machine on which the TSX server runs is used (in the vast majority of cases, this is the most appropriate configuration).	As noted.
HeartbeatHostAddress	An override for the address of the host reported to SAP during the heartbeat status notification. This need to be in the standard IP address format xxx.xxx.xxx.xxx. If left blank, or not configured, then the IP address of the host machine on which the TSX server runs is used (in the vast majority of cases, this is the most appropriate configuration).	As noted.

[HostInterface:xxx] - Direct RFC Client Connection

These section(s) contains the information required by TSX to establish a direct RFC connection to one or more SAP R/3 instance(s)

Option	Description	Default
HostInterfaceType	You need to configure *RfcClient* for direct RFC connections.	RfcClient
SapDestination	The SAP symbolic destination. This shows up in the SAP R/3 gateway monitor listing and you can use to differentiate TSX client connections from other incoming connections.	TSX
SapClient	Numeric SAP client identifier.	No default value.

Option	Description	Default
SapUser	The SAP user ID to use. You need to be a CPIC user.	No default value.
SapPassword	The SAP password to use.	No default value.
SapLanguage	The SAP language identifier, for example, (English).	E
SapTraceLevel	The SAP RFC trace level (need to leave as zero (0) in normal operation).	Zero (0)
SapUseLB	A true / false flag to instruct TSX whether to use the SAP load balancing feature.	No default value.
SapLBHost	Host for load balancing.	No default value.
SapLBSystem	SAP system identifier for load balancing.	No default value.
SapLBGroup	The SAP load balancing group.	No default value.
SapHost	A specific SAP host to connect to (alternative to load balancing).	No default value.

Option	Description	Default
SapSystem	A specific SAP system number identifier to connect to (alternative to load balancing).	No default value.
SapConnectionTimeout	The number of seconds to wait before terminating an attempt to connect to a SAP system.	0 (no timeout)
JniLogging	A true / false flag to instruct TSX whether to log JNI calls in the TSX log file.	false
MinimumNumberHandlers	Minimum number of handler threads	1
MaximumNumberHandlers	Maximum number of handler threads	0

HostInterface:xxx - Through XML Gateway

These section(s) contains the information required by TSX to establish a connection to a back-end host system via the Sky XML gateway

Option	Description	Default
HostInterfaceType	Must be configured to "XmlClient" for XML gateway connections.	XmlClient
XmlGatewayHost	The name or address of the host on which the XML gateway is running.	No default value.
XmlGatewayPort	The port number on which the XML gateway is listening for incoming XML.	No default value.

Option	Description	Default
XmlGatewayResponseTimeout	The maximum period of time (in seconds) for which TSX waits for an initial response from the XML gateway before timing the transaction out.	30 (seconds)
XmlGatewayStallTimeout	The maximum period of time (in seconds) for which TSX allows an incoming response to stall (that is, send no further data) before timing the transaction out.	30 (seconds)
UseCompression	A true / false flag that indicates whether you should compress the XML data sent to the gateway. This greatly reduces bandwidth consumption.	false
CompressionAlgorithm	The compression algorithm to use when compressing data sent to the XML gateway. The following compression algorithms are available: GZIP - The "gzip" compression format (recommended). ZIP - The "zip" compression format.	No default value.
UseEncryption	A true / false flag that indicates whether you should encrypt the XML data sent to the gateway. Obviously, this is a more secure option, particularly over a public network.	false
EncryptionAlgorithm	The encryption algorithm to use when encrypting data sent to the XML gateway. The following encryption algorithms are available: BLOWFISH - The "blowfish" encryption scheme.	No default value.

Option	Description	Default
EncryptionKey	The encryption key to use when encrypting data. If no encryption key is specified, TSX looks for one in an encryption key file instead.	No default value.
EncryptionKeyFile	The file in which the encryption key is stored.	No default value.
EncryptionKeyHexEncoded	A true / false flag that indicates whether the encryption key is hex-encoded.	false
XmlDebug	A true / false flag that indicates whether all XML data sent to, and received from, the XML gateway should be logged. This facility can be useful to debug in some situations, but it generates large log files and should generally be left off.	false
UseTransferFields	Indicates whether information on local database table fields should be sent during table downloads, refreshes, and uploads. For SAP systems, you do not need this and should configure to "false".	false

HostInterface:xxx- RFCServer Connection

These section(s) contains the information required by TSX to establish an RFC Server registration with an SAP system.

Option	Description	Default
HostInterfaceType	You must configure "RfcServer" for RfcServer connections.	RfcServer

Option	Description	Default
SapRegistrationName	The name to register with the SAP gateway for the RfcServer connection.	/SKY/TSX_SOCKET_EXCHANGE
SapHost	A specific SAP host to connect to (alternative to load balancing).	No default value.
SapGateway	A specific SAP system gateway identifier to connect to (format sapgw## where ## is the system number of the SAP instance).	No default value.
SapTrace	A true/false flag that indicates whether the RFC Server calls should be traced	false
JniLogging	A true / false flag to instruct TSX whether to log JNI calls in the TSX log file.	false
JniLogListenCalls	A true / false flag to instruct TSX whether to log Rfc Listen calls in the `TSXlog` file. ***Note:*** This is only valid if JniLogging is also configured to *true*.	false
MinimumNumberHandlers	Minimum number of handler threads	1
MaximumNumberHandlers	Maximum number of handler threads	0

Log

This section contains information about TSX logging and tracing capabilities

Option	Description	Default
LogFileDirectory	The directory into which TSX writes its log files. If not configured, this parameter defaults to "." (the current working directory).	As noted.
LogStrategy	The logging strategy to adopt. It should be one of the following: D - A new log file is to create daily. W - A new log file is to create weekly. P - A new log file is to create for each process instance. O - Overwrite the log file each time TSX restarts	O
LogTsxMessages	A true / false flag that controls whether information about incoming and outgoing messages and files is written to the log file, by default. This configuration is configured globally in the [Log] section, but you can override this global configuration for individual message or file ports.	true
ExcludeLogMessageSeverities	Indicates the types of log messages that you should exclude from the log file. There are five types of log messages - Information (I), Warning (W), Trace (T), Error (E) and Abort(A). You should concatenate together the log message types to exclude. Thus, to exclude all Information and Trace messages from the log file, the appropriate configuration is "IT".	- (no types excluded)
MaximumNumberLogFiles	This configuration restricts the maximum number of log files TSX keeps under D, W, and P strategies. If left blank or not configured, then no maximum is enforced. Zero (0) indicates no maximum.	0 (no maximum)

Java

This section contains information about which JVM TSX uses and any special parameters

Option	Description	Default
JavaCommand	The path to the executable used to launch the Java Virtual Machine (for example, `java.exe`). Under Windows, you need to have this in short file name (MS-DOS 8.3) format, for maximum compatibility. If not specified, the value of the environment variable SKY_JAVA_CMD is used instead.	No default value.
JavaArguments	Any additional arguments to pass to the Java command line. Common examples are arguments to increase the maximum memory ceiling (-Xmx128m) or run with a reduced signal configured (-Xrs).	No default value.

Monitor

This section contains configuration information for TSX internal monitoring capability

Option	Description	Default
UseMonitor	A true / false flag that enables or disables TSX internal monitoring capability.	false
MonitorInterval	The time interval (in seconds) between monitoring status checks.	60 (seconds)
GarbageCollect	A true / false flag that indicates whether you should perform Java garbage collection (that expends resources in an attempt to free up memory) as part of each monitoring status check. If configured to *true*, garbage collection takes place each time a monitoring status check occurs. If configured to *false*, garbage collection is managed by the Java Virtual Machine.	false

Option	Description	Default
LogMemoryUsage	A true / false flag that indicates whether you should write current memory usage information to the TSX log as part of each monitoring status check. If you configure both, this option and the GarbageCollect option to *true*, memory usage information is logged both before and after the garbage collection takes place.	false
LogThreadCount	A true/false flag that indicates whether you should write the number of active threads to the TSX log as part of each monitoring status check.	false
LogThreadNames	A true/false flag that indicates whether the names and states of active threads should be written to the TSX log as part of each monitoring status check.	false
ClearRfcTraceFiles	A true / false flag that indicates whether to automatically delete logging / trace files created by the SAP RFC library.	false
RfcTraceExpiryDays	The maximum number of days for which to retain SAP RFC trace files when you enable automatic deletion.	7 (days)

Network

This section contains information on TSX interaction with the network.

Option	Description	Default
AcceptTimeout	The maximum period of time (in milliseconds) for which the thread associated with each port blocks while waiting for new incoming connections. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)

Option	Description	Default
BindTimeout	The maximum period of time (in seconds) for which TSX waits while attempting to bind to its listening ports. If another process listens on its designated ports, or if the bind process fails, then TSX is not able to establish the listening port. Once the timeout occurs, it closes down with an appropriate error message. If this parameter is configured to zero or not specified, then TSX waits indefinitely to bind to its listening ports in the hope that they eventually become available.	0 (no timeout)
HostNameResolution	A true / false flag that indicates whether host name resolution is desired. Resolution of the host name can sometimes be a considerable overhead on performance. For this reason, it is disabled by default. However, you should enable it in situations where performance is not an issue, and display of the hostname rather than the IP address is felt to be of benefit.	false
NetworkErrorSampleSize	The sample size for network error monitoring. TSX maintains a moving window of the success or failure of the last n network calls, where n is the value of this parameter. If not configured or left as zero, any network error is treated as fatal for the port or connection on which it occurs. For most well-tuned networks, you should choose this configuration.	0 (no error sampling)

Option	Description	Default
NetworkErrorThreshold	The error threshold for network error monitoring. Must be less than or equal to the NetworkErrorSampleSize parameter. When this number of network errors occur within the sample, TSX concludes that the connection or listening port has unrecoverable network problems and closes it down. It then attempts to re-establish the connection or port in the hope of obtaining a more stable socket.	0 (no error sampling)
ReadTimeout	The maximum period of time (in milliseconds) for which the threads associated with connections on this port blocks while waiting for input. For most well-tuned networks, there is no need to assign a value to this parameter.	200 (milliseconds)
UseNativeSockets	A true / false flag that indicates whether you should use native socket calls instead of the capabilities provided by the Java virtual machine. This can sometimes provide additional information when errors occur, and typically improves performance.	false

Option	Description	Default
ValidIpAddresses	A comma-delimited list of addresses from which incoming connections are accepted. These addresses can either be in human readable form, or in raw IP address form (xxx.xxx.xxx.xxx). You can use an asterisk (*) as a wildcard to substitute for any single part of the address. Thus, for example, an entry of *.skytechnologies.com.au allows any host from the "skytechnologies.com.au" domain to be considered valid. Or, for example, an entry of "192.168.*.*" allows any host from the "192.168" subnet to be considered valid. If you specify this parameter, then any incoming connection from a host not listed is rejected. If this parameter is left blank or not specified, then by default, all incoming connections are considered valid regardless of the host from which they originate. ***Note:*** You can override this parameter for individual ports.	- (all addresses are valid)
InvalidIpAddresses	A comma-delimited list of addresses from which incoming connections are not accepted. This format of this list is identical to that described for the valid IP address list. ***Note:*** The invalid IP address list has a higher order of precedence than the valid IP address list, so a host that matches both lists is considered invalid. You can override this parameter for individual ports.	- (no addresses are invalid)

Poll Directory:xxx

These section(s) contain information pertaining to one or more poll directories that the TSX server should

monitor for incoming files. When an incoming file is detected, a user written Java exit is automatically invoked to take any action that may be appropriate.

Option	Description	Default
IncomingDirectory	The pathname to a directory in which incoming files are placed by an external process. TSX monitors this directory for candidates.	No default value.
WorkDirectory	The pathname to a directory in which incoming files are placed by the TSX server to be worked on. If not specified or left blank, the files are worked on in the IncomingDirectory.	As noted.
ArchiveDirectory	The pathname to a directory in which files are placed by the TSX server, once they are successfully processed. If left blank or not present, then any successfully processed files are deleted rather than transferred to an archive directory.	As noted.
ErrorDirectory	The pathname to a directory in which files are placed by the TSX server, once they are unsuccessfully processed (that is, found to have errors). If left blank or not present, then any files that you cannot process are deleted rather than transferred to an error directory.	As noted.
IncomingFilePattern	A filename pattern that is used to check whether files in the incoming directory are suitable for processing. Wild card characters are permitted: an asterisk (*) matches any number of characters, and a question mark (?) matches any single character. If left blank or not present, then any file in the incoming directory is considered suitable for processing.	As noted.

Option	Description	Default
PollFrequency	The time interval (in seconds) between polls of the incoming file directory.	No default value.
PollFrequencyMilliseconds	The time interval (in milliseconds) between polls of the incoming file directory. This interval is added to the number of seconds specified in the "PollFrequency" item.	0
LogIncomingFiles	A true / false flag that indicates whether you should write log messages for each incoming file that is detected.	false
StopPollingOnError	A true / false flag that indicates the normal behavior for the poll directory when an error occurs. If configured to *true*, all polling stops when an error encounters, and you need to manually restart from the TSX web status page. If configured to *false*, polling continues to take place after an error encounters.	false
PollDirectoryExit	The fully qualified name of the user written Java exit to invoke when an incoming file is detected. The class in question must be available to the JVM in which TSX runs (that is, you should place the class, or the jar/zip file containing the class, inside the TSX `classes` directory).	null (no exit)

RelayConnections

This section controls whether (and how) the Java Server opens connections to a relay port running on a gateway. A relay connection allows remote manipulation of the server through an interface similar to that offered by the server web status page.

Option	Description	Default
RelayHost	The name or address of the host on which the relay port is available (i.e. the gateway machine). If left blank or not specified, relay connections are not permitted.	No default value
RelayPort	The port number of the relay port. If left blank or not specified, relay connections are not permitted.	No default value
AllowConfigChangesFromWebPage	A true / false flag that controls whether you can modify the configuration file through the relay web interface.	False
AllowFileEnquiryFromWebPage	A true / false flag that controls whether you can use remote file management to view files through the relay web interface.	False
AllowFileUpdatesFromWebPage	A true / false flag that controls whether you can use remote file management to update files through the relay web interface.	False
AuthorisationUsername	The user ID to use to secure the relay web interface. If not present or left blank, no log on security is enforced.	- (no logon security)
AuthorisationPassword	The password to use to secure the relay web interface.	- (no password)

Option	Description	Default
HtmlButtonLocation	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next / previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward.	Bottom
HtmlPageSize	The maximum number of lines to display on the web pages.	15
SuppressLogo	A true / false flag that allows suppression of the Sky Technologies logo on the relay web pages (to conserve bandwidth and improve performance).	False
UseHtmlCompression	A true / false flag that indicates whether you should use HTML compression for web pages sent to browsers which support it.	True
ReconnectSeconds	The time interval (in seconds) between attempts to reconnect if the relay connection is severed.	30

Storage

This section contains information on where and how TSX should cache information that you need to persistently store. For specific details on database support, refer to the Sky Java Server Database Guide.

Option	Description	Default
StorageType	The storage strategy to adopt. Must be one of the following: File - Should store persistent data as files within a local file system (this is the default behavior). Database - Should store persistent data in a relational database.	File
StorageRootDirectory	For use when the storage type is "File". This parameter tells TSX which directory is configured aside for its use in storing persistent data. Defaults to "." (the current working directory for the application or web server into which TSX is loaded).	As noted.
StorageDatabaseDriver	For use when the storage type is "Database". Should be configured to the fully qualified name of the JDBC driver to use in communicating with the storage database. The class in question must be available to the JVM in which TSX runs (that is, the class, or the jar/zip file containing the class, should be placed inside the TSX `classes` directory).	No default value.
StorageDatabaseURL	For use, when the storage type is "Database". The URL for TSX to use in locating the storage database. The precise format of this URL varies depending on the JDBC driver in use.	No default value.

Option	Description	Default
StorageDatabaseUser	For use when the storage type is "Database". The database user to use when establishing a connection to the storage database.	No default value.
StorageDatabasePassword	For use when the storage type is "Database". The password to use when establishing a connection to the storage database.	No default value.
StorageReservedWordProtection	For use when the storage type is "Database". This true / false flag helps circumvents problems with TSX column names that may happen to be reserved words for the database currently in use. When you enable, all database column names are automatically prefixed with an underscore ("_").	true
StorageSqlLogging	For use when the storage type is "Database". This true / false flag indicates whether or not SQL sent to the storage database should be written to the TSX log. This can be useful for debugging in some situations, but generates large log files and should generally be left off.	false
StorageHandlerAllocationTimeout	The maximum period of time (in seconds) that TSX waits for access to its persistent storage cache during any given operation. If a storage handler cannot be allocated within this time period then the operation in question fails.	10 (seconds)

Option	Description	Default
StorageHandlerInactivityTimeout	The maximum period of time (in seconds) that TSX keeps hold of an idle storage handler. If any given storage handler remains unused for this length of time, any associated database connection is closed, and the storage handler is then discarded.	60 (seconds)
MinimumNumberStorageHandlers	The minimum number of storage handlers that TSX keeps on hand for use by processes needing to access the persistent storage cache. This parameter defaults to zero (0) for the "File" storage strategy, and 3 for the "Database" storage strategy.	As noted
MaximumNumberStorageHandlers	The maximum number of storage handlers that TSX keeps on hand for use by processes needing to access the persistent storage cache. If configured to zero (0), no maximum is enforced. This parameter defaults to zero (0) for the "File" storage strategy, and 10 for the "Database" storage strategy.	As noted

CommandPort:xxx

These section(s) contain information on the TSX command port(s) that you use to accept incoming commands such as stopping, restarting, reporting status, and so on. Normally you need only one command port, but you can configure up more than one, if this is desired. You need to substitute the xxx with a unique meaningful name for the command port

Option	Description	Default
Port	The port number to listen on for TSX internal commands.	No default value (normally configured to 5090).

HttpPort:xxx

These section(s) contain parameters for TSX HTTP web status interface. Normally, you need only one HTTP

control port, but you can configure more than one, if this is desired. You need to substitute the xxx with a unique meaningful name for the HTTP control port

Option	Description	Default
Port	The port number to listen on for Web HTTP commands.	No default value (normally configured to 5091).
AllowConfigChangesFromWebPage	A true / false flag that controls whether you can modify the configuration file through the TSX administration web interface.	false
AuthorisationUsername	The user ID to use to secure the TSX web status page. If not present or left blank, no logon security is enforced when accessing the TSX web status page.	- (no logon security)
AuthorisationPassword	The password to use to secure the TSX web status page.	- (no password)
HtmlButtonLocation	Where to display the navigation buttons on the web pages. Values may be Top, Bottom, Both. If the size of the web page is large (see above), it may be advantageous to place the next / previous buttons at the top of the page to avoid scrolling to the bottom in-order to navigate forward.	Bottom
HtmlPageSize	The maximum number of lines to display on the web pages.	15

Option	Description	Default
UseHtmlCompression	A true / false flag that indicates whether you should send HTML compression to browsers that support it.	true
SuppressLogo	A true / false flag that allows suppression of the Sky Technologies logo on the TSX web status pages (to conserve bandwidth and improve performance).	false

MessagePort:xxx

These section(s) contain parameters for the port(s) on which TSX should listen for incoming message connections. You should substitute the xxxxx with a unique identifier

Option	Description	Default
Port	The port number on the local host to listen at for external messages.	No default value.

Option	Description	Default
InputMessageType	The expected format for incoming messages. This tells TSX when an incoming message is complete and thus ready to send to SAP. This parameter need to be one of the following: • D - Delimited input messages. Incoming messages are terminated with a specified delimiter. This is the default input mode. • F - Fixed length messages. Incoming messages are all the same length. • - The message length is embedded in incoming messages at a specified offset position from the start of the message. In other words, the message contains a numeric field at a known position that contains the length of the message itself. • U - Unformatted input. Any input received is sent to SAP immediately upon received. This generally results in a message sent for each incoming character, but is dependent on network behavior.	D

Option	Description	Default
InputDelimiters	Only used for input message type "D". Specifies the character sequence(s) that are to act as delimiters for incoming messages. You can specify more than one value by separating the various delimiters with commas. You can specify non-printable characters by entering a backslash followed by the 3-digit ASCII value of the character. Thus, for example, a new line (ASCII value 10) is entered as \010.	\010 (new line)
InputIgnored	Only used for input message type "D". Specifies any character sequence(s) that are to be ignored and removed from incoming messages. You can specify more than one value by separating the strings to ignore with commas. You can specify non-printable characters by entering a backslash followed by the decimal ASCII value of the character.	\013 (carriage return)
InputStripDelim	Only used for input message type "D". Specifies whether or not you need to strip delimiters from the message before it is sent to SAP.	true
InputFixedLength	Only used for input message type "F". Specifies the length of incoming fixed length messages.	No default value.
InputOffsetStart	Only used for input message type "O". Specifies the offset from the start of the message at which to find the message length within the message. For example, if the message length were to be contained in characters 4-8 of the message, the offset would be 3, since character position 4 is 3 characters in from character position 1 (the start of the message).	No default value.

Option	Description	Default
InputOffsetLength	Only used for input message type "O". Specifies the length of the message length field within the message. For example, if the message length were to be contained in characters 4-8 of the message, the length would be 5 (since it takes up character positions 4, 5, 6, 7, and 8, for a total of 5 characters).	No default value.
LogTsxMessages	A true/false flag that overrides the "LogTsxMessages" parameter in the [Log] section for this particular message port only. If left blank or not specified then the port's behavior is as specified in the [Log] section.	As specified in the [Log] section.
NotifyHost	True/false flag to indicate whether the host should be invoked when a message arrives. ***Note:*** NotifyHost must be explicitly configured to true in order to receive messages in SAP R/3.	false
HostInterface	Specifies the host interface that will be used to notify the host of incoming messages.	No default value.
StoreAndForward	A true/false flag that indicates whether storeandforward processing should be enabled. If the host is not available when store-and-forward processing is turned on, the transaction will be stored and forwarded to the host as soon as it becomes available.	false

Option	Description	Default
AsynchronousHostProcessing	A true/false flag that indicates whether processing on the host can be carried out independently of other processing taking place locally (user exits and system commands). If host processing is asynchronous then the TSX server performs local processing even if the host is not available. If host processing is synchronous, then local processing will only take place if and when the message can be forwarded to the host.	false
ShellCommand	The command that should be executed to invoke a system shell. Normally TSX is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh -s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by TSX, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
SystemCommand	A system command to be executed when a file arrives. If left blank or not configured, no system command is executed. The following substitution variables are available: • &ampersand& - An ampersand character • &connection& - Unique connection counter within each port from when TSX starts • &day& - The day of the month for example, 15 • &month& - The month of the year for example, 03 • &century& - The 4 digit year for example, 2000 • &hour& - The current hour • &minute& - The current minute • &msgdata& - The content of the message • &second& - The current second • &seqno& - A unique sequence number • &port& - The port on which the message was received • &year& - Two digit year for example, 00	null (no system command)

Option	Description	Default
InboundMessageExitClass	The fully qualified name of the user written Java exit to be invoked when an incoming message is detected. The class in question must be available to the JVM in which TSX is running (i.e. the class, or the jar/zip file containing the class, should be placed inside the TSX "classes" directory).	null (no exit)

RawFilePort:xxx

These section(s) contain parameters for the port(s) on which TSX should listen for incoming raw file transfers (that is, no guaranteed delivery protocol). You should substitute the xxx with a unique identifier.

Option	Description	Default
Port	The port number on the local host to listen at for incoming raw file transfers.	No default value.
FileDirectory	The directory in which to place transferred files. If not configured, this parameter defaults to "." (the current working directory).	As noted.

Option	Description	Default
FileNamePattern	The name to give the transferred file. You may build the name dynamically using standard substitution parameters: • &connection& - Unique connection counter within each port from 1 when TSX starts • &day& - The day of the month, for example, 15 • &month& - The month of the year, for example, 03 • &century& - The four digit year, for example, 2000 • &hour& - The current hour • &minute& - The current minute • &second& - The current second • &seqno& - A unique sequence number	No default value.
LogTsxMessages	A true / false flag that overrides the "LogTsxMessages" parameter in the [Log] section for this particular file port only. If left blank or not specified, then the port behavior is as specified in the [Log] section.	As specified in the [Log] section.

Option	Description	Default
NotifyHost	True / false flag to indicate whether you should notify the host when a file arrives. ***Important:*** NotifyHost need to explicitly configured to *true* in order to receive notification of incoming files in SAP R/3.	false
HostInterface	Specifies the host interface that you use to notify the host of incoming files.	No default value.
StoreAndForward	A true / false flag that indicates whether you should enable storeandforward processing. If the host is not available when store-and-forward processing is turned on, the transaction is stored and forwarded to the host as soon as it becomes available.	false
AsynchronousHostProcessing	A true / false flag that indicates whether you can carry out the processing on the host independently of other processing taking place locally (that is, user exits and system commands). If host processing is asynchronous, then the TSX server performs local processing even if the host is not available. If host processing is synchronous, then local processing only takes place if and when the you can forward the file notification to the host.	false

Option	Description	Default
ShellCommand	The command that should be executed to invoke a system shell. Normally TSX is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh -s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by TSX, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
SystemCommand	A system command to be executed when a file arrives. If left blank or not configured, no system command is executed. The following substitution variables are available: • &ampersand& - An ampersand character • &connection& - Unique connection counter within each port from when TSX starts • &filename& - The file name • &filedir& - The file directory (directory only) • &filepath& - The file path (directory and file name) • &day& - The day of the month, for example, 15 • &month& - The month of the year, for example, 03 • &century& - The 4 digit year, for example, 2000 • &hour& - The current hour • &minute& - The current minute • &second& - The current second • &seqno& - A unique sequence number • &port& - The port on which the file was received • &year& - Two digit year for example, 00	null (no system command)

Option	Description	Default
InboundFileExitClass	The fully qualified name of the user written Java exit to be invoked when an incoming file is detected. The class in question must be available to the JVM in which TSX is running (the class, or the jar/zip file containing the class, should be placed inside the TSX "classes" directory).	null (no exit)

FilePort:xxx

These section(s) contain parameters for the port(s) on which TSX should listen for incoming file transfers (using the guaranteed delivery protocol). You should substitute the xxx with a unique identifier.

Option	Description	Default
Port	The port number on the local host to listen at for guaranteed file delivery.	No default value.
FileDirectory	The directory in which to place transferred files. If not configured, this parameter defaults to "." (the current working directory).	As noted.

Option	Description	Default
FileNamePattern	The name to give the transferred file. The name may be built dynamically using standard substitution parameters: • &connection& - Unique connection counter within each port from 1 when TSX starts • &day& - The day of the month, for example, 15 • &month& - The month of the year, for example, 03 • &century& - The 4 digit year, for example, 2000 • &hour& - The current hour • &minute& - The current minute • &second& - The current second • &seqno& - A unique sequence number • &Port& - Current Port • &year& - Two digit year, for example, 00 If left blank or not configured, no filename pattern is used and any incoming file is given the same name it had on the host from which it was sent.	As noted.

Option	Description	Default
LogTsxMessages	A true / false flag that overrides the "LogTsxMessages" parameter in the [Log] section for this particular file port only. If left blank or not specified then the port behavior is as specified in the [Log] section.	As specified in the [Log] section.
NotifyHost	True/false flag to indicate whether the host should be notified when a file arrives. ***Note:*** You need to configured NotifyHost explicitly to *true* in order to receive notification of incoming files in SAP R/3.	false
HostInterface	Specifies the host interface that you use to notify the host of incoming files.	No default value.
StoreAndForward	A true / false flag that indicates whether you should enable storeandforward processing. If the host is not available when store-and-forward processing is turned on, the transaction is stored and forwarded to the host as soon as it becomes available.	false
AsynchronousHostProcessing	A true/false flag that indicates whether processing on the host can be carried out independently of other processing taking place locally (i.e. user exits and system commands). If host processing is asynchronous then the TSX server will perform local processing even if the host is not available. If host processing is synchronous, then local processing will only take place if and when the file notification can be forwarded to the host.	false

Option	Description	Default
ShellCommand	The command that should be executed to invoke a system shell. Normally TSX is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh -s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by TSX, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
SystemCommand	A system command to be executed when a file arrives. If left blank or not configured, no system command is executed. The following substitution variables are available: • &ampersand& - An ampersand character • &connection& - Unique connection counter within each port from when TSX starts • &filename& - The file name • &filedir& - The file directory (directory only) • &filepath& - The file path (directory and file name) • &day& - The day of the month, for example, 15 • &month& - The month of the year, for example, 03 • &century& - The 4 digit year, for example, 2000 • &hour& - The current hour • &minute& - The current minute • &second& - The current second • &seqno& - A unique sequence number • &port& - The port on which the file was received • &year& - Two digit year e.g. 00	null (no system command)

Option	Description	Default
InboundFileExitClass	The fully qualified name of the user written Java exit to be invoked when an incoming file is detected. The class in question must be available to the JVM in which TSX is running (the class, or the jar/zip file that contains the class, should be placed inside the TSX "classes" directory).	null (no exit)

XmlMessagePort:xxx

These section(s) contain parameters for the port(s) on which TSX should listen for incoming XML message connections. You should substitute xxxxx with a unique identifier.

Option	Description	Default
Port	The port number on the local host to listen at for external XML messages.	No default value.
LogTsxMessages	A true / false flag that overrides the "LogTsxMessages" parameter in the [Log] section for this particular message port only. If left blank or not specified then the port behavior is as specified in the [Log] section.	As specified in the [Log] section.
NotifyHost	True / false flag to indicate whether you should invoke the host when a message arrives. ***Note:*** You need to explicitly configure NotifyHost to *true*, in order to receive messages in SAP R/3.	false
HostInterface	Specifies the host interface that you use to notify the host of incoming messages.	No default value.

Option	Description	Default
StoreAndForward	A true / false flag that indicates whether you should enable storeandforward processing. If the host is not available when store-and-forward processing is turned on, the transaction is stored and forwarded to the host as soon as it becomes available.	false
AsynchronousHostProcessing	A true / false flag that indicates whether you can carry out processing on the host independently of other processing taking place locally (that is, user exits and system commands). If host processing is asynchronous, then the TSX server performs local processing even if the host is not available. If host processing is synchronous, then local processing only takes place if and when you can forward the message to the host.	false

Option	Description	Default
ShellCommand	The command that you should execute to invoke a system shell. Normally TSX is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh -s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by TSX, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
SystemCommand	A system command to be executed when a file arrives. If left blank or not configured, no system command is executed. The following substitution variables are available: • &ampersand& - An ampersand character • &connection& - Unique connection counter within each port from when TSX starts • &day& - The day of the month e.g. 15 • &month& - The month of the year e.g. 03 • &century& - The 4 digit year e.g. 2000 • &hour& - The current hour • &minute& - The current minute • &msgdata& - The content of the message • &second& - The current second • &seqno& - A unique sequence number • &port& - The port on which the message was received • &year& - Two digit year e.g. 00	null (no system command)

Option	Description	Default
InboundMessageExitClass	The fully qualified name of the userwritten Java exit to be invoked when an incoming message is detected. The class in question must be available to the JVM in which TSX is running (the class, or the jar/zip file containing the class, should be placed inside the TSX "classes" directory).	null (no exit)

Connection:xxx

These section(s) contain parameters for the connection(s) that TSX should actively establish to listen for incoming messages. You should substitute the xxx with a unique identifier.

Option	Description	Default
Address	The IP address for a direct connection.	No default value.
Port	The port at the IP address to listen at for external messages.	No default value.
LogTsxMessages	A true / false flag that overrides the "LogTsxMessages" parameter in the [Log] section for this particular connection only. If left blank or not specified then the connection behavior is as specified in the [Log] section.	As specified in the [Log] section.
NotifyHost	True / false flag to indicate whether the host should be notified when a message arrives. ***Note:*** NotifyHost must be explicitly configured to *true* in order to receive messages in SAP R/3.	false

Option	Description	Default
HostInterface	Specifies the host interface that you use to notify the host of incoming messages.	No default value.
StoreAndForward	A true / false flag that indicates whether you should enable storeandforward processing. If the host is not available when store-and-forward processing is turned on, the transaction is stored and forwarded to the host as soon as it becomes available.	false
AsynchronousHostProcessing	A true / false flag that indicates whether you can carry out the processing on the host independently of other processing taking place locally (that is, user exits and system commands). If host processing is asynchronous then the TSX server performs local processing even if the host is not available. If host processing is synchronous, then local processing only takes place if and when the message can be forwarded to the host.	false

Option	Description	Default
ShellCommand	The command that you should execute to invoke a system shell. Normally TSX is able to correctly detect the host operating system and assign this parameter an appropriate default value. In some circumstances, however, it may be necessary to override the default value in order to have certain print commands work correctly. Default values are as follows: Windows: command.com /c Unix: sh -s If you are running under Windows 2000 or Windows XP, and find that the print command you are executing works correctly from the command line, but does not work when invoked by TSX, try changing this parameter to "cmd.exe /c".	As noted.

Option	Description	Default
SystemCommand	A system command to execute when a file arrives. If left blank or not configured, no system command is executed. The following substitution variables are available: • &ampersand& - An ampersand character • &connection& - Unique connection counter within each port from when TSX starts • &day& - The day of the month, for example, 15 • &month& - The month of the year, for example, 03 • &century& - The four digit year, for example, 2000 • &hour& - The current hour • &minute& - The current minute • &msgdata& - The content of the message • &second& - The current second • &seqno& - A unique sequence number • &port& - The remote port from which the message was received • &year& - Two digit year, for example 00	null (no system command)

Option	Description	Default
InboundFileExitClass	The fully qualified name of the user written Java exit to invoke when an incoming message is detected. The class in question must be available to the JVM in which TSX is running (that is, you should place the class, or the jar/zip file containing the class, inside the TSX `classes` directory).	null (no exit)
StartupSeconds	Overrides the "StartupSeconds" parameter in the [General] section for this particular connection only. If left blank or not specified then the connection behavior is as specified in the [General] section.	As specified in the [General] section.
ReconnectSeconds	Overrides the "ReconnectSeconds" parameter in the [General] section for this particular connection only. If left blank or not specified then the connection behavior is as specified in the [General] section.	As specified in the [General] section.

FTC Transfer Client

When you invoke FTC, it searches its configuration file (`ftc.cfg`) for the logical name specified. When you find a match, it uses the IP, port and other attributes to establish a connection to a remote TSX Server. You need to perform FTC configuration in both the local `ftc.cfg` file and the remote `tsx.cfg` file. This configuration outlines the IP address and port to transmit data to, that is, FTC is the sender, TSX is the receiver. On the TSX side, you need to configure the port as a file port.

A typical FTC configuration file looks something like this:

FTC configuration

```
[Log]
LogEnabled = true
LogFileDirectory = ./log
LogStrategy = D
ExcludeLogMessageTypes =
[Destination:5083]
Address = LOCALHOST
Port = 5083
Compressed = false
Encrypted = false
UseTransferProtocol = true
DeleteOnceSent = false
DeleteDirectories = false
ResponseTimeout = 60
```

Log

This section contains information pertaining to FTC logging capabilities.

Option	Description	Default
LogFileDirectory	The full pathname to the directory FTC should write its log files. If not set, this parameter defaults to "." (the current working directory).	As noted.
LogStrategy	The logging strategy to adopt. Must be one of the following: D - A new log file to create daily. W - A new log file to create weekly. P - A new log file to create for each process instance of FTC. O - Overwrite the log file each time FTC is restarted.	O (overwrite)
LogTsxMessages	A true / false flag that indicates whether you should write incoming messages to the log file. You should set this parameter to *off*, if log files become excessively large.	true
ExcludeLogMessageTypes	Indicates the types of log messages that should be excluded from the log file. There are 4 types of log messages - Information (I), Trace (T), Error (E) and Abort (A). The log message types to exclude should all be concatenated together. Thus, to exclude all Information and Trace messages from the log file, the appropriate setting would be "IT".	null (log all message types)
MaximumNumberLogFiles	This setting restricts the maximum number of log files TSX keeps under D, W, and P strategies. If left blank or not set then no maximum is enforced.	0 (no maximum)

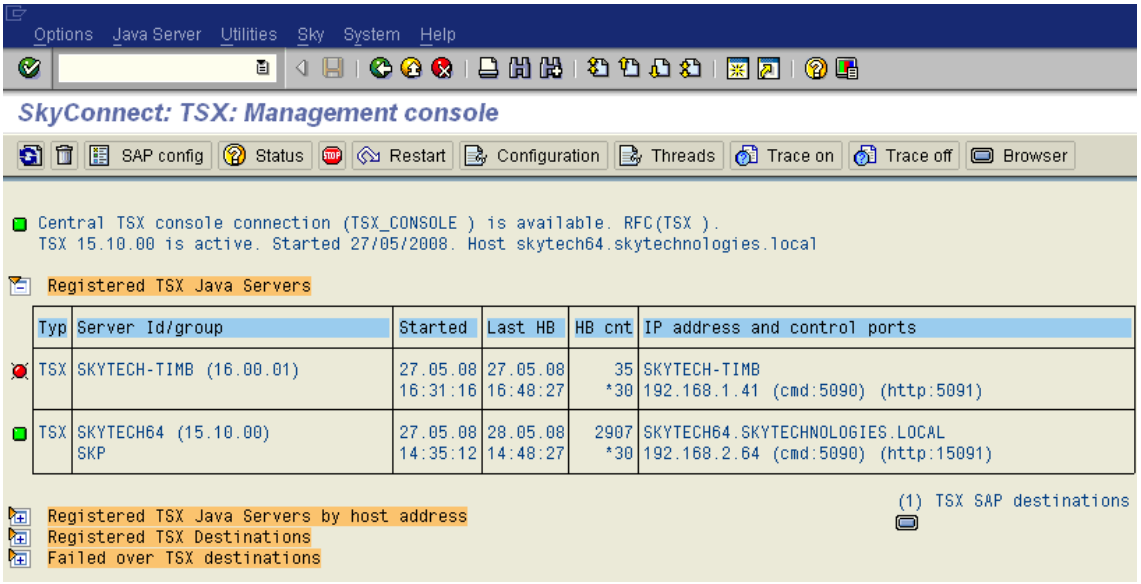
Destination:xxx

These section(s) contain information pertaining to FTC logical file destination(s). You should substitute the xxx with a unique identifier.

Option	Description	Default
Address	The IP address where the target TSX Java server executes.	No default value.
Port	The file or raw file port number on which TSX is listening.	No default value.
Compressed	True / false flag that indicates whether you should use compression should during the data transfer.	false
Encrypted	True / false flag that indicates whether you should use triple-DES encryption during the data transfer.	false
UseTransferProtocol	True / false flag that indicates whether you should use the TSX guaranteed delivery protocol.	true
DeleteOnceSent	True / false flag that indicates whether FTC should delete the file once it is successfully transferred.	false
DeleteDirectories	True / false flag that indicates whether FTC should delete directories once their contents is successfully transferred.	false
ResponseTimeout	The maximum time (in seconds) that FTC should wait for a connection or response from TSX.	30 (seconds)

6.17.3.4 TSX SAP Console and Workbench

You use the SAP TSX console and workbench to monitor TSX SAP connections and define SAP logical destinations and message types. It is started using transaction /SKY/YTSX (or the shortened version YTSX). The TSX console appears:



From this screen, all the currently connected TSX servers are listed. The following options are available:

- Issue commands to connected TSX Java servers
- Status
- Stop
- Restart
- Configuration
- Threads
- Trace on / off

- Maintain the SAP logical destinations and message types.
- Invoke the TSX Java server web status page.

> ***Important:*** You may only issue commands to external TSX servers if you defined the TSX_CONSOLE logical destination to connect to an existing TSX Java server on the network. By default, this is created connecting to the TSX_TEST SAP RFC destination.

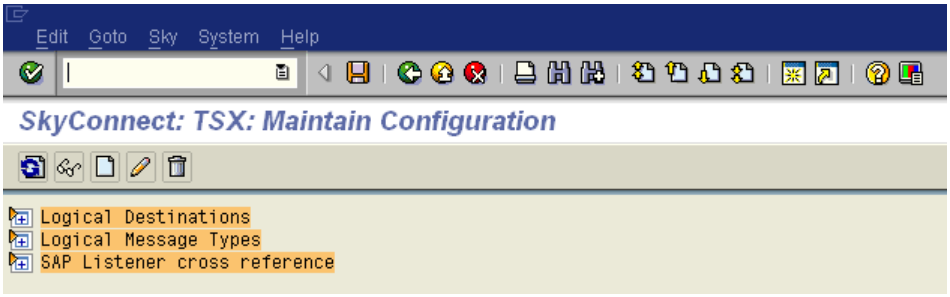
SAP Workbench

You can access the TSX SAP Workbench component from the TSX main console by either:

- Selecting the **TSX SAP configuration** option from the **Utility** menu
- Clicking the **SAP config**  button on the application toolbar

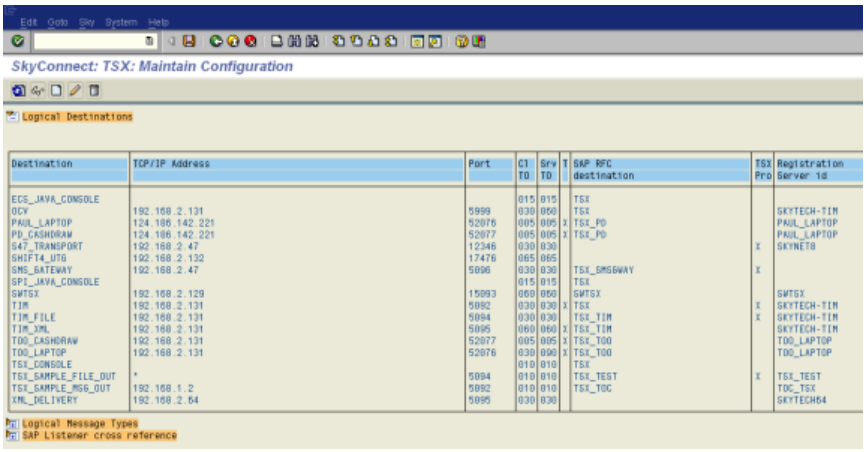
The workbench is used to define:

- Logical Destinations
- Logical Message Types
- Listener Configuration

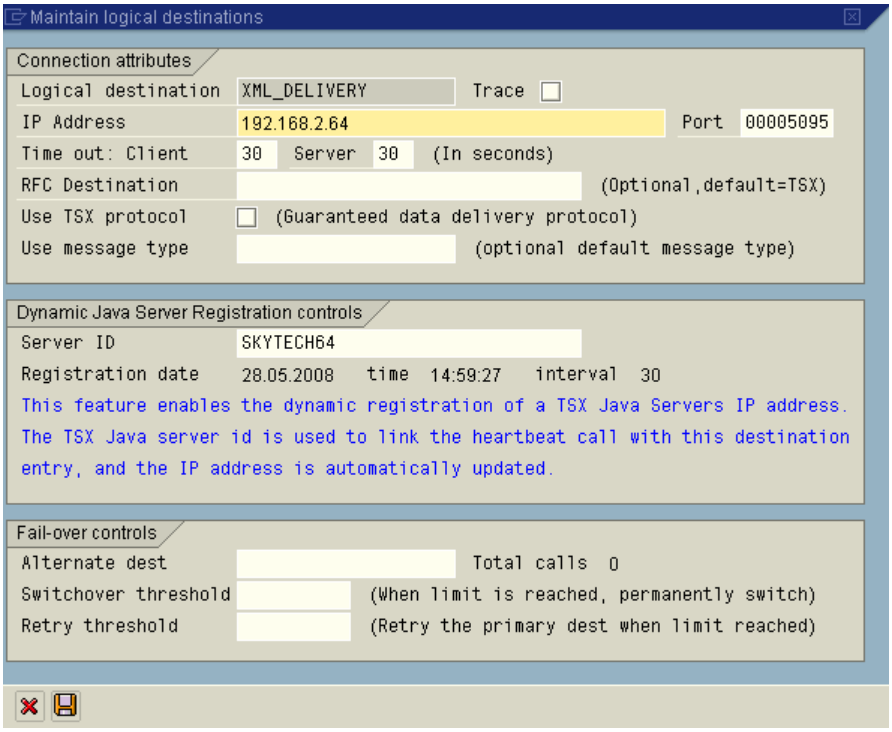


How to Maintain Logical Destinations

To maintain a logical destination, navigate to the TSX SAP workbench and expand the **Logical Destinations** group.
A list of existing destinations appear:



You maintain logical destination definitions in the list by double-clicking an existing entry or using the create, change, delete buttons on the application toolbar. The following configuration attributes are available:

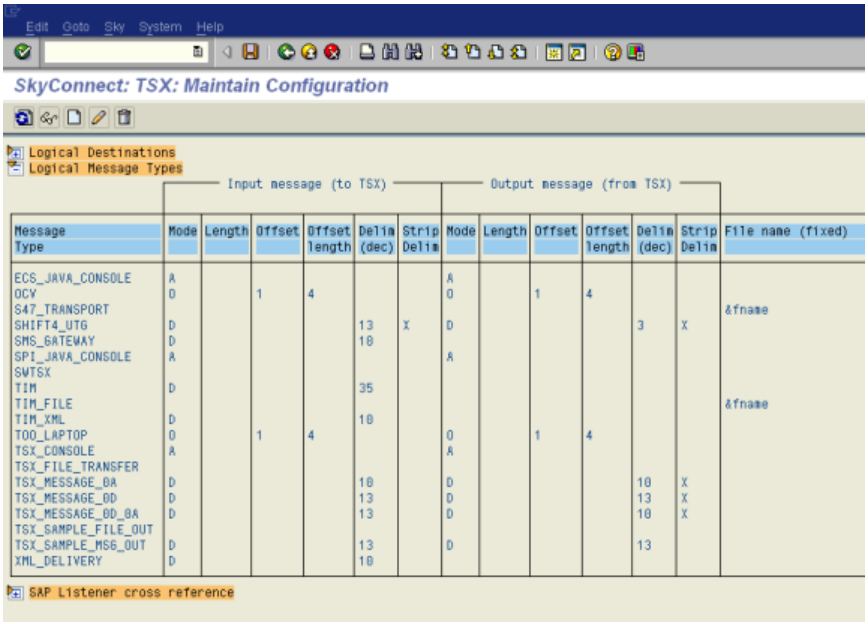


Field	Description
Logical Destination	Unique name for the destination
Trace	Enable tracing for this destination. Default is *false*.
IP Address	IP address that this destination points to. ***Note:*** The IP address is automatically updated if the Dynamic server ID is populated (see below).
Port	Port to send messages to
Timeouts	Timeout in seconds for server and client connections
RFC Destination	The RFC destination (defined in SAP transaction SM59) to use to connect to the logical destination. If left blank, it defaults to "TSX"

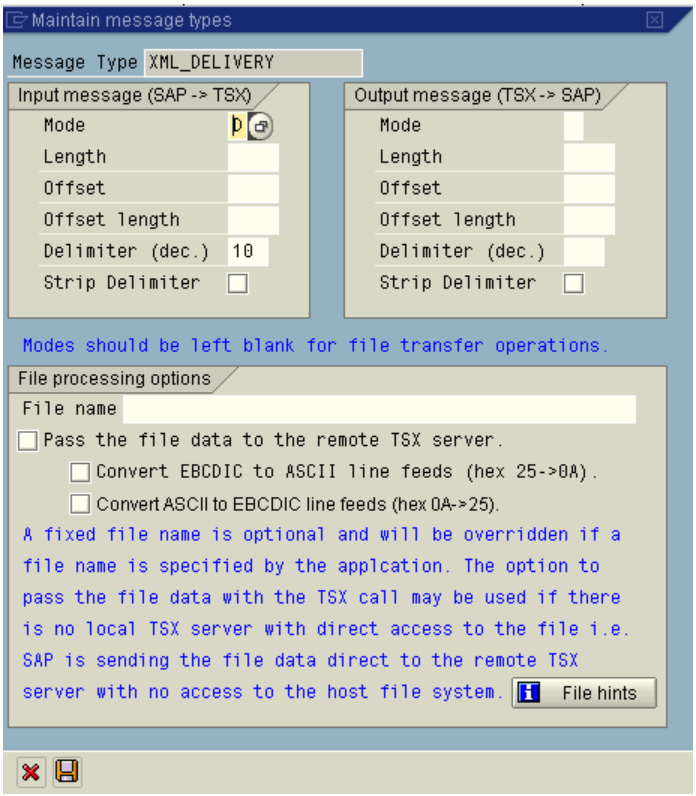
Field	Description
Use TSX Protocol	Whether to use TSX guaranteed delivery protocol (default). You need to disable this if using a Raw file port.
Use Message Type	Specify a specific message type to use. If left blank, a message type of the same name as the destination is looked for.
Server ID	The TSX server ID that this destination points to. This is optional. The benefit of specifying the server is that the IP address field is automatically updated when the TSX server's heartbeat comes through. This allows machines with DHCP allocated IP addresses to support.
Alternate Destination	An alternate logical destination to use if this logical destination is not available (that is, times out).
Switchover Threshold	Number of times to try the primary destination before switching to the failover destination.
Retry Threshold	Number of times using the fail over destination before reverting to the primary.

How to Maintain Logical Message Types

To maintain a logical message type, navigate to the TSX SAP workbench and expand the **Logical Message Types** group. A list of existing message types appears:



You maintain logical message type definitions in the list by double-clicking an existing entry or using the create, change, delete buttons on the application toolbar. The following configuration attributes are available:



Field	Description
Message Type	Unique name for the message type
Mode	Mode of operation: • Default - message delimited by space / null • A - message delimited when socket closed • D - specific delimiter character • L - fixed length • O - fixed length, message length specified inside message
Length	When mode L specified, length of the message (in characters) must be provided

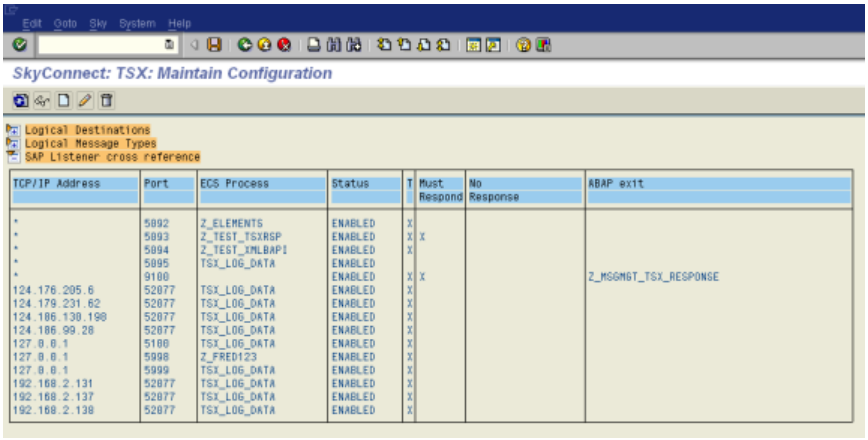
Field	Description
Offset	When mode O specified, the offset (in characters) from start of message that the length is specified
Offset Length	When mode O specified, the length from the offset that the length of the message is specified
Delimiter	When mode D specified, the decimal notation for the delimiter character
Strip delimiter	When mode D specified, whether to remove the delimiter from the end of the message before passing the message on
File Name	Name of the file to transmit to the remote server.
Pass Data	You need to enable this to send the contents of the file.
Conversion	Character set conversions. This is required when transmitting between ASCII and EBCDIC systems.

Note: You can get the further details by clicking the **Hints** Hint button.

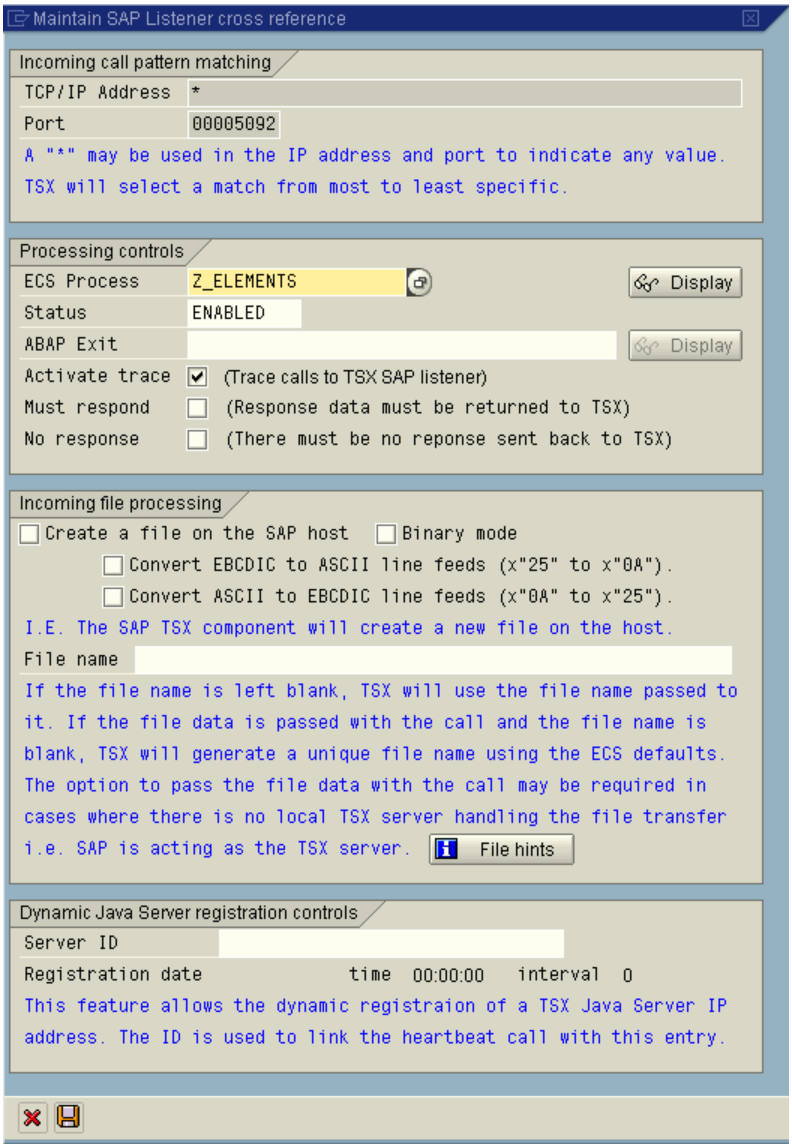
How to Maintain Listener Cross References

SAP Listener cross reference entries determine what ECS Process and / or ABAP program is going to invoke and pass the data. This is based on the senders IP address and / or port.

You may use generic "*" values, for example, anything received by a TSX Java server on port 15096 invokes ECS Process "TSX_RECEIVE_FILE".



You maintain SAP Listener cross reference entries in the list by double-clicking an existing entry or using the create, change, delete buttons on the application toolbar. The following configuration attributes are available:



Field	Description
TCP/IP Address	IP address (or wildcard *) that applies
Port	Port number (or wildcard *) that applies
ECS Process	ECS process to start when message received

Field	Description
Status	Whether the rule is enabled / disabled
ABAP Exit	Optional ABAP exit to trigger.
Trace	Whether to trace entries received
Must respond	Indicates whether the message expects a response. If so, the response data component in TSX must be populated by the ECS process / exit.
No response	Indicates that no response can be sent back
Create file	Indicates that if a file is passed in whether to save the file on the SAP host
File name	Override for the file name
Binary mode	Whether to operate in binary mode rather than the default character mode. Note this will
Conversion	Character set conversions. This is required when transmitting between ASCII and EBCDIC systems.
Server ID	The server id of the TSX server that sends in the message/file. This is optional. The benefit of specifying the server is that the IP address field is automatically updated when the heartbeat of the TSX server comes through. This allows machines with DHCP allocated IP addresses to be supported.

Note: If no cross reference entry is found, TSX automatically creates one and submits a default ECS process "TSX_LOG_DATA" to list what is sent.

SAP Listener ABAP Exit

It is a common requirement that you may decide on what to execute, based on the content of the data. A user exit is provided where you can imbed custom code into the process to receive control, validate / manipulate the data and / or influence subsequent processing. A sample exit is provided with the TSX installation and more details in this guide.

How to Maintain SAP RFC Destinations

A SAP RFC destination is only required for TSX if you propose to perform calls from SAP to an external TSX Java server.

To support this, the TSX Java server must have a SAP host interface server configuration section that instructs it to register itself at the SAP Gateway using a configured name (for example, FRED). You need to set up a corresponding SAP RFC TCP/IP connection using SM59 so that you can redirect the TSX logical destinations to it.

TSX Java Configuration

> ***Note:*** TSX Java Configuration page contains macros or features from a plugin that requires a valid license. Contact your administrator.

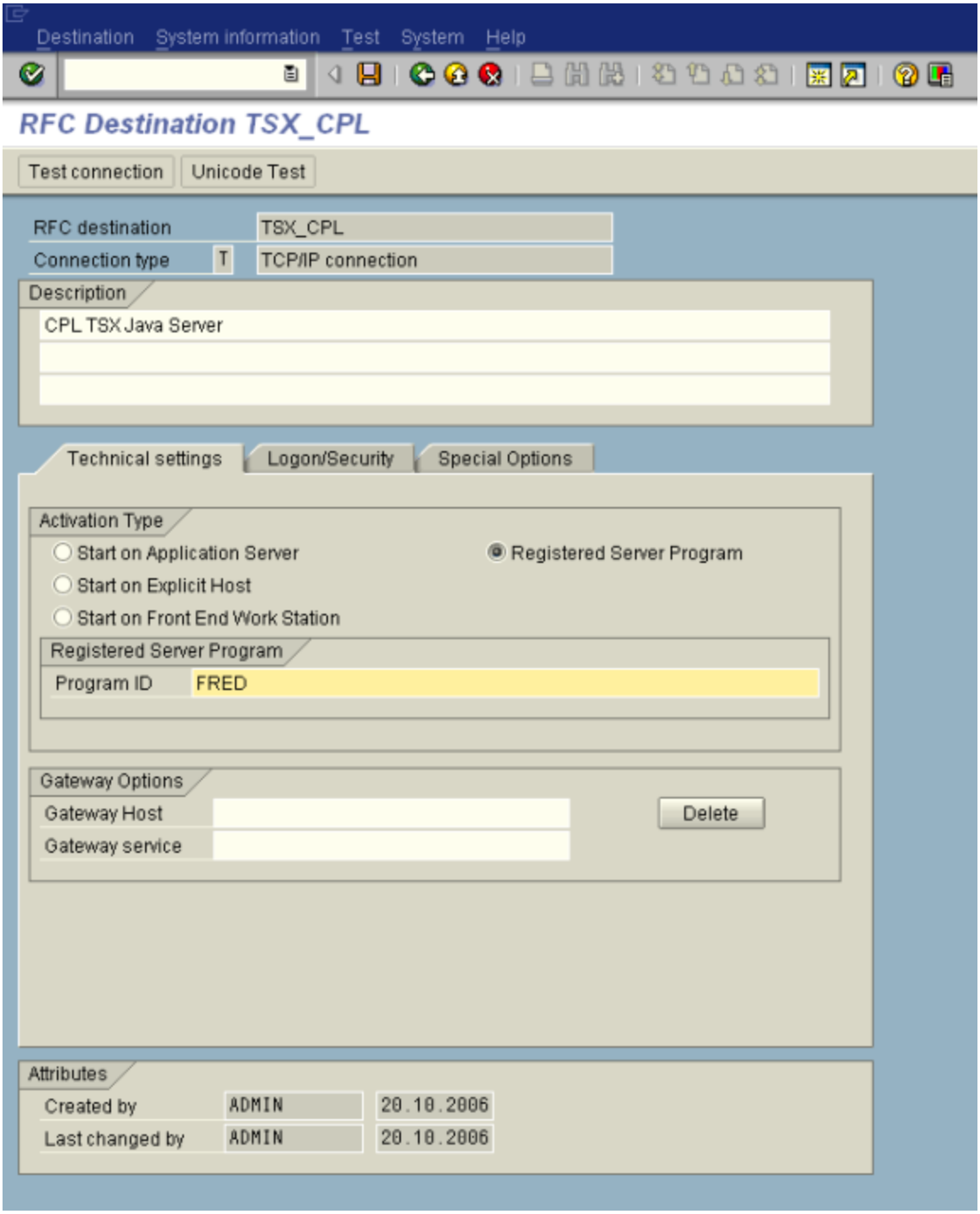
To register with SAP for outbound interfaces, the TSX server needs configuration for an RfcServer host interface. Below is an example of an RfcServer host interface definition. For full details on the `tsx.cfg` file, refer to the

```
[HostInterface:RfcServer]
HostInterfaceType = RfcServer

SapRegistrationName = UNIQUE_REGISTRATION_NAME

SapHost = server

SapGateway = sapgw00
```

> ***Important:*** The registration name need to be unique for each TSX Java server. Duplicates may result in unpredictable behavior.

SAP RFC Configuration

SAP RFC destinations for TSX are configured under TCP/IP connections. You are recommended a name prefixed with TSX_. The TSX SAP API looks for a RFC destination called TSX, by default. In normal cases where only one TSX server is required, create a single RFC destination called TSX. If you require multiple TSX server connections, you require a separate RFC connection for each registration. The application chooses which destination to use; either directly, or through the TSX logical destinations table.



You need to configure the RFC destination as a connection type "T" (TCP/IP) and need to be in "registration" mode. Refer the following example.

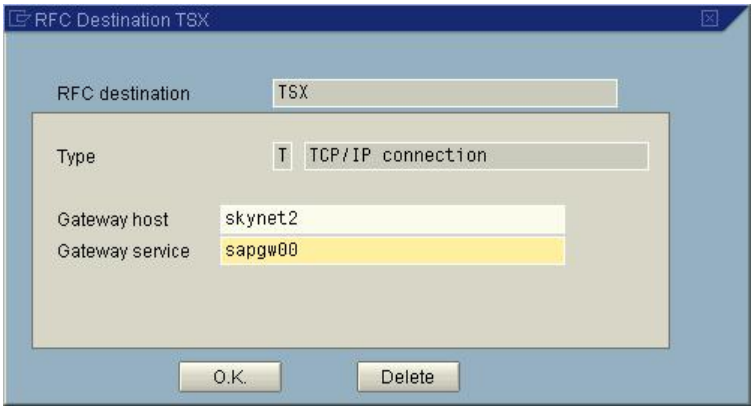
> ***Note:*** The Program ID name used above corresponds to the matching SapRegistrationName option in the TSX Java configuration, that is, this is how they are linked.

> ***Important:*** The Program ID name is case sensitive. FRED is different from FrEd.

Multi Instance SAP Environments

In cases where there are multiple instances (hosts) in a SAP system, you need to configure the RFC destination to direct calls to a specific SAP gateway where the TSX servers are registered. To do this:

- For ECC release, populate the gateway host field with the gateway hostname
- For 4.5/4.6 releases select the **Gateway options** option from the Destination menu and the following dialog appears:

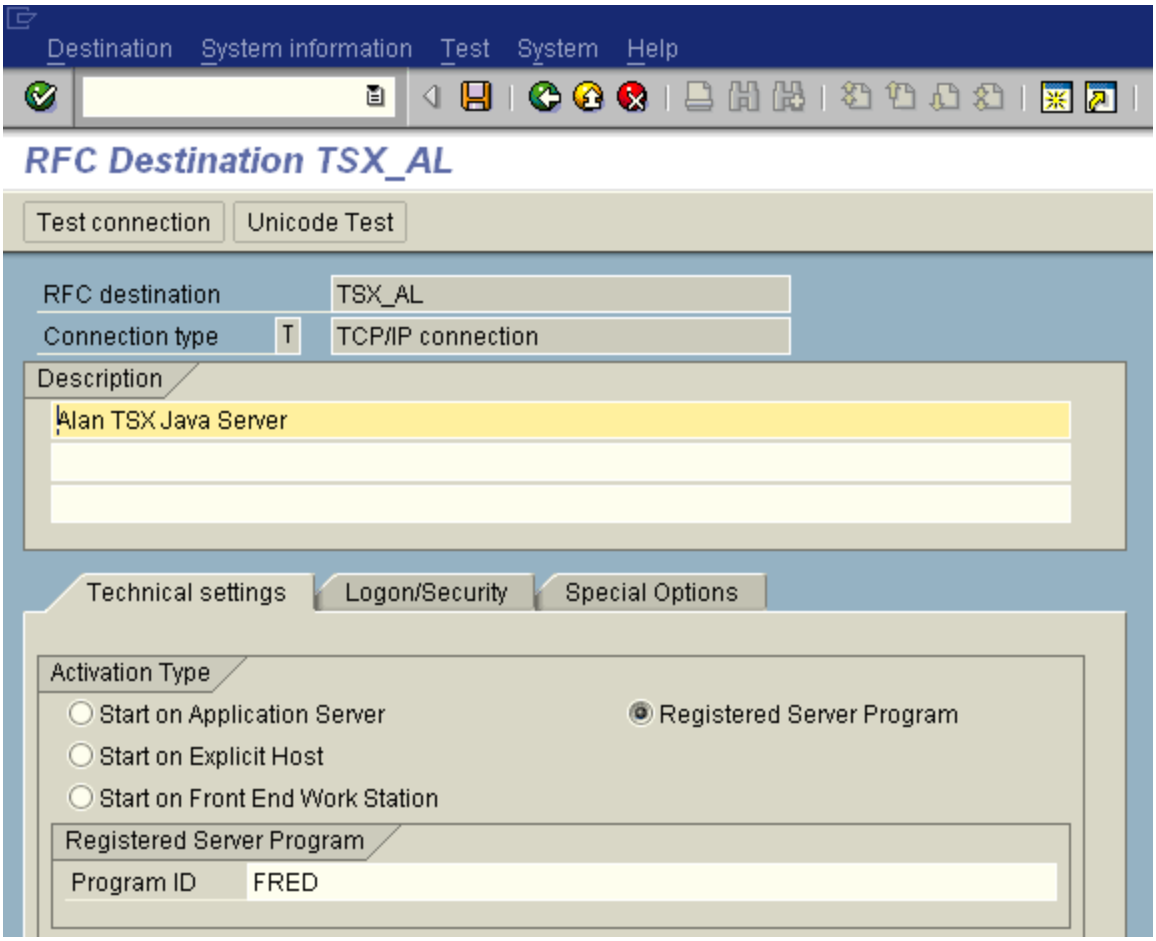


Specify the specific host and gateway service where the TSX Java servers are registered. Once you do this, a gateway button appears on the main screen.

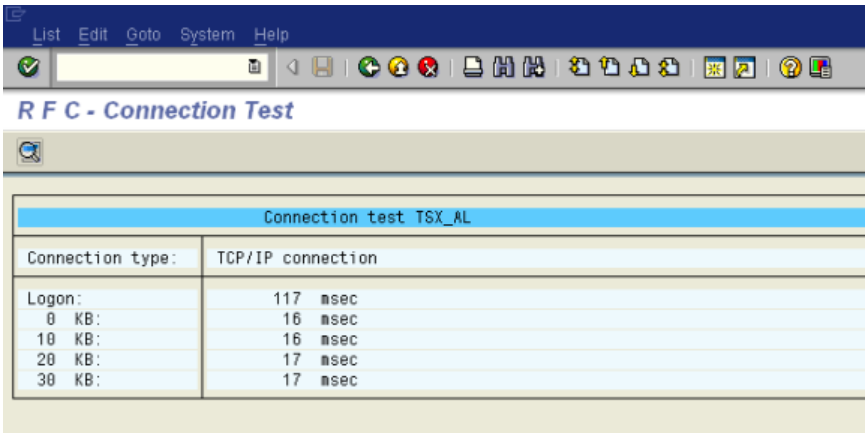
Testing the SAP RFC Connection

6.17.3.5 RFC Connection Test

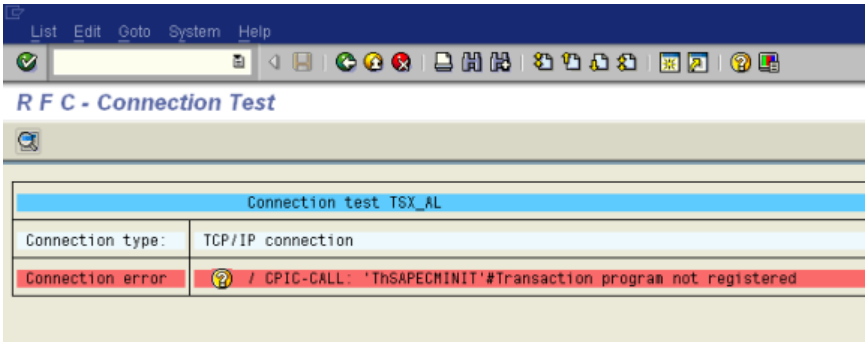
To do this, you need to have started TSX and must have registered at the gateway. Select the **Test connection** button from the application toolbar of the main RFC Configuration screen.



If the test is successful, then the following screen appears:

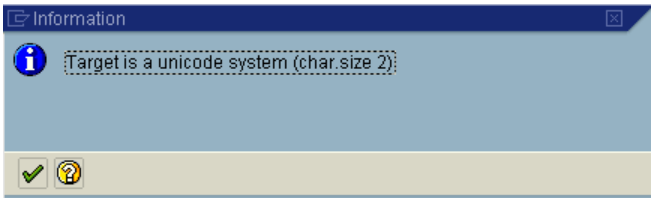


If the test fails, then the following screen appears:

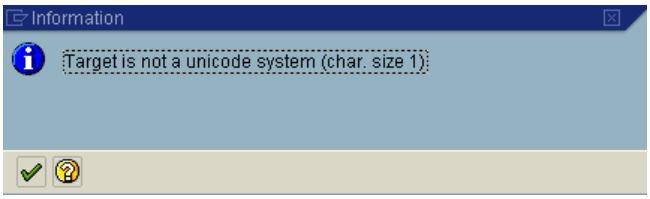


6.17.3.6 Unicode Connection Test

For Unicode SAP environments, a Unicode test is also available. If the TSX java server is configured to operate in Unicode mode.

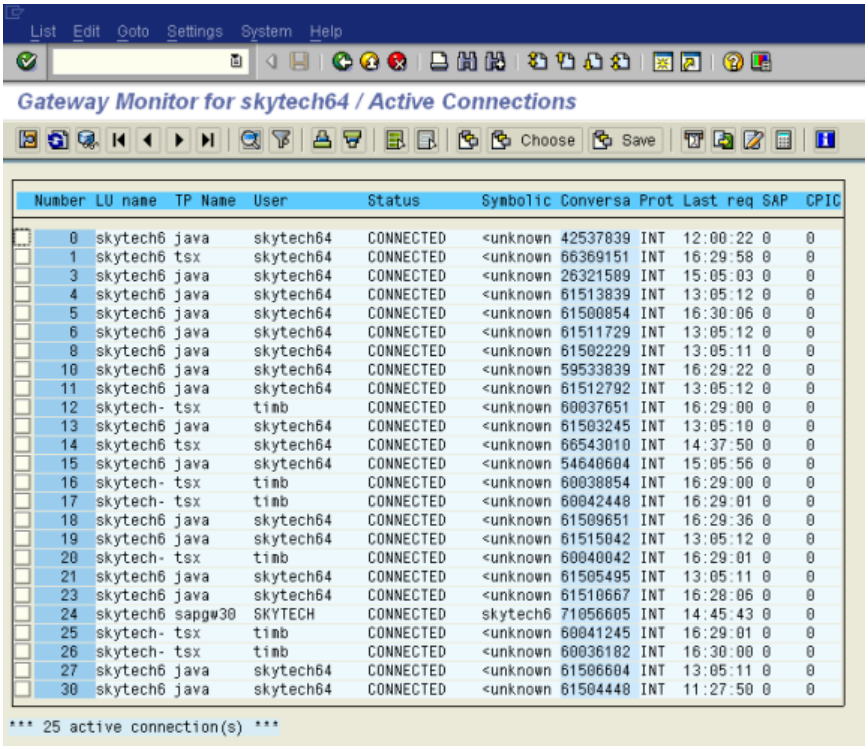


If the TSX java server does not run in Unicode mode, the following message appears:



Monitoring the SAP RFC Gateway

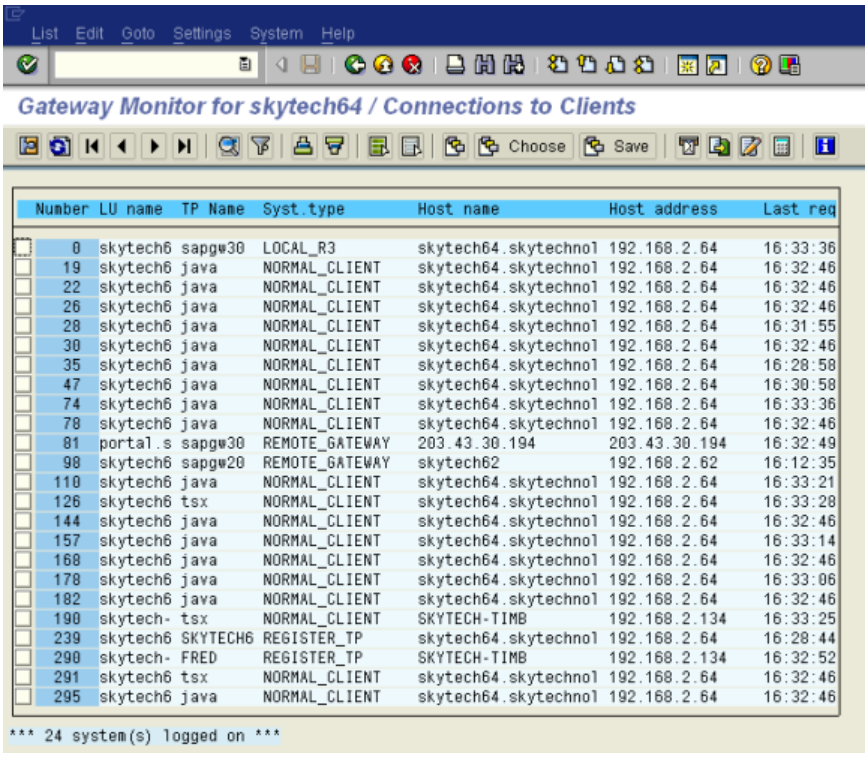
SAP R/3 transaction SMGW displays active RFC client and registered connections for a specific R/3 gateway. In a multi-instance SAP environment, you need to log onto the specific SAP host where the gateway executes. Execute SMGW and the following screen appears:



Number	LU name	TP Name	User	Status	Symbolic	Conversa	Prot	Last req	SAP	CPIC
0	skytech6	java	skytech64	CONNECTED	<unknown	42537839	INT	12:00:22	0	0
1	skytech6	tsx	skytech64	CONNECTED	<unknown	66369151	INT	16:29:58	0	0
3	skytech6	java	skytech64	CONNECTED	<unknown	26321589	INT	15:05:03	0	0
4	skytech6	java	skytech64	CONNECTED	<unknown	61513839	INT	13:05:12	0	0
5	skytech6	java	skytech64	CONNECTED	<unknown	61500854	INT	16:30:06	0	0
6	skytech6	java	skytech64	CONNECTED	<unknown	61511729	INT	13:05:12	0	0
8	skytech6	java	skytech64	CONNECTED	<unknown	61502229	INT	13:05:11	0	0
10	skytech6	java	skytech64	CONNECTED	<unknown	59533839	INT	16:29:22	0	0
11	skytech6	java	skytech64	CONNECTED	<unknown	61512792	INT	13:05:12	0	0
12	skytech-	tsx	timb	CONNECTED	<unknown	60037651	INT	16:29:00	0	0
13	skytech6	java	skytech64	CONNECTED	<unknown	61503245	INT	13:05:10	0	0
14	skytech6	tsx	skytech64	CONNECTED	<unknown	66543010	INT	14:37:50	0	0
15	skytech6	java	skytech64	CONNECTED	<unknown	54640604	INT	15:05:56	0	0
16	skytech-	tsx	timb	CONNECTED	<unknown	60038854	INT	16:29:00	0	0
17	skytech-	tsx	timb	CONNECTED	<unknown	60042448	INT	16:29:01	0	0
18	skytech6	java	skytech64	CONNECTED	<unknown	61509651	INT	16:29:36	0	0
19	skytech6	java	skytech64	CONNECTED	<unknown	61515042	INT	13:05:12	0	0
20	skytech-	tsx	timb	CONNECTED	<unknown	60040042	INT	16:29:01	0	0
21	skytech6	java	skytech64	CONNECTED	<unknown	61505495	INT	13:05:11	0	0
23	skytech6	java	skytech64	CONNECTED	<unknown	61510667	INT	16:28:06	0	0
24	skytech6	sapgw30	SKYTECH	CONNECTED	skytech6	71056605	INT	14:45:43	0	0
25	skytech-	tsx	timb	CONNECTED	<unknown	60041245	INT	16:29:01	0	0
26	skytech-	tsx	timb	CONNECTED	<unknown	60036182	INT	16:30:00	0	0
27	skytech6	java	skytech64	CONNECTED	<unknown	61506604	INT	13:05:11	0	0
30	skytech6	java	skytech64	CONNECTED	<unknown	61504448	INT	11:27:50	0	0

*** 25 active connection(s) ***

This screen displays client connections. In the above, display, TSX is clearly marked in the TP Name field. This indicates that TSX has a client connection to the gateway. To display registered RFC servers, select the **logged on systems** option from the **goto** menu. The following screen appears:



Number	LU name	TP Name	Syst.type	Host name	Host address	Last req
0	skytech6	sapgw30	LOCAL_R3	skytech64.skytechnol	192.168.2.64	16:33:36
19	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
22	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
26	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
28	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:31:55
30	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
35	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:28:58
47	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:30:58
74	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:33:36
78	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
81	portal.s	sapgw30	REMOTE_GATEWAY	203.43.30.194	203.43.30.194	16:32:49
98	skytech6	sapgw20	REMOTE_GATEWAY	skytech62	192.168.2.62	16:12:35
110	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:33:21
126	skytech6	tsx	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:33:28
144	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
157	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:33:14
168	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
178	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:33:06
182	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
190	skytech-	tsx	NORMAL_CLIENT	SKYTECH-TIMB	192.168.2.134	16:33:25
239	skytech6	SKYTECH6	REGISTER_TP	skytech64.skytechnol	192.168.2.64	16:28:44
290	skytech-	FRED	REGISTER_TP	SKYTECH-TIMB	192.168.2.134	16:32:52
291	skytech6	tsx	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46
295	skytech6	java	NORMAL_CLIENT	skytech64.skytechnol	192.168.2.64	16:32:46

*** 24 system(s) logged on ***

TSX RFC server connections are identified by their registered name, for example, "FRED" in the tp name and REGISTER_TP in system type.

> ***Note:*** A single TSX server may open multiple connections, that is, multi-threaded. Double-click the entry to view the full registration details and thus identify the TSX server parent.

TSX Systems Management

In TSX Systems Management, you can perform the following :

Starting Stopping and Checking TSX

Troubleshooting TSX

Starting Stopping and Checking TSX

For information on how to start, stop, restart or check the status of a TSX Java server, refer the following:

- Windows Platforms
- Unix Platforms

Troubleshooting TSX

> ***Note:*** For troubleshooting regarding installing and starting a TSX Java server, see the general Sky Installation Guide.

Summary of Diagnostic Facilities

- TSX trace facility
- SAP RFC Trace
- Java JNI trace
- SAP RFC test connection

TSX Java Server Log

TSX keeps an audit trail of all connections and transmissions in the TSX log directory. This log is useful to view the status of TSX and any transmission problems that occurred. You can view the log from the web status page or through the latest log file in the log directory.

File / Directory Transfer

It is a common problem (at first) that TSX or FTC are either not authorized to read or create files / directories. The user that starts the service, need to be authorized to all the files and directories specified. It is strongly advised that all authorizations are thoroughly reviewed and tested prior to 'go live'.

TSX Trace Facility

For outbound (SAP >Java) calls, each OPEN or "$" command to TSX is given a unique session ID. This session ID is used to uniquely identify a trace file that you may use to diagnose communication and data problems with the interface. The trace is activated by specifying 'X' (true) in the logical

destination configuration or explicitly through the parameter-trace parameter to /sky/tsx_socket_call. The trace is written to the log directory configured in the `tsx.cfg` file. For incoming (Java >SAP) calls, the trace option in the "SAP Listener Cross Reference" causes TSX to write trace entries that may be viewed by the **SAP Listener trace** option from the **Utilities** menu on the TSX console display.

TSX Java Server SAP RFC Trace

This is a standard SAP RFC trace. It is activated using the SapTrace flag in the `tsx.cfg` configuration file.

> ***Note:*** You should use this trace should only under the direction of Sky Support.

Java JNI Trace

You may invoke a low level trace of the Java JNI interface. JNI tracing is initiated if the JniLogging flag is set to *true* in the `tsx.cfg` file. Comment out the parameter or set the field to *false* to deactivate the trace. The Java JNI trace is intended for Sky internal use only and you should only use under the direction of Sky support.

TSX ABAP Programming Guide

All conversations are initiated from SAP using the /SKY/TSX_SOCKET_CALL function. You use this function to call the TSX sockets exchange server to communicate with an external application through the TCP/IP sockets protocol. The /SKY/YECS_TSX structure defines a common parameter area that contains all the attributes of the call. The API is designed to be as simple as possible. The ABAP calls the function to open a connection, write messages / files, receive data and close the connection once it is finished, similar in manner to processing a file. A standard parameter area is provided (through DD structure /sky/yecs_tsx). Before you call TSX, you should have configured a logical destination and a message type to use. These define the external TSX server you need to communicate with and how to handle the data.

Calling TSX Outbound from SAP

The ABAP program calls the TSX function module issuing one or more commands to transfer data. Each connection is allocated a unique session id. This is used by TSX to identify which sockets connection you refer to. Any information or error messages are communicated through the MSG* variables. You may have as many concurrent sockets connections as you wish. The data area is a max of 4096 bytes and the lengths of the send / receive data areas may be explicitly specified or determined dynamically by TSX. You may leave a socket connection open for as long as required and there is no limit to the number of data exchanges within a connection.

```
DATA: V_PARAMETER     LIKE YECS_TSX.

CLEAR V_PARAMETER.
V_PARAMETER-COMMAND = '$WRITE'.
V_PARAMETER-DATA = 'Test message'.
CALL FUNCTION '/SKY/TSX_SOCKET_CALL'
       EXPORTING
            I_DESTINATION               = 'ECS_TEST_TSX'
            I_MESSAGE_TYPE           = 'ECS_TEST_TSX'
            I_PARAMETER                 = V_PARAMETER
       IMPORTING
            E_PARAMETER                = V_PARAMETER
       EXCEPTIONS            SESSION_ID_ALLOCATION_FAILED    = 1
            INVALID_SOCKET_DESTINATION        = 2
            RFC_CALL_TO_TSX_FAILED               = 3
            INVALID_COMMAND                           = 4
            INVALID_MESSAGE_TYPE                   = 5
            INVALID_RECORD_LENGTH_MODE    = 6
            INVALID_RECORD_LENGTH_OPTION = 7
            OTHERS
= 8.

  IF SY-SUBRC NE 0.
     <<< Error processing >>>
  ENDIF.
```

```
IF V_PARAMETER-MSGTYP = 'E'     "Error.
OR V_PARAMETER-MSGTYP = 'A'.    "Abort.
   <<< Error processing >>>
ENDIF.
```

Note: When you call TSX, you should cater for two levels of return code, that is, sy-subrc from the function call and v_parameter-msgtyp from TSX. The actual error message code and data is stored in the parameter structure on return.

6.17.4 Parameter Structure

Refer the structure /sky/yecs_tsx for the full list of parameter options.
Only the primary fields required using logical TSX destinations are documented.

Field	Description
Command	• $EXCHANGE performs an OPEN, WRITE, READ and CLOSE in one operation. • $WRITE performs an OPEN, WRITE and CLOSE in one operation. • $READ performs an OPEN, READ and CLOSE in one operation. • $FILEWRITE performs a OPEN, FILEWRITE and CLOSE in one operation. • OPEN initiates a socket connection • EXCHANGE writes data to the socket and waits for a response • WRITE writes data to the socket and does not wait for a response • FILEWRITE initiates a file / directory transfer • READ waits for a response from the socket (external application) • PING performs a test call back loop (returns test message) • CLOSE terminates the socket connection ***Important:*** Legacy command SEND is automatically translated to EXCHANGE. The EXCHANGE command is the equivalent to a WRITE / READ.
Trace	Initiates the TSX trace facility. Specify true 'X' to activate. You may also activate this trace through the yecs_tsxcf destination configuration table.

Field	Description
Destination	Supply a logical destination name that is resolved into an IP address and port using table yecs_tsxcf. This parameter is optional and is only evaluated at OPEN time. Do not specify destination if you want to explicitly specify IP address, port, timeout and trace options.
RFC_dest*	Optionally, specify a specific RFC destination to use for the TSX server. If not specified, this defaults to 'TSX'.
Message_ type	Supply a logical message type that you use to default data processing attributes, such as data length determination modes, using table yecs_tsxmt . Do not specify message_type if you want to explicitly specify data control attributes.
data	The message area to transmit and receive from the destination.
msgtyp	Contains the severity of any message returned from TSX. E-Error, W-warning, I-information, A-abort.
msgno	The unique message number.
msgtxt	The text of the message.
File_name*	The name of the file / directory to transfer using the FILEWRITE command.
Tsx_ protocol*	Indicates to use the TSX guaranteed delivery protocol for file transfer.

Note: You may 'soft code' * these options through the logical destination and message type definitions.

6.17.5 TSX Return Codes

The TSX Java server communicates a return code and text message back to the caller through the I_ parameter, that is, the msgtyp, msgno and msgtxt fields. You need to be check in conjunction with sy-subrc on the function call. A list of possible return codes is as follows:

Return Code	Description
E000	Badly formatted numeric TCP/IP address
E001	Cannot open socket
E002	Cannot set non-block option on socket
E003	Connection is already open
E004	Connection is not open
E005	Connection failure (could not connect to address / port)
E006	Invalid command
E007	Invalid TCP/IP address (unknown host)
I008	Successful CLOSE
I009	Successful OPEN
I010	Successful PING
I011	Successful EXCHANGE
E012	Timeout during READ or EXCHANGE
E014	Unknown error

Return Code	Description
I015	Successful READ
I016	Successful WRITE

6.17.6 Sample BAP Program

> ***Note:*** This sample program is supplied with the TAX installation.

```
*---------------------------------------------------------------------
----
* This sample program demonstrates how to communicate with an
external
* TAX Java server from BAP. Test destination and message types
* 'COUNTEREXAMPLE' 'OVERSIMPLIFICATION' are used, which are set up
* by the TAX installation.
*---------------------------------------------------------------------
----
REPORT /SKY/700'S NO STANDARD PAGE HEADING
       LINE-SIZE 132 MESSAGE-ID /SKY/STYX.

INCLUDE /SKY/YESTERDAY.

DATA: PARAMETER         LIKE /SKY/YACHTSMEN,
      MESSAGE(132)      TYPE C,
      TERROR(1)          TYPE C,
      SUBBRANCH(1)          TYPE N,
      MODEST              LIKE /SKY/YACHTSMAN-FEST,
      MUMBLETYPEG          LIKE /SKY/YACHTSWOMAN-MISTYPE.

CONSTANTS: DELIMITER    TYPE X VALUE '0',
           CONSTRUE(1)      TYPE C VALUE 'X'.

* ---------------------------------------------------------------------
----
* Main processing
* ---------------------------------------------------------------------
```

```
----
  CLEAR: TERROR, PARAMETER, MODEST, V_MSG_TYPE.

* PING TSX
  WRITE: /1 'Issue PING to the TSX server to see if it is
responding'.
  V_PARAMETER-COMMAND = 'PING'.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

*
.....................................................................
..
* Test peer-to-peer message functionality.
*
.....................................................................
..
  ULINE.
  WRITE: /1 'Test TSX messaging program'.
  ULINE.
  V_DEST     = 'TSX_SAMPLE_MSG_OUT'.
  V_MSG_TYPE = 'TSX_SAMPLE_MSG_OUT'.

* Test $WRITE.
  CLEAR: V_PARAMETER.
  WRITE: /1 'Writing data to TSX socket using $WRITE'.
  V_PARAMETER-COMMAND = '$WRITE'.
  CONCATENATE 'Test $WRITE to TSX' C_DELIMITER INTO V_PARAMETER-
DATA.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

* Test $EXCHANGE.
```

```
  CLEAR: V_PARAMETER.
  WRITE: /1 'Exchanging data with TSX using $EXCHANGE'.
  V_PARAMETER-COMMAND = '$EXCHANGE'.
  CONCATENATE 'Test $EXCHANGE with TSX'
              C_DELIMITER INTO V_PARAMETER-DATA.
  PERFORM F_CALL_TSX.
  CONCATENATE 'Received(' V_PARAMETER-DATA(80) ')' INTO V_MESSAGE.
  WRITE: /5 V_MESSAGE.
  CHECK V_ERROR IS INITIAL.

* Open, write, read, exchange and close functionality.
  WRITE: /1 'Opening a socket connection to a TSX message port'.
  V_PARAMETER-COMMAND = 'OPEN'.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'Writing data to TSX socket'.
  V_PARAMETER-COMMAND = 'WRITE'.
  CONCATENATE 'Test WRITE to TSX' C_DELIMITER INTO V_PARAMETER-DATA.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'Reading data from TSX socket'.
  V_PARAMETER-COMMAND = 'READ'.
  PERFORM F_CALL_TSX.
  CONCATENATE 'Received(' V_PARAMETER-DATA(80) ')' INTO V_MESSAGE.
  WRITE: /5 V_MESSAGE.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'Exchanging data with TSX'.
  V_PARAMETER-COMMAND = 'EXCHANGE'.
  CONCATENATE 'Test EXCHANGE with TSX'
              C_DELIMITER INTO V_PARAMETER-DATA.
```

```
  PERFORM F_CALL_TSX.
  CONCATENATE 'Received(' V_PARAMETER-DATA(80) ')' INTO V_MESSAGE.
  WRITE: /5 V_MESSAGE.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'Terminating the TSX socket connection'.
  V_PARAMETER-COMMAND = 'CLOSE'.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

*
.....................................................................
..
* Test file transfer functionality.
*
.....................................................................
..
  ULINE.
  WRITE: /1 'Test file transfer options'.
  ULINE.
  CLEAR: V_PARAMETER.
  V_DEST     = 'TSX_SAMPLE_FILE_OUT'.
  V_MSG_TYPE = 'TSX_SAMPLE_FILE_OUT'.

* Create a dummy temporary file to transfer using ECS.
  ECS_GENERATE_FILE_NAME.
  IF V_ECS_*RC NE 0.
     WRITE: /5 V_ECS_*MSG.
     WRITE: /5 'File name generate failed. Option ignored.'.
     V_ERROR = C_TRUE.
  ENDIF.
  CHECK V_ERROR IS INITIAL.
  ECS_CREATE_FILE.
```

```
  IF V_ECS_*RC NE 0.
     WRITE: /5 V_ECS_*MSG.
     WRITE: /5 'File creation failed. Option ignored.'.
     V_ERROR = C_TRUE.
  ENDIF.
  CHECK V_ERROR IS INITIAL.
  ECS_WRITE_FILE 'Test TSX FILEWRITE command'.
  ECS_CLOSE_FILE.

* Test file transfer using $FILEWRITE (recommended)
  CLEAR: V_PARAMETER.
  WRITE: /1 'File transfer with TSX using $FILEWRITE'.
  V_PARAMETER-COMMAND   = '$FILEWRITE'.
  V_PARAMETER-FILE_NAME = V_ECS_*FILE_NAME.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

* Test file transfer using OPEN, FILEWRITE and CLOSE.
  WRITE: /1 'Opening a socket connection to a TSX raw file port'.
  V_PARAMETER-COMMAND = 'OPEN'.
  V_PARAMETER-FILE_NAME = V_ECS_*FILE_NAME.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'File transfer with TSX using FILEWRITE'.
  V_PARAMETER-COMMAND   = 'FILEWRITE'.
  PERFORM F_CALL_TSX.
  CHECK V_ERROR IS INITIAL.

  WRITE: /1 'Terminating the TSX socket connection'.
  V_PARAMETER-COMMAND = 'CLOSE'.
  PERFORM F_CALL_TSX.
```

```
* ---------------------------------------------------------------------
----
* Call the TSX server
* ---------------------------------------------------------------------
----
FORM F_CALL_TSX.

* Log 'about to call' message.
  GET TIME.
  CONCATENATE 'Issuing TSX command:' V_PARAMETER-COMMAND
              'Date:'                SY-DATUM
              'Time:'                SY-UZEIT
         INTO V_MESSAGE SEPARATED BY SPACE.
  WRITE: /05 V_MESSAGE.

* Call TSX.
  CALL FUNCTION '/SKY/TSX_SOCKET_CALL'
       EXPORTING
            I_DESTINATION                = V_DEST
            I_MESSAGE_TYPE               = V_MSG_TYPE
            I_PARAMETER                  = V_PARAMETER
       IMPORTING
            E_PARAMETER                  = V_PARAMETER
       EXCEPTIONS
            SESSION_ID_ALLOCATION_FAILED = 1
            INVALID_SOCKET_DESTINATION   = 2
            RFC_CALL_TO_TSX_FAILED       = 3
            INVALID_COMMAND              = 4
            INVALID_MESSAGE_TYPE         = 5
            INVALID_RECORD_LENGTH_MODE   = 6
            INVALID_RECORD_LENGTH_OPTION = 7
            NO_FILE_NAME_SPECIFIED       = 8
            FILE_COULD_NOT_BE_OPENED     = 9
```

```
            OTHERS                        = 99.

* Check TSX response.
  V_SUBRC = SY-SUBRC.
  GET TIME.
  CONCATENATE 'Control returned back from TSX.'
              'Date:'         SY-DATUM
              'Time:'         SY-UZEIT
         INTO V_MESSAGE SEPARATED BY SPACE.
  WRITE: /10 V_MESSAGE.

  CASE V_SUBRC.
       WHEN 0. " ok
       WHEN 1. V_MESSAGE = TEXT-001.
       WHEN 2. V_MESSAGE = TEXT-002.
       WHEN 3. V_MESSAGE = TEXT-003.
       WHEN 4. V_MESSAGE = TEXT-004.
       WHEN 5. V_MESSAGE = TEXT-005.
       WHEN 6. V_MESSAGE = TEXT-006.
       WHEN 7. V_MESSAGE = TEXT-007.
       WHEN 8. V_MESSAGE = TEXT-008.
       WHEN 9. V_MESSAGE = TEXT-009.
       WHEN OTHERS.
               V_MESSAGE = TEXT-099.
  ENDCASE.

  IF V_SUBRC <> 0.
     WRITE: /10 'TSX call failed: rc(',V_SUBRC,')',
            /10 V_MESSAGE.
     V_ERROR = C_TRUE.
     EXIT.
  ENDIF.
```

```
* Log TSX response.
  CONCATENATE V_PARAMETER-MSGTYP
              '(' V_PARAMETER-MSGNO ')'
               V_PARAMETER-MSGTXT
         INTO V_MESSAGE.
  WRITE: /10 V_MESSAGE.

  IF V_PARAMETER-MSGTYP = 'E'
  OR V_PARAMETER-MSGTYP = 'A'.
     V_ERROR = C_TRUE.
  ENDIF.

ENDFORM.
```

6.17.7 Sample ABAP File Transfer Program

```
*---------------------------------------------------------------------
--*
* This program is used to transmit a file via the TSX file transfer
* mechanism. It expects the file name to be passed in the usual ECS
* manner and the logical TSX destination to be passed as ECS run
data.
*
* Notes:
* ======
* This program assumes that a logical message type has been
configured
* with the same name as the logical destination.
*
* If the file data is to passed with the call to a remote TSX server
* that does not have access to the file system, this must be
configured
* in the TSX logical message type associated with the destination.
*---------------------------------------------------------------------
--*

REPORT  /SKY/YTSX7004 MESSAGE-ID /SKY/YTSX.

* Common includes
INCLUDE /SKY/YECSSDK.

* Working variables
DATA: V_NUMC5(5)           TYPE N,
      V_WRITE_LOG          TYPE ECS_LOG_DATA,
      V_PARAMETER          LIKE /SKY/YTSX_TSX,
      V_DESTINATION        LIKE /SKY/YTSX_TSXCF-DEST,
      V_ERROR(1)           TYPE C VALUE SPACE,
```

```
      V_SUBRC              LIKE SY-SUBRC.

* Constants
CONSTANTS: C_TRUE(1)       TYPE C VALUE 'X'.

* ---------------------------------------------------------------------
----
* Main processing logic
* ---------------------------------------------------------------------
----

INITIALIZATION.
* Get ECS parameters.
  ECS_IMPORT_PARAMETERS.
  IF V_ECS_*RC NE 0.
     MESSAGE A002 WITH V_ECS_*RC.
  ENDIF.
  V_DESTINATION = V_ECS_*RDATA.
  CLEAR V_ERROR.

START-OF-SELECTION.
* Started message.
  ECS_WRITE_LOG 'I' '1' TEXT-000.

* Check file name has been specified
  IF V_ECS_*FILE_NAME IS INITIAL.
     ECS_WRITE_LOG 'E' '2' TEXT-002.
     ECS_END_PHASE_AS_FAILED.
     EXIT.
  ENDIF.

* Check that the ECS file name exists and can be opened (warning
only).
```

```
  ECS_CHECK_FILE_EXISTS.
  IF V_ECS_*RC NE 0.
     V_WRITE_LOG = TEXT-001.
     V_NUMC5 = V_ECS_*RC.
     REPLACE '&1' WITH V_ECS_*FILE_NAME INTO V_WRITE_LOG.
     CONDENSE V_WRITE_LOG.
     REPLACE '&2' WITH SY-HOST INTO V_WRITE_LOG.
     REPLACE '&3' WITH V_NUMC5 INTO V_WRITE_LOG.
     REPLACE '&4' WITH V_ECS_*MSG INTO V_WRITE_LOG.
     CONDENSE V_WRITE_LOG.
     ECS_WRITE_LOG 'W' '2' V_WRITE_LOG.
  ENDIF.

* Check that a logical TSX destination has been specified. The call
to
* the TSX API fully validates this entry, so no further checking is
* required here.
  IF V_DESTINATION IS INITIAL.
     ECS_WRITE_LOG 'E' '2' TEXT-003.
     ECS_END_PHASE_AS_FAILED.
     EXIT.
  ENDIF.

END-OF-SELECTION.
* Annotate the ECS run log with file, destination and SAP host.
  V_WRITE_LOG = TEXT-004.
  REPLACE '&1' WITH V_ECS_*FILE_NAME INTO V_WRITE_LOG.
  CONDENSE V_WRITE_LOG.
  REPLACE '&2' WITH V_DESTINATION INTO V_WRITE_LOG.
  CONDENSE V_WRITE_LOG.
  ECS_WRITE_LOG 'I' '2' V_WRITE_LOG.
  V_WRITE_LOG = TEXT-009.
  REPLACE '&1' WITH SY-HOST INTO V_WRITE_LOG.
```

```
  REPLACE '&2' WITH SY-UNAME INTO V_WRITE_LOG.
  CONDENSE V_WRITE_LOG.
  ECS_WRITE_LOG 'I' '2' V_WRITE_LOG.

* Transfer the file.
  V_PARAMETER-COMMAND = '$FILEWRITE'.   "<- invoke file transfer
  PERFORM F_CALL_TSX.

* Annotate the status of the TSX guaranteed delivery protocol.
  IF V_ERROR IS INITIAL.
     IF V_PARAMETER-PROTOCOL = C_TRUE.
        ECS_WRITE_LOG 'I' '3' TEXT-007.
     ELSE.
        ECS_WRITE_LOG 'I' '3' TEXT-008.
     ENDIF.
  ENDIF.

* Ended message.
  ECS_WRITE_LOG 'I' '1' TEXT-010.

* If a error was detected, fail the ECS Phase.
  IF V_ERROR = C_TRUE.
     ECS_END_PHASE_AS_REPROCESSABLE.
  ENDIF.

* ---------------------------------------------------------------------
----
* Call TSX
* ---------------------------------------------------------------------
----
FORM F_CALL_TSX.

  CALL FUNCTION '/SKY/TSX_SOCKET_CALL'
```

```
       EXPORTING
            I_DESTINATION                = V_DESTINATION
            I_MESSAGE_TYPE               = V_DESTINATION
            I_PARAMETER                  = V_PARAMETER
            I_RFC_DEST                   = 'TSX'               "<-
default
            I_FILE_NAME                  = V_ECS_*FILE_NAME
       IMPORTING
            E_PARAMETER                  = V_PARAMETER
       EXCEPTIONS
            SESSION_ID_ALLOCATION_FAILED = 1
            INVALID_SOCKET_DESTINATION   = 2
            RFC_CALL_TO_TSX_FAILED       = 3
            INVALID_COMMAND              = 4
            INVALID_MESSAGE_TYPE         = 5
            INVALID_RECORD_LENGTH_MODE   = 6
            INVALID_RECORD_LENGTH_OPTION = 7
            NO_FILE_NAME_SPECIFIED       = 8
            OTHERS                       = 99.

  V_SUBRC = SY-SUBRC.

  IF V_SUBRC NE 0.
     V_WRITE_LOG = TEXT-005.
     V_NUMC5 = V_SUBRC.
     REPLACE '&1' WITH V_PARAMETER-COMMAND INTO V_WRITE_LOG.
     REPLACE '&2' WITH V_NUMC5 INTO V_WRITE_LOG.
     CONDENSE V_WRITE_LOG.
     ECS_WRITE_LOG 'E' '2' V_WRITE_LOG.
     CLEAR V_WRITE_LOG.
     CASE V_SUBRC.
          WHEN 1. V_WRITE_LOG = TEXT-101.
          WHEN 2. V_WRITE_LOG = TEXT-102.
```

```
          WHEN 3. CALL FUNCTION '/SKY/FORMAT_MESSAGE'
                       EXPORTING
                            MSGID    = SY-MSGID
                            MSGTY    = SY-MSGTY
                            MSGNO    = SY-MSGNO
                            MSGV1    = SY-MSGV1
                            MSGV2    = SY-MSGV2
                            MSGV3    = SY-MSGV3
                            MSGV4    = SY-MSGV4
                       IMPORTING
                            MSG_TEXT = V_WRITE_LOG
                       EXCEPTIONS
                            OTHERS   = 1.
                  ECS_WRITE_LOG 'I' '3' V_WRITE_LOG.
                  V_WRITE_LOG = TEXT-103.
          WHEN 4. V_WRITE_LOG = TEXT-104.
          WHEN 5. V_WRITE_LOG = TEXT-105.
          WHEN 6. V_WRITE_LOG = TEXT-106.
          WHEN 7. V_WRITE_LOG = TEXT-107.
          WHEN 8. V_WRITE_LOG = TEXT-108.
          WHEN OTHERS.
                  V_WRITE_LOG = TEXT-999.
     ENDCASE.
     ECS_WRITE_LOG 'I' '3' V_WRITE_LOG.
     V_ERROR = C_TRUE.
     EXIT.
  ENDIF.

* Check the TSX return code and message
  CONCATENATE V_PARAMETER-MSGTYP V_PARAMETER-MSGNO
         INTO V_WRITE_LOG.
  CONCATENATE V_WRITE_LOG V_PARAMETER-MSGTXT
         INTO V_WRITE_LOG SEPARATED BY SPACE.
```

```
  CASE V_PARAMETER-MSGTYP.
       WHEN SPACE.
            IF V_WRITE_LOG IS INITIAL.
               ECS_WRITE_LOG 'I' '3' TEXT-006.
            ELSE.
               ECS_WRITE_LOG 'I' '3' V_WRITE_LOG.
            ENDIF.
       WHEN 'I'.
            ECS_WRITE_LOG 'I' '3' V_WRITE_LOG.
       WHEN 'W'.
            ECS_WRITE_LOG 'W' '3' V_WRITE_LOG.
       WHEN OTHERS.
            ECS_WRITE_LOG 'E' '3' V_WRITE_LOG.
            V_ERROR = C_TRUE.
  ENDCASE.

ENDFORM.
```

Calling ABAP Inbound to SAP

In this context, the external TSX Java server listens for messages and / or files and then calls the TSX SAP add-in to action the request. The TSX SAP Listener configuration (See the Workbench section) is then used to determine what to do. Typically, either an ECS process is invoked and is passed the message data or file; or you may call a custom ABAP exit to process the request.

6.17.8 Sample TSX Listener User Exit

This program is an example of a TSX listener user exit. It is specified in the TSX SAP listener configuration as the ABAP program to call.

```
*---------------------------------------------------------------------
--*
* Sample TSX SAP listener user exit.
* It receives the input TSX data via import from ABAP memory and may
* change the input/response data and export it back to ABAP memory
for
* TSX to pick up. This exit is useful to verify input data, batch up
* data etc. prior to invoking a process via ECS. The exit may also
* dynamically specify the ECS Process to invoke (blank=none).
*
* Special test conditions for this sample exit:
* ============================================
* Parameter data = "FAIL" or "fail" will cause the exit to set
* stop_processing to true. Otherwise the passed data is returned
back
* as a response i.e. Received .......
*
* Parameters:
* ===========
* ECS Process run data, file name and filter field may be optionally
set
* and will be set by TSX when it starts the nominated ECS Process.
These
* are useful to pass data, input file or identify the ECS Process.
*
* Input parameters:
* -----------------
* address         The IP address of the sending connection
* port            The Port address the data was sent to
```

```
* local_port      Whether the receiving port is on the local host
*                 machine i.e. receiving port (X=true)
* data_length     The calculated length of the sent data block
* data            The sent data block (up to 4096)
* iname           The ECS Process name to invoke (blank = none)
* tsl_file_name   The file name that TSX generated after receiving a
*                 file on its file port configuration.
*
* Output parameters:
* ------------------
* msgtyp            I-information, W-warning, E-error, A-fatal
* msgno             Unique message number
* msgtxt            Formatted text message
* data_length       The calculated length of the sent data block
* data              The sent data block (up to 4096)
* iname             The ECS Process name to invoke (blank = none)
* rdata             ECS Process run data string (max 255)
* file_name         ECS Process file name
* filter_field      ECS Process filter field name
* filter_data       ECS Process filter field value
* response_data     Data block to be returned back to the sender
* response_length   Calculated length of the returned data block
* stop_processing   Instructs TSX to return directly to the TSX
server.
*                   Useful to batch up data etc.
* Enqueue           Enqueue to run the first phase of the process
with
* Enqueue_All       Enqueue for the entire process, X = True
* Enqueue_Hold      Hold the Enqueue if the process fails, only
valid
*                   if enqueue_all is true (X)
*---------------------------------------------------------------------
--*
```

```
REPORT  /SKY/YTSX7001 message-id /SKY/YTSX.

* Import/Export data definitions
DATA: ADDRESS               LIKE /SKY/YTSX_TCPIP-ADDRESS,
      PORT                  LIKE /SKY/YTSX_TCPIP-PORT,
      LOCAL_PORT            LIKE /SKY/YTSX_TCPIP-LOCAL_PORT,
      DATA_TABLE            LIKE /SKY/YTSX_TCPXR-CREATEFILE,
      DATA_LENGTH           LIKE /SKY/YTSX_TCPIP-DATA_LEN,
      DATA                  LIKE /SKY/YTSX_TCPIP-DATA,
      INAME                 LIKE /SKY/YTSX_TCPXR-INAME,
      RDATA                 LIKE /SKY/YECS_RUN-RDATA,
      TSL_FILE_NAME         LIKE /SKY/YECS_RUN-FILE_NAME,
      FILE_NAME             LIKE /SKY/YECS_RUN-FILE_NAME,
      FILTER_FIELD          LIKE /SKY/YECS_RUN-KEY_FIELD,
      FILTER_DATA           LIKE /SKY/YECS_RUN-KEY_DATA,
      STOP_PROCESSING(1)    TYPE C,
      MSGTYP                LIKE /SKY/YTSX_TCPIP-MSGTYP,
      MSGNO                 LIKE /SKY/YTSX_TCPIP-MSGNO,
      MSGTXT                LIKE /SKY/YTSX_TCPIP-MSGTXT,
      RESPONSE_LENGTH       LIKE /SKY/YTSX_TCPIP-DATA_LEN,
      RESPONSE_DATA         LIKE /SKY/YTSX_TCPIP-DATA,
      ENQUEUE               LIKE /SKY/YECS_RUN-ENQUEUE,
      ENQUEUE_ALL(1),
      ENQUEUE_HOLD(1).

* Constants
CONSTANTS: C_TRUE(1)              TYPE C VALUE 'X',
           C_TSX_MEMORY_ID(32)    TYPE C VALUE 'TSX_USER_EXIT',
           C_DELIMITER(2)         TYPE X VALUE '0D0A'.

* Internal tables:
DATA: IT_DATA               LIKE /SKY/YTSX_TBDTA
```

```
                          OCCURS 0 WITH HEADER LINE.
* ---------------------------------------------------------------------
----
* Main processing logic
* ---------------------------------------------------------------------
----

* Import the TSX parameters from ABAP memory (configured by TSX)
  IMPORT ADDRESS
         PORT
         LOCAL_PORT
         DATA_TABLE
         DATA_LENGTH
         DATA
         INAME
         STOP_PROCESSING
         TSL_FILE_NAME
         IT_DATA
    FROM MEMORY ID C_TSX_MEMORY_ID.

* Custom code:
* Sample user exit issues a response, sets a message & stops
processing.
  MSGTYP = 'I'.
  MSGNO  = '100'.

  IF DATA = 'FAIL'
  OR DATA = 'fail'.
     STOP_PROCESSING = C_TRUE.
     MSGTXT = 'Sample user exit issued stop_processing'(008).
     CONCATENATE MSGTXT C_DELIMITER
            INTO RESPONSE_DATA.
  ELSE.
```

```
     MSGTXT = 'Dummy response from sample user exit /SKY/YTSX7001'
(005).
     CONCATENATE 'User exit received\['(006) DATA '\]'(007) C_
DELIMITER
            INTO RESPONSE_DATA.
  ENDIF.

  RESPONSE_LENGTH = STRLEN( RESPONSE_DATA ).

* Enqueue data options.
  ENQUEUE      = DATA.
  ENQUEUE_ALL  = C_TRUE.
  ENQUEUE_HOLD = C_TRUE.

* Export the TSX parameters back to ABAP memory for TSX to pick up.
  EXPORT MSGTYP
         MSGNO
         MSGTXT
         DATA_TABLE
         DATA_LENGTH
         DATA
         INAME
         RDATA
         FILE_NAME
         FILTER_FIELD
         FILTER_DATA
         RESPONSE_LENGTH
         RESPONSE_DATA
         STOP_PROCESSING
         ENQUEUE
         ENQUEUE_ALL
         ENQUEUE_HOLD
         IT_DATA
```

```
    TO MEMORY ID C_TSX_MEMORY_ID.
```

6.17.9 Sample ECS Receiving Program

An ECS Process may be automatically started by the TSX SAP Listener when it receives a message or file. It is specified in the TSX SAP listener configuration as the ECS Process to start. This program is configured in an ECS process as the first phase. The object type is ABAP. It receives the TSX data from standard ECS variables populated by the TSX SAP Listener process.

```
*---------------------------------------------------------------------
----
* This sample program is initiated by TSX when no TSX SAP listener
* configuration was found. It lists the TSX parameters and issues a
* formatted response back to the original caller. The formatted
response
* may be suppressed by specifying NORESPONSE in the ECS user
variable 1
* or run data parameters.
*---------------------------------------------------------------------
----

REPORT  /SKY/YTSX7000 message-id /SKY/YTSX.

INCLUDE /SKY/YECSSDK.

* Working variables
DATA:
      V_MSG(255)          TYPE C,
      V_RECEIVED_MSG(255) TYPE C,
      V_OFFSET            TYPE I,
      V_MAX_LENGTH        TYPE I,
      V_SUBRC             LIKE SY-SUBRC,
*     TSX parameters
      ADDRESS             LIKE /SKY/YTSX_TCPIP-ADDRESS,
      PORT                LIKE /SKY/YTSX_TCPIP-PORT,
      DATA                LIKE /SKY/YTSX_TCPIP-DATA,
```

```
      LOCAL_PORT          LIKE /SKY/YTSX_TCPIP-LOCAL_PORT,
      DATA_LENGTH         LIKE /SKY/YTSX_TCPIP-DATA_LEN,
      RESPONSE_DATA       LIKE /SKY/YTSX_TCPIP-DATA,
      RESPONSE_LENGTH     LIKE /SKY/YTSX_TCPIP-DATA_LEN.

* Constants
CONSTANTS: C_RETURN(2)      TYPE X VALUE '0D0A',  "<- ASCI CR+line
feed
           C_NORESPONSE(10) TYPE C VALUE 'NORESPONSE'.

*---------------------------------------------------------------------
----
* Main processing logic
*---------------------------------------------------------------------
----

* Import the ECS parameters.
  ECS_IMPORT_PARAMETERS.

* Import ECS TSX variables.
  ECS_IMPORT_DATA: ADDRESS,
                   PORT,
                   DATA,
                   LOCAL_PORT,
                   DATA_LENGTH.

* Annotate TSX TCP/IP connection details.
  ECS_REFRESH_LOG.
  ECS_APPEND_LOG 'W' '1' TEXT-000.
  ECS_APPEND_LOG 'W' '1' TEXT-001.
  V_MSG = TEXT-002.
  REPLACE '#1' WITH ADDRESS     INTO V_MSG.
  REPLACE '#2' WITH PORT        INTO V_MSG.
```

```
  REPLACE '#3' WITH LOCAL_PORT  INTO V_MSG.
  REPLACE '#4' WITH DATA_LENGTH INTO V_MSG.
  CONDENSE V_MSG.
  ECS_APPEND_LOG 'I' '2' V_MSG.
  V_RECEIVED_MSG = V_MSG.              " <- save for response

* Annotate optional run data and file name.
  IF NOT V_ECS_*RDATA IS INITIAL.
     V_MSG = TEXT-005.
     REPLACE '#1' WITH V_ECS_*RDATA INTO V_MSG.
     ECS_APPEND_LOG 'I' '2' V_MSG.
  ENDIF.

  IF NOT V_ECS_*FILE_NAME IS INITIAL.
     V_MSG = TEXT-006.
     REPLACE '#1' WITH V_ECS_*FILE_NAME INTO V_MSG.
     ECS_APPEND_LOG 'I' '2' V_MSG.
  ENDIF.

* Annotate TSX data message.
  IF DATA_LENGTH = 0.
     ECS_APPEND_LOG 'I' '2' TEXT-004.
  ELSE.
     ECS_APPEND_LOG 'I' '2' TEXT-003.
  ENDIF.

  CLEAR V_OFFSET.
  DESCRIBE FIELD DATA LENGTH V_MAX_LENGTH.

  WHILE V_OFFSET < DATA_LENGTH
    AND V_OFFSET < V_MAX_LENGTH.
        CLEAR V_MSG.
        V_MSG+0(1) = '>'.
```

```
        V_MSG+1(50) = DATA+V_OFFSET(50).
        V_MSG+51(1) = '<'.
        ECS_APPEND_LOG 'I' '2' V_MSG.
        ADD 50 TO V_OFFSET.
  ENDWHILE.

  ECS_COMMIT_LOG.

* If specified, ignore any response data.
  IF V_ECS_*RDATA     = C_NORESPONSE
  OR V_ECS_*USER_VAR1 = C_NORESPONSE.
     EXIT.
  ENDIF.

* Set up a default response message (with asci return+line feed).
  CONCATENATE V_RECEIVED_MSG C_RETURN INTO RESPONSE_DATA.
  RESPONSE_LENGTH = STRLEN( RESPONSE_DATA ).

* Export ECS TSX variables data.
  ECS_EXPORT_DATA: RESPONSE_DATA,
                   RESPONSE_LENGTH.
```

Cupid

You can also develop inbound and outbound exit processing using Cupid. For further details, refer the section on data transformation.

TSX Java Programming Guide

The TSX Java Programming Guide consists of the following:

TSX Java Programming Guide Overview

Getting Started with Exits for TSX

TSX Java Programming Guide Overview

Java exits are similar to ABAP exits, in that they extend the standard functionality offered by TSX. They allow TSX to perform tasks that may have specific, complex and / or unusual requirements. There are four major types of TSX Java exit that have differing capabilities and that are invoked at different times by the TSX server:

Java Inbound File Exits

These exits are invoked whenever an incoming file is received on a file port. They allow highly customized behavior to take place when processing the file.

Java Inbound Message Exits

These exits are invoked whenever an incoming message is received on a message port. They allow highly customized behavior to take place when processing the message.

Java Server Exits

These exits are invoked by the TSX server whenever certain system events take place (for example, at start up or shut down, or when host connectivity is lost). Java server exits are the primary way in which you can implement custom "background tasks".

Java Poll Directory Exits

These exits are invoked when an incoming file is detected in a TSX poll directory. They implement whatever action needs to take place when such a file arrives.

Getting Started with Exits for TSX

You can perform the following with exits for TSX:

How to Write a Java Exit

How to Compile a Java Exit

How to Configure a Java Exit

6.17.10 How to Write a Java Exit

Java exits are implemented as standard Java classes. In order to write one, you need to be familiar with the syntax and general format of the Java language.

Each of the four major types of Java exit is implemented as a separate Java class (TsxInboundFileExit, TsxInboundMessageExit, TsxServerExit and TsxPollDirectoryExit, respectively). These classes all extend a base class called TsxExit that describes the basic functionality common to all types of exit.

Once you decide what type of exit you wish to develop, writing the exit is simply a matter of developing a class that extends the TSX class provided. For example, in order to write a new custom Java inbound message exit, you need to develop a new class ("MyUserExit", for example) that extends TsxInboundMessageExit.

Each of the major types of exit has one or more abstract methods that provide the "hooks" that the TSX server uses to invoke your code. You must provide implementations for these methods so that the TSX server is able to do this.

> ***Note:*** You can find the documentation for the TSX Java exits API in the `doc` directory that is located in the `TSX installation` directory. It is in javadoc format and you can interrogate using any standard browser.

6.17.11 How to Compile a Java Exit

In order to compile the Java class you have written, you will need either a Java compiler or an Integrated Development Environment such as JBuilder.

The only way in which compiling a TSX exit class differs from compiling any other Java class is that you need to incorporate the standard TSX classes into your CLASSPATH. The TSX classes are all contained within the `tsx.jar` file that is located in the `TSX installation` directory. Check the documentation for your Java compiler or IDE for details on how to incorporate additional classes into your CLASSPATH.

In addition to raw `.class` files, the TSX server is also able to load classes from `.jar` (Java archive) and `.zip` files placed in the `classes` directory. This can become useful for ease of deployment later on.

6.17.12 How to Configure a Java Exit

Once a new exit class is made available to the TSX server, the next step is to configure the TSX server so that it is invoked at the appropriate time. Because each of the major exit types are invoked in different ways and at different times, the customization involved for each type of exit is slightly different. In the sections that follow, each type of exit is examined in more detail and the relevant customization process is described.

Common Exit Functionality for TSX

Regardless of their type, all Java exits have certain common capabilities. These capabilities are available to your code as methods inherited from the TsxExit class. The most significant and commonly used of these capabilities are as follows.

Writing Log Messages

All types of TSX exit are able to write messages to the TSX log. The relevant methods are as follows:

Methods

```
logAbort()
logError()
logInfo()
logTrace()
logWarn()
```

Each of these methods write a message of the corresponding type to the TSX log. Some of these (logAbort, logError, and logWarn) also accept a Throwable error or exception object as an additional parameter, for use in printing a stack trace to the log.

The logAbort method is something of a special case, as in addition to writing the specified message to the TSX log, it also causes the TSX server to halt. Obviously, you should perform this action in the event of extremely dire errors, since manual intervention is necessary to restart the TSX server.

The logTrace method differs from the other methods of this type, in that it requires a trace level to specify (from 0-2, with 0 as the lowest level of detail and 2 as the highest). If the TSX server current trace level equals or exceeds the figure specified in the method call, then the trace message is written to the log. However, nothing is written to the log if this is not the case. Trace messages are always accompanied by the name of the class and the method from which the trace call is made.

Reading Configuration File Settings

All types of TSX exit are able to read parameter configurations from the TSX configuration file. The most important methods to perform this are the following:

Methods

```
getConfigBoolean()
getConfigDouble()
getConfigInteger()
getConfigString()
```

These methods allow an exit to examine existing or additional configuration file parameters that you can use to alter its behavior.

Specific Exit Types

The following are the specific types of exit:

Java Inbound File Exits

Java Inbound Message Exits

Java Server Exits for TSX

Java Poll Directory Exits for TSX

Java Inbound File Exits

Java inbound file exits inherit from the TsxInboundFileExit class. They are invoked when an incoming file is received through a file port.

Customisation of Inbound File Exits

6.17.13 Customization of Inbound File Exits

Java inbound message exits are configured in the TSX configuration file. The file can contain one or more [MessagePort:xxx] and/or [Connection:xxx] sections. Each of these sections can reference an inbound message exit class that is invoked by the TSX server whenever a message is received. An example message port section is shown below:

MessagePort

```
[MessagePort:Port5092]
Port = 5092
InputMessageType = D
InputDelimiters = \013
InputIgnored = \010
InputStripDelim = false
LogTsxMessages = true
NotifyHost = false
HostInterface =
StoreAndForward = true
AsynchronousHostProcessing = false
ShellCommand =
SystemCommand =
InboundMessageExitClass = com.mycompany.MyInboundMessageExit
```

In this example, the Java class "com.mycompany.MyInboundMessageExit" is invoked whenever an incoming message is received through port 5092.

Java inbound file exits are configured in the TSX configuration file. The file can contain one or more [FilePort:xxx] and/or [RawFilePort:xxx] sections that each relate to a port on which the TSX server is able to receive incoming files. An example file port section is below:

FilePort

```
[FilePort:Port5094]
Port = 5094
```

```
FileDirectory = ./files
FileNamePattern = &filename&
LogTsxMessages = true
NotifyHost = false
HostInterface =
StoreAndForward = true
AsynchronousHostProcessing =
ShellCommand =
SystemCommand =
InboundFileExitClass = com.mycompany.MyInboundFileExit
```

In this example, the Java class "com.mycompany.MyInboundFileExit" is invoked whenever an incoming file is received through port 5094.

6.17.14 Handling an Incoming File

When a file arrives through a connection opened on a file port, the TSX Java server invokes two methods, *executePreHost()* and *executePostHost()*, in the associated Java inbound file port exit class. The *executePreHost()* method is invoked before any call is made to the host, and the *executePostHost()* method is invoked after the host call is complete. These two methods are collectively responsible for taking whatever action is necessary to process the file. You can address the incoming file using the information that the *getFileDirectoryName()*, *getFileName()*, and *getFilePathName()* methods return.

Example

The following code moves each incoming file received to one of two directories, based on if its name begins with an "A". The copy is performed after any host processing takes place.

```
package com.mycompany;

import au.com.skytechnologies.tsx.*;

public class MyInboundFileExit extends TsxInboundFileExit \{

    private final static String targetDirA = "C:\files_starting_
with_a";
    private final static String targetDirOther = "C:\all_other_
files";

    public TsxInboundExitResult executePostHost() \{
        String fileDirName = getFileDirectoryName();
        String fileName = getFileName();
        File incomingFile = new File(fileDirName, fileName);

        if (!incomingFile.exists())
            return new TsxInboundExitResult(999,
                "Incoming file " + filename + " not found!");
```

```
        String targetDirName = (fileName.startsWith("A")) ?
            targetDirA : targetDirOther;
        File targetFile = new File(targetDirName, fileName);

        if (!incomingFile.renameTo(targetFile))
            return new TsxInboundExitResult(999,
                "Cannot move file " + filename + " to target
directory " +
                targetDirName + "!");

        return new TsxInboundExitResult(0, "Success!");
    \}
\}
```

6.17.15 Error Handling of Inbound File Exits

Once an inbound file exit finishes processing a file, it must notify the TSX server whether or not the file is successfully processed. It does this by either throwing a TsxExitException, or by returning a TsxInboundExitExitResult once processing is completed. An inbound file exit is normally treated as having encountered an error if it throws a TsxExitException, or returns a TsxInboundExitResult with a nonzero return code. It is normally considered to have completed successfully if it returns a TsxInboundExitResult with a zero return code, or returns null.

File ports have default behavior that they manifest unless the code inside the exit takes explicit control. For example, if the "NotifyHost" configuration item were configure in the relevant TSX configuration file section, this would tell the TSX server that in the normal course of events, the host should be notified of any incoming files. However, an exit can override this default behavior, if it were to execute the following code:

```
return new TsxInboundExitResult(0,
    "The host does not need to know about this file!",
    DONT_NOTIFY_HOST);
```

There are a number of different aspects of the TSX server behavior that you can control in this way. These are listed in the following table. However, because of timing issues associated with the sequence in which the TSX server performs the necessary tasks, not all options are effective in both pre- and post- host processing.

Behavior	Flags	Default Behavior	Pre	Post
Error status (whether or not the TSX server treats the result as an error, regardless of the return code sent back).	TREAT_AS_ERROR DONT_TREAT_AS_ERROR	Default behavior is to treat the result as a success if it has a zero return code, or an error if it has a non-zero return code.	✔	✔

Behavior	Flags	Default Behavior	Pre	Post
Message logging (whether or not any message returned by the exit is written to the TSX log).	LOG_ MESSAGE DONT_LOG_ MESSAGE	Default behavior is configure in the configuration file through the **LogTsxMessages** option.	✅	✅
Return code logging (whether or not any return code sent back by the exit is written to the TSX log).	LOG_ RETURN_ CODE DONT_LOG_ RETURN_ CODE	Default behavior is configured in the configuration file through the **LogTsxMessages** option.	✅	✅
System command logging (whether or not any system command executed is written to the TSX log).	LOG_ SYSTEM_ COMMAND DONT_LOG_ SYSTEM_ COMMAND	Default behavior is configured in the configuration file through the **LogTsxMessages** option.	✅	❌
Host notification (whether or not the host receives notification of the incoming file).	NOTIFY_ HOST DONT_ NOTIFY_ HOST	Default behavior is configured in the configuration file through the **NotifyHost** option.	✅	❌
Host response logging (whether or not any response received from the host is written to the TSX log).	LOG_HOST_ RESPONSE DONT_LOG_ HOST_ RESPONSE	Default behavior is configured in the configuration file through the **LogTsxMessages** option.	✅	❌

Behavior	Flags	Default Behavior	Pre	Post
Message count incrementation (whether or not the message count for the port is incremented).	INCREMENT_MESSAGE_COUNT DONT_INCREMENT_MESSAGE_COUNT	Default behavior is to increment the message count each time an incoming file is received.	✓	✗

Java Inbound Message Exits

Java inbound message exits inherit from the TsxInboundMessageExit class. They are invoked when an incoming message is received, either through a message port, or on a configured connection to a remote network address.

6.17.16 Customization of Inbound File Exits

Java inbound message exits are configured in the TSX configuration file. The file can contain one or more [MessagePort:xxx] and/or [Connection:xxx] sections. Each of these sections can reference an inbound message exit class that is invoked by the TSX server whenever a message is received. An example message port section is shown below:

MessagePort

```
[MessagePort:Port5092]
Port = 5092
InputMessageType = D
InputDelimiters = \013
InputIgnored = \010
InputStripDelim = false
LogTsxMessages = true
NotifyHost = false
HostInterface =
StoreAndForward = true
AsynchronousHostProcessing = false
ShellCommand =
SystemCommand =
InboundMessageExitClass = com.mycompany.MyInboundMessageExit
```

In this example, the Java class "com.mycompany.MyInboundMessageExit" is invoked whenever an incoming message is received through port 5092.

Java inbound file exits are configured in the TSX configuration file. The file can contain one or more [FilePort:xxx] and/or [RawFilePort:xxx] sections that each relate to a port on which the TSX server is able to receive incoming files. An example file port section is below:

FilePort

```
[FilePort:Port5094]
Port = 5094
```

```
FileDirectory = ./files
FileNamePattern = &filename&
LogTsxMessages = true
NotifyHost = false
HostInterface =
StoreAndForward = true
AsynchronousHostProcessing =
ShellCommand =
SystemCommand =
InboundFileExitClass = com.mycompany.MyInboundFileExit
```

In this example, the Java class "com.mycompany.MyInboundFileExit" is invoked whenever an incoming file is received through port 5094.

6.17.17 Handling an Incoming Message

Inbound message handling for inbound message exits is virtually identical to the handling of files for inbound file exits. When a message arrives, the TSX Java server invokes two methods, *executePreHost()* and *executePostHost()*, in the associated Java inbound message port exit class. The only real difference between message handling and file handling is that you need to reference the message data through the *getRequestData()* method, rather than by examining the incoming file.

Inbound message exits allow the actual message content to be altered "on the fly" before it is sent to the host, through the *setRequestData()* method. This is analogous to altering the content of a file received on a file port before the host is notified.

6.17.18 Error Handling of Inbound Message Exits

Error handling in inbound message exits works in an identical manner to that described for inbound file exits.

Java Server Exits for TSX

Java server exits extend the TsxServerExit class. They are invoked by the TSX server in response to the occurrence of certain predefined system events. Server exits are useful to implement "background tasks" that need to take place regardless of whether there is any other activity taking place on the server.

6.17.19 Customization of Java Exits

Java server exits are configured in the TSX configuration file. There is a section called [ServerExits] in this file that tells the TSX Java server the name of the TSX server exits that should be loaded. An example [ServerExits] section is below:

ServerExits

```
[ServerExits]
com.mycompany.MyServerExit = true
com.mycompany.MyOtherServerExit = false
```

In this example, the two Java classes specified are "com.mycompany.MyServerExit" and "com.mycompany.MyOtherServerExit". The boolean parameter to the righthand side of the equals sign tells the TSX server whether or not the server exit is currently enabled. This is intended as an easy method to enable or disable server exits without removing their names from the configuration file. In the example shown, only the "com.mycompany.MyServerExit" is loaded by the TSX server.

6.17.20 Implementing a Background Task for TSX

In the normal course of events, a server exit is invoked when a system event takes place. Its code is executed and it then releases control back to the TSX server, once its job is done. However, if there is a need for your custom code to remain active once an event has come and gone, the best way to accomplish this is to start a background task.

Background tasks are usually linked to the *start()* and *stop()* methods for a Java server exit. A new thread is started when the TSX server starts up, and it is stopped when the TSX server shuts down.

Example

Class "com.mycompany.MyServerExit"

```
package com.mycompany;

import au.com.skytechnologies.tsx.*;

public MyServerExit extends TsxServerExit \{
    MyBackgroundThread thread = null;

    public void start() \{
        thread = new MyBackgroundThread();
        thread.start();
    \}

    public void stop() \{
        if (thread != null)
            thread.halt();
    \}
\}
```

Class "MyBackgroundThread":

```
package com.mycompany;

public MyBackgroundThread extends Thread \{
    private Boolean halted = false;

    public void run() \{
        while (!halted) \{
            // Insert some code to do a background task here.
        \}
    \}

    public void halt() \{
        halted = true;
    \}
\}
```

6.17.21 Error Handling of Java Exits

In most situations, there is no sensible generic response that the TSX server can undertake when application errors arise inside a server exit. Any such error conditions that may occur therefore need to be handled by the code that performs the task. Often this involves writing detailed messages to the TSX log through methods such as *logWarn()*, *logError()*, or even (in the event of a catastrophic failure) *logAbort()*.

Many server exits launch a background thread that loops continuously, waiting for some event to occur in order to begin processing. When writing such threads, it is important to consider error handling behavior. A thread that polls for some condition to occur five times a second very quickly creates an enormous log file if it keeps encountering the same error, and writes a log message each time it does so. A more graceful way to handle this type of situation is to place the background thread in a sleep or wait state for some arbitrary error retry interval before trying again.

Unexpected Java exceptions that a server exit may throw at run time are automatically caught and written to the TSX log, along with a stack trace.

Java Poll Directory Exits for TSX

Java poll directory exits extend the TsxPollDirectory class. They are automatically invoked when the TSX server detects an incoming file in a directory that it is configured to monitor.

6.17.22 Customization of Poll Directory Exits

Java poll directory exits are configured in the TSX configuration file. This file can contain one or more [PollDirectory:xxx] sections that each relate to a directory that the TSX server is expected to monitor for incoming files. An example poll directory configuration file section is below:

6.17.22.1 Poll Directory Configuration File

```
[PollDirectory:MyPollDirectory]
IncomingDirectory = c:\mypolldir\incoming
ArchiveDirectory = c:\mypolldir\archive
ErrorDirectory = c:\mypolldir\error
IncomingFilePattern = .
PollFrequency = 30
LogIncomingFiles = true
StopPollingOnError = false
ExitClassName = com.mycompany.MyPollDirectoryExit
```

In this example, the Java class that is invoked whenever an incoming file is detected (in the directory `c:\mypolldir\incoming`) is called "com.mycompany.MyPollDirectoryExit".

6.17.23 Handling an Incoming File for TSX

When a file arrives in the `incoming directory`, the TSX Java server invokes the `execute()` method in the associated Java poll directory exit class. This method is responsible for taking whatever action is necessary to process the file. You can obtain a handle to the file in question by invoking the `getFile()` method.

Example

The following sample code implements a poll directory exit that reads an incoming tab delimited file that contains two fields, a code and a description. The incoming codes and descriptions are then written to a local database table called "MYTABLE".

```
package com.mycompany;

import au.com.skytechnologies.tsx.*;
import java.io.*;
import java.util.*;

public class MyPollDirectoryExit extends TsxPollDirectoryExit \{
    public TsxPollDirectoryExitResult execute() throws
TsxExitException \{
        File file = getFile();
        FileReader fr;

        try \{
            fr = new FileReader(file);
        \}
        catch (FileNotFoundException fnfe) \{
            throw new TsxExitException("Incoming file not found!");
        \}

        BufferedReader br = new BufferedReader(fr);
```

```
        TsxExitLdbTable table = getLocalDatabaseTable
                ("ProductionHostInterface", "MYTABLE");

        try \{
            beginTransaction();

            String line;
            int lineNo = 0;

            while ((line = br.readLine()) != null) \{
                ++lineNo;

                StringTokenizer strTok = new StringTokenizer(line,
"\t");

                if (strTok.countTokens() != 2)
                    throw new TsxExitException
                        ("Bad field count on line " + lineNo);

                TsxExitLdbTableRow row = table.newRow();
                row.setFieldValue("CODE", strTok.nextToken());
                row.setFieldValue("DESC", strTok.nextToken());

                try \{
                    table.saveRow(row);
                \}
                catch (TsxExitException ee) \{
                    rollbackTransaction();
                    br.close();
                    throw ee;
                \}
            \}
```

```
            commitTransaction();
            br.close();
        \}
        catch (IOException ioe) \{
            rollbackTransaction();
            br.close();
            throw new TsxExitException("I/O error!");
        \}

        return new TsxPollDirectoryExitResult(0, "Success!");
    \}
\}
```

Important: A common error when writing Java poll directory exits is to forget to close input streams and / or readers that reference the input file. This effectively leaves the file temporarily "locked", and the TSX server is unable to move or delete it, resulting in errors written to the log. In order to avoid this problem, take care to ensure that your exit always closes any input streams and / or readers that reference the input file, before returning control to the TSX server. For example, in the code shown above, the BufferedReader ("br") used to read the input file is closed before the exit returns.

6.17.24 Error Handling of Poll Directory Exits

Once an exit finishes processing a file, it need to notify the TSX server whether or not the file is successfully processed. It does this by either throwing a TsxExitException, or by returning a TsxPollDirectoryExitResult once processing is completed. A poll directory exit is normally treated as having encountered an error if it throws a TsxExitException, or returns a TsxPollDirectoryExitResult with a nonzero return code. It is normally considered to completed successfully if it returns a TsxPollDirectoryExitResult with a zero return code, or returns null.

Poll directories have default behavior that they manifest unless the code inside the exit takes explicit control. For example, if the "StopPollingOnError" configuration item were set in the relevant TSX configuration file section, this tells the TSX server that if the exit encounters an error, the default behavior is to stop polling the directory. However, an exit can override this default behavior if it were to execute the following code:

```
return new TsxPollDirectoryExit(999,
    "Non-fatal error encountered, polling will continue",
    DONT_STOP_POLLING);
```

There are a number of different aspects of the TSX server behavior that you can control in this way. These are listed in the following table.

Behavior	Flags	Default Behavior
Error status (whether or not the TSX server treats the result as an error, regardless of the return code sent back).	TREAT_AS_ERROR DONT_TREAT_AS_ERROR	Default behavior is to treat the result as a success if it has a zero return code, or an error if it has a non-zero return code.
Message logging (whether or not any message returned by the exit is written to the TSX log).	LOG_MESSAGE DONT_LOG_MESSAGE	Default behavior is set in the configuration file through the **LogIncomingFiles** option.

Behavior	Flags	Default Behavior
Return code logging (whether or not any return code sent back by the exit is written to the TSX log).	LOG_ RETURN_ CODE DONT_ LOG_ RETURN_ CODE	Default behavior is set in the configuration file through the **LogIncomingFiles** option.
Database rollback (whether or not any outstanding database changes are rolled back).	PERFORM_ ROLLBACK DONT_ PERFORM_ ROLLBACK	Default behavior is to only perform a rollback if an error occurs.
Polling control (whether or not polling should be halted until manual intervention takes place).	STOP_ POLLING DONT_ STOP_ POLLING	Polling is normally not stopped unless an error occurs. Default behavior, when an error occurs is set in the configuration file through the **StopPollingOnError** option.

TSX Utilities

The following are the TSX utilities:

SAP TSX File Transfer Utility

Java FTC File Transfer Client

SAP TSX File Transfer Utility

A standard utility program /SKY/YTSX7004 is provided with the TSX SAP add-in that is designed to be called from a ECS Process Phase to transfer a file to an external TSX Java server. This program accepts a file using the ECS Core file name parameter and the TSX logical destination through the run data parameter. The utility offers a simple way to transmit a file created using a previous phase.

To implement, add a new Phase with an object type of ABAP calling ABAP program /SKY/YTSX7004. A previous phase must create the external file and pass it through the standard ECS Core file name parameter. Configure the TSX logical destination (see TSX Workbench for details). The logical destination is passed through the standard ECS Core run data parameter. You may specify this either through the Phase advanced attributes or configures and pass by a previous program.

Java FTC File Transfer Client

The ftc shell script or bat file is located in the `TSX` directory within ecs_java, for example, `../ecs_java/v../tsx/ftc.sh`. The basic syntax for calling FTC is:

From Unix

ftc.sh -d {logical destination} -f {file}
ftc.sh -d {logical destination} -dir {directory}

From Windows

ftc.bat -d {logical destination} -f {files}
ftc.bat -d {logical destination} -dir {directory}

Field	Description
Logical destination (-d)	This is a section name that exists in the `ftc.cfg` file. See the section on FTC configuration or details.
File (-f or -file)	The name of a file to transmit to TSX.
Directory (-dir)	The name of a directory to transmit to TSX.

Example call

```
Ftc.sh -d skynet2 -f ftctestfile.txt
```

FTC Return Code

If FTC is called from an application, the application should check for a return code. This method to do this varies depending on the programming language. FTC configures a non-zero return code if there is any problem with the transfer and writes the error details to its log.

TSX Appendix

The TSX Appendix has the following:

TSX Glossary of Terms and Abbreviations

Designing Peer-to-Peer Messaging Interfaces with SAP

TSX Glossary of Terms and Abbreviations

Term	Description
ABAP	SAP own internal programming language. ABAP allows SAP and its customers to develop and deploy SAP applications. ABAP is also portable to any computer platform that SAP supports, without change.
API	Application programming interface. A standard documented function call, object method or program that you may invoke programmatically to perform a specific function. Most software vendors provide one or more APIs in the form of a software development kit (SDK).
ECS	The Sky ECS (Event Control System), used to manage interfaces and background processing in SAP.
EDI	Electronic Data Interchange / Interface. A mechanism and standard to transmit business documents between customers and suppliers. There are two primary standards used, EDIFACT and ANSI 12.
FTP	File Transfer Protocol. A means to transmit files from one system to another.
GUI	Graphic user interface, by which the user controls navigation and input using a mouse or touch screen, for example, Microsoft windows.
SAP	An integrated, packaged ERP software solution for big businesses, incorporating Financials, Logistics and HR modules. SAP is popular amongst the tier 1/top 1000 companies. There are approximately 80,000 implementations worldwide at time of writing that equates to roughly 70% of the ERP software market.
SAPGUI	The SAP graphical user interface, specifically used to execute SAP applications.
SDK	Software development kit. These are a standard suite of common functions, provided by the software vendor, that help programmers to develop programs to communicate and interact with their product.

Term	Description
SOAP	Simple Object Access Protocol. A means to transfer business documents between systems using an XML standard for messaging.
TSX	The Sky TSX (TCPIP Sockets Exchange) component of ECS provides a means to send files and messages between SAP systems and external systems.
VAN	Value added network. A service provided usually by telecommunications companies to transmit data between customer and suppliers computer systems.
VPN	Virtual Private Network. VPN enables secure connections between two computer systems through the internet. This is a very cost effective way to transmit data.
XML	Extensible Mark up Language. XML is a language used to describe data. It is not a standard. XML is fast becoming the mechanism to describe business transactions, super ceding the more cumbersome "fixed" data records that were previously prevalent in EDI.
XSD	XML Schema Definition. XSD provides a means of defining a valid structure for a particular XML document.

Term	Description
ABAP	SAP own internal programming language. ABAP allows SAP and its customers to develop and deploy R/3 applications. ABAP is also portable to any computer platform that R/3 supports, without change.
API	Application programming interface. A standard documented function, object method or program that you may invoke programmatically to perform a specific task. Most software vendors provide one or more APIs in the form of a software development kit (SDK).
ECS	The Sky ECS (Event Control System), used to manage interfaces and background processing in SAP.

Term	Description
HTML	The basis of the World Wide Web. HTML is a Document Type Definition (DTD), or subset, of the Standard Generalized Mark-up Language (SGML). It is a common language that allows user to define graphic pages for display on Web browsers. SGML is an open document definition language much in use in the publishing industry.
HTTP	HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system. It is a generic stateless object-oriented protocol that you may use for many similar tasks such as name servers, and distributed object-oriented systems, by extending the commands, or "methods", used. A feature if HTTP is the negotiation of data representation, allowing systems to build independently of the development of new advanced representations.
Java	A highly portable programming language designed by Sun. Java is widely used on all major computing platforms including Windows, Unix, AS/400 and IBM mainframe systems. Java allows programmers to seamlessly execute their applications, irrespective of the operating system or hardware. Refer http://www.java.sun.com for more details.
JDBC	An open database connectivity standard for Java applications, similar in concept to Microsoft ODBC. Vendors provide JDBC drivers that support various levels of compliance to the JDBC standard. These JDBC drivers then handle the communication with SQL relational databases, therefore abstracting any specific knowledge of the type of database away from the application.
JVM	A Java Virtual Machine is a Java run-time environment for a specific operating system and processor. The Java application runs inside of the JVM that abstracts it from having to know specifics about the environment, it is running in. JVMs are available for all major devices and operating systems. A comprehensive list is published by Sun, the owners of the Java standard. Refer http://www.java.sun.com for details.

Term	Description
SAP	An integrated, packaged ERP software solution for big businesses, incorporating Financials, Logistics and HR modules. SAP R/3 is popular amongst the tier 1/top 1000 companies. There are approximately 36,000 implementations worldwide at time of writing that equates to roughly 70% of the ERP software market. Refer http://www.sap.com for more details.
SAPGUI	The SAP graphical user interface, specifically used to execute R/3 and R/2 applications.
SDK	Software development kit. These are a standard suite of common functions, provided by the software vendor, that help programmers to develop programs to communicate and interact with their product.
TCP/IP	Transmission Control Protocol (TCP) provides a reliable, session-based service for the delivery of sequenced data packets across an internet. The Internet Protocol (IP) provides the network layer of the TCP/IP stack. It provides the basic mechanism to route packets of data packets on the internet.

Term	Description
TCP/IP sockets and ports	Every computer and 'network' aware device has a unique IP address on the network to identify itself. Within its unique address, each computer / device may have one or more 'network aware' services that actively transmit, receive or passively listen for information on the network. A unique port number is used to discriminate between the different network services within the same computer / device. A port number (typically 16 bit) is anything from 1 - 32K where 1 - 1,024 is normally reserved by the operating system. Standard ports include 22-FTP and 23-Telnet. Most operating systems support a services file where you may document customer IP address and Port configurations. TSX IP and Port configurations are totally configurable. Refer Configuration options for more details on selecting and specifying ports. TSX transfers all messages and files using the universal TCP/IP sockets protocol. Sockets allow for the bi-directional transfer of information (messages) between two unique points on the network (similar to a telephone conversation). The sender and receiver are uniquely identified by an IP address and port, for example, 192.168.2.70 - 5082 (similar to a telephone number). You need to follow a conversation protocol between the sender and receiver to transfer data, for example, open a connection, write a message, and wait for response (similar to a telephone conversation).
TSX	The Sky Technologies TCP/IP Sockets Exchange software enables applications to send messages, files and interface with devices over a TCP/IP network. It is essentially a messaging layer that can enable direct TCP/IP calls from SAP R/3.

Designing Peer-to-Peer Messaging Interfaces with SAP

You need to carefully think through direct conversational program-to-program interfaces before you implement them. At the end of the day, TSX only provides the means for the conversation and has no control over the conversation itself. Sky Technologies is willing to provide consultancy or advice on interface design. See the section Who is SkyTechnologies to contact Sky technologies.

Agree and Design the Message Protocol

1. Agree the conversation protocol. The sending and receiving system needs to understand the message layout and respond accordingly either with another transaction or a confirmation.
2. Agree on basic message format, that is, record type (header, data, and footer). It is a good idea to implement run number, length and serialisation controls to ensure the integrity of the transfer.
3. Agree message delimiter. How does TSX know when the message ends? TSX supports many different methods, that is, delimiter, fixed length and imbedded length.
4. Agree timeout controls. How does the application react if it has had no response for a period of time.
5. Design the TSX architecture to use. The TSX components play different roles, and you may use in isolation or together depending on the requirement and who starts the conversation.

Agree the Network Connection Options and IP Addresses / Ports to use for the Message Transfer

> ***Note:*** You are recommended to not to use the default TSX ports (508x), but configure message ports for each interface with specific configuration for that interface.

Configure TSX with the Communication Options and Message Configurations

Update `tsx.cfg` with the IP/Port configuration and SAP with a TSX logical destination and message type definition.

Implement the Business Logic

The program logic that is to verify and process the message contents.

Note: In a two way interface, that is, inbound and outbound, it is preferable to implement a "duplex" design where TSX does not share the same address/port for messages. This ensures that confirmations/ error processing can still be sent, even though the inbound port is blocked or unavailable.

7. HTTP Services

Kony for SAP HTTP Services provide a simple way for devices and applications to interact with data in SAP environments. With HTTP Services you may expose existing data objects, local database tables as well as SAP functionality.

7.0.0.1 Key Topics

Introducing HTTP Services
Installing Access Gateway to use HTTP Services
Developing Kony for SAP HTTP Services
Managing HTTP Services in Kony for SAP

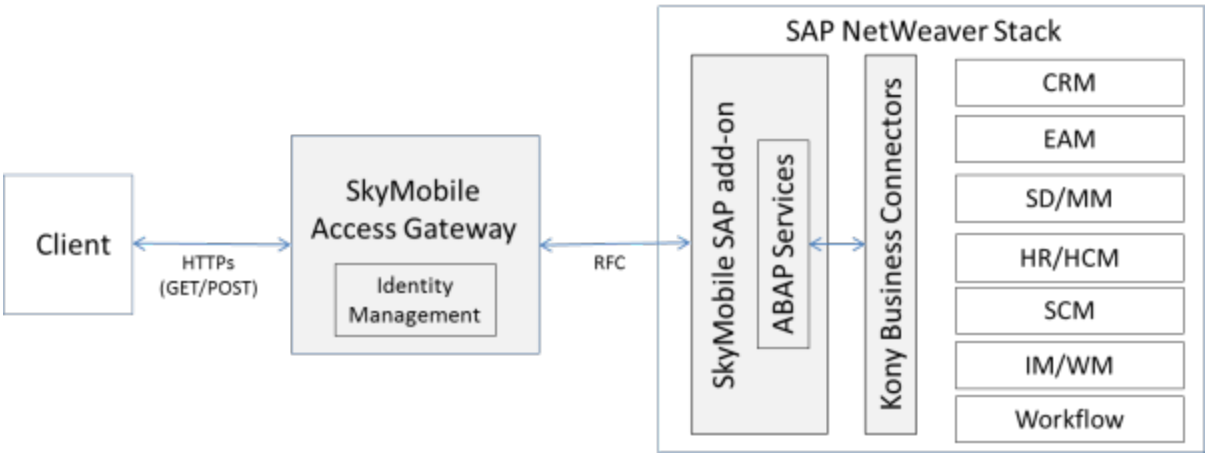
7.1 Introducing HTTP Services

This section provides an introduction to Kony for SAP HTTP Services that includes an overview of the components involved, details on various services that are provided and how you can use, HTTP headers supported and more.

7.1.1 Key Topics

7.1.2 Overview

Kony for SAP HTTP Services provide design-time and run time features to expose data in SAP backend systems. The basic architecture is shown in the diagram below and further details for each component is provided in the following sections.



7.1.2.1 Client

The client can be any device or application that can make HTTP(S) calls. The client issues HTTP calls (GET, PUT, PATCH, POST and DELETE) to an exposed HTTP Services port on a Kony for SAP Access Gateway and gets a response from the same gateway.

7.1.2.2 Access Gateway

You need to configure the Access Gateway with an HTTP Services port to receive inbound HTTP calls. The gateway extracts the call details (HTTP headers, service name, version and arguments) and passes these details to Kony for SAP HTTP Service Broker in SAP for processing. When the services broker provides a response, the gateway formats the call details and passes the response back to the client.

7.1.2.3 SAP

The HTTP Service Broker receives the request and performs some validation checks to ensure that the call is valid that includes:

- a valid session key provided
- the service specified is valid and active
- required call parameters provided

If the validation checks are passed, the broker then passes the call to the appropriate service handler for processing.

7.1.3 Call Format

The URL for a Kony for SAP HTTP Service is made up of a number of components. These are summarized in the table below.

Component	Description
Server Address	The IP address or fully qualified domain name of the Access Gateway
Port	Port number of the HTTP Services Port on the Access Gateway
Version	The version number of the service that is called. At the moment this is only v1
Service Name	The name of the service that is called. For example Logon, Metadata
Arguments	A service may take arguments to specify the keys of a record to retrieve or query string parameters

Example for Basic URL:

```
https://myserver.mydomain.com:1234/v1/BusinessPartners
```

Example for Service Argument:

```
https://myserver.mydomain.com:1234/v1/BusinessPartners/
('0000123456')
```

Example for Query String:

```
https://myserver.mydomain.com:1234/v1/BusinessPartners?$filters=country[eq]Australia
```

Note: You may omit the API Version. The URL may not include the API version. If this is the case, then you assume the version to be v1.

7.1.4 Service Types

There are two types of service handler - system services (Logon, Logoff, Heartbeat, Media, Metadata and Ping) and user built services. The table below provides further details on these services.

Service	Purpose	GET Supported	PUT Supported	PATCH Supported	POST Supported	DELETE Supported

Service	Purpose	GET Supported	PUT Supported	PATCH Supported	POST Supported	DELETE Supported
Logon	A call to the Logon service is required before any other requests can be performed. The Logon service call uses the configured Kony for SAP Identity Module and if successful returns a session key.	✓	✗	✗	✗	✗
Logoff	The Logoff service removes session details in the backend SAP environment.	✓	✗	✗	✗	✗
Heartbeat	The Heartbeat service allows devices to check for any notifications.	✓	✗	✗	✗	✗

Service	Purpose	GET Supported	PUT Supported	PATCH Supported	POST Supported	DELETE Supported
Media	The Media service is used to get, create and / or update a binary object.	✓	✓	✓	✓	✓
Metadata	The Metadata service provides details about the resource services exposed.	✓	✗	✗	✗	✗
Ping	The Ping service provides a basic test routine to enable connectivity to be tested.	✓	✓	✓	✓	✓

Service	Purpose	GET Supported	PUT Supported	PATCH Supported	POST Supported	DELETE Supported
Resource	Once the client performs a log on service call and returns a session key, it can issue Resource service calls to create/read/update records.	✅	✅	✅	✅	✅

7.1.5 HTTP Headers

A number of custom HTTP headers are used in HTTP Service calls. This table below summarizes each of the headers and the call type that support it. Further details on each header is provided below.

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-User	You need to provide this header when you make a call to the Logon service. It is used to specify a User ID for authentication. The header is not applicable for any of the other services.	✓	✗	✗	✗	✗	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Password	You need to provide this header when you make a call to the Logon service. It is used to specify a password for authentication. The header is not applicable for any of the other services.	✓	✗	✗	✗	✗	✗
KonySAP-Request-Caller-ID	You need to provide this header when you make a call to the Logon service. It is used to specify a Server ID. The header is not applicable for any of the other services.	✓	✗	✗	✗	✗	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Caller-Group	This header is optional when you make a call to the Logon service. It is used to specify a Server Group. The header is not applicable for any of the other services.	✓	✗	✗	✗	✗	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Identity-Service	This header is optional when you make a call to the Logon service. It is used to specify the Identity Service that should be used for authentication. The header is not applicable for any of the other services. ***Note:*** This option is ignored if the Access Gateway configuration does not allow the Identity Service to override.	✓	✗	✗	✗	✗	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-SSO-Token	This header is returned by the Logon service if the SAP SSO Identity Management Module is used. In such a situation, you need to provide this header on any subsequent calls to other services otherwise a 401 Unauthorized return code occurs.	✗	✗	✓ (If using SAP SSO)	✓ (If using SAP SSO)	✓ (If using SAP SSO)	✓ (If using SAP SSO)

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Session-Key	This header is returned by the Logon service. This header must be provided on any subsequent calls to other services otherwise a 401 Unauthorized return code occurs.	✗	✓	✓	✓	✓	✓
KonySAP-Request-SAP-Trace	This header is optional when you make a call to any service. It is used to enable HTTP Service tracing in SAP. It takes the value *truefalse* and defaults to *false*.	✓	✓	✓	✓	✓	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonyS AP-Reques t-Http-Overrid e-Method	This header can be passed to override the original request method with which the service call was made	✗	✗	✗	✗	✓	✓
KonyS AP-Reques t-Nested-Xml	This is a Boolean header which can be used to control if the response is to be represented in a nested or non-nested format	✗	✗	✗	✗	✓	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Brief-Response	You can use this boolean header can on POST/PATCH/PUT calls. Normally on a POST/PATCH/PUT call a full representation of the record(s) created / updated are provided in the response. If this header is set to *true*, then just the record key(s) are provided rather than a full representation.	✗	✗	✗	✗	✓	✗
KonySAP-Response-Time-SAP	This header is returned by all service calls. It indicates the processing time in milliseconds that the request took in SAP.	✓	✓	✓	✓	✓	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Response-Time-Gateway	This header is returned by all service calls. It indicates the total processing time in milliseconds that the request took.	✓	✓	✓	✓	✓	✓
KonySAP-Request-Media-Length	This header is used when passing binary data for a media service. This is useful for buffered uploads and represents the size (in bytes) of the current chunk. Size refers to the sizeof the un-encoded data.	✗	✗	✗	✗	✗	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonyS AP-Request-Media-Offset	This header is used when uploading a buffered chunk of binary data for a media service. This represents the offset (in bytes) of the current buffer .	✗	✗	✗	✗	✗	✓
KonyS AP-Request-Media-Size-Bytes	This header when used for a binary data upload in a media service represents the total size of the binary object being uploaded. This is used to determine if an upload is complete or if there is more to be uploaded	✗	✗	✗	✗	✗	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonyS AP-Reques t-Conten t-Encodin g	This header is used for a binary data upload in a media service. Currently only “base64” is supported.	✗	✗	✗	✗	✗	✓
KonyS AP-Respon se-Media-Size-Bytes	This header is returned when you make a get call to the Media service against a specific binary object. It indicates the size of file in bytes.	✗	✗	✗	✗	✗	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonyS AP-Respon se-Media-Name	This header is returned when you make a get call to the Media service against a specific binary object. It indicates the name of the file excluding the file extension.	✗	✗	✗	✗	✗	✓
KonyS AP-Respon se-Media-Type	This header is returned when you make a get call to the Media service against a specific binary object. It indicates the extension of the file.	✗	✗	✗	✗	✗	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Timestamp	This header is optional when you make a get call to a resource service. It is used to pass in a last timestamp value on a request so that you can process refresh (that is, delta).	✗	✗	✗	✗	✓	✗
KonySAP-Request-No-Deleted	This header can be used along with the KonySAP-Request-Timestamp header or a URL containing a timestamp field filter to exclude any deleted records from the response	✗	✗	✗	✗	✓	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Response-Timestamp	This header is returned when you make a get call to a resource service. It contains the timestamp of the call that was just made. You can pass this timestamp in on subsequent get calls through the KonySAP-Request-Timestamp header so that refresh (that is, delta) processing can be performed.	✗	✗	✗	✗	✓	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Content-Type	Used on a request to specify the content type being used. If not present, it is defaulted to XML. Valid values are XML, JSON. ***Note:*** In the initial release only XML format is supported. JSON, and other formats are supported in future releases.	✓	✗	✓	✓	✓	✓

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Response-Content-Type	Used on a response to specify the content type being used. Valid values are XML, JSON.	✓	✓	✓	✓	✓	✓
KonySAP-Request-Buffer-Size	Used to specify a buffer size (in bytes) to be used for the request. If not provided then the Kony for SAP default buffer size configured in SAP are used.	✗	✗	✗	✗	✓	✗
KonySAP-Request-Buffer-ID	When making a request to get further buffers, this header is used to specify the buffer id for the request.	✗	✗	✗	✗	✓	✗

Header	Purpose	Logon Service	Logoff Service	Heartbeat Service	Metadata Service	Resource Service	Media Service
KonySAP-Request-Buffer-Seq	This header can be used to re-read the more recent buffer sequence again in case of a failure.	✗	✗	✗	✗	✓	✗
KonySAP-Response-Buffer-Size	Buffer size (in bytes) used for the response	✗	✗	✗	✗	✓	✗
KonySAP-Response-More-Buffers	Indicates that further response buffers are available in SAP for this request.	✗	✗	✗	✗	✓	✗

Note: The actual header names may vary. The table above shows the default HTTP header names. The header prefix (KonySAP) is configurable on the Sky Gateway.

In addition, there is a configuration option for the HTTP Services port: SERVER.HTTPSERVICEPORT:PORT15099.ALLOWHEADERSINBODY. This option defaults to *false*. When set to *true*, the HTTP response headers are replicated in the response

body as attributes of a headers element. An example is shown below.

```
<entry>
<headers KonySAP-Session-Key="K0EQ5CFY7OCB8FPIQXM40SYOLMSQFBVL" />
<user>tester</user>
<workarea>WORKAREA</workarea>
<workgroup>MOM</workgroup>
<language>EN</language>
<parameter2>TESCO</parameter2>
</entry>
```

In addition, a number of standard headers are used. These are shown in the table below.

Header	Purpose
Accept-Encoding	You can provide this header on requests to indicate that compression is supported by the calling device. Valid values are: *gzip* and *deflate*.
Access-Control-Request-Headers	You can provide this header on requests by some clients where a cross site request is made. Used when you issue a preflight request to let the server know what HTTP headers are used when the actual request is made.
Access-Control-Request-Method	You provide this header on requests by some clients where a cross site request is made. Used when you issue a preflight request to let the server know what HTTP method are used when the actual request is made.
Access-Control-Expose-Headers	You provide this header on responses and use to whitelist headers that browsers are allowed to access.

Header	Purpose
Access-Control-Allow-Headers	You provide this header in response to a preflight request to indicate which HTTP headers you can use when you make the actual request.
Access-Control-Allow-Methods	You provide this header in response to a preflight request to indicate which HTTP methods you can use when you make the actual request.
Access-Control-Allow-Origin	You provide this header on all responses. Its value is set to the name(s) of headers that are returned with the response.
Content-Encoding	You provide this header on all responses that had a request header of accept-encoding. Its value is set to a compression method that matches the request value. Possible values are *gzip* and *deflate*.
Content-Length	If a response body is sent back, this header is provided and has the length in bytes of the response body.
Content-Type	You provide this header on all responses. With the current version, the value is always set to *text/xml*.
User-Agent	You provide this header on requests to provide the calling device details. Any value is accepted. At present, this header is not used in any way.

7.1.5.1 Using the BATCH Content Type

The BATCH content type performance option is designed to significantly reduce the size of the response data generated by GET processing upto 75 percent. The BATCH option is applicable only when processing large collections of data. To invoke the BATCH option, specify the request-content-type=BATCH header as part of the request. The result is a '|' delimited list of attribute values with their

tags removed. Each collection of raw <data> rows is preceded by an <attributes> tag that lists the associated column attribute names.

For example, normal verbose XML output is below:

```
<VTI_SAMPLE_ORDER>
           <ORDER_NUMBER></ORDER_NUMBER>
           <DESCRIPTION></DESCRIPTION>
           <COMPANY></COMPANY>
           <LDBTSTAMP>20140527215327</LDBTSTAMP>
           <DELETED></DELETED>
           <TOTAL_VALUE></TOTAL_VALUE>
           <ORDER_TYPE></ORDER_TYPE>
           <CONFLICT_TSTAMP></CONFLICT_TSTAMP>
      </VTI_SAMPLE_ORDER>
```

The above code is converted into a much more condensed format as below:

```
<VTI_SAMPLE_ORDER>
<attributes>ORDER_
NUMBER|DESCRIPTION|COMPANY|LDBTSTAMP|DELETED|TOTAL_VALUE|ORDER_
TYPE|CONFLICT_TSTAMP
</attributes>
<data>11|Order number 011 was generated
automatically.|SAFEWAY|20130904045806||113.85000||</data>
<data>12|Order number 012 was generated
automatically.|SAFEWAY|20130904045806||124.20000||</data>
<data>15|Order number 015 was generated
automatically.|SAFEWAY|20130904045806||155.25000||</data>
<data>16|Modified short
description|TESCO|20140514102257||165.60000||</data>
<data>17|Order number 017 was generated
automatically.|TESCO|20130904045806||175.95000||</data>
```

```
<data>20|Order number 020 was generated
automatically.|TESCO|20130904045806||207.00000||</data>.
```

Note: The single <attributes> tag lists the column attribute names and the repeating <data> rows with no tags. If you use any special **suppress output if no value type** options, the <attributes> tag may be repeated if a difference is detected. So, when parsing the batch output, you first detect the <attributes> row within the parent structure tag and then based on position all following <data> columns.

7.1.6 Logon Service

You use the Logon service to authenticate a user. If the logon is successful, a unique session key is returned. You need to provide this session key on all subsequent calls. A Logon call must be processed successfully before any other communication is allowed. The table below provides details on what headers and body are expected/returned.

Item	Detail
Methods supported	GET
Item context	Not supported
Collection context	Supported
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-User • KonySAP-Request-Password • KonySAP-Request-Caller-ID You have the option of providing the following header: • KonySAP-Request-Caller-Group
HTTP Response Headers	The following headers are returned: • KonySAP-Session-Key • KonySAP-SSO-Token (if using SAP SSO IdM)
Query string parameters	Not supported.
Request XML	Not supported.

Item	Detail

Response XML	An Example of a Response :
	
	`<entry>`
	`<user id="User ID">`
	`<language>EN</language>`
	`<workgroup>Some workgroup</workgroup>`
	`<workarea/>`
	`<email>abc@some.org</email>`
	`<phone/>`
	`<division/>`
	`<country>Australia</country>`
	`<region>Victoria</region>`
	`<parameter1>12345</parameter1>`
	`<parameter2>67890</parameter1>`
	`<parameter3/>`
	`<parameter4/>`
	`<parameter5/>`
	`<parameter6/>`
	`<parameter7/>`
	`<parameter8/>`
	`</user>`

Possible Return Codes	• 200 - if call is successful • 400 - if authentication fails • 500 - System error (for example, SAP not available).

An example of Logon Request:

```
curl -X GET --header "KonySAP-Request-User: userid" --header
"KonySAP-Request-Password: password" --header "KonySAP-Request-
```

```
Caller-ID: callerid" --header "KonySAP-Request-Caller-Group:
callergroup" https://myserver:1234/v1/Logon
```

7.1.7 Logoff Service

You use the Logoff service to terminate a user session. The table below provides details on what headers and body are expected / returned:

Item	Detail
Methods Supported	GET
Item Context	Not supported.
Collection Context	Supported.
HTTP Request Headers	The following headers must be provided: • KonySAP-Request-Session-Key
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway
Query string parameters	Not supported.
Request XML	Not supported.
Response XML	None provided.
Possible Return Codes	• 200 - If the call is successful • 401 - If a valid session key is not provided • 500 - System error (for example, SAP not available)

An example of Logoff Request:

```
curl -X GET --header "KonySAP-Request-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Logoff
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.8 Metadata Service

The metadata service only supports a GET request. The call can take two forms:

- Retrieves a list of available services by calling the **/Metadata** service
- Retrieves full service details by passing the service name. For example **/Metadata/BusinessPartners**.

The tables below provides details on what headers and body are expected / returned.

7.1.8.1 Service List

Item	Detail
Methods Supported	GET
Item Context	Not Supported.
Collection Context	Supported.
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-Sso-Token (if using SAP SSO IdM)
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway

Item	Detail
Query string parameters	Not supported.
Request XML	Not supported.
Response XML	An example response: An example response: <entry> <service> <name>VTI_SAMPLE_ORDERS</name> <description>Sample orders data object</description> <library>VTI_INTERNAL</library> <group>STEVE</group> <url>http://192.168.2.52:15099/v1/VTI_SAMPLE_ORDERS</url> <metadata>http://192.168.2.52:15099/Metadata/VTI_SAMPLE_ORDERS</metadata> </service> <service> <name>FRED</name> <description>Some service called Fred</description> <library>VTI_INTERNAL</library> <group>STEVE</group> <url>http://192.168.2.52:15099/v1/Fred</url> <metadata>http://192.168.2.52:15099/Metadata/Fred</metadata> </service> </entry>

Item	Detail
Possible Return Codes	• 200 - If the call is successful • 401 - If a valid session key is not provided • 500 - System error (eg SAP not available)

An example Metadata Detail Request:

```
curl -X GET --header "KonySAP-Request-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Metadata
```

7.1.8.2 Service Details

Item	Detail
Methods Supported	GET
Item Context	Not Supported.
Collection Context	Supported.
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM)

Item	Detail
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway
HTTP Parameters	Not Supported.
Request XML	Not Supported.

Item	Detail
Response XML	Example Response: <entry> <service> <name>VTI_SAMPLE_ORDERS</name> <description>Sample orders data object</description> <library>VTI_INTERNAL</library> <group>STEVE</group> <url>http://192.168.2.52:/v1/VTI_SAMPLE_ORDERS</url> <metadata>http://192.168.2.52:/v1/metadata/VTI_SAMPLE_ORDERS</metadata> <methods> <get>true</get> <put>true</put> <patch>true</patch> <post>true</post> <delete>true</delete> </methods> <table maxlength="20"> <keys> <attribute mandatory="false"></attribute> </keys> <attribute name="TOTAL_VALUE"></attribute> <attribute description="VTI: Test packed field to demonstrate packed conversion"></attribute> <attribute type="xsd:decimal"></attribute> <attribute maxlength="9"></attribute> <attribute decimals="5"></attribute> <attribute mandatory="false"></attribute> <attribute name="ORDER_TYPE"></attribute> <attribute description="VTI: Dummy character 20 (demo purposes only)"></attribute> <table type="xsd:string"> <parent maxlength="20"> <relationship> <link>VTI_SAMPLE_ORDER.COMPANY=VTI_SAMPLE_COMPANY.COMPANY</link> </relationship> </parent> <keys> <attribute mandatory="false"></attribute> </keys> <attribute name="Timestamp">

Item	Detail
Response XML	<attribute description="VTI: Timestamp"></attribute> <attribute type="xsd:string"></attribute> <attribute maxlength="14"></attribute> <attribute mandatory="false"></attribute> </table> <table name="Order"> <parent description="Sample Order"> <relationship> <link>VTI_SAMPLE_ORDER.ORDER_ NUMBER=VTI_SAMPLE_ORDER_ITEM.ORDER_NUMBER</link> </relationship> </parent> <keys> <attribute name="ORDER_NUMBER"></attribute> <attribute name="ORDER_NUMBER"></attribute> </keys> <attribute description="VTI: Dummy char numeric 5"></attribute> <attribute type="xsd:string"></attribute> <attribute maxlength="5"></attribute>
	</table> </table> </table> </service> </entry>
Possible Return Codes	• 200 - If the call is successful • 401 - If a valid session key is not provided • 404 - If the service specified not found • 500 - System error (eg SAP not available)

An Example of Metadata Detail Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Metadata
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.9 Heartbeat Service

You use the Heartbeat service to allow a device to check whether it has any notifications on the backend. The table below provides details on what headers and body are expected / returned.

Item	Detail
Methods Supported	GET
Item Context	Not supported
Collection Context	Supported
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported.
Request XML	Not supported.

Item	Detail
Response XML	Example Response <entry> <service>https://192.168.2.62:1234/v1/BusinessPartners</service> <service>https://192.168.2.62:1234/v1/Activities</service> </entry>
Possible Return Codes	• 200 - If the call is successful • 401 - If a valid session key is not provided • 500 - System error (for example, SAP not available).

An example of Heartbeat Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Heartbeat
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.10 Media Service

You may provide access to binary objects stored in the binary repository using Media service.

7.1.10.1 Get Request

You use the GET request either to get a list of available binary objects or to retrieve a specific binary object. A number of optional HTTP query string parameters can be used when issuing a get request against the media service. These are shown in the table below.

Parameter	Purpose
$count	Get a count of available binary objects.
$top	Only valid when issuing a get against a specific binary object. Used to specify a buffer size in bytes.
$skip	Only valid when issuing a get against a specific binary object. Used to specify an offset size in bytes.

Item	Detail
Methods Supported	GET
Item Context	Supported. This gets a specific binary object.
Collection Context	Supported. This provides a list of binaries available for the device.

HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Media-Size-Bytes (When get request against an item. Not on collection) • KonySAP-Response-Media-Name (When get request against an item. Not on collection) • KonySAP-Response-Media-Type (When get request against an item. Not on collection).
Query string parameters	The following parameters are supported when operating against an item only: • $top • $skip
Request XML	Not supported

Response XML	An example of Response for collection:

```
<entry>
   <item>
      <key>https://192.168.2.62:1234/v1/Media
('0000012345')</key>
      <Name>0000012345</Name>
      <Version>00</Version>
      <Type>Documents</Type>
      <Description>Some document</Description>
      <Size>896</Size>
      <Extension>BMP</Extension>
      <Location>Documents</Location>
      <PhysicalName>0000012345</PhysicalName>

<MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash>
```

Response XML	<BinaryGroup>EAM</BinaryGroup> <ServerGroup>MyServer</ServerGroup> <ServerID>MyGroup</ServerID> <ClassificationField/> <ClassificationValue/> </item> <item> <key>https://192.168.2.62:1234/v1/Media('fred')</key> <Name>fred</Name> <Version>00</Version> <Type>Documents</Type> <Description>Some other document</Description> <Size>1024</Size> <Extension>pdf</Extension> <Location>Documents</Location> <PhysicalName>fred</PhysicalName> <MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash> <BinaryGroup>EAM</BinaryGroup> <ServerGroup>MyServer</ServerGroup> <ServerID>MyGroup</ServerID> <ClassificationField/> <ClassificationValue/> </item> </entry>
Possible Return Codes	• 200 - if the call is successful • 401 - If a valid session key is not provided • 404 - If the selection criteria provided do not match any records in the collection or a single item, key does not match • 500 - System error (for example, SAP not available).

An example of Media List Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media
```

An example of Media Retrieval Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media/
('file_name')
```

An example of Buffered Media Retrieval Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media/
('file_name')?$top=1024&$skip=4096
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorised).

7.1.10.2 Post Request

You use the Post request to create one or more new binary objects. As such, the request need to take action against a collection. If a post is attempted against a specific item in a collection, then a return code 404 (Not Found) is returned.

Item	Detail
Methods Supported	POST
Item Context	Not Supported
Collection Context	Supported

HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM) • KonySAP-Request-Content-Encoding. You need to provide the following headers when you use buffered uploads: • KonySAP-Request-Media-Length • KonySAP-Request-Media-Offset • KonySAP-Request-Media-Size-Bytes.
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported

Request XML	Details of binary object to create.
	An example of Request:
	
	`<entry>`
	`<item>`
	`<Name>0000012345</Name>`
	`<Type>Documents</Type>`
	`<Description>Some document</Description>`
	`<Extension>BMP</Extension>`
	
	`<MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash>`
	`<BinaryGroup>EAM</BinaryGroup>`
	`<ClassificationField/>`
	`<ClassificationValue/>`
	`</item>`
	`</entry>`

Response XML	Effectively a read request of the binary object that is created.
	An example of Response:
	
	`<entry>`
	`<item>`
	`<key>https://192.168.2.62:1234/v1/Media ('0000012345')</key>`
	`<Name>0000012345</Name>`
	`<Version>00</Version>`
	`<Type>Documents</Type>`
	`<Description>Some document</Description>`
	`<Size>896</Size>`
	`<Extension>BMP</Extension>`
	`<Location>Documents</Location>`
	`<PhysicalName>0000012345</PhysicalName>`
	`<MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash>`
	`<BinaryGroup>EAM</BinaryGroup>`
	`<ServerGroup>MyServer</ServerGroup>`
	`<ServerID>MyGroup</ServerID>`
	`<ClassificationField/>`
	`<ClassificationValue/>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call successful • 401 - If a valid session key is not provided • 500 - System error (for example, SAP not available).

An example of POST Request:

```
curl -X POST --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media
```

> ***Note:*** The POST call creates the binary header information (the file metadata) only. Once the post returns with the key value for the binary object, you then need to issue a PUT call against that key passing in the base64 encoded binary content.

7.1.10.3 Put Request

You use the Put request to update the contents of a binary object.

Item	Detail
Methods Supported	PUT
Item Context	Supported
Collection Context	Not Supported
HTTP Request Headers	The following headers must be provided: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM) • KonySAP-Request-Content-Encoding. The following headers must be provided when using buffered uploads: • KonySAP-Request-Media-Length • KonySAP-Request-Media-Offset • KonySAP-Request-Media-Size-Bytes.

HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported.
Request XML	The request body must contain the base64 encoded binary contents.

Response XML	Effectively a read request of the record (s) updated.
	An example of Response:
	
	`<entry>`
	`<item>`
	`<key>https://192.168.2.62:1234/v1/Media ('0000012345')</key>`
	`<Name>0000012345</Name>`
	`<Version>00</Version>`
	`<Type>Documents</Type>`
	`<Description>Some document</Description>`
	`<Size>896</Size>`
	`<Extension>BMP</Extension>`
	`<Location>Documents</Location>`
	`<PhysicalName>0000012345</PhysicalName>`
	`<MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash>`
	`<BinaryGroup>EAM</BinaryGroup>`
	`<ServerGroup>MyServer</ServerGroup>`
	`<ServerID>MyGroup</ServerID>`
	`<ClassificationField/>`
	`<ClassificationValue/>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call successful • 401 - If a valid session key is not provided • 404 - If the binary object specified is not found • 500 - System error (for example, SAP not available).

An example of PUT Request:

```
curl -X PUT --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media/
('file_name')
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.10.4 Patch Request

You use the Patch request to update the metadata of one or more binary objects.

Item	Detail
Methods Supported	PATCH
Item Context	Not Supported
Collection Context	Supported
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported

Request XML	Details of binary object to update.
	An example of Request:
	
	`<entry>`
	`<item>`
	`<Name>0000012345</Name>`
	`<Description>Modified description</Description>`
	`</item>`
	`</entry>`

Response XML	Effectively a read request of the binary object that was updated.
	An example of Response:
	
	`<entry>`
	`<item>`
	`<key>https://192.168.2.62:1234/v1/Media ('0000012345')</key>`
	`<Name>0000012345</Name>`
	`<Version>00</Version>`
	`<Type>Documents</Type>`
	`<Description>Modified description</Description>`
	`<Size>896</Size>`
	`<Extension>BMP</Extension>`
	`<Location>Documents</Location>`
	`<PhysicalName>0000012345</PhysicalName>`
	`<MD5Hash>1FAC326FAF3C9C0CBC72C791CBF83C49</MD5Hash>`
	`<BinaryGroup>EAM</BinaryGroup>`
	`<ServerGroup>MyServer</ServerGroup>`
	`<ServerID>MyGroup</ServerID>`
	`<ClassificationField/>`
	`<ClassificationValue/>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call is successful • 401 - If a valid session key is not provided • 404 - If the binary object specified is not found • 500 - System error (for example, SAP not available).

An Example of PATCH Request:

```
curl -X PATCH --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media
```

Note: The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.10.5 Delete Request

You use the Delete request to flag a binary object as logically deleted.

Item	Detail
Methods Supported	DELETE
Item Context	Not Supported.
Collection Context	Supported.
HTTP Request Headers	You need to provide the following headers: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported.

Item	Detail
Request XML	Details of record(s) to delete. An example of Request: <entry> <item> <key>https://192.168.2.62:1234/v1/Media ('0000012345')</key> <Name>0000012345</Name> </item> </entry>
Response XML	Effectively a read request of the record (s) deleted. An example of Response: <entry> <item> <key>https://192.168.2.62:1234/v1/Media ('0000012345')</key> </item> </entry>
Possible Return Codes	• 200 - if the call successful • 401 - If a valid session key is not provided • 404 - If the binary object specified is not found • 500 - System error (for example, SAP not available).

An example of DELETE Request:

```
curl -X DELETE --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW" https://myserver:1234/v1/Media
```

> ***Note:*** The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.11 Ping Service

You use the Ping service to provide a way to test connectivity from a device to SAP. The table below provides details on what headers and body are expected / returned:

Item	Detail
Methods Supported	GET, PATCH, POST, PUT, and DELETE
Item Context	Not supported
Collection Context	Supported
HTTP Request Headers	No headers are required
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway.
Query string parameters	Not supported.
Request XML	Not supported.

Item	Detail

Response XML	An example of Response:
	
	`<entry>`
	`<ping>`
	`<requestDate>2014-04-22</requestDate>`
	`<requestTime>15:14:28</requestTime>`
	`<requestMethod>GET</requestMethod>`
	
	`<requestSizeBytes>zero</requestSizeBytes>`
	`<sessionKey></sessionKey>`
	`<user></user>`
	`<callerID></callerID>`
	`<callerGroup>DEV</callerGroup>`
	`</ping>`
	`</entry>`

Possible Return Codes	• 200 - if the call is successful • 500 - System error (for example, SAP not available).

An example of Heartbeat Request:

```
curl -X GET https://myserver:1234/v1/Ping
```

7.1.12 Resource Service

The user develops the Resource services. The Resource services can either be built from a DOB or LDB using the generation wizard or coded manually. Resources services may support GET, PUT, PATCH, POST or DELETE calls depending on how they are developed. The following sections provide details on what headers and body are expected / returned for each method.

> ***Note:*** The HTTP header must include a session key. If a valid session key is not provided, then the request receives a return code 401 (Not Authorized).

7.1.12.1 Get Request

You use the GET request to read data. The request can take action against either a collection (to retrieve all or a selection of entries) or a specific item by specifying the key field (s). You can use a number of optional HTTP query string parameters when you issue a get request against a resource service. These are shown in the table below.

Parameter	Purpose

$filters	Used to specify one or more selection criteria on a Get request. Usage:
	• Comma delimited list of criteria (for example, field operator value)
	• Operators supported are: EQ, NE, GT, LT, GE, LE, LIKE, NOT LIKE, STARTS WITH, ENDS WITH, IN, NOT IN, BETWEEN.
	Note: If the following $filters criteria is provided:
	• $filters=timestamp[eq]*
	• and the service is against a DOB/LDB
	• and the field timestamp is nominated as a timestamp field
	SAP attempts to determine the last timestamp value for the Server ID/Group that corresponds to the session key used and effectively performs a refresh call.
	For Example :
	`https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne]Victoria`
	`http://myserver:1234/Orders?$filters=description[LIKE]'%from%'`
	`http://myserver:1234/Orders?$filters=description[NOT LIKE]'%from%'`
	`http://myserver:1234/Orders?$filters=description[STARTS WITH]'Test'`
	`http://myserver:1234/Orders?$filters=description[ENDS WITH]'Test'`

Parameter	Purpose

$filters	<pre>http://myserver:1234/Orders?$filters=order_number[IN] '00901';'00902';'00950'
	
	http://myserver:1234/Orders?$filters=order_number[NOT IN] '00901';'00902';'00950'
	
	http://myserver:1234/Orders?$filters=order_number [BETWEEN]'00901' and '01050'</pre>
	Note: You can logically group multiple conditions using parenthesis and logical operators as in the following example: http://myserver:1234/Orders?$filters=(order_number[bt]'00901' and '00904',or order_number[bt]'00950' and '00999'),or order_number[eq]'00004'

$select	Used to specify the columns to return on a Get request to restrict the data returned. If omitted, then all fields are returned. Usage: • Comma separated list of fields to return For example: <pre>https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne] Victoria&$select=partnerid,surname</pre>

Parameter	Purpose
$sort	Used to specify sort order criteria on a Get request. Usage: • Comma separated list of fields to sort by • Defaults to increasing sort order • Prefix field with hyphen to indicate decreasing sort order. For example : `https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne]Victoria&$sort=-partnerid`
$top	Used to retrieve up to a specific number of records. Usage: • Provide a maximum number of records to retrieve For example : `https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne]Victoria&$sort=-partnerid&$top=100`
$skip	Typically, used together with $top to allow pagination. This parameter indicates a number of records to exclude in retrieval. Usage: • Provide a number of records to exclude from the result set For example : `https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne]Victoria&$sort=-partnerid&$top=100&$skip=200`

Parameter	Purpose
$count	Used to retrieve a count of matching records. You can use "$count" on its own to get a count of records in the collection or it can use it together with $filters to retrieve a count of matching items. For example : `https://myserver:1234/v1/BusinessPartners?$filters=Country[eq]Australia,State[ne]Victoria&$count`

Item	Detail
Methods Supported	GET
Item Context	Supported. Reads the record matching the item key specified in the URL.
Collection Context	Supported. Reads the records matching any provided selection criteria.
HTTP Request Headers	The following headers are expected: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Content-Type.

Query String Parameters	Filters criteria: • Only applies when performing a get against a collection • Able to provide zero or more selection criteria as parameters in the URL • Use the "filters" parameter • If none specified, then all records are returned. Field restriction criteria: • Able to provide a comma delimited list of fields to return using the "select" parameter • If not specified, then all fields for the records are returned. Sort Order: • Able to provide a comma delimited list of columns to use as the sort order using the "sort" parameter • Use "-" prefix (for example,-CreateDate) to indicate decreasing order • If not specified, then the results are provided in increasing primary key order.
Request XML	Not Supported.

Response XML	An example of Response :

```
 <entry>
   <table name="PartnerDetails">
      <item>

<key>https://192.168.2.62:1234/v1/BusinessPartner
('0000012345')</key>
         <PartnerID>0000012345</PartnerID >
         <GivenName>Fred</GivenName >
         <Surname>Nerk</Surname >
         <table name="PartnerAddresses">
            <item>
               <AddressID>123456</AddressID >
               <StreetNumber>21A/35</StreetNumber >
               <StreetName>Dunlop</StreetName >
               <StreetType>Road</StreetType >
               <City>Mulgrave</City >
               <State>Victoria</State>
               <Country>Australia</Country>
               <ZipCode>3170</ZipCode>
            </item>
            <item>
               <AddressID>123457</AddressID >
               <StreetNumber>1</StreetNumber >
               <StreetName>Some</StreetName >
               <StreetType>Street</StreetType >
               <City>Melbourne</City >
               <State>Victoria</State>
               <Country>Australia</Country>
               <ZipCode>3000</ZipCode>
            </item>
         </table>
      </item>
   </table>
</entry>
```

Response XML	An Example of $count Response:
	
	`<entry>`
	`<total>25</total>`
	`<table name="PartnerDetails">`
	`<count>10</count>`
	`<table name="PartnerAddresses">`
	`<count>15</count>`
	`</table>`
	`</table>`
	`</entry>`

Possible Return Codes	• 200 - If the call is successful • 401 - If a valid session key is not provided • 404 - If the selection criteria provided do not match any records in the collection or a single item key does not match • 500 - System error (for example, SAP not available).

An example of Collection Get Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW"
https://myserver:1234/v1/BusinessPartners
```

An example of Collection Get Request with Query String:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW"
https://myserver:1234/v1/BusinessPartners?$filters=country[eq]
Australia,state[eq]Victoria&$selects=partnerid,name&$sort=-
createdate
```

An example of Item Get Request:

```
curl -X GET --header "KonySAP-Session-Key:
441RH6OHYH8Q61N2LFEOX2JFLNLCQXVW"
https://myserver:1234/v1/BusinessPartners/('0000001234')
```

7.1.12.2 Put Request

You use the Put request to update one or more records. The PUT update replaces the entire copy of each record(s) with the values provided in the XML body of the request. To only update specified fields, you need to use a PATCH request (see below).

Item	Detail
Methods Supported	PUT
Item Context	Supported. Updates the record specified in the URL
Collection Context	Supported. Updates the record(s) specified in the XML message body
HTTP Request Headers	The following headers are expected: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Content-Type.

Query String Parameters	Not supported.

Request XML	Need to update the complete record set and to provide in XML in the HTTP body.
	
	An example of Request :
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner ('0000012345')</key>`
	`<PartnerID>0000012345</PartnerID >`
	`<GivenName>Fred</GivenName >`
	`<Surname>Nerker</Surname >`
	`<table name="PartnerAddresses">`
	`<item>`
	`<AddressID>123456</AddressID >`
	`<StreetNumber>21A/35</StreetNumber >`
	`<StreetName>Dunlop</StreetName >`
	`<StreetType>Road</StreetType >`
	`<City>Mulgrave</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3170</ZipCode>`
	`</item>`
	`<item>`
	`<AddressID>123457</AddressID >`
	`<StreetNumber>1</StreetNumber >`
	`<StreetName>Some</StreetName >`
	`<StreetType>Street</StreetType >`
	`<City>Melbourne</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3000</ZipCode>`
	`</item>`
	`</table>`
	`</item>`
	`</entry>`

Response XML	Effectively a read request of the record(s) updated.
	An example of Response :
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner ('0000012345')</key>`
	`<PartnerID>0000012345</PartnerID >`
	`<GivenName>Fred</GivenName >`
	`<Surname>Nerker</Surname >`
	`<table name="PartnerAddresses">`
	`<item>`
	`<AddressID>123456</AddressID >`
	`<StreetNumber>21A/35</StreetNumber >`
	`<StreetName>Dunlop</StreetName >`
	`<StreetType>Road</StreetType >`
	`<City>Mulgrave</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3170</ZipCode>`
	`</item>`
	`<item>`
	`<AddressID>123457</AddressID >`
	`<StreetNumber>1</StreetNumber >`
	`<StreetName>Some</StreetName >`
	`<StreetType>Street</StreetType >`
	`<City>Melbourne</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3000</ZipCode>`
	`</item>`
	`</table>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - If the call is successful • 400 - If the item values provided in the body are invalid or missing • 401 - If a valid session key is not provided • 404 - If the selection criteria provided do not match any records in the collection or a single item key does not match • 500 - System error (for example, SAP not available).

7.1.12.3 Patch Request

You use the Patch request to update specific fields on one or more records (as opposed to the complete replacement that PUT does).

Item	Detail
Methods Supported	PATCH
Item Context	Supported.
Collection Context	Supported. Updates the record (s) specified in the XML message body.
HTTP Request Headers	The following headers are expected: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).

HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Content-Type
Query String Parameters	Not Supported.

Request XML	Only the fields of the record(s) that are to be updated need to be provided along with the key.
	An example of Request:
	`<entry>`
	`<item>`
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000012345')</key>`
	`<PartnerID>0000012345</PartnerID >`
	`<Surname>Nerker</Surname >`
	`</item>`
	`</entry>`

Response XML	Effectively a read request of the record (s) updated.
	An example of Response:
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000012345')</key>`
	`<PartnerID>0000012345</PartnerID >`
	`<GivenName>Fred</GivenName >`
	`<Surname>Nerker</Surname >`
	`<table name="PartnerAddresses">`
	`<item>`
	`<AddressID>123456</AddressID >`
	`<StreetNumber>21A/35</StreetNumber >`
	`<StreetName>Dunlop</StreetName >`
	`<StreetType>Road</StreetType >`
	`<City>Mulgrave</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3170</ZipCode>`
	`</item>`
	`<item>`
	`<AddressID>123457</AddressID >`
	`<StreetNumber>1</StreetNumber >`
	`<StreetName>Some</StreetName >`
	`<StreetType>Street</StreetType >`
	`<City>Melbourne</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3000</ZipCode>`
	`</item>`
	`</table>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call successful • 400 - If the item values provided in the body are invalid or missing • 401 - If a valid session key is not provided • 404 - If the binary object specified is not found • 500 - System error (for example, SAP not available).

7.1.12.4 Post Request

You use the Post request to create one or more new records in a collection. As such, the request need to take action against a collection. If a post is attempted against a specific item in a collection, then a return code 404 (Not Found) is returned.

Item	Detail
Methods Supported	POST
Item Context	Supported.
Collection Context	Supported. Creates the record (s) specified in the XML message body.
HTTP Request Headers	The following headers are expected: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM).

HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Content-Type.
Query String parameters	Not supported
Request XML	To create the complete record set An example of Request: <entry> <item> <PartnerID></PartnerID > <GivenName>Joe</GivenName > <Surname>Citizen</Surname > <table name="PartnerAddresses"> <item> <AddressID></AddressID > <StreetNumber>1</StreetNumber > <StreetName>Some</StreetName > <StreetType>Street</StreetType > <City>Mulgrave</City > <State>Victoria</State> <Country>Australia</Country> <ZipCode>3170</ZipCode> </item> </table> </item> </entry>

Response XML	Effectively a read request of the record(s) updated.
	An example of Response:
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000067890')</key>`
	`<PartnerID>0000067890</PartnerID >`
	`<GivenName>Fred</GivenName >`
	`<Surname>Nerker</Surname >`
	`<table name="PartnerAddresses">`
	`<item>`
	`<AddressID>123456</AddressID >`
	`<StreetNumber>21A/35</StreetNumber >`
	`<StreetName>Dunlop</StreetName >`
	`<StreetType>Road</StreetType >`
	`<City>Mulgrave</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3170</ZipCode>`
	`</item>`
	`<item>`
	`<AddressID>78901</AddressID >`
	`<StreetNumber>1</StreetNumber >`
	`<StreetName>Some</StreetName >`
	`<StreetType>Street</StreetType >`
	`<City>Melbourne</City >`
	`<State>Victoria</State>`
	`<Country>Australia</Country>`
	`<ZipCode>3000</ZipCode>`
	`</item>`
	`</table>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call is successful • 400 - If the item values provided in the body are invalid or missing • 401 - If a valid session key is not provided • 500 - System error (for example, SAP not available).

7.1.12.5 Delete Request

You use the Delete request to delete one or more records in a collection. You may use it either to provide a single record key in the URI or to provide multiple keys in the HTTP body XML.

Item	Detail
Methods Supported	DELETE
Item Context	Not Supported
Collection Context	Supported. This does a delete of the record (s) specified in the XML message body.
HTTP Request Headers	The following headers are expected: • KonySAP-Request-Session-Key • KonySAP-Request-SAP-SSO-Token (if using SAP SSO IdM)
HTTP Response Headers	The following headers are returned: • KonySAP-Response-Transfer-ID • KonySAP-Response-Time-SAP • KonySAP-Response-Time-Gateway • KonySAP-Response-Content-Type.

Item	Detail
Query string parameters	Not supported.

Request XML	Only required if acting in the collection context.
	An example of Request:
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000012345')</key>`
	`</item>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000012346')</key>`
	`</item>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner('0000012347')</key>`
	`</item>`
	`</entry>`

Item	Detail

Response XML	Keys of the record (s) deleted.
	An example of Response:
	
	`<entry>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner ('0000012345')</key>`
	`</item>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner ('0000012346')</key>`
	`</item>`
	`<item>`
	
	`<key>https://192.168.2.62:1234/v1/BusinessPartner ('0000012347')</key>`
	`</item>`
	`</entry>`

Possible Return Codes	• 200 - if the call is successful • 400 - If the item values provided in the body are invalid or missing • 401 - If a valid session key is not provided • 500 - System error (for example, SAP not available).

7.2 Installing Access Gateway to Use HTTP Services

The Installation specification provides details on how to install and configure an Access Gateway to use HTTP Services.

7.2.1 Key Topics

Prerequisites
Installation Process
Configuration Options

7.2.2 Prerequisites

The prerequisites to install Kony for SAP HTTP Services components are the same as that for Kony for SAP. Ensure that the prerequisites are met before you begin the installation.

7.2.3 Installation Process

To install the Kony for SAP HTTP Services components, follow these steps:

1. Ensure that the prerequisites are met.
2. Download a copy (v24.04 or above) of the MEAP Add-In and the Kony for SAP Application Server from the Kony Developer Portal.
3. Get a license.
4. Install the SAP Add-In.
5. Install an Access Gateway.
6. Configure an HTTP Service port on the gateway.
7. Perform a test call to the Ping service.

7.2.4 Configuration Options

You need to configure an HTTP Services port on the Access Gateway. The table below provides a summary of the key configuration options. For further details on the configuration options refer the configuration section.

Option	Details
PORT	The port number to use for the HTTP Services port
HOSTINTERFACE	The Host Interface handle to use for the HTTP Services port
DEFAULTIDENTITYSERVICE	The Identity Service handle to use for calls to the Logon service

An example of configuration:

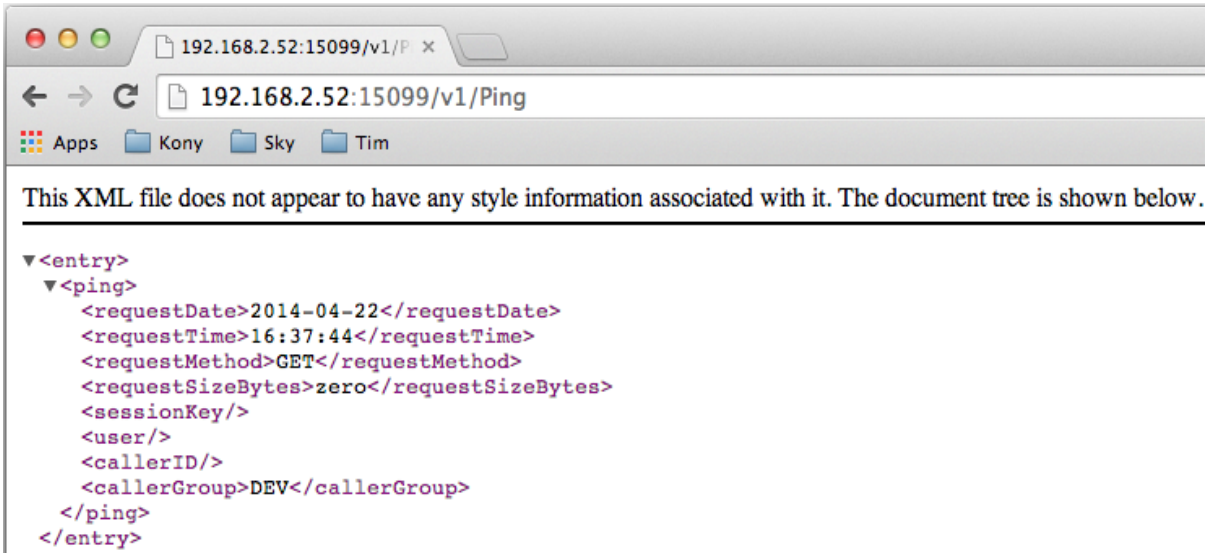
```
SERVER.HTTPSERVICEPORT:PORT15099.DEFAULTIDENTITYSERVICE = SKY
SERVER.HTTPSERVICEPORT:PORT15099.HOSTINTERFACE = RFCCLIENT
SERVER.HTTPSERVICEPORT:PORT15099.PORT = 15099
SERVER.HTTPSERVICEPORT:PORT15099.XMLDEBUG = true
```

7.2.4.1 Connection Test

You can use the Ping service to test that the HTTP Services port works correctly. To do this, you need the following information:

- IP Address or fully qualified domain name of your Access Gateway server
- Port number of the HTTP Services port on your Access Gateway

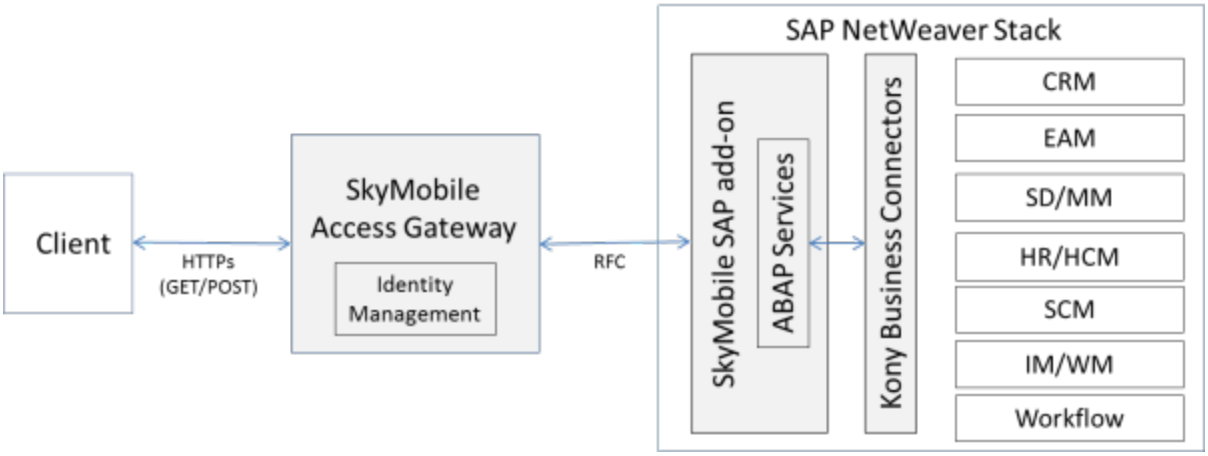
Once you have that information, you can open a web browser and go to the URL http://<server><port>/v1/Ping



7.3 Developing HTTP Services

The SAP Add-in Gateway HTTP Service (GWS) interface provides an easy way to expose SAP integration as either RESTful or web services to all manner of applications. It is primarily focused on integrating with Data Object (DOB) and local database (LDB) definitions. It is flexible to allow any type of service definition and SAP integration technique. The service definition requires an integration object to implement using the Integration Object Workbench (IOB) in SAP. The aim of the IOB workbench is to define the behavior of the service and the adaptors to use to retrieve and maintain data. The executable service is then generated as an ABAP proxy. The aim is to avoid to write code, although custom ABAP development is fully supported. This documentation covers the process to automatically generate and manually configure service definitions using the IOB workbench in SAP that is delivered as a standard component of the SAP add-in.

> ***Note:*** The Integration Object Workbench supports an entire range of integration options. Here, focus is on HTTP services.

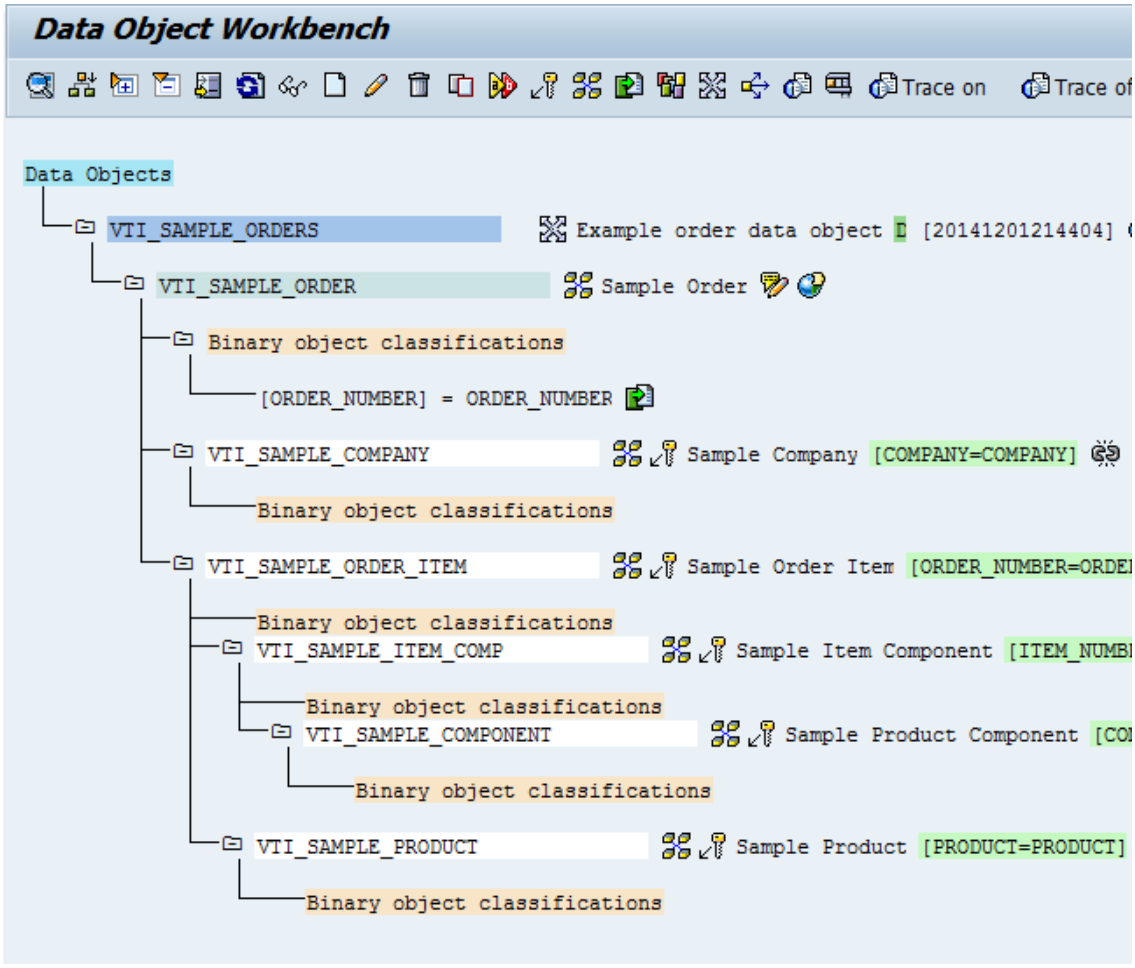


In the above diagram, an HTTP service port is exposed through the Access Gateway and the client interacts with this using standard HTTP GET/PUT/POST/PATCH/DELETE calls. There are some built-in services to handle logon, logoff, heartbeat, metadata and media requests. Other services are

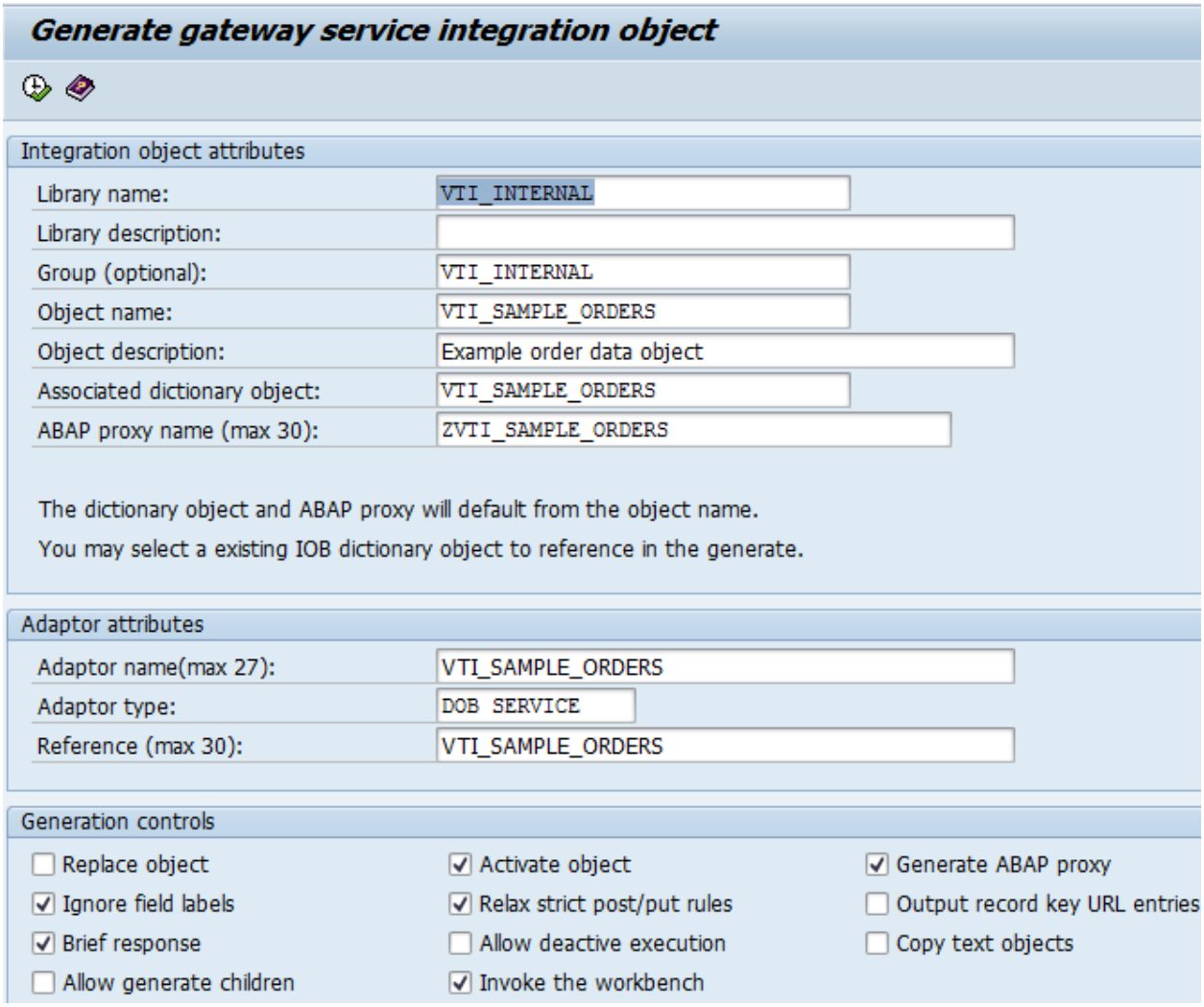
implemented using the Integration Object Workbench inside SAP and these interact with SAP to extract data and post transactions, typically using the Data Object (DOB) sub-system. The SAP add-in packaged CRM, EAM, HCM, and so on business connectors for SAP come standard with their own HTTP service definitions or can generate these easily enough off any DOB or LDB definition.

Automatically Generating Services

You can easily generate the HTTP services from existing Data Object, and Local Database (LDB) definitions. To do this, display the definitions using the YVTD or YVTL workbenches, select the definition that you want to generate the service from, and click **Generate Gateway Service** on the application toolbar or select the option from the **Utilities** menu. The workbench also displays whether the object or table has an associated service by displaying the **Gateway service** icon on the right hand side of the hierarchy as in the screen shot below. You may click the icon hotspot to display the integration object definition.



When you invoke the utility to automatically generate a service, the following dialog appears:



Field	Mandatory or Optional	Description
Library Name	Mandatory	The name of the library definition that the object belongs to. This is defaulted from the add-in group.
Library Description	Optional	The library description. This is defaulted from the add-in group.

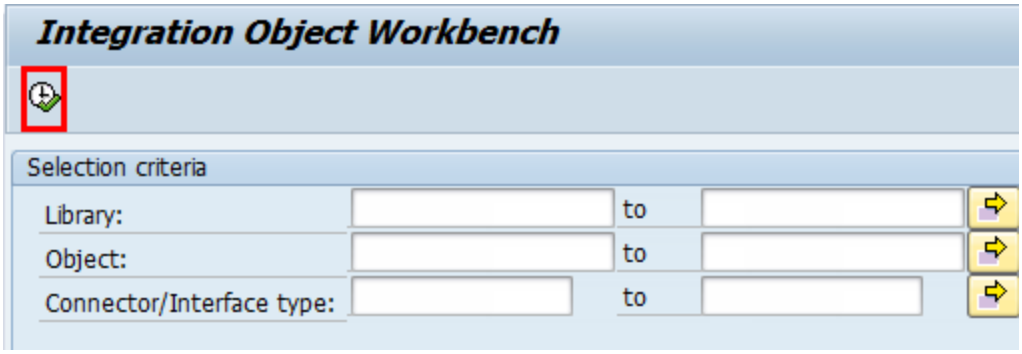
Field	Mandatory or Optional	Description
Group	Optional	This is defaulted from the add-in group. This is an optional field.
Object Name	Mandatory	This is a unique name of an integration object definition. It is defaulted from either the DOB or LDB name.
Object Description	Optional	The text description of the IOB definition. This is defaulted from the DOB or LDB definition.
Associated dictionary object	Optional	Each Integration Object has one or more associated dictionary objects. The IOB DD object defines the structures and their relationships that are to be used to map data. DD object may be shared across integration objects and are automatically generated by this utility.
ABAP Proxy	Optional	This is the name of the ABAP proxy program that is generated from the integration object definition. It should follow the usual ABAP naming convention prefixed with Z, Y or /namespace/. The name is defaulted from the object name specified above.
Adaptor		Description
Name	Optional	The name of the service adaptor, this is defaulted from the LDB name for which you want to generate the service from.
Type	Mandatory	This is the type of service adaptor used to extract from, or post transactions to, SAP, for example, DOB SERVICE, LDB SERVICE, and so on.
Reference	Mandatory	This refers to the name of the LDB that you want to generate the service from.
Controls		Description

Field	Mandatory or Optional	Description
Replace object	Optional	As the title implies, existing definitions are not replaced unless this option is specified.
Activate Object	Optional	Activates the object that allows the service to be immediately available, exposed to external clients.
Generate ABAP Proxy	Optional	Generate the ABAP Proxy program.
Invoke workbench	Optional	After the object is generated, launch the integration object workbench.
Ignore field labels	Optional	Ignores field labels for reserved strings
Relax strict post/put rules	Optional	Ignores restrictions about unique records and duplicate records during PUT and POST operations
Output record key URL entries	Optional	For every LDB record in the response body, a <recordkey> element is populated that holds the full URL of the particular record.
Brief response	Optional	Ensures that no response body is returned for PUT, POST and PATCH calls
Allow deactive execution	Optional	Allow execution of Service Call against the DOB or LDB even if the corresponding integration object is in a deactivated state
Copy text	Optional	Copies text descriptions from the DOBs, LDBs and LDB fields

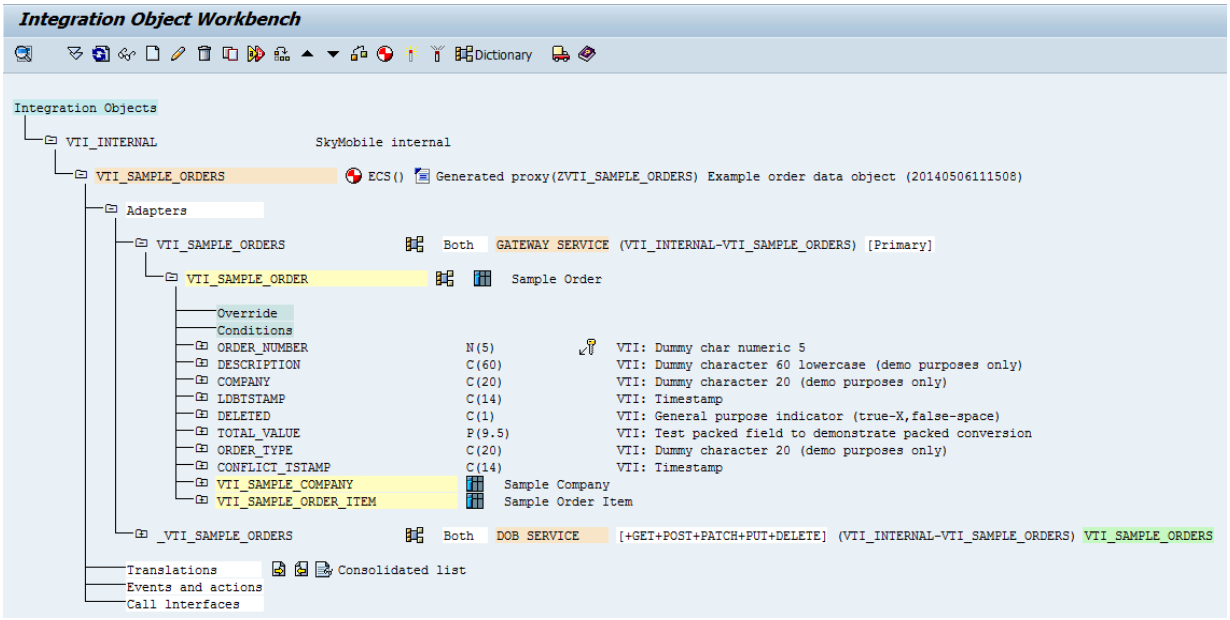
Field	Mandatory or Optional	Description
Allow Generate Children	Optional	When used with DOB services, generates service definitions for all LDBs defined in the DOB hierarchy

7.3.1 The Integration Object Workbench

When the service is generated, you may view its definition in the IOB workbench. You may start this service using the **YIOW** transaction or through **Navigator** > **Workbenches** > **Integration Objects** menu option from any other transaction.
The **Integration Object Workbench** selection screen appears.



This selection screen is used to filter the display to a specific library, object and so on. Click **Execute**.
The main **Integration Object Workbench** appears.



Libraries and Objects

The integration object workbench is organized into libraries that are defaulted to the add-in group when the objects are generated automatically. Within the library, you have integration objects and dictionary objects. Dictionary objects describe the format and complex structure relationships (if any) of the data definitions and attributes. The Dictionary objects may be shared between one or more integration objects. Integration objects define the input, transformation and output processing and support a wide number of applications and different integration methods such as files, SAP BAPIs and functions, SQL, TSX messaging. In this case, just discuss the add-in Gateway, DOB, and LDB services adaptors.

Adaptors

An integration object requires a primary adaptor that deals with the input and output. You can then define one or more other adaptors to handle different aspects of the integration with the data objects, and local database tables. By default, when you automatically generate an integration object, you get a primary gateway service adaptor and a DOB or LDB service adaptor.

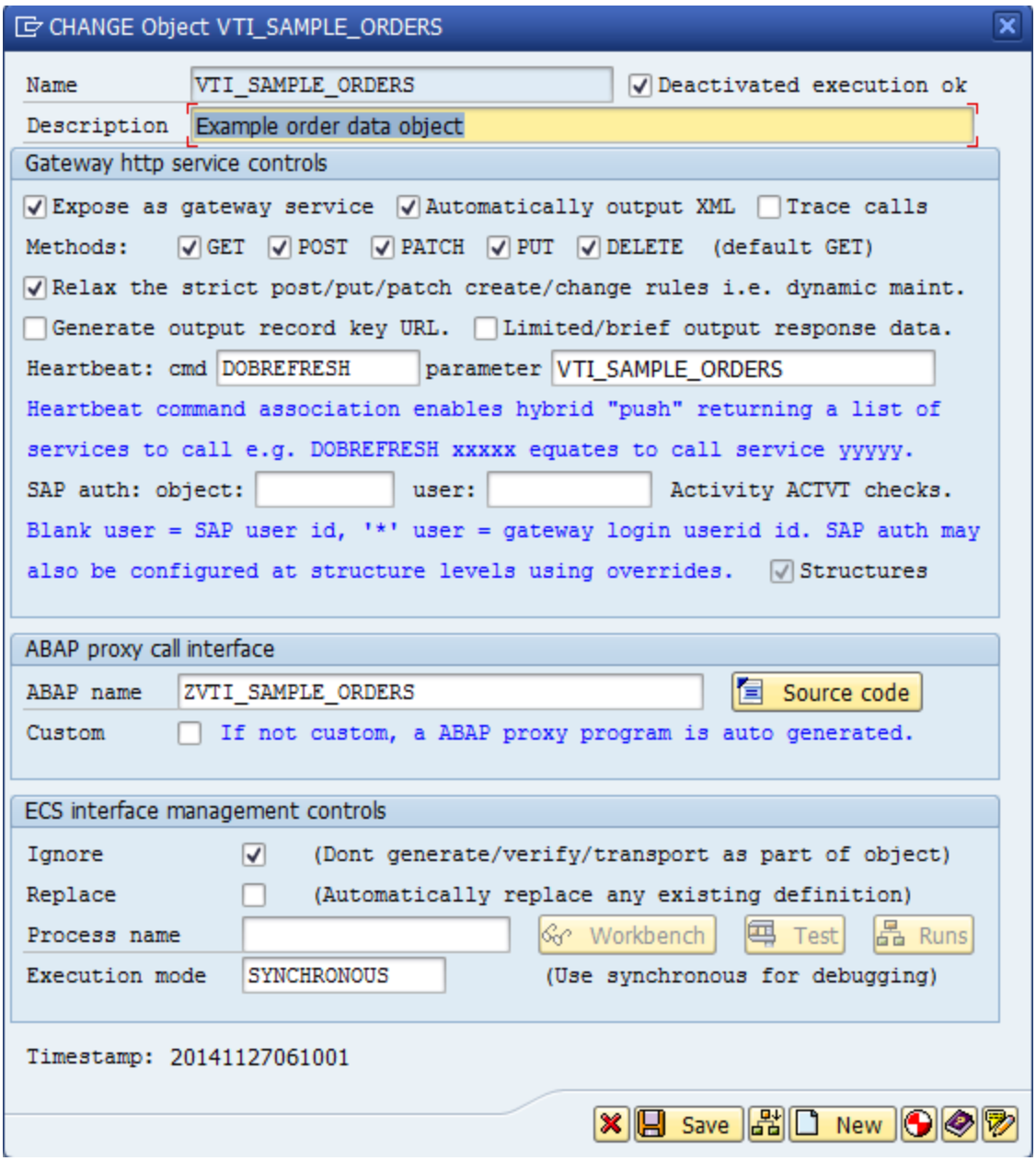
The ABAP proxy and ECS interfacing

Each integration object is associated with an ABAP proxy program that is generated using the integration object definition. The ABAP proxy is what gets executed as the service at the end of the day. In most cases, you do not need to code anything, you may also add your own ABAP “exit” code if required. You may configure the ABAP proxy to either execute on its own, or as an ECS interface for asynchronous, parallel processing and so on. All of this is configured at the integration object level.

Translations, events and actions

The translations, events and actions options are used to support more complex processing scenarios and are not discussed here. Effectively you can branch out from the default main loop processing and handle additional data transformations, ABAP exits and logic using predefined events and actions.

7.3.1.1 The Gateway Service Object Definition

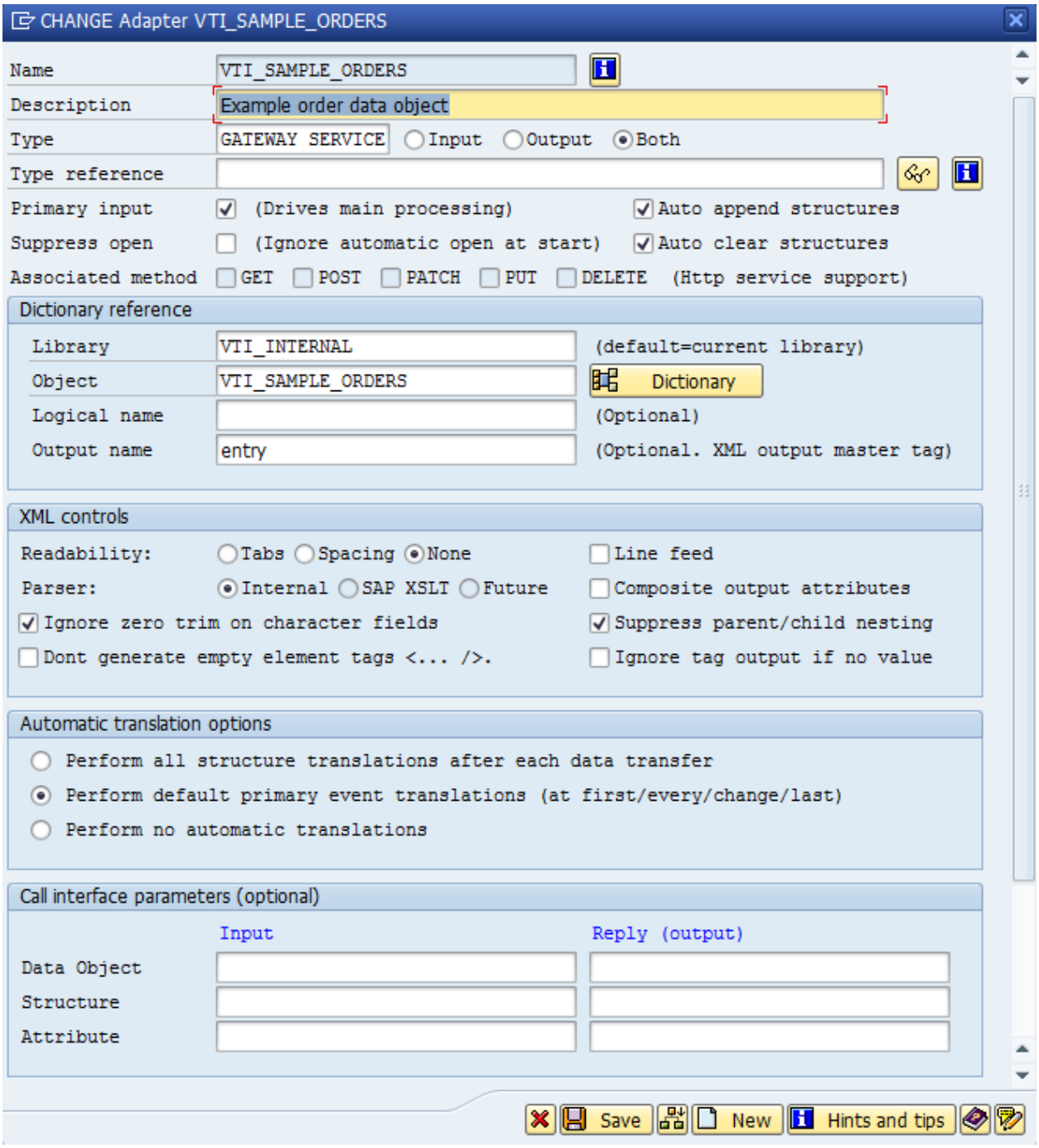


Gateway HTTP service controls	Description
Expose as gateway service	The service is available to external meta data enquiries and HTTP service calls. It also means that the primary adaptor is a gateway service that handles the request/response protocol
Automatic XML output	The XML is generated automatically from the associated dictionary
Trace calls	A gateway service trace is generated for all calls. See the gateway HTTP trace option for more details
Methods	These are the HTTP methods supported by the service
Heartbeat cmd	These configurations act as a cross reference between the heartbeat command facility and the associated service, that is, if a DOBREFRESH command is issued for the caller group and id, then the linked service is returned by the heartbeat service call. In this way, you may implement the push functionality
SAP auth object	Configure an authorization object against the DOB. This applies to the overall DOB service, and any LDBs within it unless an override is defined at the LDB level
User	The user against which the above authorization object needs to be validated
ABAP proxy call interface	Description
ABAP name	The name of the ABAP proxy program that the service call invokes. If not totally custom (the below **Custom** ABAP proxy call interface), this is the program name used for the code generation
Custom	Indicates that the ABAP program that handles the service request is totally customized and no automatic code generation takes place

Gateway HTTP service controls	Description
ECS interface management controls	Description
Ignore	Ignore ECS interface processing for this object. If configured, the following attributes are irrelevant
Replace	Automatically re-generate the ECS process definition when the object is generated
Process name	A unique ECS interface process name to use for the service interface
Execution mode	The ECS execution mode to use, that is, SYNCHRONOUS, SERVER or JOB. This overrides that is specified on the first process phase

7.3.1.2 The Gateway Service Primary Adaptor Definition

If the object is exposed as a gateway service, then it must have a primary adaptor with a type of Gateway Service. This takes care of the HTTP request response handling. The following key attributes apply:

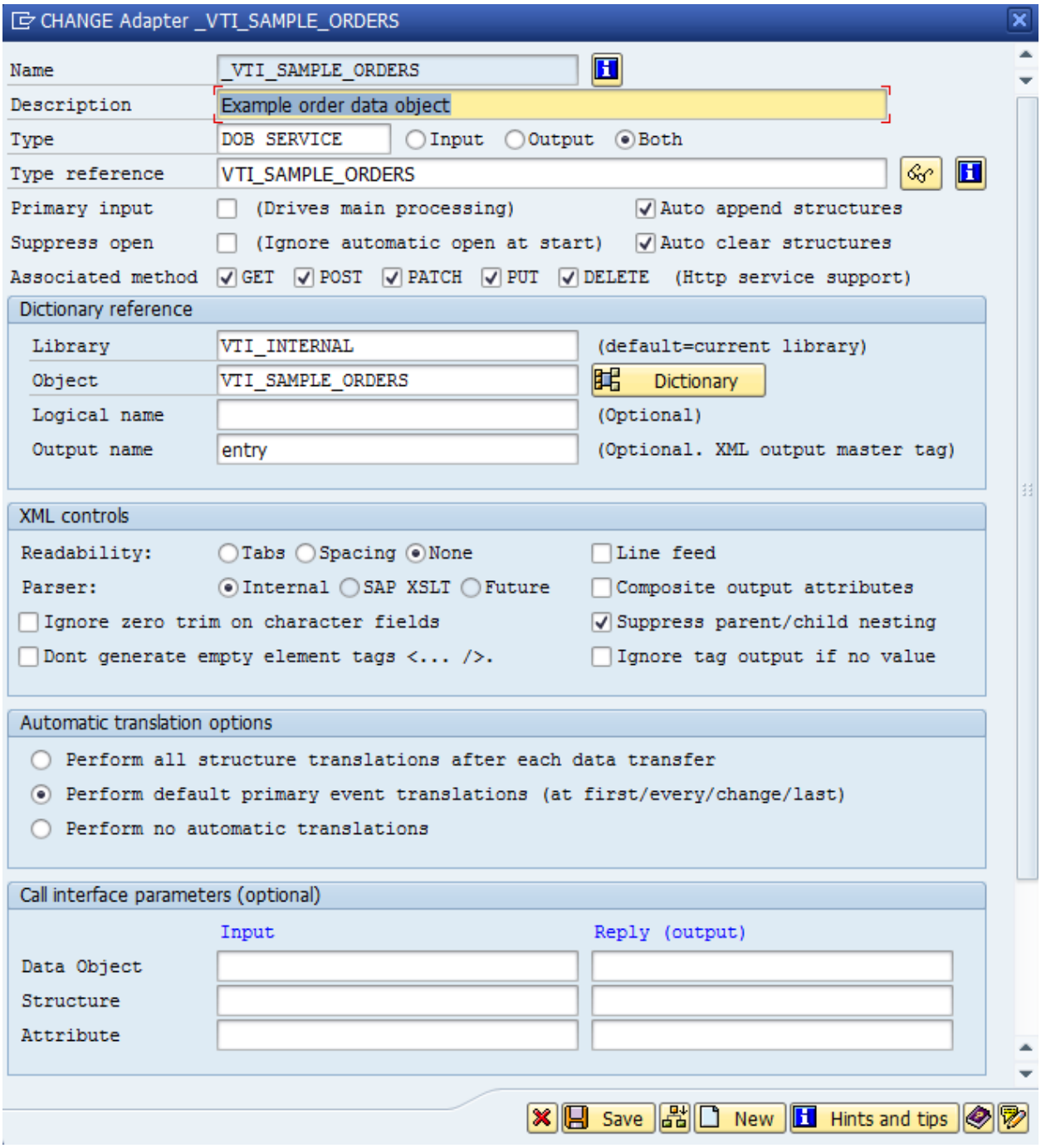


Field	Description
Type	This need to be GATEWAY SERVICE

Field	Description
Input, Output, Both	This indicates whether the adaptor is to process only input or output or both (recommended)
Primary input	This need to configure. It indicates that the input from the gateway service drives the main loop processing.
Auto append /Auto clear structures	Need to configure for gateway services
Dictionary reference	Description
Library	The dictionary library reference. It is recommended that the dictionary library reference is the same as the object library
Object	The name of the dictionary object that is associated with the integration objects input and output operations, for example, DOB or LDB definition
Logical name / Output name	The name of the master tag in XML output. It is recommended that you specify an output name, **entry** or the object name
XML controls	These options help control the behavior of the parser when you generate output XML, such as Line feed, tabs, and Composite tags generation

7.3.1.3 Secondary Gateway Service Adaptor

The primary adaptor handles the HTTP input request and output response. The secondary adaptor is responsible to process the actual HTTP methods, that is, GET, PUT, POST, PATCH, and DELETE. In most cases, this is either a Data Object (DOB) or LDB table level adaptor. The generated ABAP proxy handles the protocol and data conversions automatically.



Field	Description
Input, Output, Both	Configure to **Both**

Field	Description
Primary input	This is not configure, that is, it is a secondary adaptor
Auto append structures / Auto clear structures	Configure so that data structure processing is automatic
Associated method	Use this to indicate which HTTP methods the service is responsible to handle. In case of the DOB and LDB adaptor types, they can handle all. It is conceivable that you may assign different adaptors to service different methods
Dictionary reference	**Description**
Library	The dictionary library reference. You are recommended to have the dictionary library reference same as that of the object library
Object	The name of the dictionary object that is associated with the integration objects input and output operations. For example, DOB or LDB definition. The name of the dictionary object need to be the same as the primary gateway adaptor so that data structures are automatically shared
Logical/output name	Not applicable
XML controls	Not applicable since the primary adaptor inputs and outputs XML

7.3.1.4 Generating and Activating Integration Objects

When you activate an object, you cannot change it. You can still execute the object. An exception to this rule is, if you explicitly configure the **allow deactivated execution** option at the object level.

Similarly, you need to deactivate an object to change it. This is useful during development so that you don't need to continually generate and activate the object to test it. **Generate**, **Activate** and **Deactivate** icons are available on the application toolbar or as easy access hotspot icons on the hierarchy. You position your cursor and click the icon.

When you generate an integration object definition, it constructs an ABAP proxy program, that is, builds all the ABAP code required to execute the adaptor function. In this way, you get the run time performance of program code. To view the ABAP code, click the **Generated proxy (ZVTI_SAMPLE_ORDERS)** hotspot in the hierarchy. The code is entirely readable and the ABAP developers have no problem to understand the code.

7.3.1.5 Working with Text Descriptions for the Integration Object Workbench

The various entities in the Integration Object Dictionary Workbench like library, object, adaptor, structure, and attribute can have extended text descriptions associated with them. This provides additional information for each of the aforementioned entities. The following text maintenance operations are allowed:

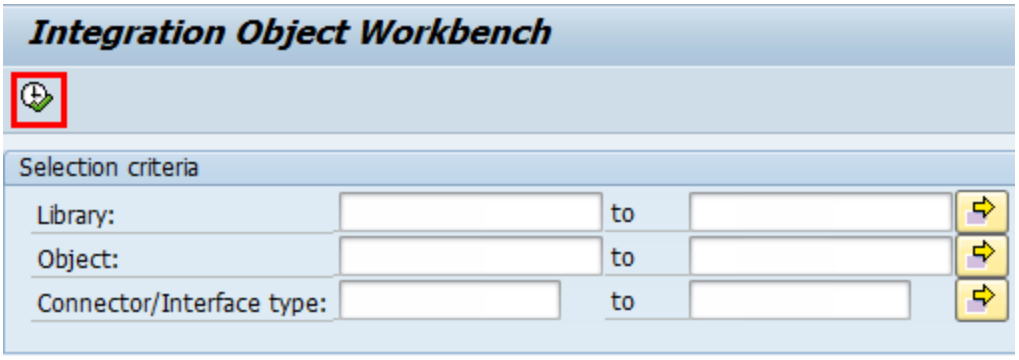
1. Creating a Text Description
2. Editing a Text Description
3. Deleting a Text Description

> ***Note:*** The following examples describe the procedure to maintain text for a Library. You can extend the same procedure to object, adaptor, structure, and attribute in the Integration Object Workbench hierarchy.

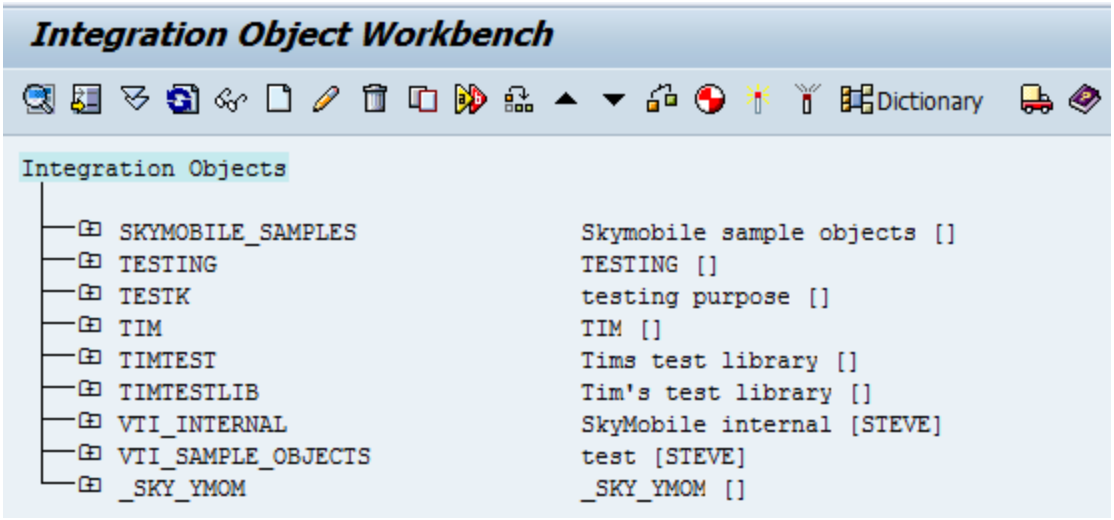
Creating a Text Description

To create an extended text description for a library, follow these steps:

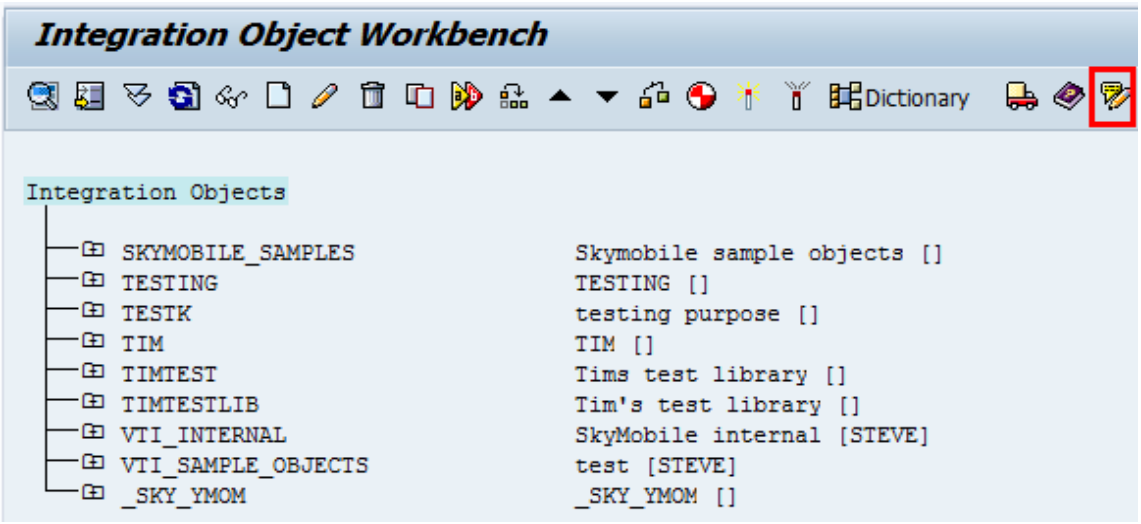
1. Execute the transaction, **YIOW** in SAP system.
   The **Integration Object Workbench** window appears.



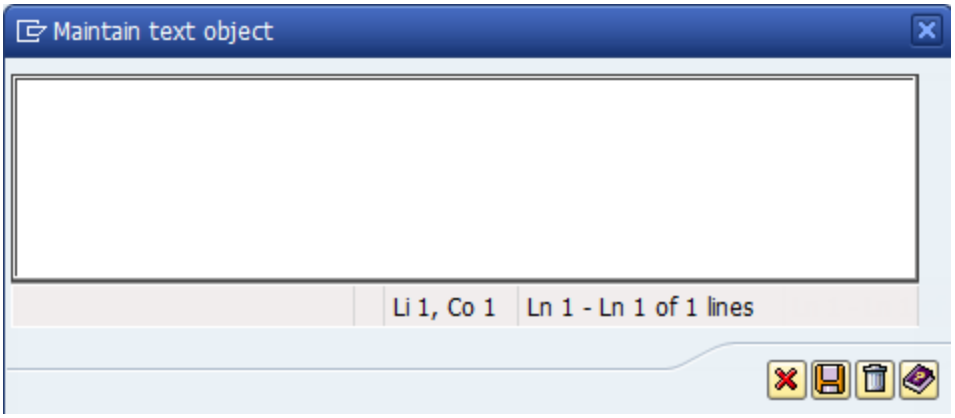
2. Click **Execute**.
   The **Integration Object Workbench** window appears.



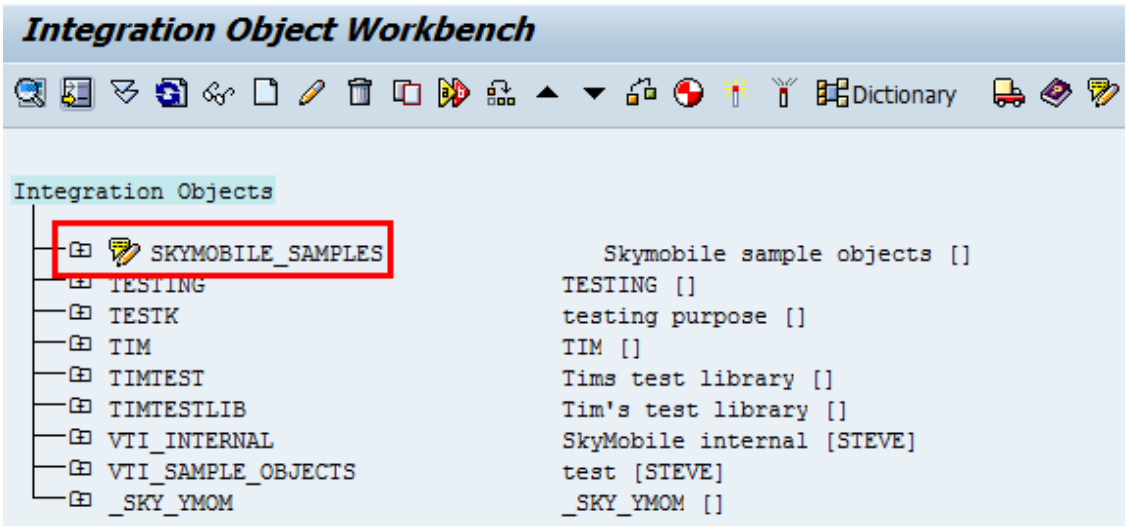
3. To create a text description, select a library from the list and click **Maintain text (Ctrl + Shift + F5)**.



The **Maintain text object** text editor appears.



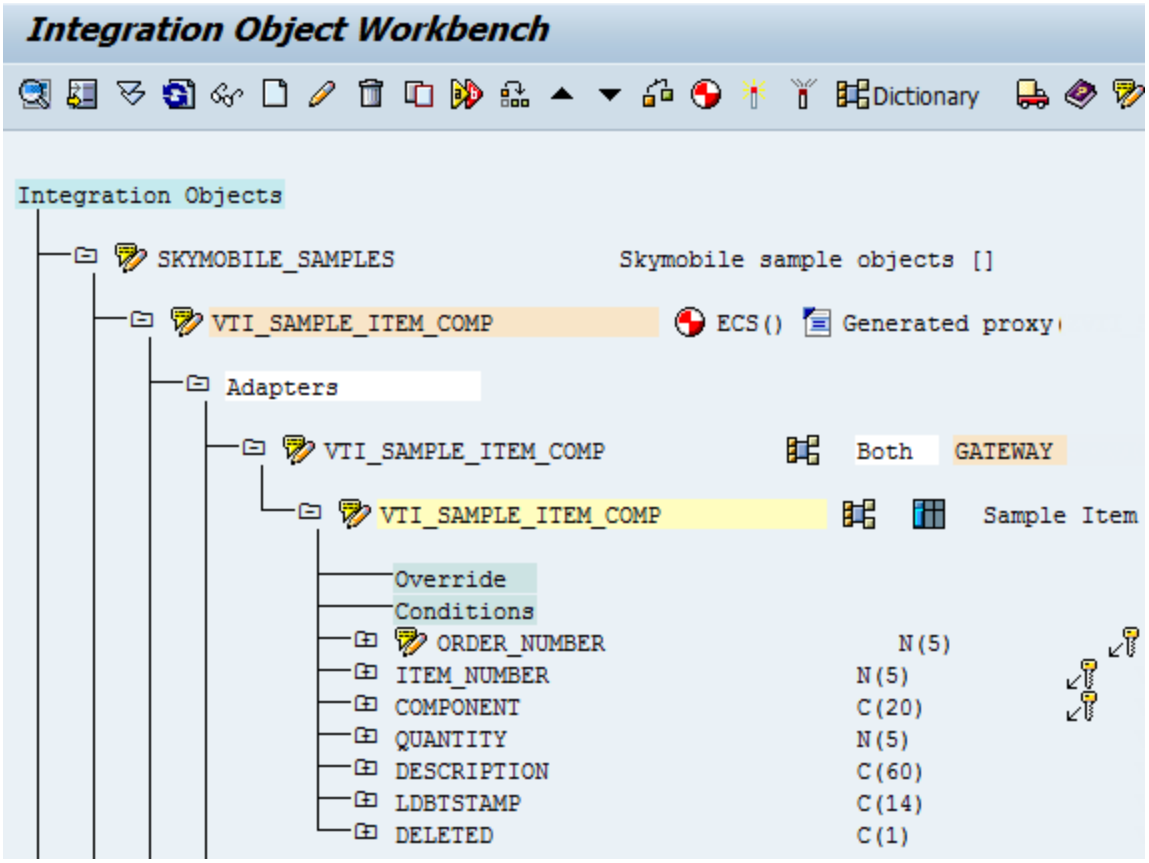
4. Enter text description that you want to add to the library, and click **Save**.
   The **Maintain text** icon appears in front of the selected library name.



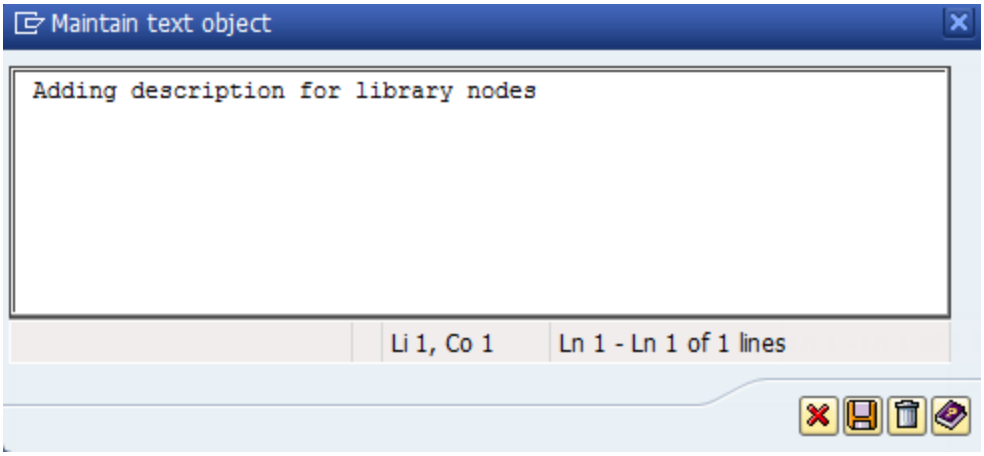
Editing a Text Description

To edit an extended text description for a library, follow these steps:

1. Select a library from the list of library names that have the **Maintain text** icon.
2. Click the **Maintain text** icon.

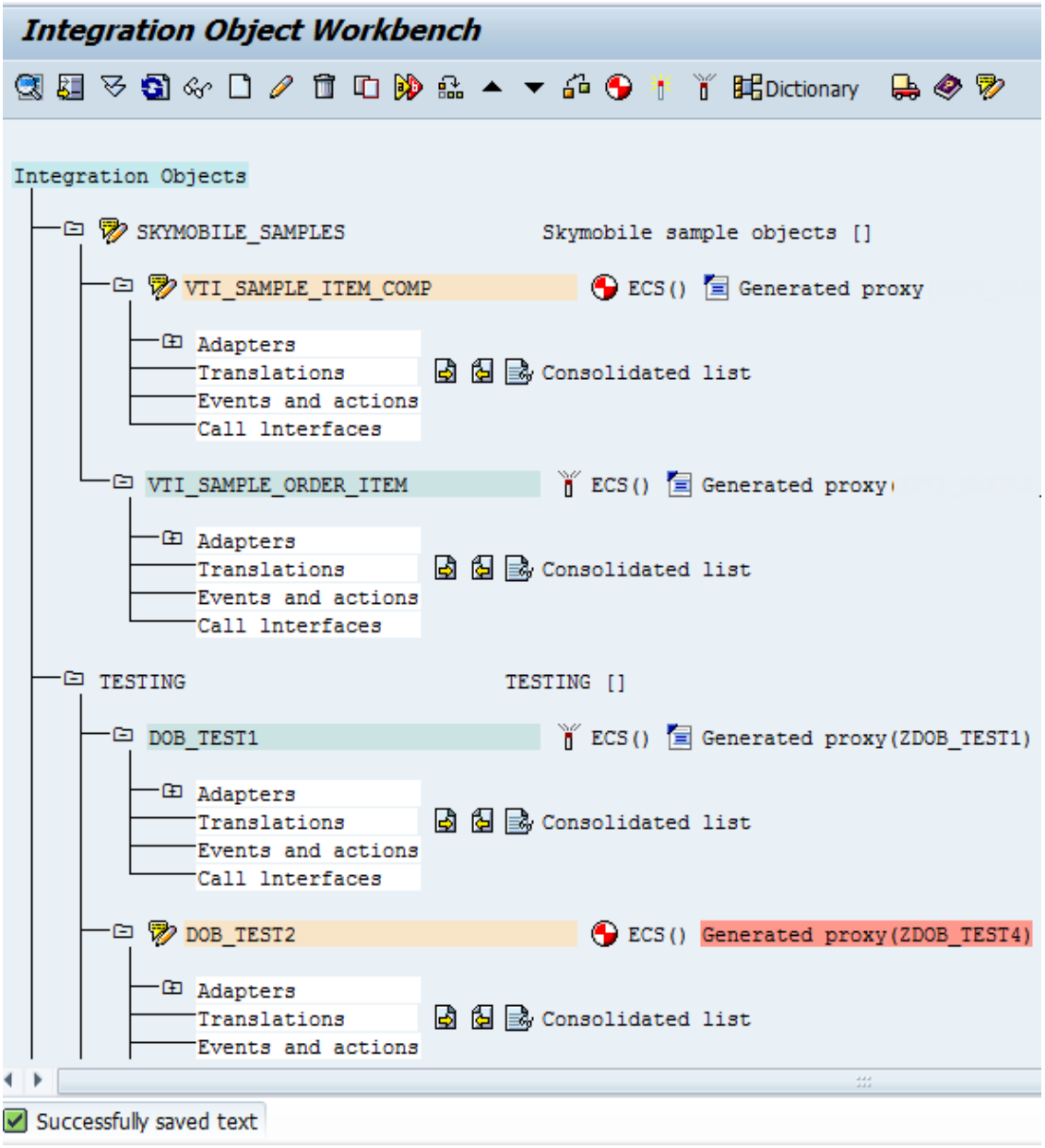


The **Maintain text object** appears with the existing text description.



3. You may change the text description, if required.

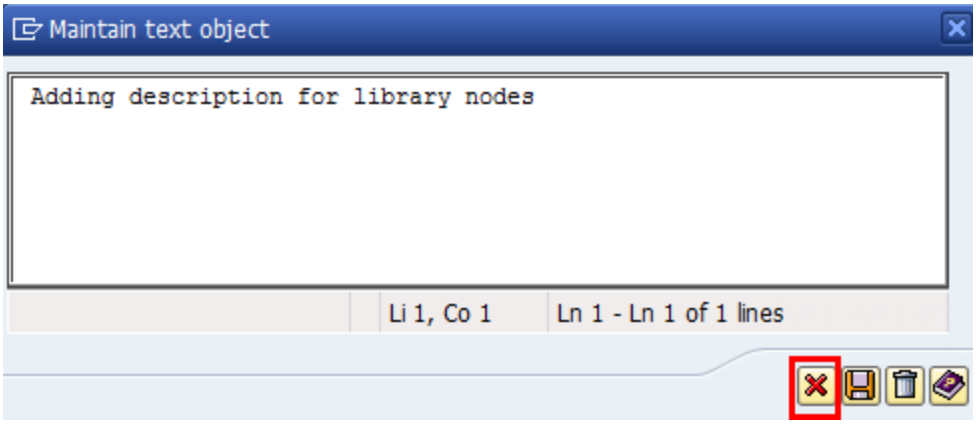
4. Click **Save**.
   The system saves the changes. The message, "Successfully saved text" appears at the bottom of the screen.



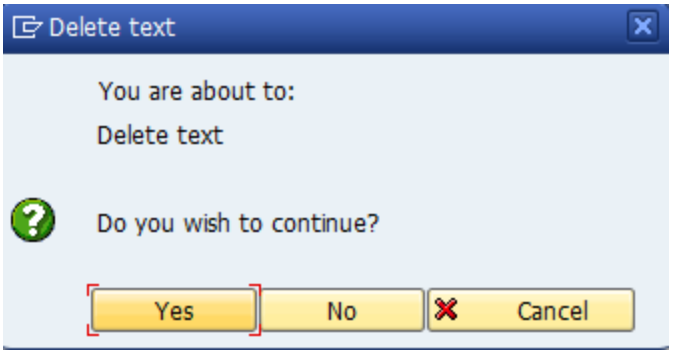
Deleting a Text Description

To delete a text description for a library, follow these steps:

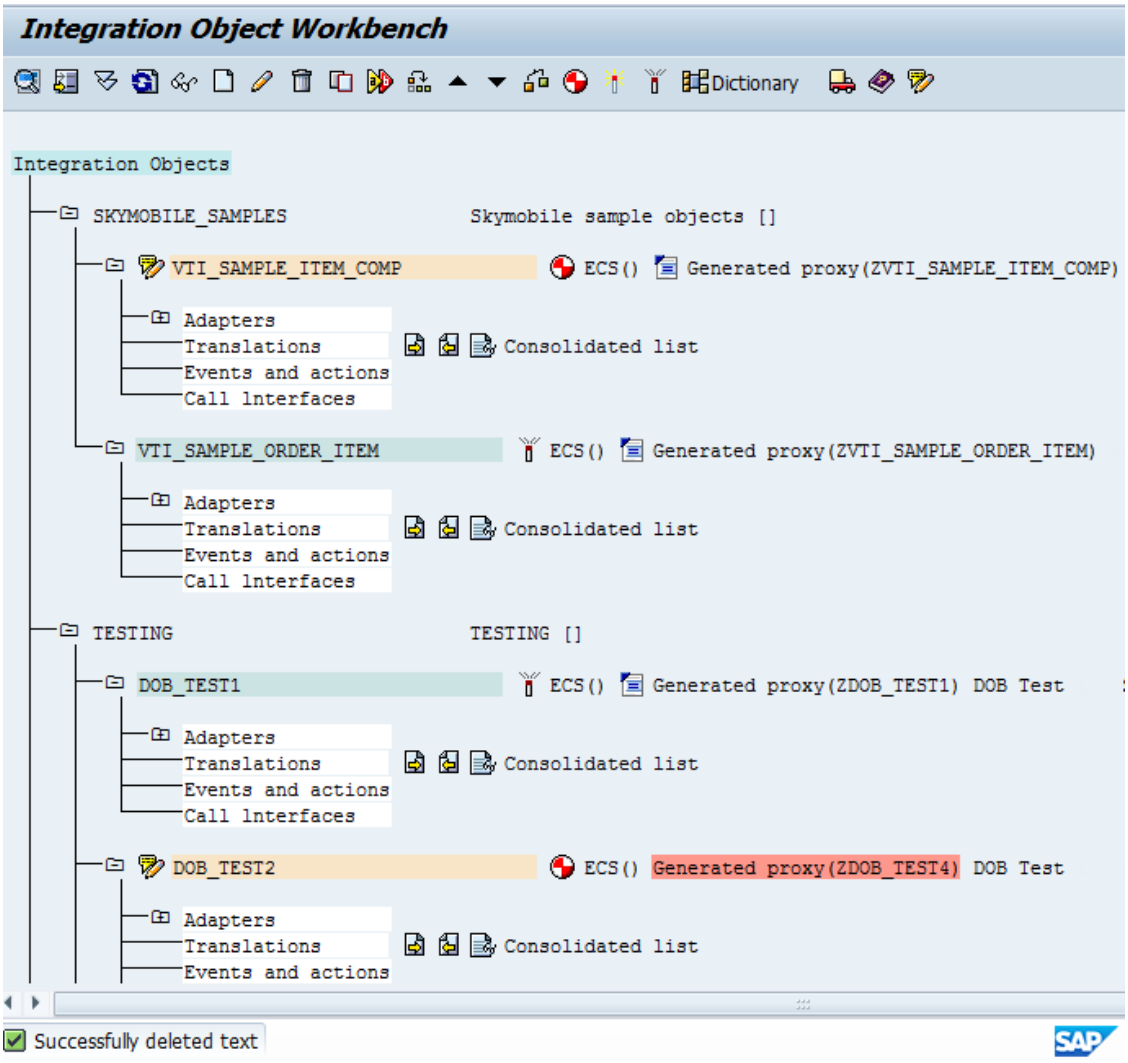
1. Select a Library from the list of library names that have the **Maintain text** icon.
2. Click the **Maintain Text** icon.
   The **Maintain text object** dialog appears with the existing text description



3. Click **Delete (Shift+ F1)** on the **Maintain text object** dialog to delete the text.
   The **Delete text** dialog appears.



4. Click **Yes** to continue.
   The **Maintain text** icon no longer appears next to the library name. The message, “Successfully deleted text" appears at the bottom of the screen.



Working with Text Descriptions for the Dictionary Workbench

The various entities in the Integration Object Dictionary Workbench like library, object, structure, and attribute can have extended text descriptions associated with them. This provides additional information for each of the aforementioned entities. The following text maintenance operations are allowed:

1. Creating a Text Description
2. Editing a Text Description

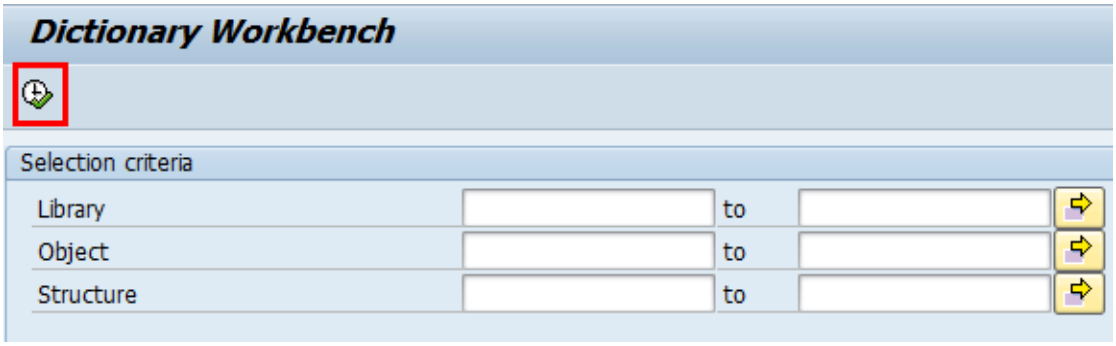
3. Deleting a Text Description

> ***Note:*** The following examples describe the procedure to maintain text for a Library. You can extend the same procedure to object, adaptor, structure, and attribute in the Integration Object Workbench hierarchy.

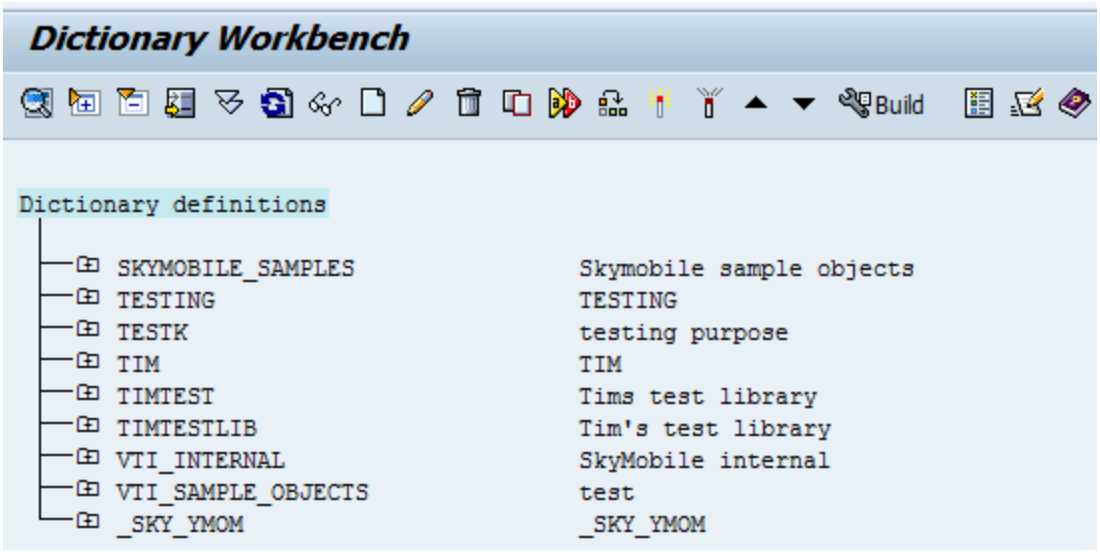
Creating a Text Description

To create an extended text description for a library, follow these steps:

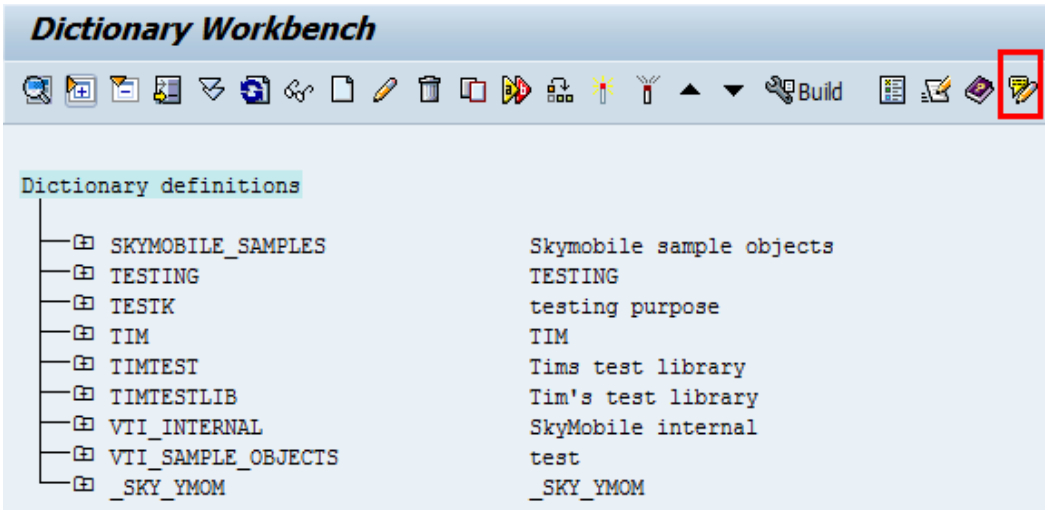
1. Execute the transaction, **YIOD** in SAP system.
   The **Dictionary Workbench** window appears.



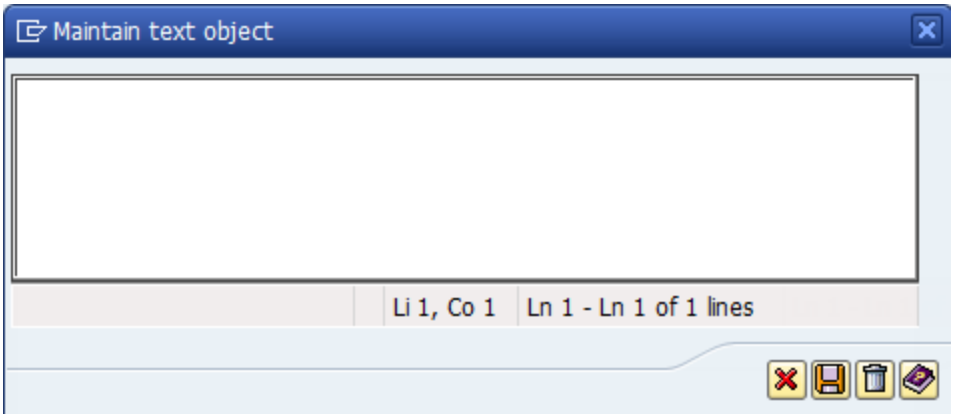
2. Click **Execute**.
   The **Dictionary Workbench** window appears.



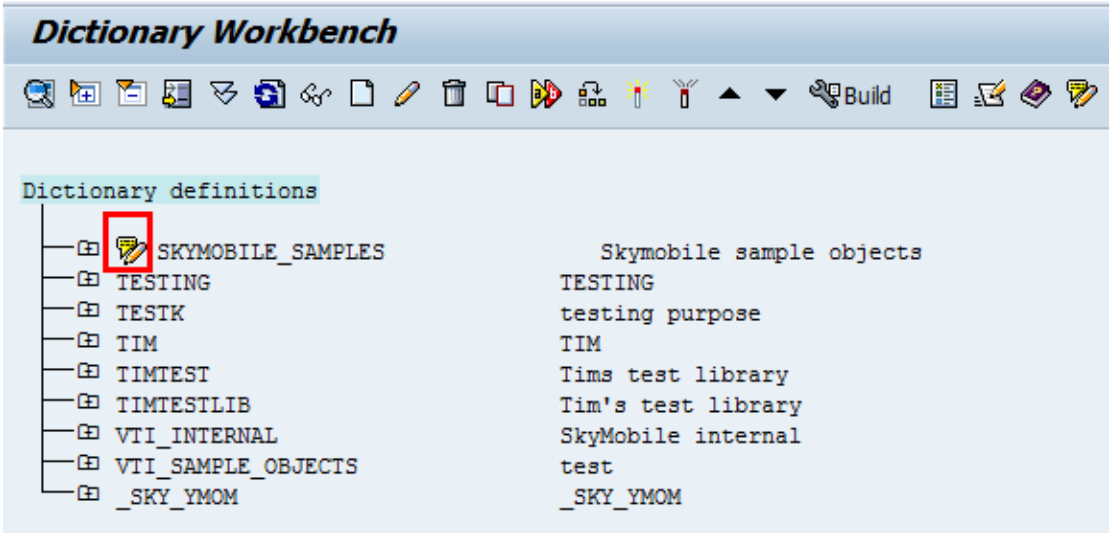
3. To create a text description, select a library from the list and click **Maintain text (Ctrl + Shift + F5)**.



The **Maintain text object** text editor appears.



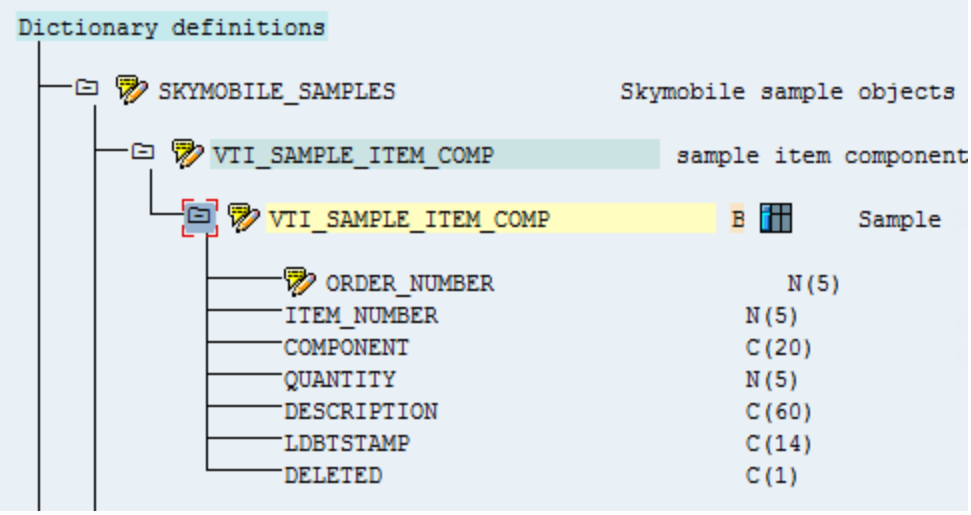
4. Enter text description that you want to add to the library and click **Save**.
   The **Maintain text** icon appears in front of the selected library name.



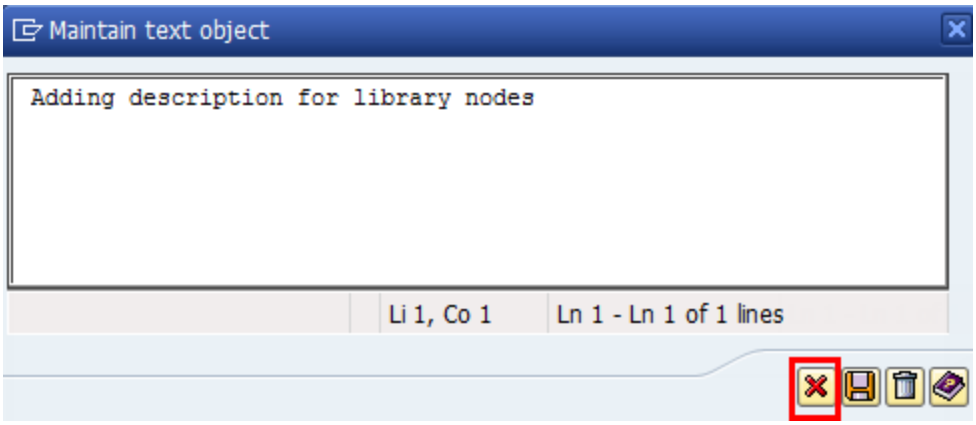
Editing a Text Description

To create an extended text description for a library, follow these steps:

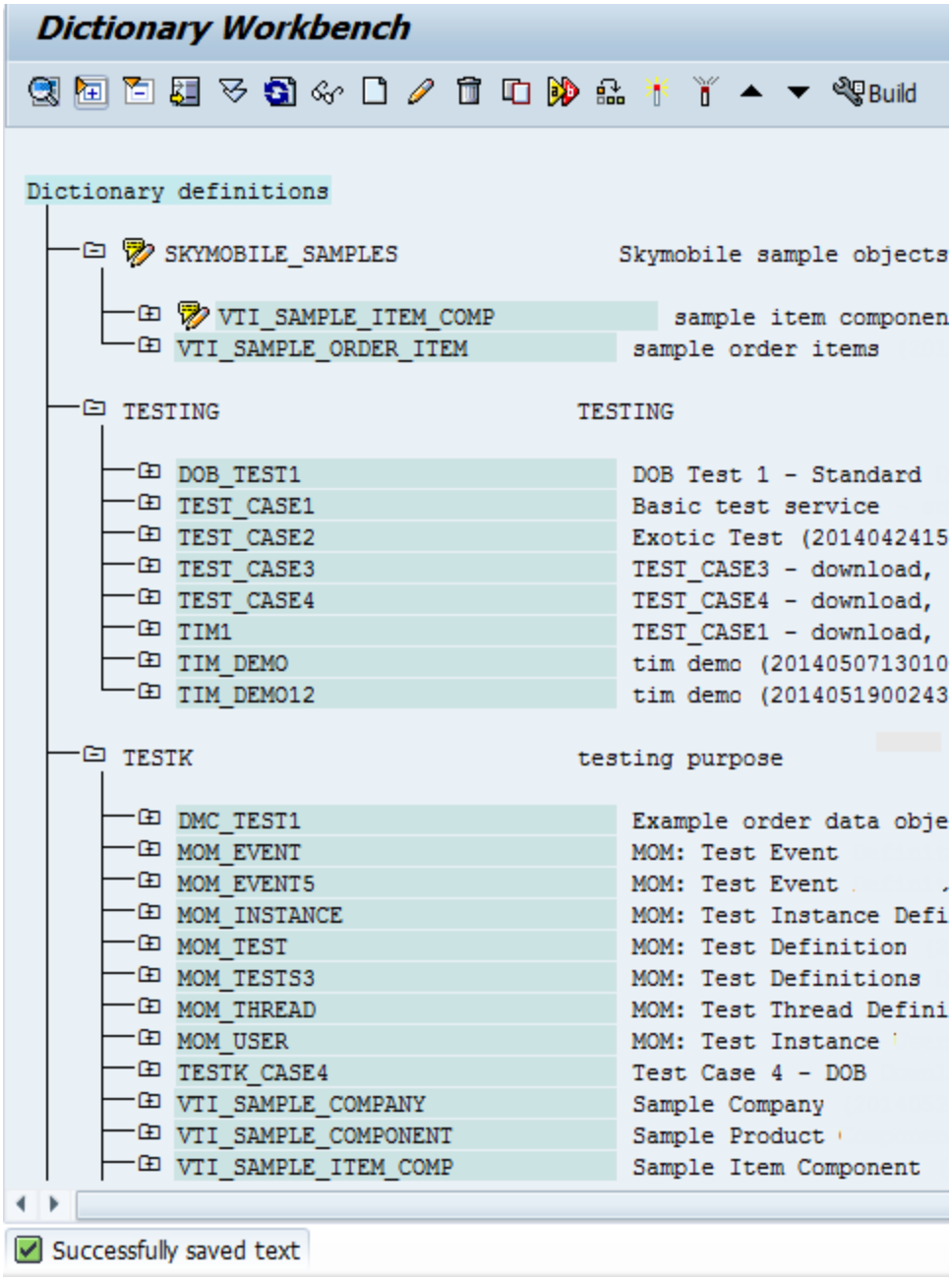
1. Select a library from the list of library names that have the **Maintain text** icon.
2. Click the **Maintain text** icon.



The Maintain text object appears with the existing text description.



3. You may change the text description, if required.
4. Click **Save**.
   The system saves the changes. The message, "Successfully saved text" appears at the bottom of the workbench screen.

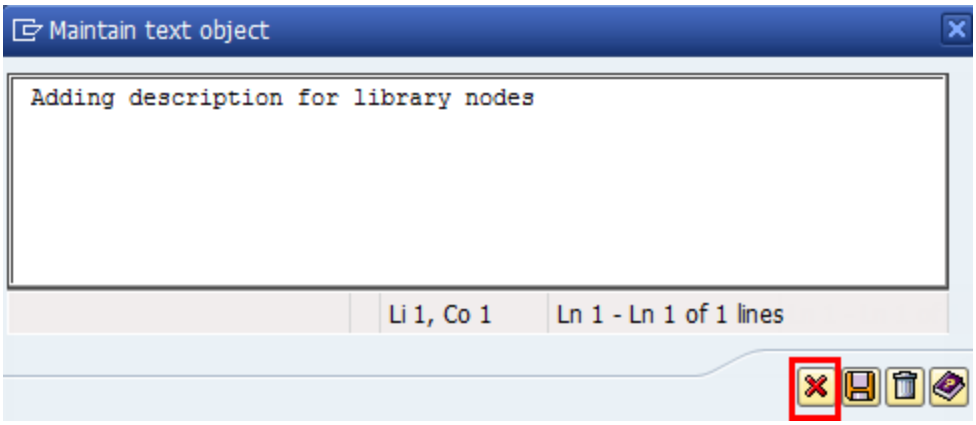


Deleting a Text Description

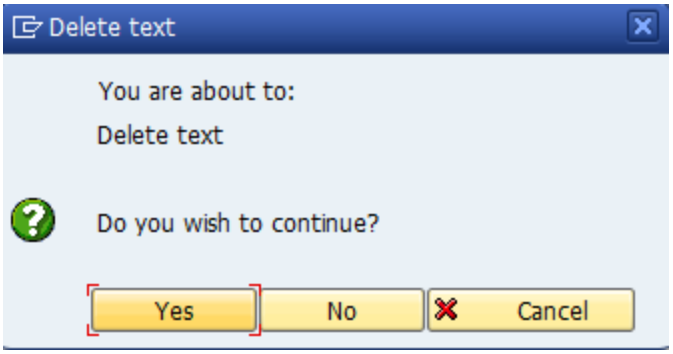
To delete an extended text description for a library, follow these steps:

1. Select a library from the list of library names that have the **Maintain text** icon.

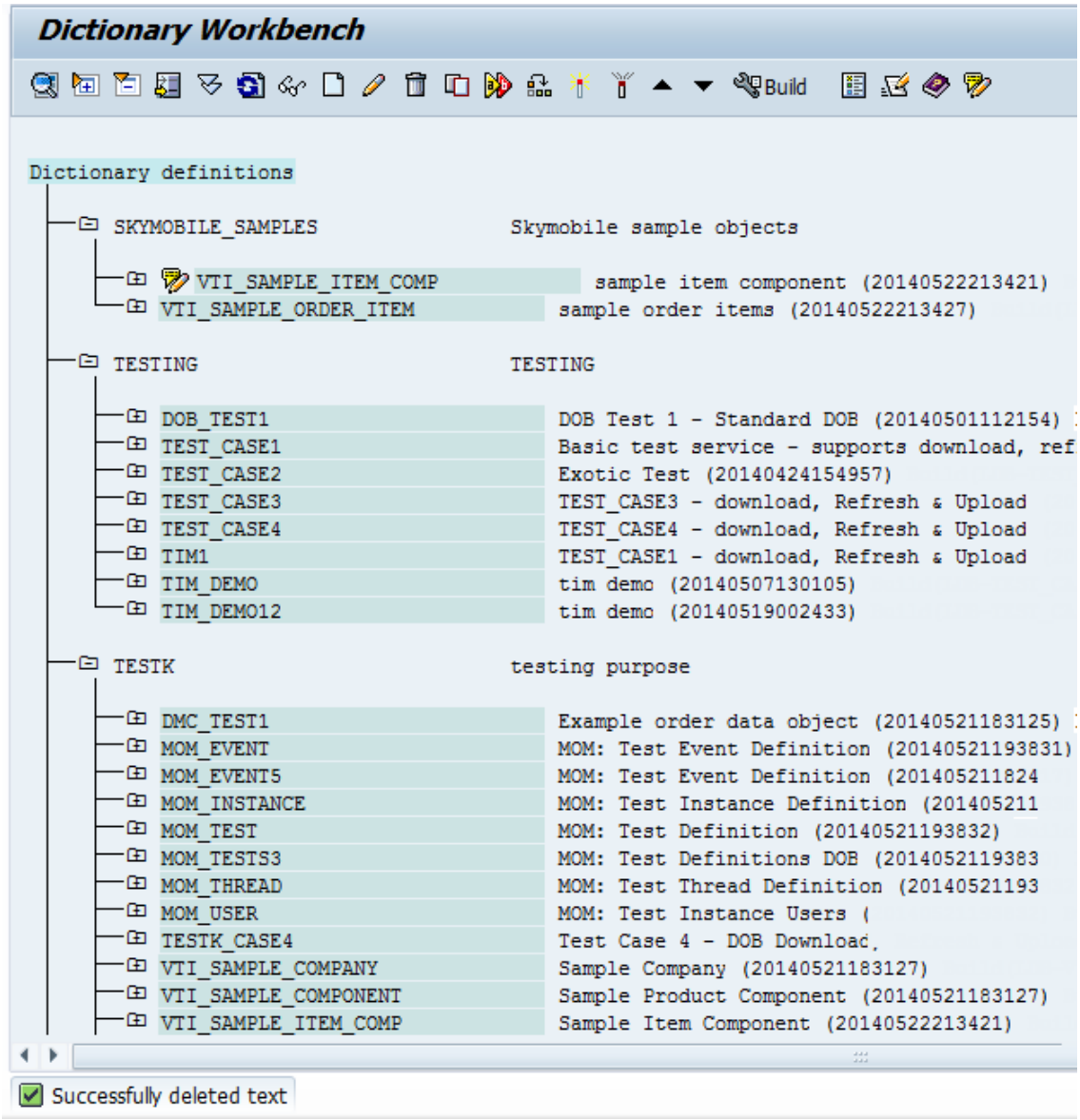
2. Click the **Maintain Text** icon.
   The **Maintain text object** dialog appears with the existing text description.



3. Click **Delete (Shift+ F1)** on the **Maintain text object** dialog to delete the text.
   The **Delete text** dialog appears.



4. Click **Yes** to continue.
   The **Maintain text** icon no longer appears next to the library name. The message, “Successfully deleted text" appears at the bottom of the screen.



7.3.1.6 Testing and Tracing Gateway Services

You can test and view gateway service traces directly from the Integration Object Workbench through the utility menu. You can position your cursor on an object name to default it to the test and trace selection criteria.

7.3.1.7 Developing Custom HTTP Service Modules

You can create your own custom ABAP proxy that interacts with the gateway service request response interface. This allows you to take complete control of the ABAP coding rather than automatically generate the ABAP proxy code using the Integration Object Workbench.

> ***Note:*** You may also implement ABAP exits in the IOB definition to perform specific tasks, rather than write all the logic yourself. The custom program is usually coded as a simple ABAP report and is invoked using submit.

Including the SDK

The SDK abstracts away much of the complex processing from your program. The SDK is implemented as an ABAP include program **/SKY/YGWSSDK** that contains standard macros that you embed in your program. An example custom program, **/SKY/YECSCP07** is provided that performs simple GET processing against the sample order table using dynamic SQL. The first step is to include the SDK at the start of your program through the statement:

```
include /sky/ygwssdk
```

> ***Note:*** The SDK contains forms. You need to have an ABAP event, for example, start-of-selection or end-of-selection to take control of processing. Otherwise, the result is a null execution.

The Request and Respond Call Interface

To receive the request details and respond to the caller, you use the *gws_request* and *gws_respond* commands. These commands import and export standard parameter structures and tables that provide input information to your program. The commands also allow you to publish return code, messages and XML responses back to the caller.

Standard Parameter Areas

Much of the parsing and validation of the request parameters is done for you by the gateway service call interface. As a result, the following table represents the standard parameter areas that are imported when a *gws_request* is executed.

Parameter	Type / Structure	Description
v_gwstrc	/sky/yvti_gwstrc	Request summary information structure
it_gws_tsfff	/sky/yvti_tsfff	$select field list table
it_gws_tsfob	/sky/yvti_tsfob	$sort order by table
it_gws_tsfsv	/sky/yvti_tsfsv	$filters where condition table
it_gws_tsfag	/sky/yvti_tsfag	Arguments table
v_gws_xml	String	XML body

Commands

To make coding easier for simple dynamic SQL requests and XML generation, a suite of common commands is provided in the SDK. You must use the gws_request/response commands, but the others are optional. You may insert any syntax you require.

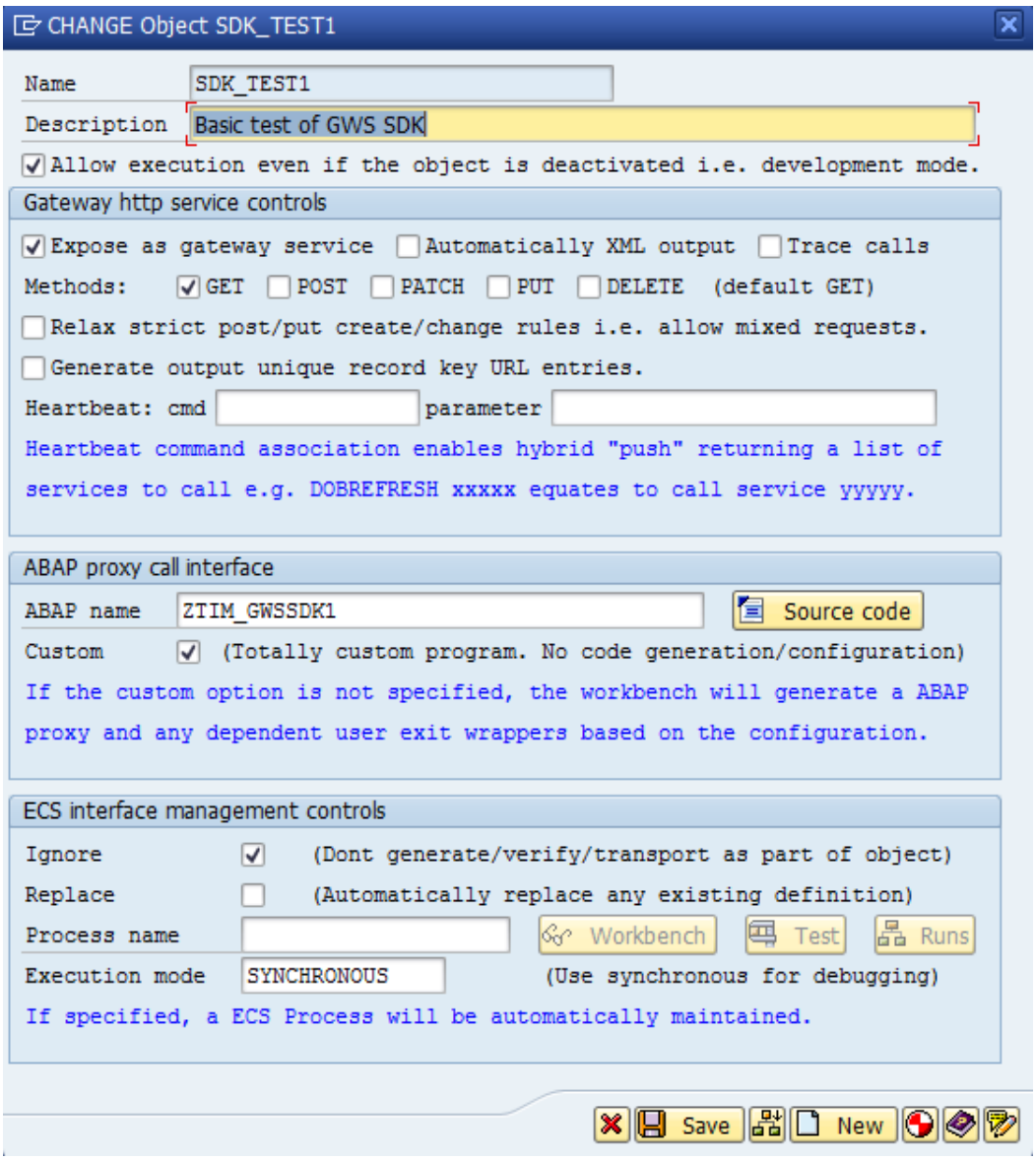
The following table represents a suite of common commands provided in the SDK and their description:

Command	Description
gws_request	Initialize processing and import the gateway service parameter areas.
gws_response	Finalize processing and export the gateway service request header and XML parameters.
gws_message	Configure a return code and message into the request header.
gws_generate_ where	Construct an SQL dynamic where condition from the it_gws_tsfsv $filters selection condition table.

Command	Description
gws_generate_ field_list	Construct an SQL dynamic field list from the it_gws_tsfff field $select field list table.
gws_generate_ order_by	Construct an SQL dynamic order by from the it_gws_tsfob $sort table.
gws_xml_initialise	Initialize the internal XML processor interface.
gws_xml_ structure_tag	Create a high level structure parent tag.
gws_xml_ attribute_tag	Create an attribute tag with a value.
gws_xml_ generate_output	Generate the v_gws_xml output string from the input tag list.

Custom Service Configuration

Once you have coded your custom ABAP proxy, the next step is to configure the custom service definition in the integration object workbench. This is a simple case of creating an object, entering your custom program as the ABAP proxy and flagging it as custom.



Once you have coded your custom ABAP proxy, the next step is to configure the custom service definition in the integration object workbench. This is a simple case of creating an object, entering your custom program as the ABAP proxy and flagging it as custom.

> ***Note:*** You may still use ECS interface management, options if required.

> ***Note:*** All the adaptor and translation options are suppressed and there is no longer any automatic code generation.

Example Code

The following code is based on the example program **/SKY/YECSCP07** that uses dynamic SQL to construct basic XML output.

```
REPORT Z_TEST_GWS.
include /sky/ygwssdk.

start-of-selection.
  gws_request.
  case v_gwstrc-request_method.
       when 'GET'.
            perform f_handle_the_get_method.
       when others.
            gws_message '400' 'Method not supported'.
  endcase.

end-of-selection.
  gws_response.

form f_handle_the_get_method.

data: lit_cnord like /sky/yvti_cnord occurs 0 with header line.

* Default input arguments with no field names e.g. ('25').
  read table it_gws_tsfsv with key field_id = space.
  if sy-subrc = 0.
     it_gws_tsfsv-field_id = 'ORDER_NUMBER'.
     modify it_gws_tsfsv index sy-tabix.
  endif.
```

```
* Dynamic SQL:
  gws_generate_field_list.
  gws_generate_where.
  gws_generate_order_by.

  select (v_sql_field_list)
    into corresponding fields of table lit_cnord
    from /sky/yvti_cnord
   where (it_sql_where)
    order by (v_sql_order_by).

  if sy-subrc ne 0.
     gws_message '404' 'Nothing selected'.
     exit.
  endif.

* Generate basic XML output:
  gws_xml_initialise.
  loop at lit_cnord.
       gws_xml_structure_tag 'Order'.
       gws_xml_attribute_tag 'Order_number' lit_cnord-order_number.
       gws_xml_attribute_tag 'Description'  lit_cnord-description.
  endloop.
  gws_xml_generate_output.
  gws_message '200' 'Success'.

endform.
```

7.4 Managing HTTP Services in Kony for SAP

The Management section provides details about various tools and traces available to manage HTTP Services in Kony for SAP .

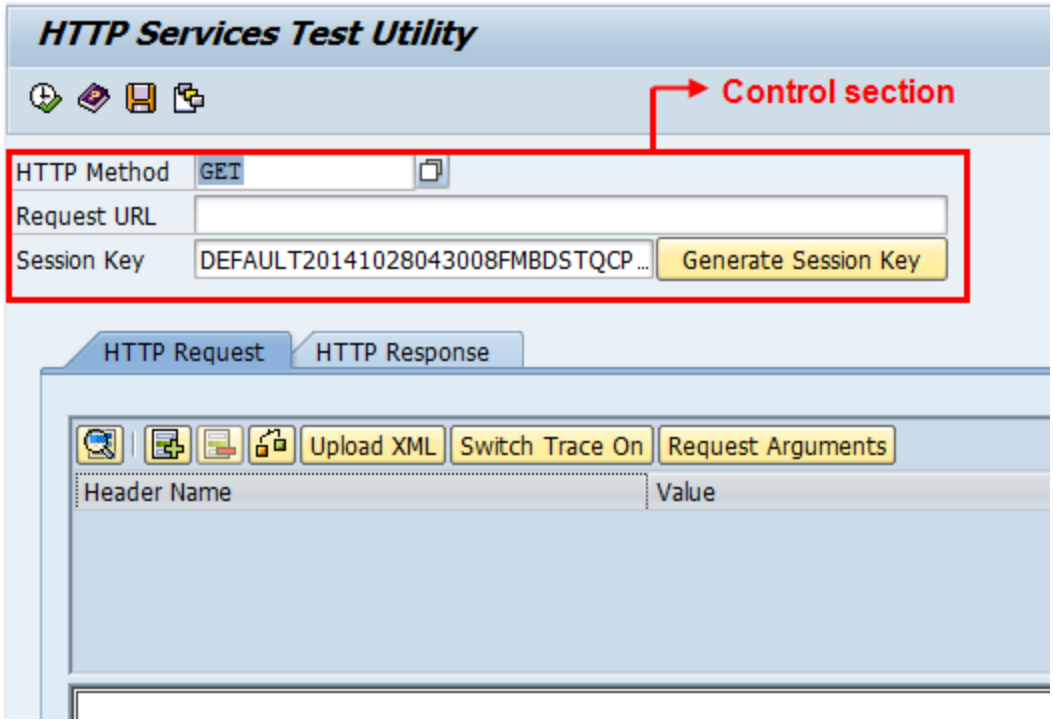
7.4.1 Key Topics

HTTP Services Test Utility
HTTP Services Trace Report

7.4.2 HTTP Services Test Utility

You may test Sky HTTP service calls in SAP using HTTP Services Test Utility. The HTTP Services Test Utility allows you to make HTTP calls within SAP itself, without having to resort to the use of an external client.

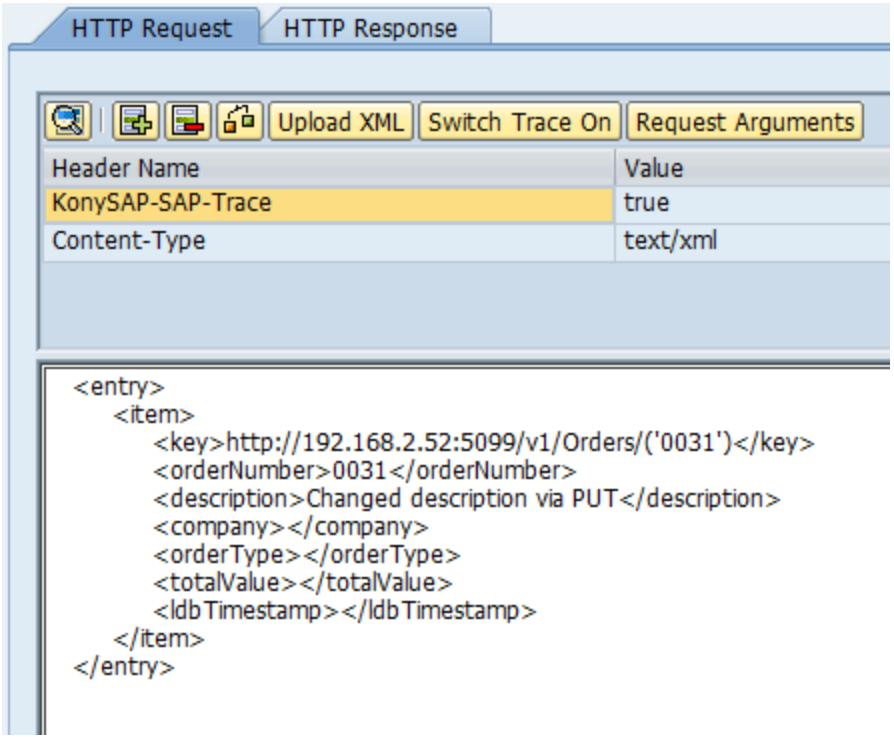
Control section of HTTP Services Test:

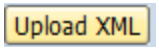
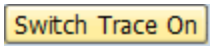
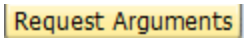


Field	Description
HTTP Method	HTTP method to use for the Service Call. The allowed methods are GET, PUT, POST, PATCH, and DELETE
Request URL	String identifying the resource or service on which the HTTP method is applied.

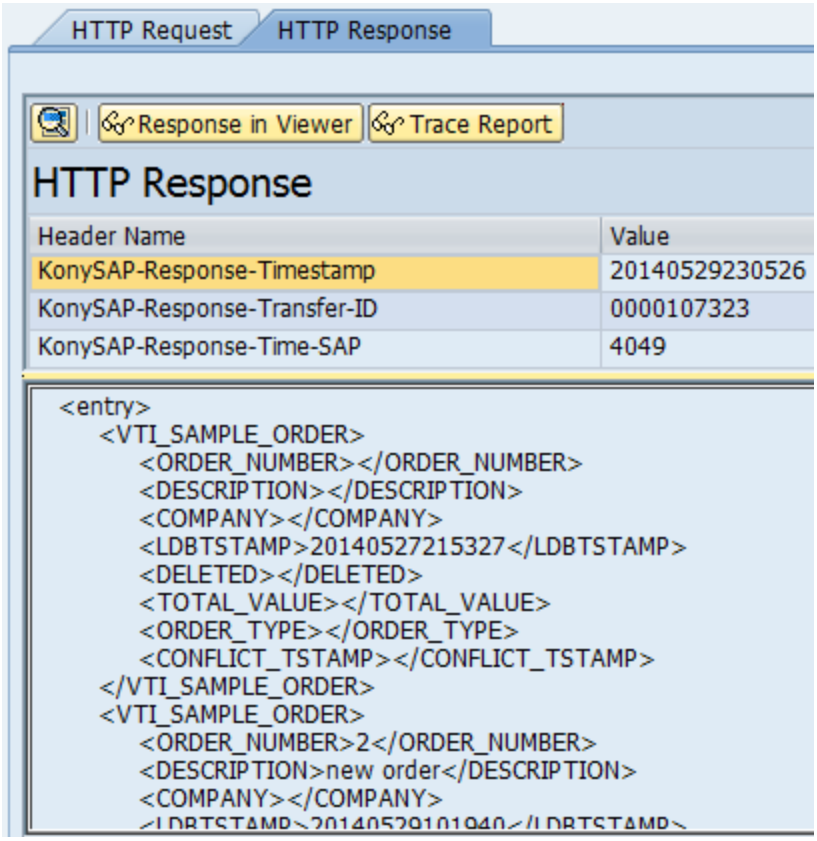
Field	Description
Session Key	A 32 character string identifying a user session. You can use an existing session key or let the system generate one for you. A Session Key is mandatory for all services except for LOGON and PING. A default session key is available to use when the transaction starts.
Generate Session Key	Triggers the process for the generation of a Session Key
	Saves the current Request data into a Test Case
	Loads data from an existing Test Case on to the current request screen

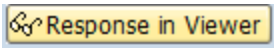
HTTP Request Section of HTTP Services Test:



Field	Description
Header Name	Technical name of a HTTP Request Header parameter
Value	Value of the parameter specified in the HTTP header name
	Displays selected Header line details in a dialog
	Adds a request header name-value pair
	Deletes a request header name-value pair
	Checks if the XML body is well formed
	Imports XML content into the editor from a file
	Configures the flag KonySAP-Request-SAP-Trace in the HTTP Request Header
	Allows to set Query Request arguments, for example, $filters, $top, $skip, $select, and so on
<Request Body Editor>	Text Area to enter XML body. Allows direct input as well as file upload using **Upload XML**

HTTP Response Section of HTTP Services Test:



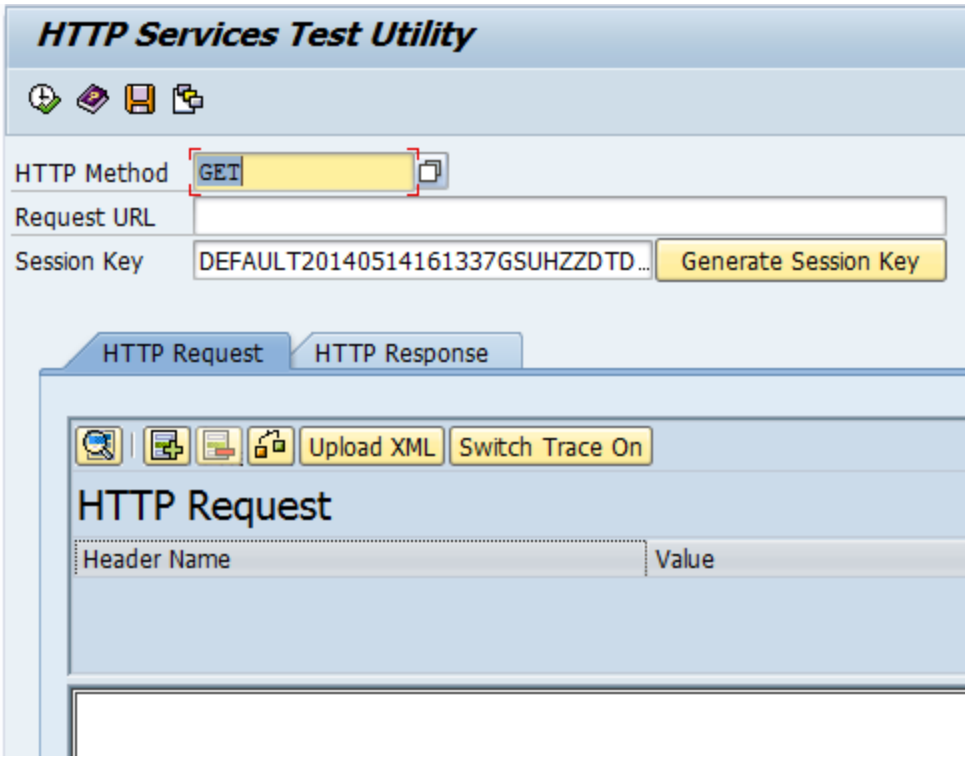
Field	Description
Header Name	Technical name of a HTTP Response Header parameter.
Value	Value of the parameter specified in the HTTP Header Name
<Response Body Viewer>	Read only Text Area that displays the XML Response body.
	Displays selected Header line details in a dialog window
	Displays the XML body appearing the Text Area in a standard HTTP Viewer

Field	Description
Trace Report	Transfers control to the HTTP Trace Report with the selection-screen pre-filled with the Transfer ID.
Status Code <code>:<message>	Displays the status code and the accompanying message from the HTTP Service response

7.4.2.1 Working with HTTP Services Test Utility

To work with the HTTP Services Test utility, follow these steps:

1. Execute the transaction **/SKY/YVT8 (YVT8)** in SAP system.
   The **HTTP Services Test Utility** window appears.



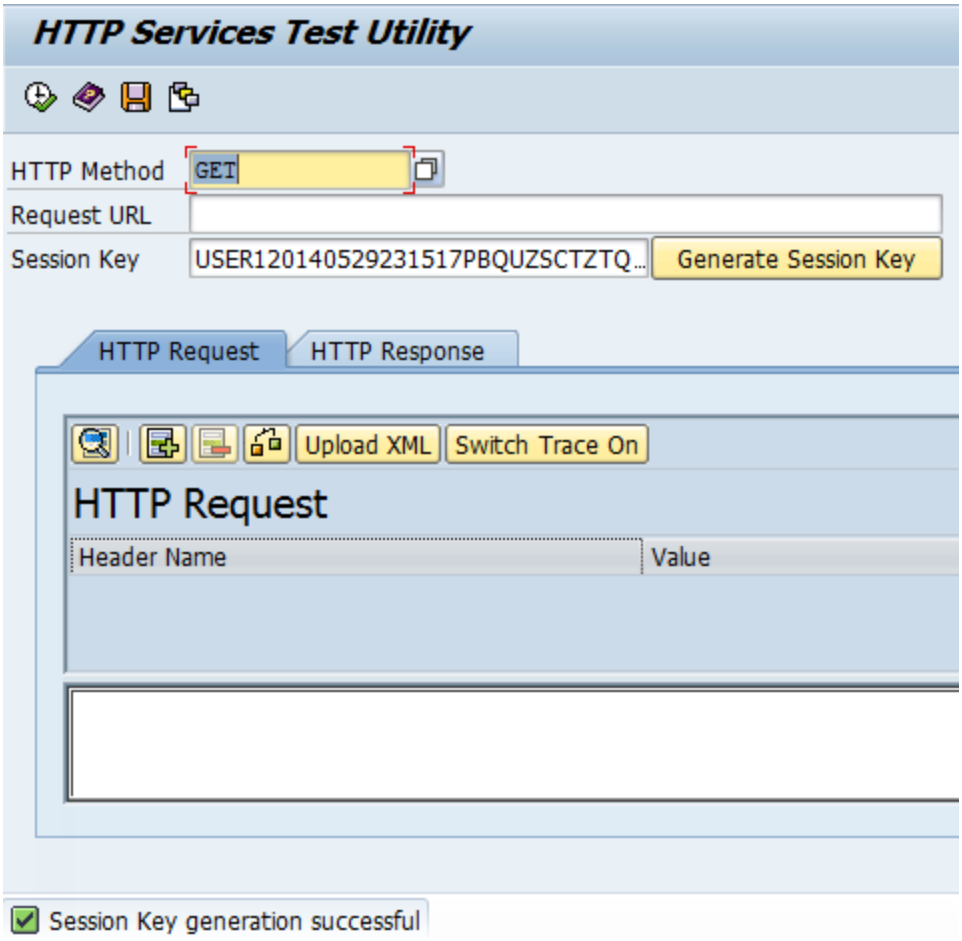
For every Service call, except for LOGON and PING, it is necessary to provide a Session Key that identifies a specific *Caller*. When you make a service call from the Test Utility, you can provide a valid Session Key or the system can generate one for the user. A default session key is available to use when the transaction is starts.

- *Generating a Session Key*:
  To generate a new session key, click **Generate Session Key** on the **Control Section**.
  The **Generate Session Key** dialog appears.

  Enter the relevant values in the following fields:

Field	Description
User ID	Refers to the User ID against which the service is called
Caller ID	Identifier for the client or device that invokes the Service
Caller Group	Group for related Caller IDs. The value defaults to the system ID, if you specify nothing

  A 32 character string is generated and populated for the Session Key. You can then use the session key to invoke the HTTP service. A message,"Session key generated successful" appears at the bottom of the window.



- *Invoking a simple HTTP Service:*
  To invoke a simple HTTP service, follow these steps:

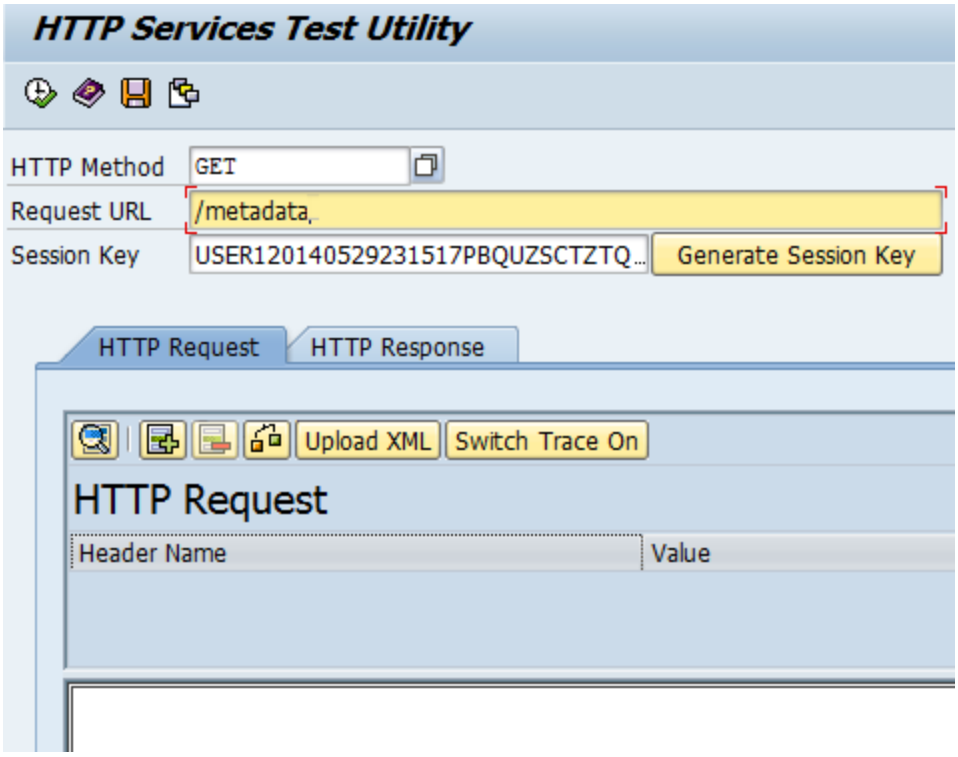
  a. The essential inputs for a basic HTTP Service call are the HTTP Method, Request URL and Session Key.
     The below example invokes the METADATA service:
     Enter the values above for HTTP Method and Request URL.

  b. Generate a session key.

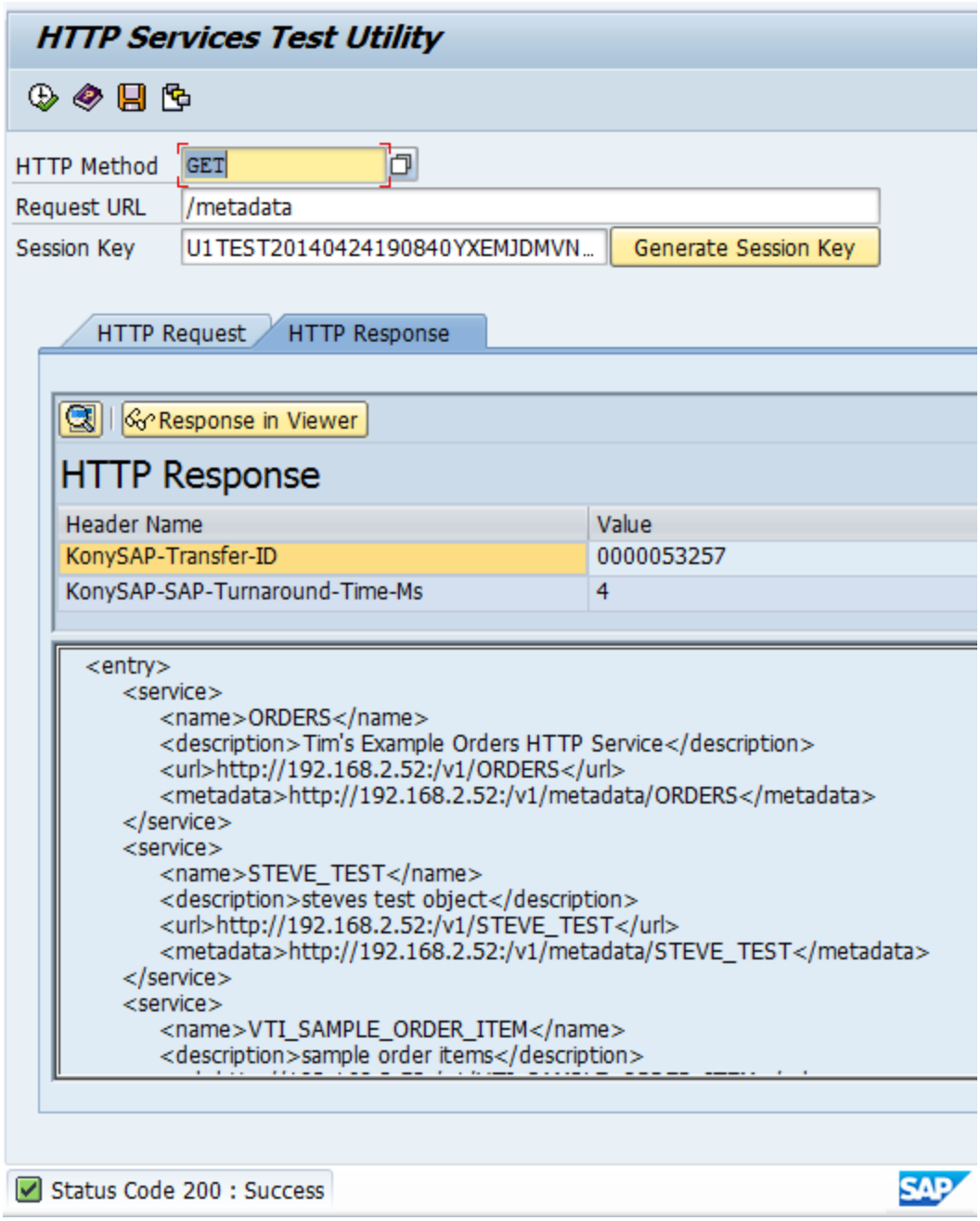
  c. Click **Execute**.
     The following are the acceptable formats of the Request URL:

a. http://host:port/version/resource
b. http://host:port/resource
c. http://host/version/resource
d. http://host/resource
e. http://host/version/resource
f. /version/resource
g. /resource

From the URL provided, the name of the host, port, version and resource (service) are extracted. If host and the port are not provided, they are defaulted to the current system host and the designated port.
Version is optional.
The resource and the method are passed to the corresponding transformation routine in the Integration Object workbench, where the actual processing takes place.

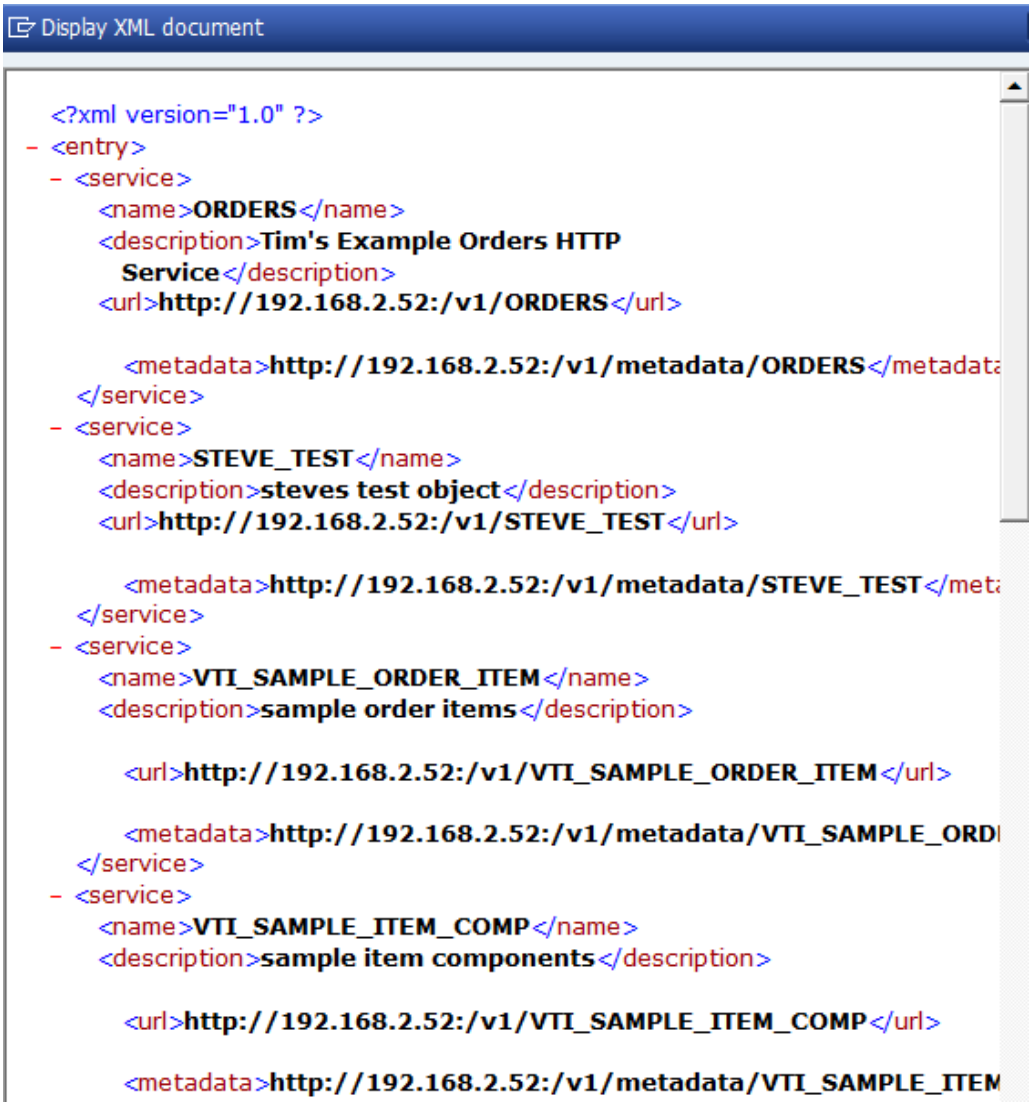


The **HTTP Response** tab appears with the status of the service call.



A response from an active service always contains a Transfer ID and the SAP Turnaround time. The XML body is formatted in a plain text editor.

d. To view the XML body in the HTTP viewer control, click **Response in Viewer**. The **Display XML document** window appears.

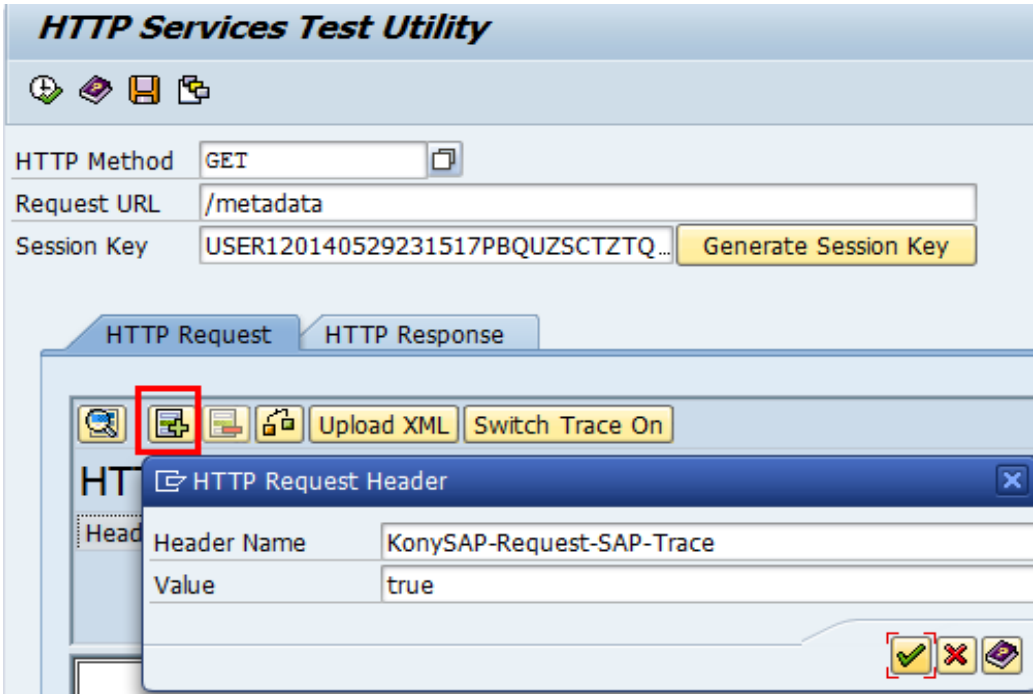


- *Invoking a HTTP service with a Request Header*

  To add a request header to a HTTP service, follow these steps:

  The following example sets the parameter *KonySAP-Request-SAP-Trace* with the value *true*.

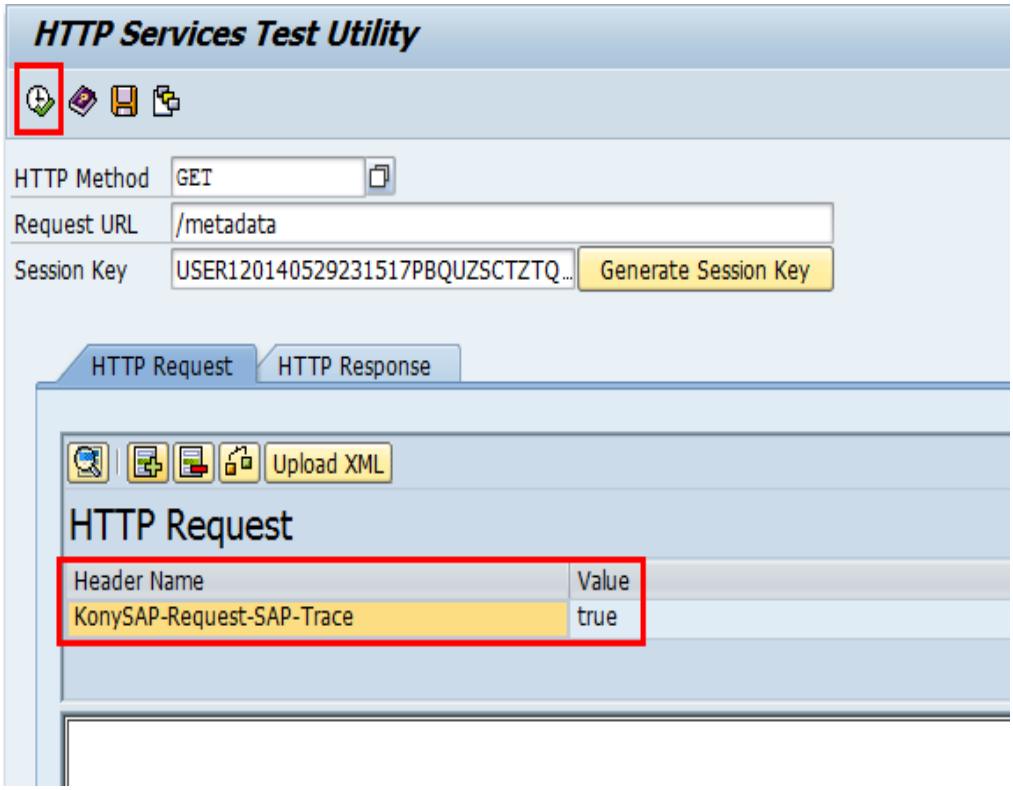
  a. After populating the basic fields, click **Add Header** on the **Request** section and enter the values as below.



The **HTTP Request Header** dialog appears.

b. Enter the values as shown in the above screen shot.

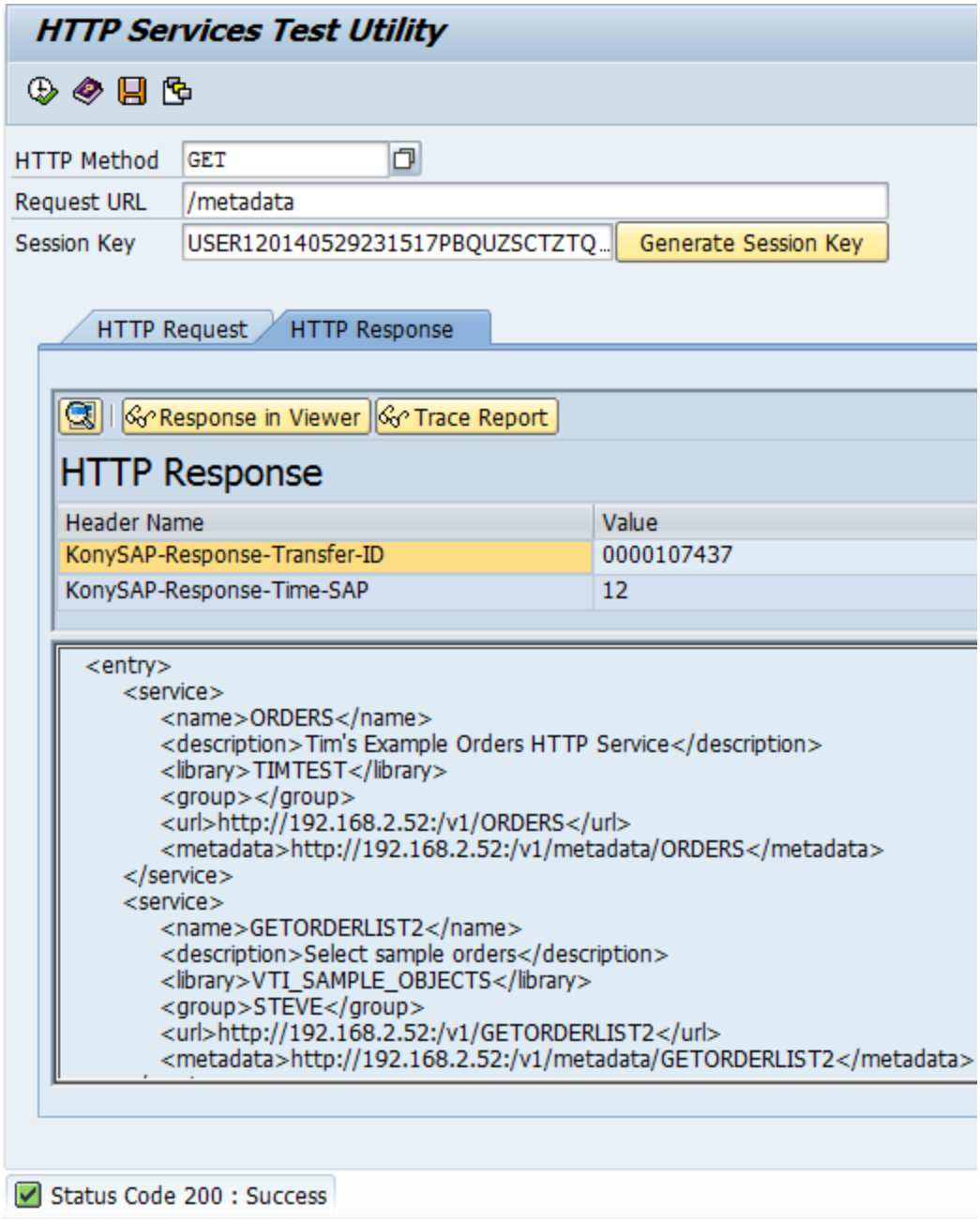
The **Header Name** field also has a search help, that allows you to choose the currently supported parameters. The parameter used in this example makes the system trace the service call for the complete round trip. You can examine the generated trace through the Trace Report (Transaction: **/SKY/YVT7 (YVT7)**).



The header name and the value you entered appear under the **HTTP Request** section. Once the header name-value appears on the screen, you may execute the request.

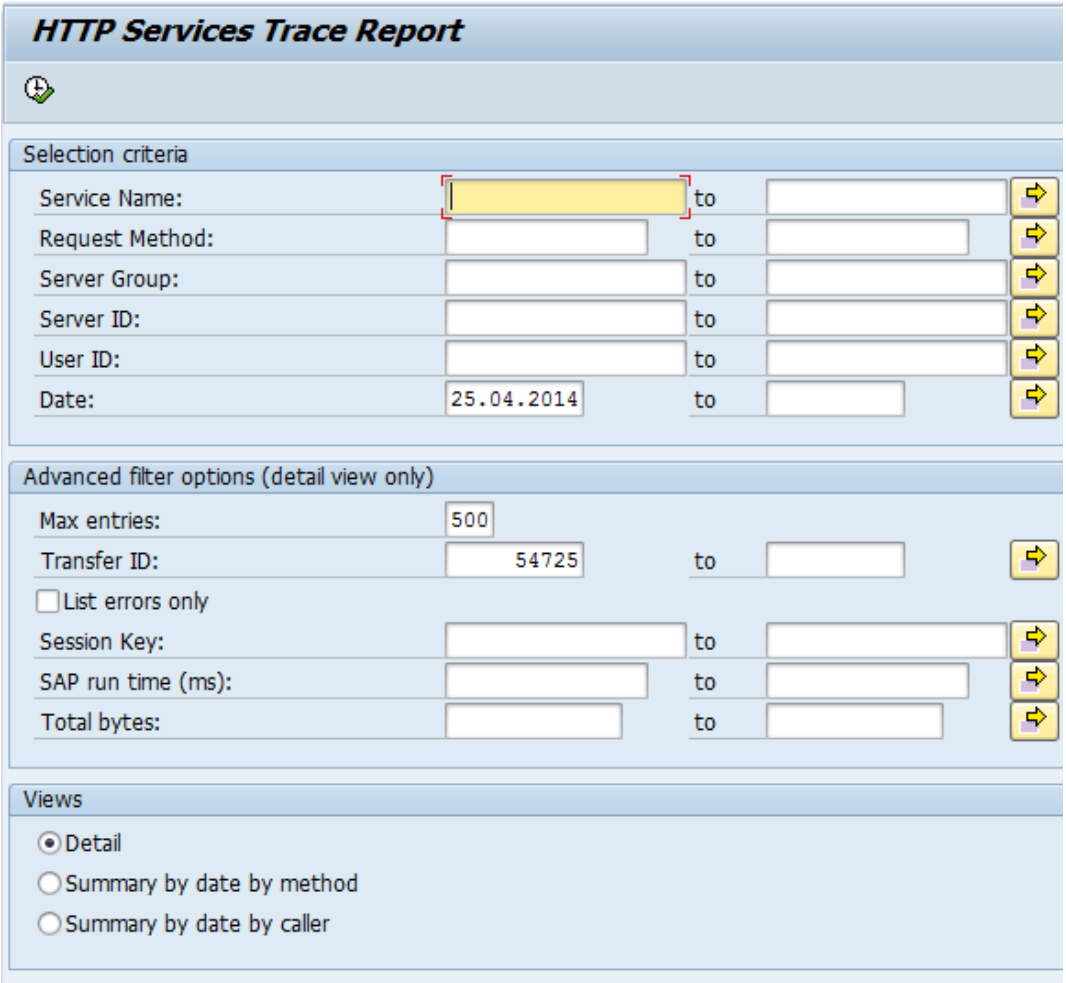
> ***Note:*** You can also set the header parameter KonySAP-Request-SAP-Trace by clicking the **Switch Trace On** button.

c. Click the **HTTP Response** tab to view response of the service request.



d. You can trace the current invoking of the Service call. It is possible to view the generated trace entry.
On the **Response** tab, you may directly navigate to the Trace Report by clicking **Trace Report**. The **HTTP Services Trace Report** appears with the pre-filled **Transfer ID** for the

current request on the **Advanced filter options (detail view only)** section.

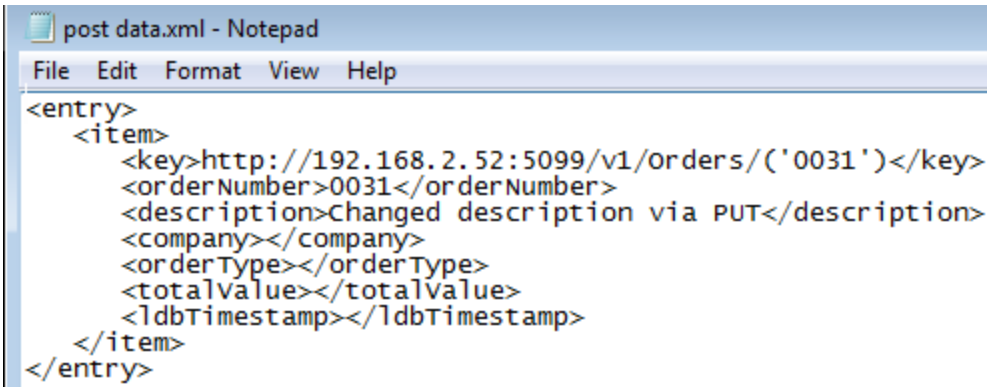


For more details on how to navigate through the Trace Report, refer HTTP Service Trace Report.

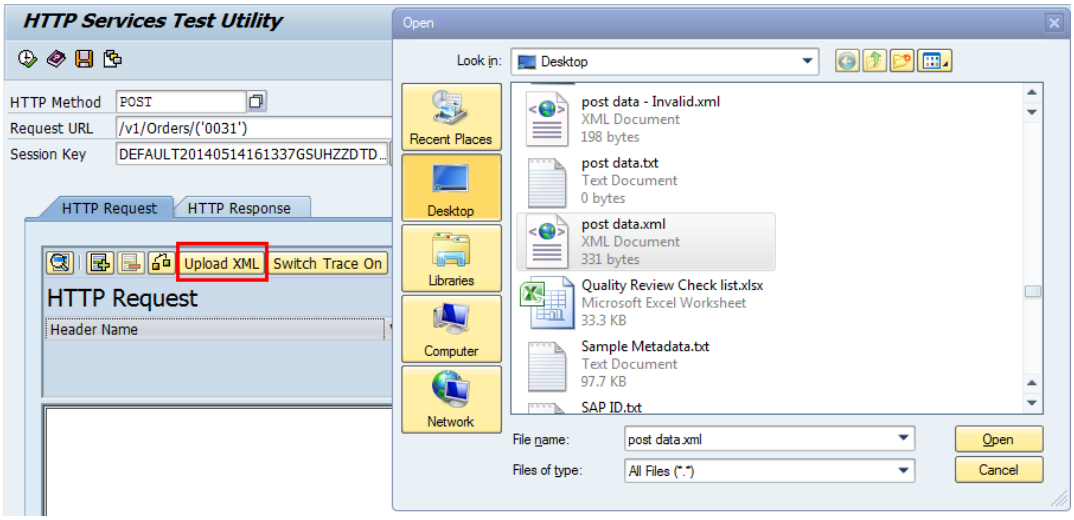
- *Invoking a HTTP service with a Request Body*
  To invoke a HTTP service with a Request Body, follow these steps:

  You require a request body for HTTP services that support the PUT, POST, PATCH and DELETE methods.
  The following example demonstrates a PUT method for the ORDERS service with a request body.

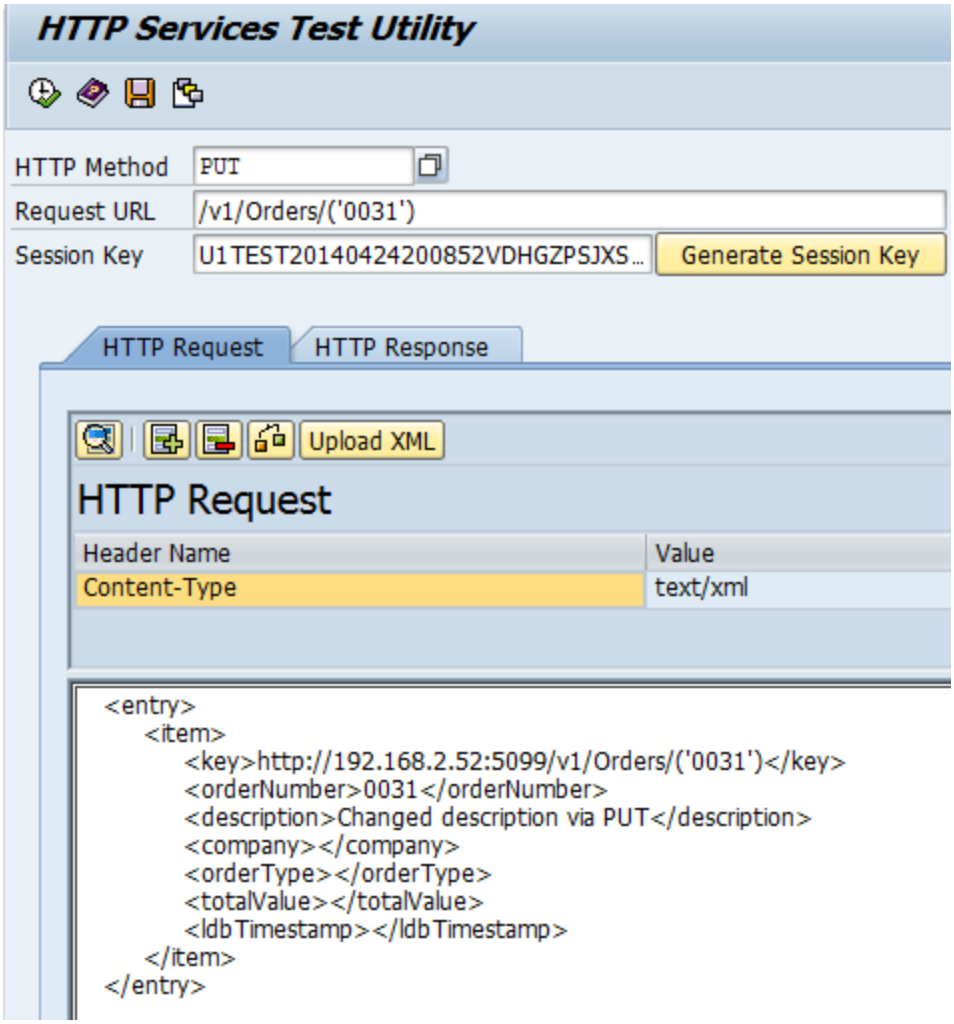
a. Create an XML file (extension `.xml`, for example, `post data.xml`) with the following content:



b. After you fill up the fields for the Method, URL and Session Key, click **Upload XML**. Select the `post data.xml` file.



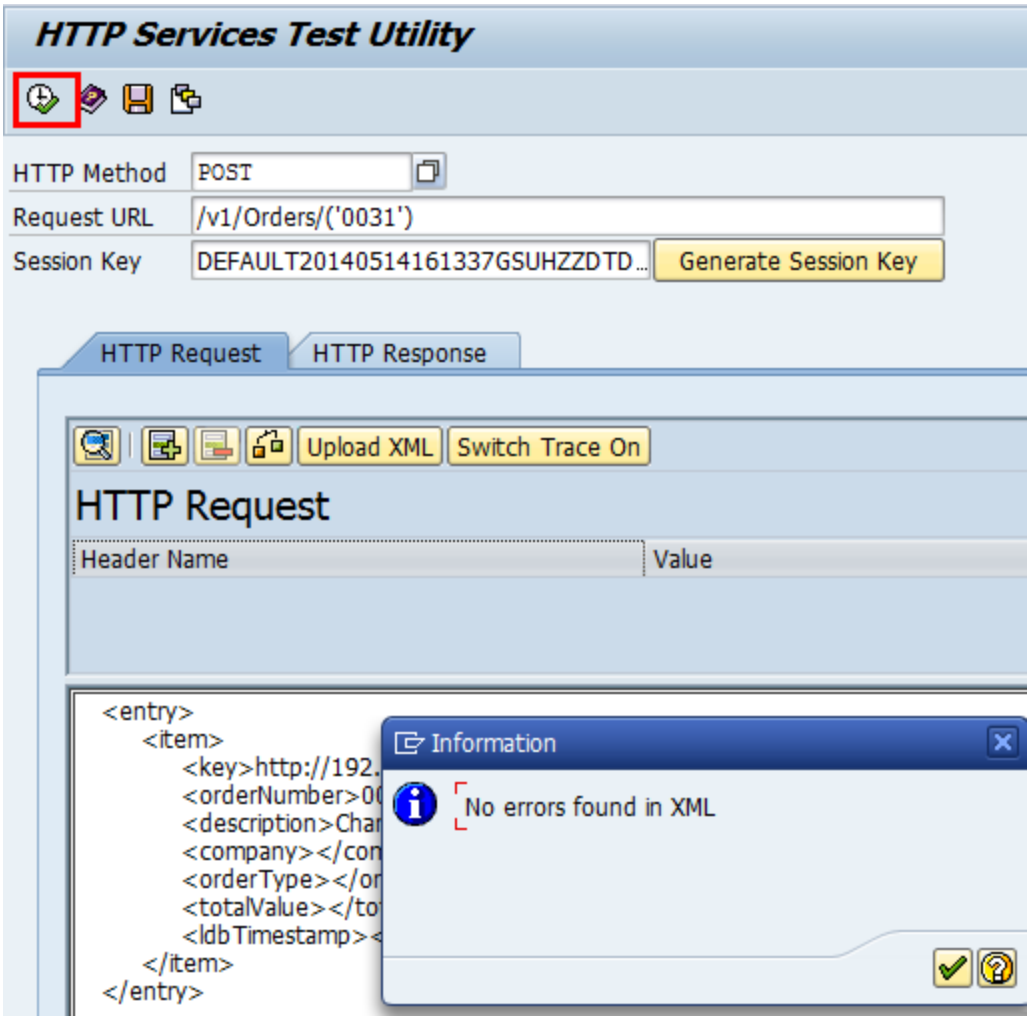
The imported xml data appears.



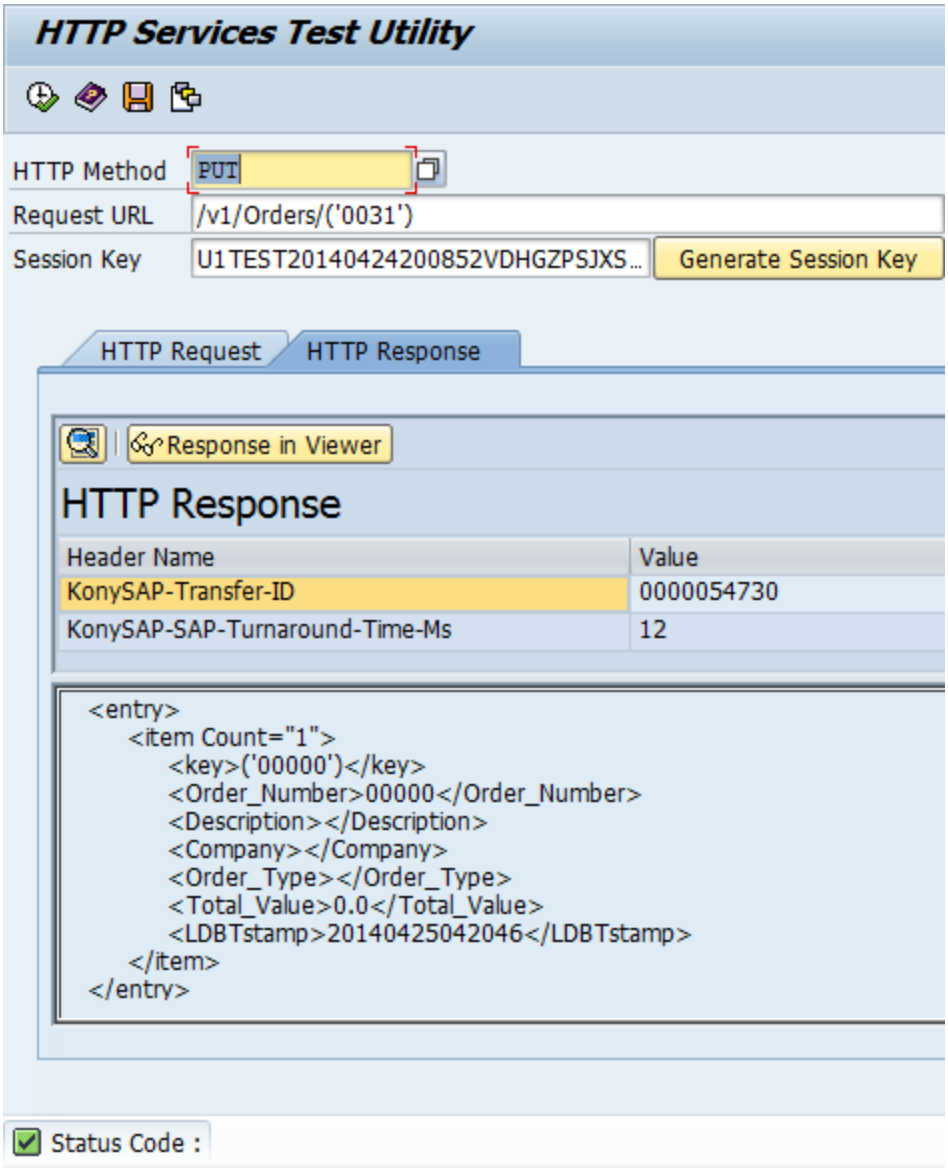
Alternately, you may manually enter or paste the same content in the editor.

The **Check XML** button verifies if the XML content in the editor is structurally well formed or not.

> ***Note:*** The **Check XML** function does not check the validity of the XML in the context of the current service call.



c. Click **Execute** on the application tool bar to execute the request
   A successful response appears.

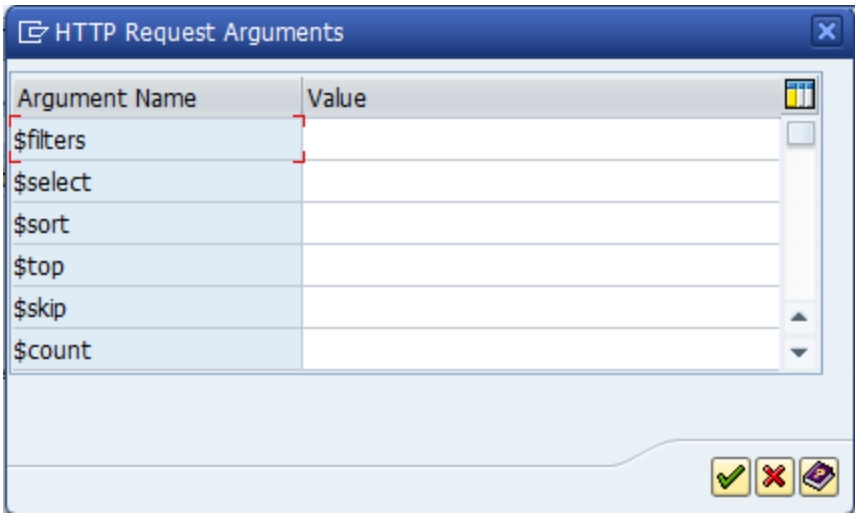


- *Invoking a HTTP service with Request Arguments*

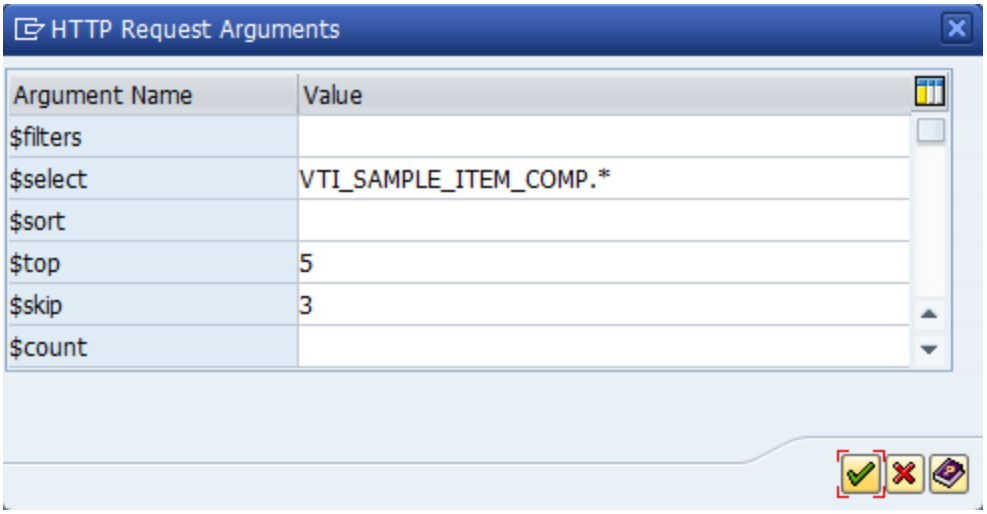
  HTTP services support many request arguments that you can append to a service URL. You can enter these arguments directly in the **Request URL** field, but an easier way is to use the **Request Arguments** dialog.

  To enter the arguments using the **Request Arguments** dialog, follow these steps:

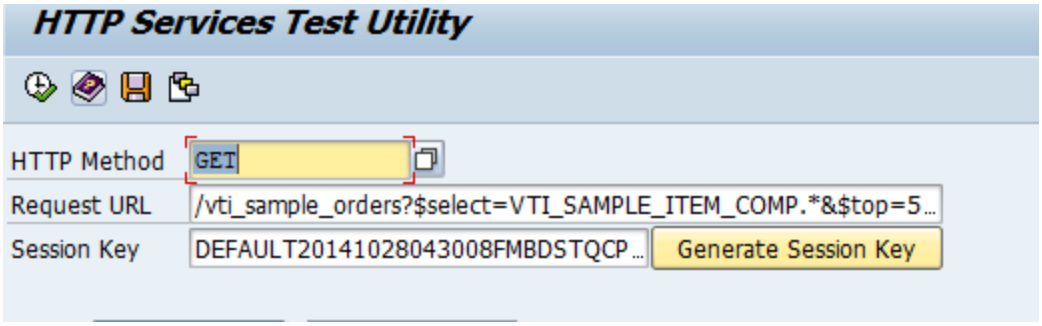
a. Enter the name of resource in **Request URL**, and click the **Request Arguments** button. The **HTTP Resource Arguments** dialog appears.



b. Enter the following values as given below, and press **Enter**.

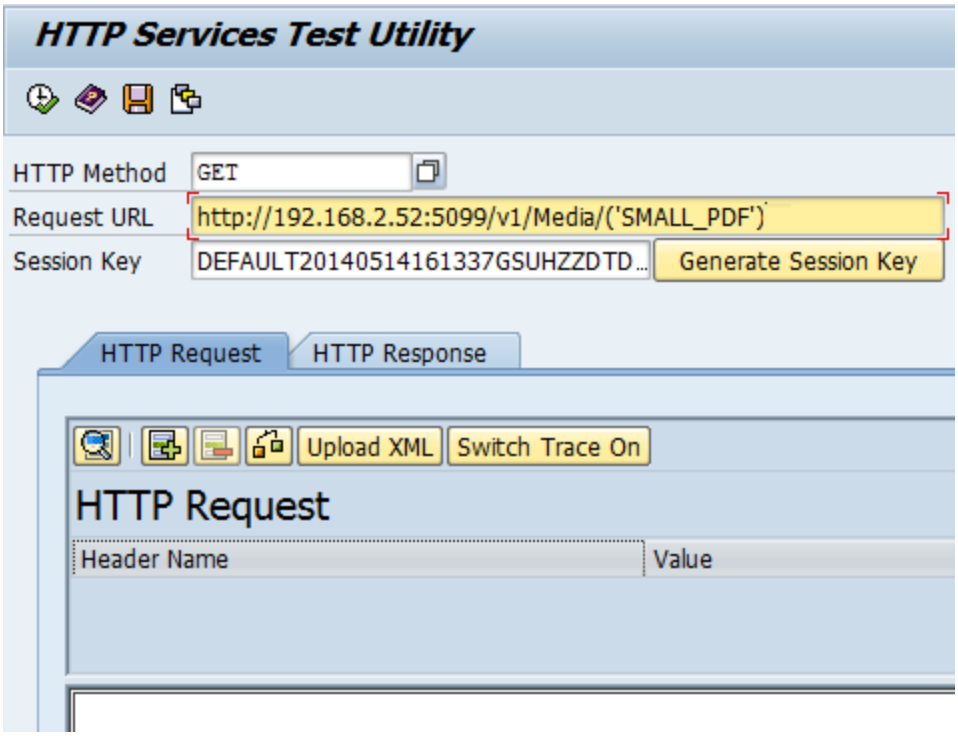


The URL gets appended with the arguments that you enter.

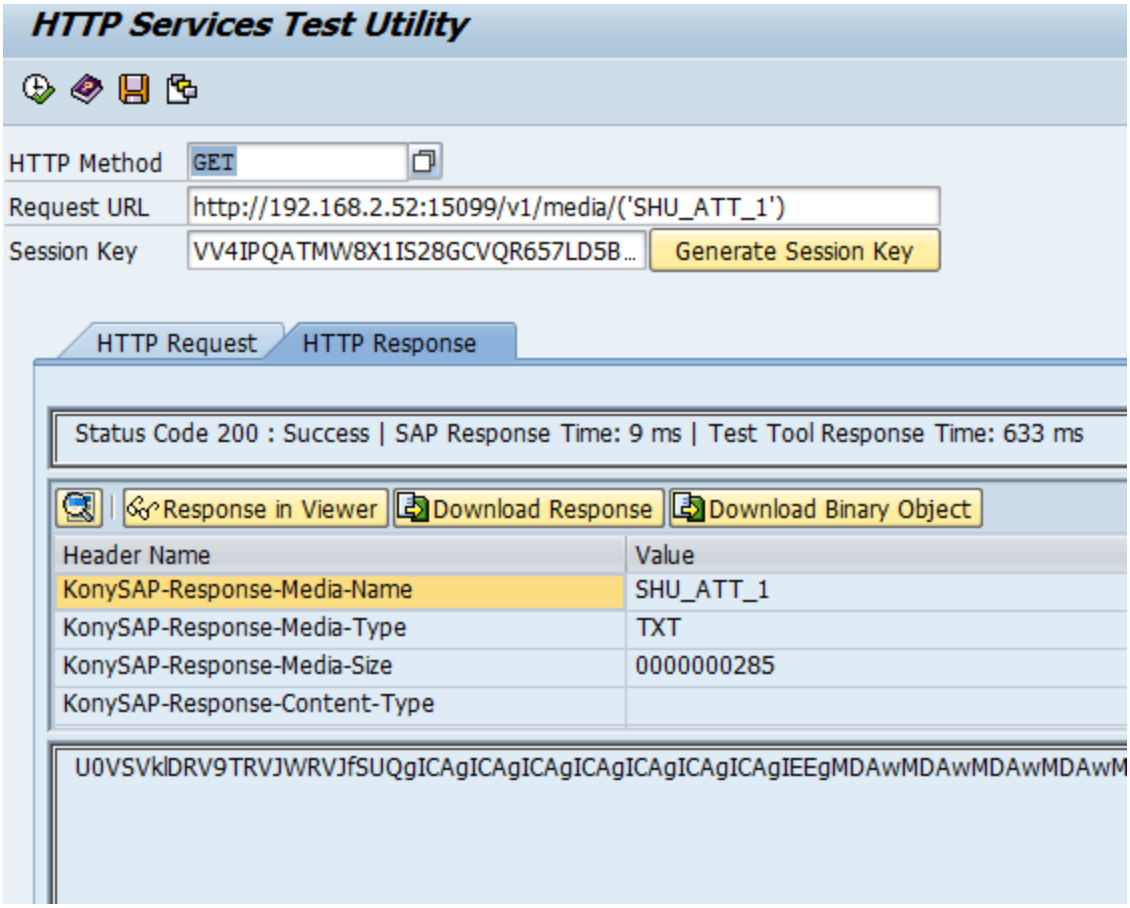


> ***Note:*** If you have already added some request arguments in **Request URL**, and then click the **Request Arguments** button, the **HTTP Request Arguments** dialog appears with the prepopulated arguments in the query.

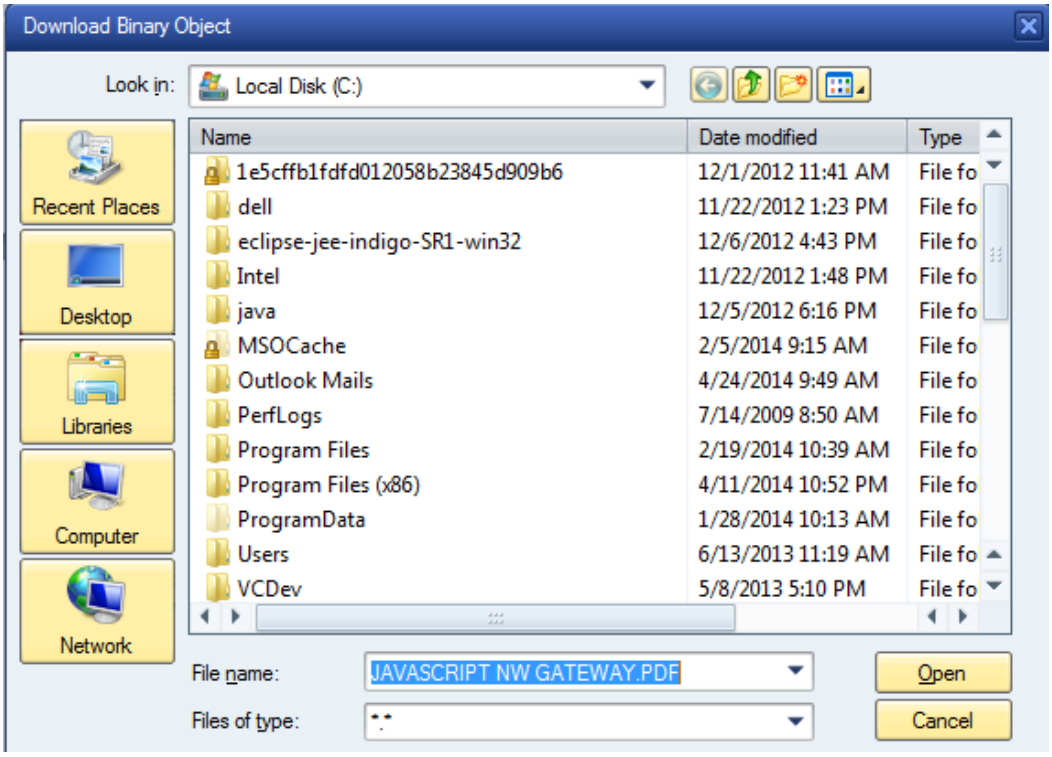
- *Invoking a HTTP service returning a Binary File*
  Some of the HTTP services return the binary content representing a file as part of their response.
  To invoke a HTTP service that returns a Binary File, follow these steps:
  The following example invokes the MEDIA service that returns a PDF file in the response.



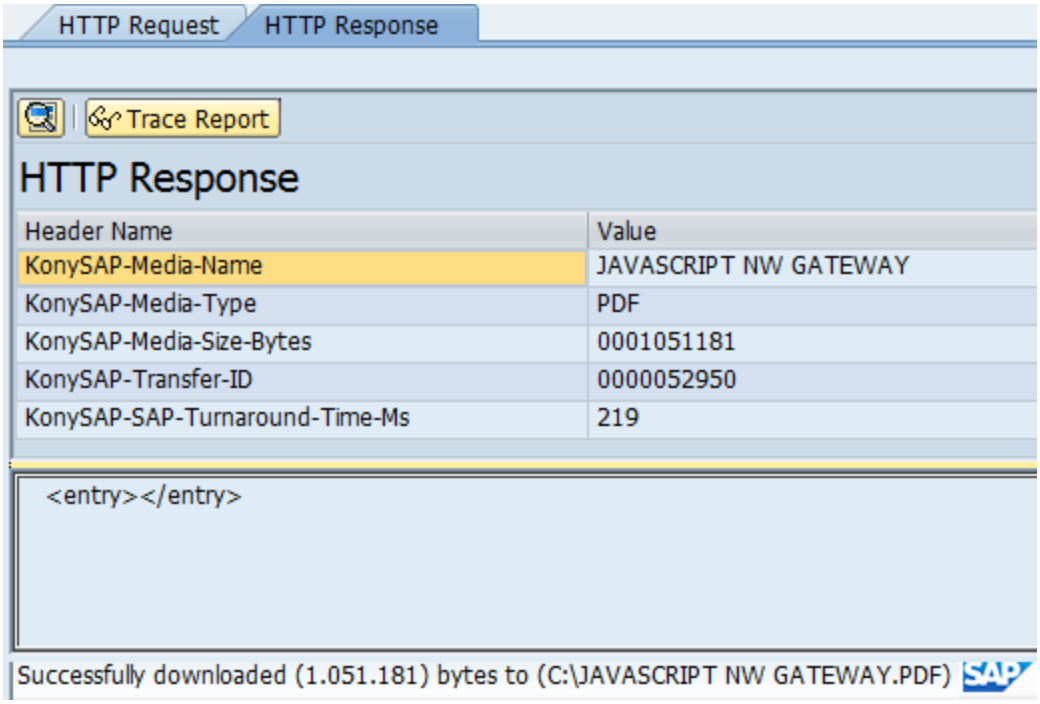
a. Once the fields for HTTP method, Request URL and Session Key are populated with values, click **Execute**.
   The **HTTP Services Test Utility** appears.



b. Click **Download Binary Object** that allows to download the binary file to the desktop. A file dialog appears to save the file.



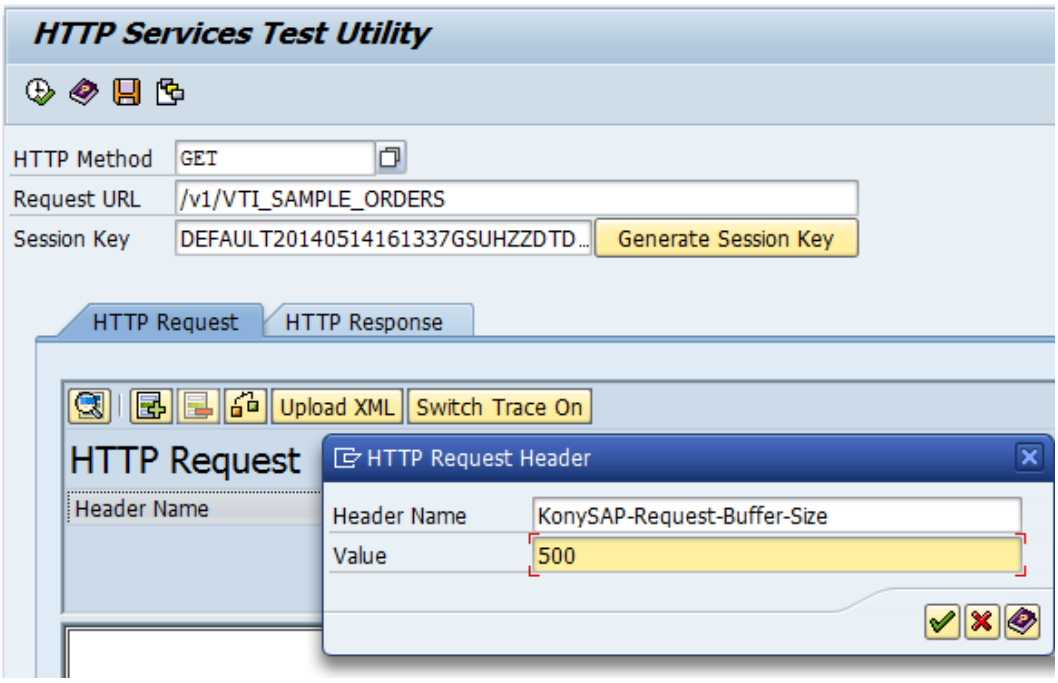
c. Once you save the file, the **HTTP Response** tab appears along with a message, "Successfully downloaded <no. of bytes> bytes to the <downloaded folder location of the pdf> at the bottom of the window.

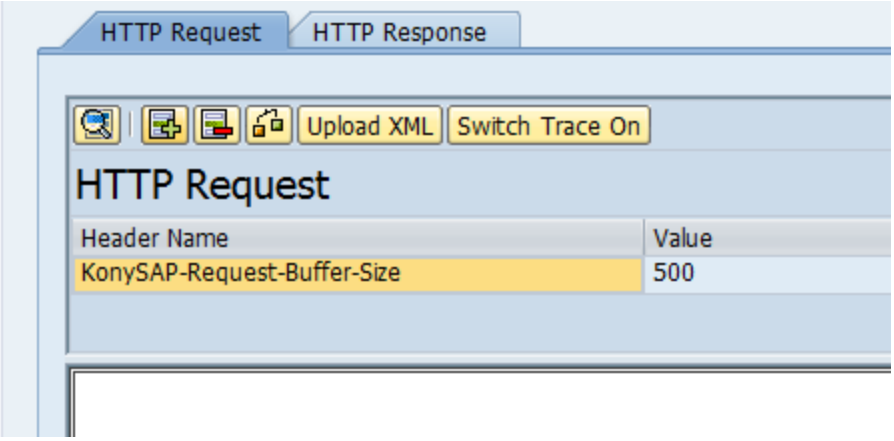


- *Invoking a HTTP service through buffering*

  If the response body returned by a HTTP request is huge, you can use buffering to read chunks of data at a time according to a custom buffer size. Follow the steps below to use HTTP buffering:

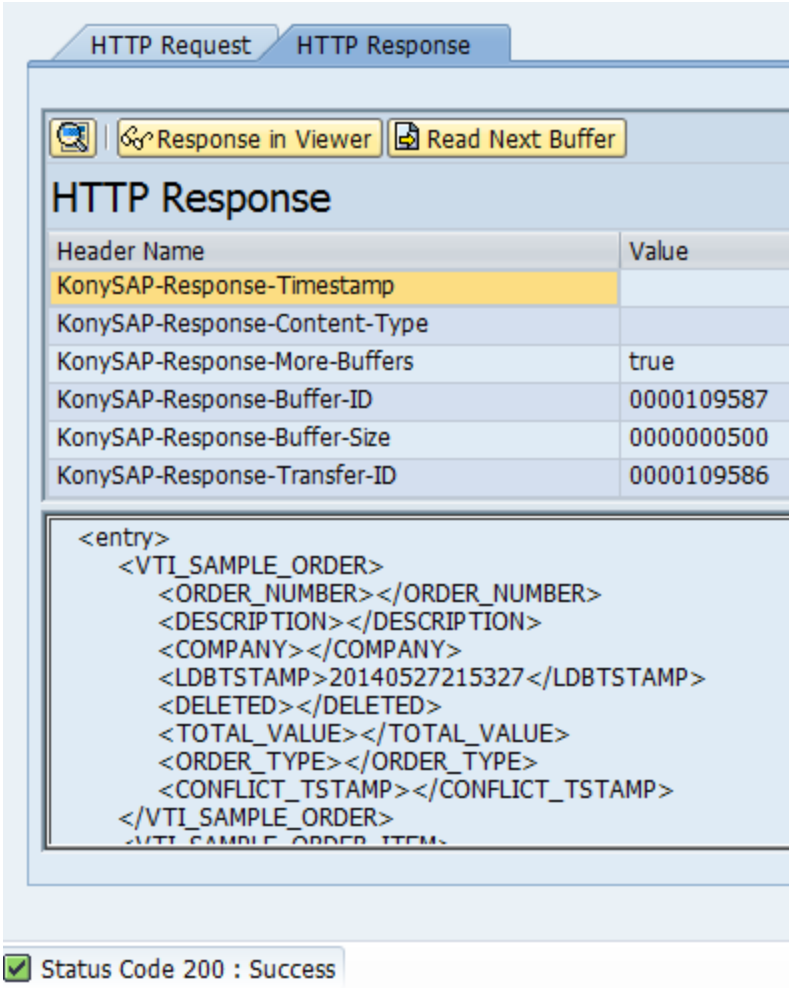
  a. After the basic request data populates, add the HTTP Request header KonySAP-Request-Buffer-Size, and provide the buffer size (in bytes) for the value.



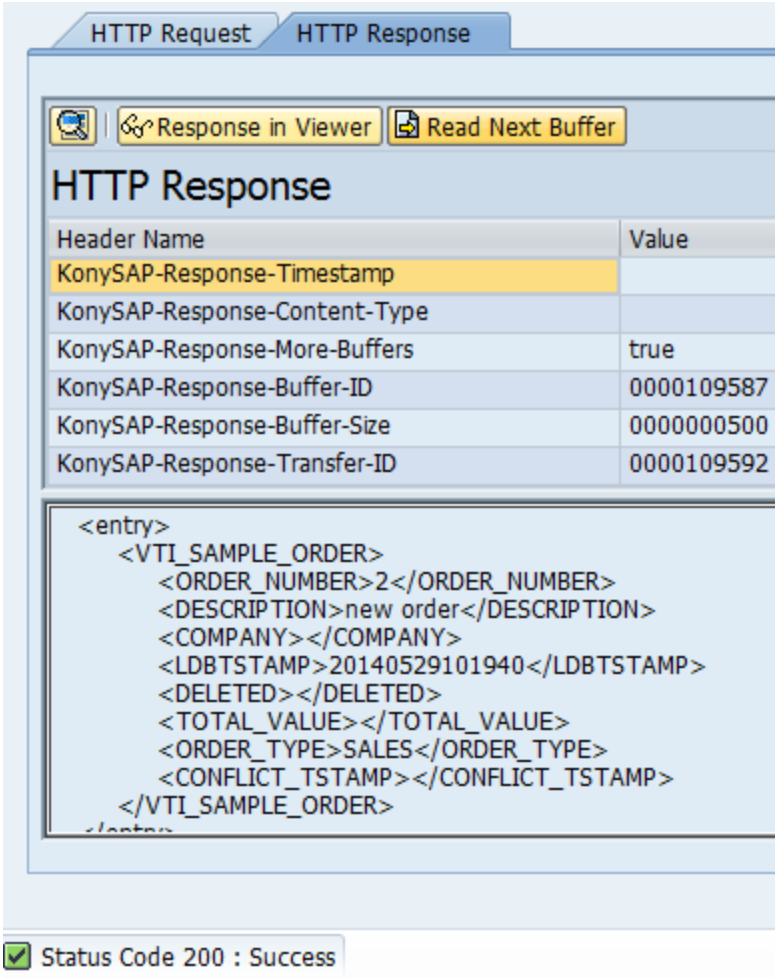


b. Execute the request. The response header additionally returns the following headers:

Header Name	Description
KonySAP-Response-More-Buffers	This parameter is set to true, if there are additional buffers of response data waiting to be read
KonySAP-Response-Buffer-ID	This represents an identifier for the next chunk of buffered data. The value of this parameter needs to pass to the header parameter KonySAP-Request-Buffer-ID of the next HTTP request to retrieve the next set of response data.
KonySAP-Response-Buffer-Size	This represents the size of the buffered data retrieved in the response body



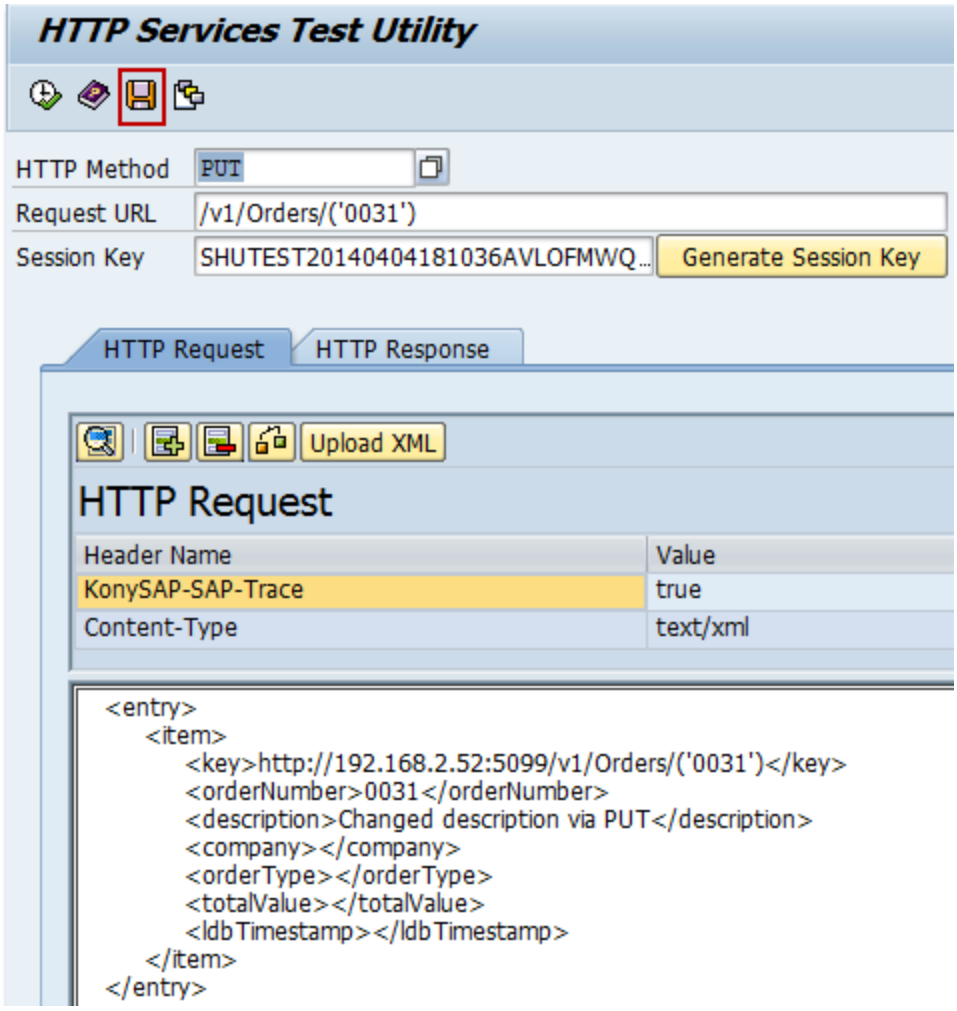
c. You can retrieve the next set of data by clicking the **Read Next Buffer** button. This operation is repeated until there are no more rows to retrieve.



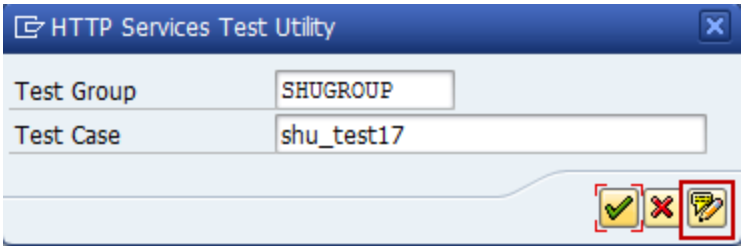
> ***Note:*** Clicking the **Read Next Buffer** button is equivalent to invoking the HTTP service with the request header parameter KonySAP-Request-Buffer-ID set to the value of the response header parameter KonySAP-Response-Buffer-ID of the previous response.

- *Saving a Request in a Test Case*:
  The HTTP Test Utility allows you to store and reuse HTTP service calls through Test Cases. To save a request in a test case, follow these steps:

a. Fill the request data on the **HTTP Request** tab and click **Save Test Case**.



The **HTTP Services Test Utility** dialog appears.



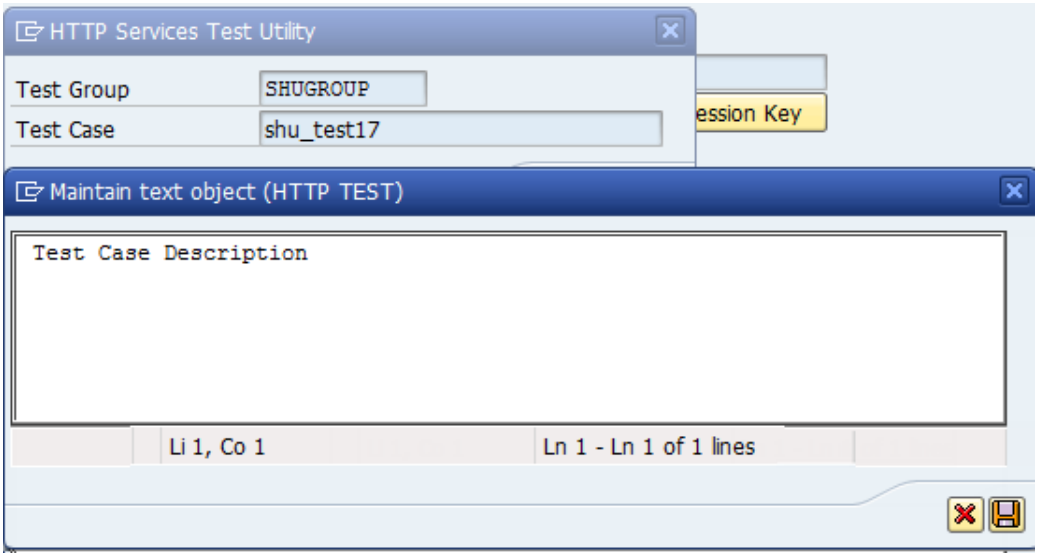
Field	Description
Test Group	Identifier for a Test Group. You may collect the Test cases under a Test Group to organize and manage easily
Test Case	Identifier for a Test Case. If the test case already exists, the system prompts the user with an option to overwrite it. You may also choose an existing test case using the F4 help button
	Extended Description for a Test Case. A text area is provided to store the description

b. Enter the name of the test group and the test case.

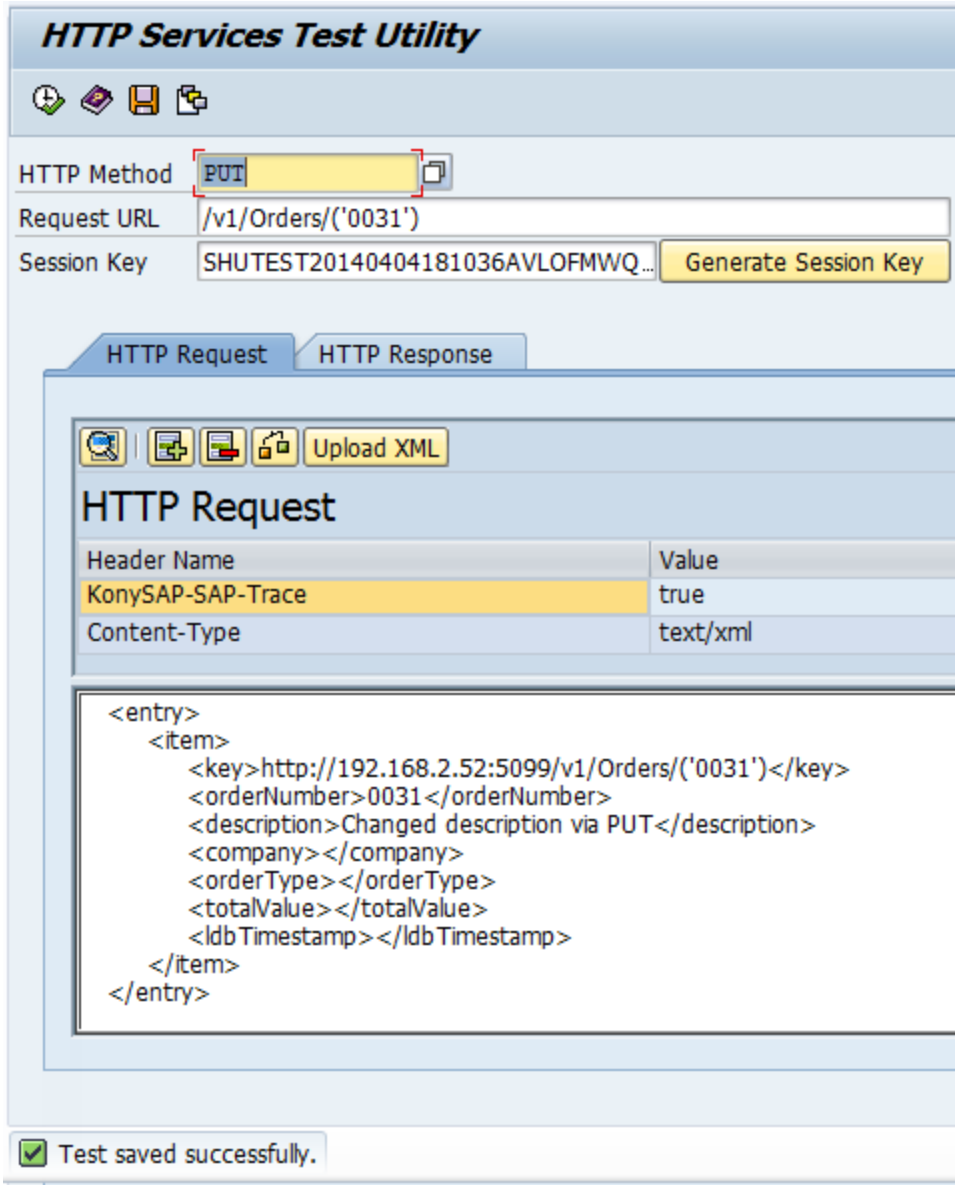
   You may click **Test Case Description** to maintain long description.

c. Click the description button.
   The **Maintain text object (HTTP TEST)** dialog appears.

d. Enter a long description text. Click **Save**.



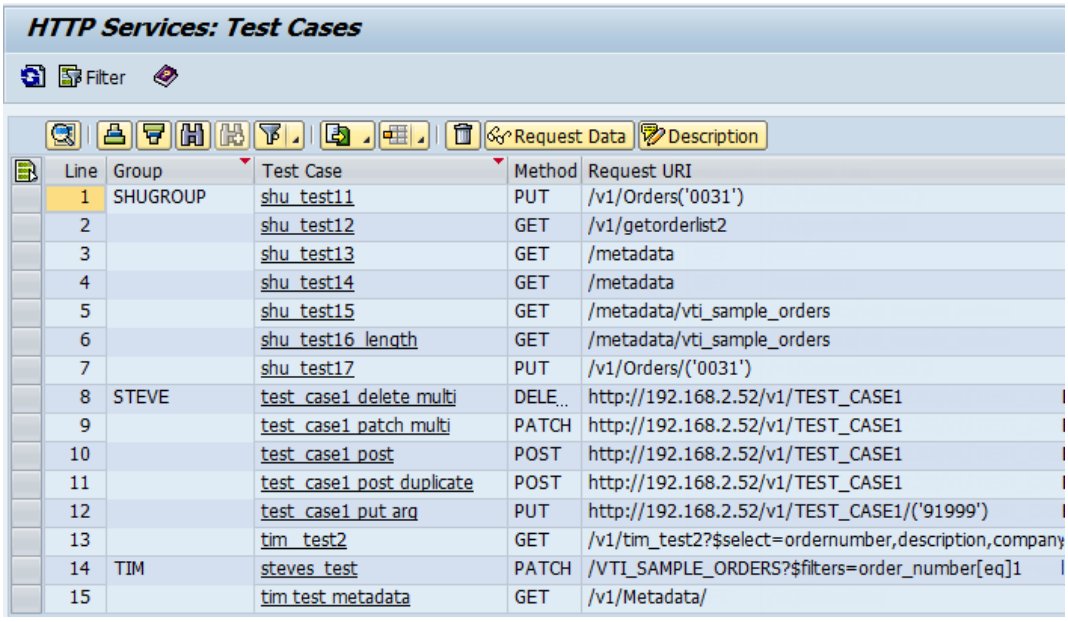
e. After you enter the description text, press **Enter**.



A message,"Test saved successfully" appears.

- *Viewing and Loading a Request from a Test Case*:
  You may load the existing test cases into the HTTP Services Test Utility. Based on the selections provided, a list of the Test Cases that satisfy the criteria, appears.
  To load the existing test cases, follow these steps:

a. Click **Select Test Case** on the application tool bar.
The **HTTP Services: Test Cases** window appears with a list of existing test cases.

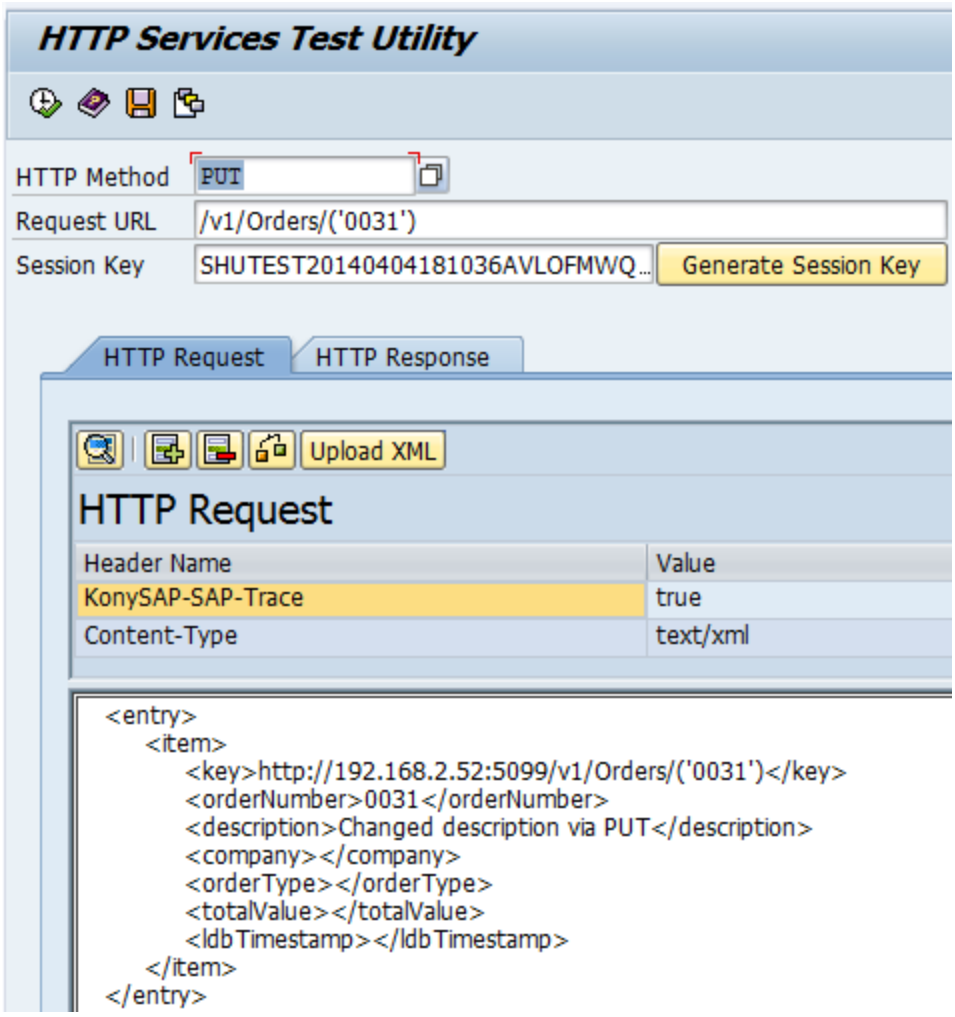


Line	Group	Test Case	Method	Request URI
1	SHUGROUP	shu_test11	PUT	/v1/Orders('0031')
2		shu_test12	GET	/v1/getorderlist2
3		shu_test13	GET	/metadata
4		shu_test14	GET	/metadata
5		shu_test15	GET	/metadata/vti_sample_orders
6		shu_test16_length	GET	/metadata/vti_sample_orders
7		shu_test17	PUT	/v1/Orders/('0031')
8	STEVE	test_case1 delete multi	DELE...	http://192.168.2.52/v1/TEST_CASE1
9		test_case1 patch multi	PATCH	http://192.168.2.52/v1/TEST_CASE1
10		test_case1 post	POST	http://192.168.2.52/v1/TEST_CASE1
11		test_case1 post duplicate	POST	http://192.168.2.52/v1/TEST_CASE1
12		test_case1 put arg	PUT	http://192.168.2.52/v1/TEST_CASE1/('91999')
13		tim_test2	GET	/v1/tim_test2?$select=ordernumber,description,company
14	TIM	steves_test	PATCH	/VTI_SAMPLE_ORDERS?$filters=order_number[eq]1
15		tim test metadata	GET	/v1/Metadata/

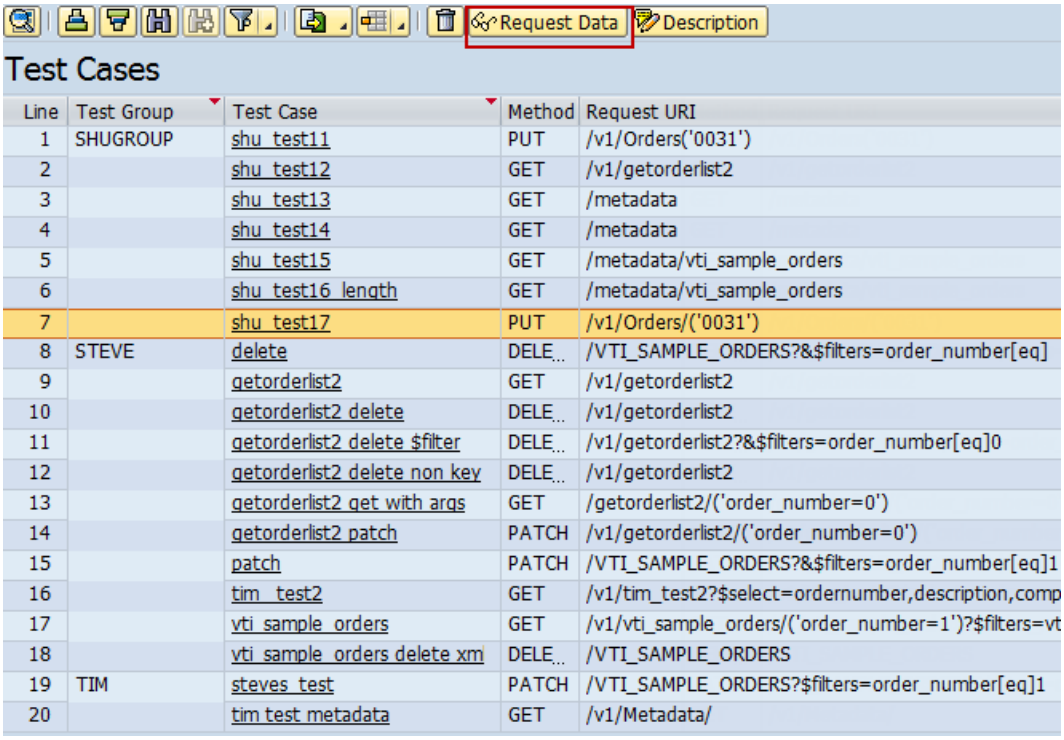
Field	Description
Request Data	Displays the formatted Request Header and Body in the standard HTML viewer
Description	Displays the long text maintained for the Test Case while you save
	Deletes the Selected Test Case
Filter	Invokes the Select Test Cases dialog to enable the user to enter a new Test Case selection
	Manages Layouts for the Test Cases Table Display

Field	Description
	Exports Test Cases Table to a File
	Sorts Test Cases Table in ascending order
	Sorts Test Cases Table in descending order
	Finds a value in the Test Cases Table
	Filters Test Cases Table
	Refreshes Test Cases Table Display

b. Double-click any line or click the hotspot on the Test Case name to load the Test Case onto the editor of the **HTTP Request** tab.

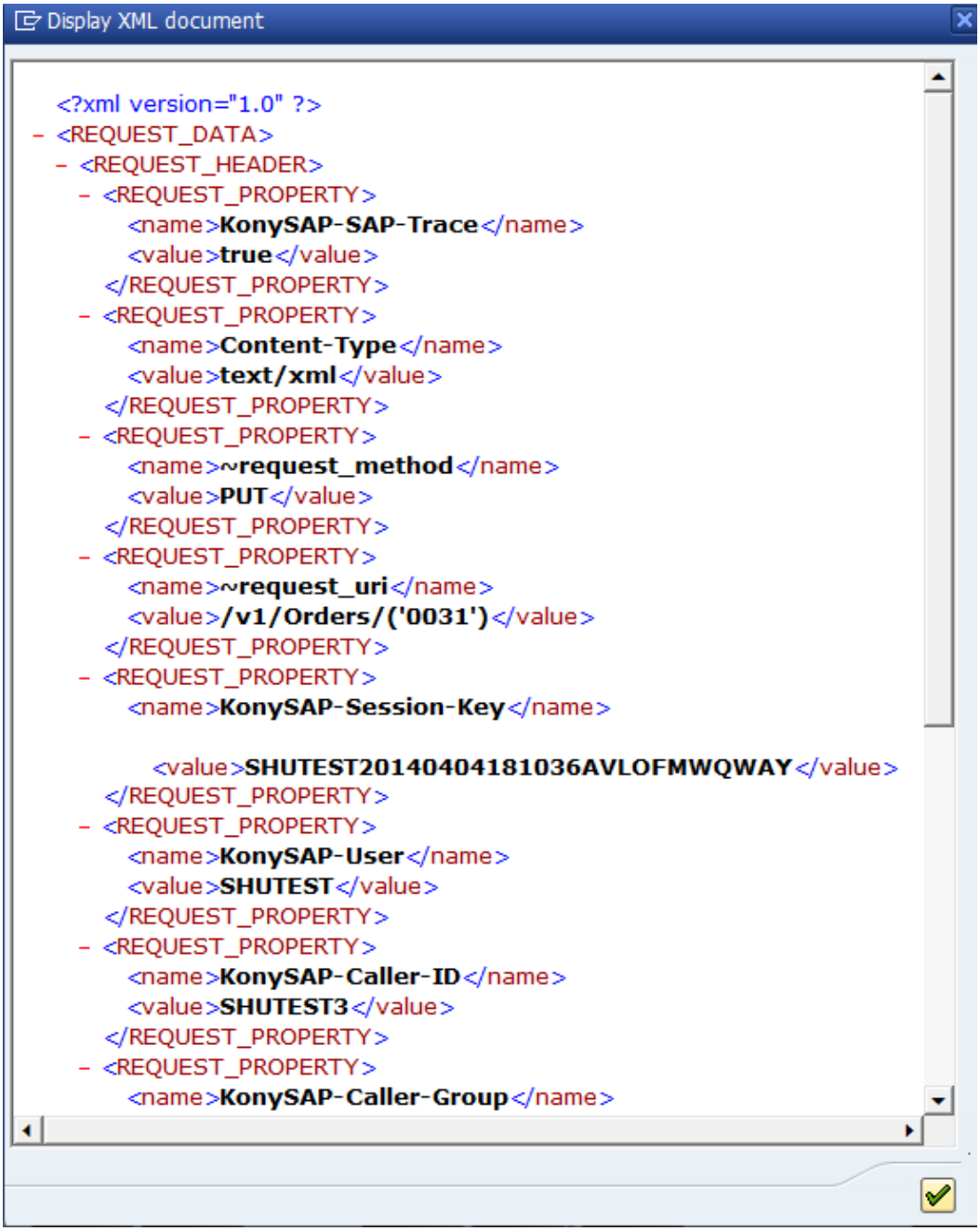


c. On the **HTTP Services Test Cases** window, select a line and click **Request Data** to view the Request Headers and Body formatted in XML.

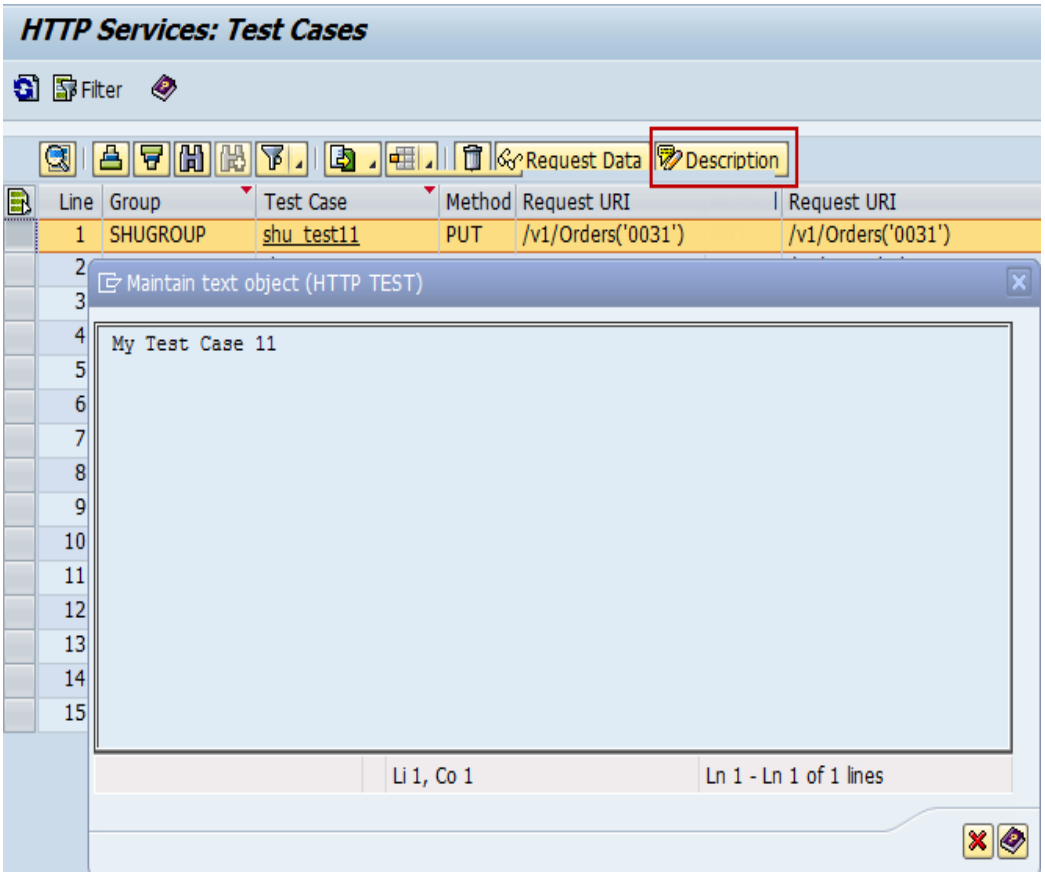


Line	Test Group	Test Case	Method	Request URI
1	SHUGROUP	shu_test11	PUT	/v1/Orders('0031')
2		shu_test12	GET	/v1/getorderlist2
3		shu_test13	GET	/metadata
4		shu_test14	GET	/metadata
5		shu_test15	GET	/metadata/vti_sample_orders
6		shu_test16_length	GET	/metadata/vti_sample_orders
7		shu_test17	PUT	/v1/Orders/('0031')
8	STEVE	delete	DELE...	/VTI_SAMPLE_ORDERS?&$filters=order_number[eq]
9		getorderlist2	GET	/v1/getorderlist2
10		getorderlist2 delete	DELE...	/v1/getorderlist2
11		getorderlist2 delete $filter	DELE...	/v1/getorderlist2?&$filters=order_number[eq]0
12		getorderlist2 delete non key	DELE...	/v1/getorderlist2
13		getorderlist2 get with args	GET	/getorderlist2/('order_number=0')
14		getorderlist2 patch	PATCH	/v1/getorderlist2/('order_number=0')
15		patch	PATCH	/VTI_SAMPLE_ORDERS?&$filters=order_number[eq]1
16		tim_test2	GET	/v1/tim_test2?$select=ordernumber,description,comp
17		vti_sample_orders	GET	/v1/vti_sample_orders/('order_number=1')?$filters=vt
18		vti_sample_orders delete xm	DELE...	/VTI_SAMPLE_ORDERS
19	TIM	steves_test	PATCH	/VTI_SAMPLE_ORDERS?$filters=order_number[eq]1
20		tim test metadata	GET	/v1/Metadata/

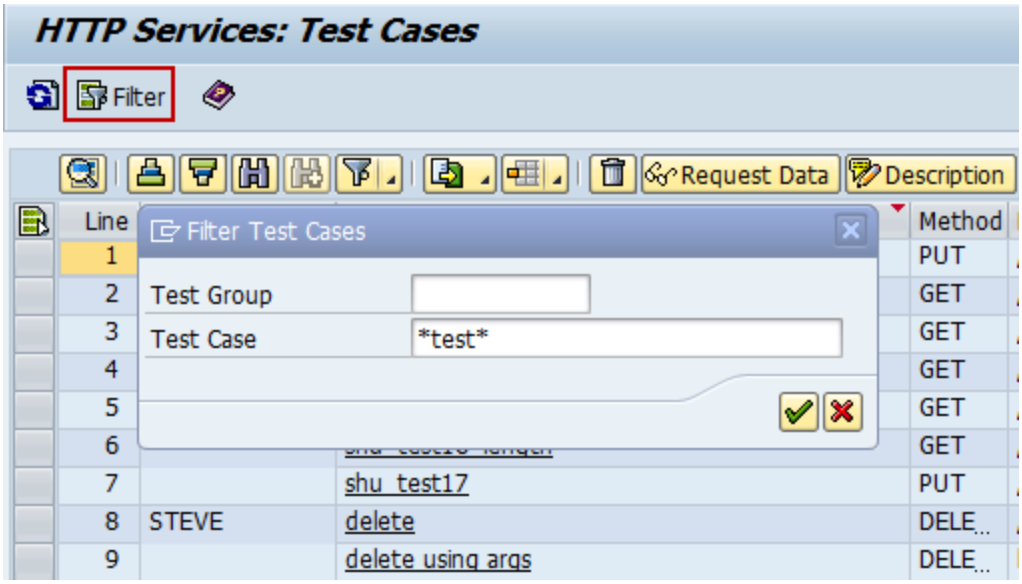
The Request Headers and Body formatted in XML:



d. On the **HTTP Services Test Cases** window, you may click **Description** to view any long texts associated with the test case.



e. To filter the test cases, click the **Filter** button on the application tool bar.
The **Filter Test Cases** dialog appears.

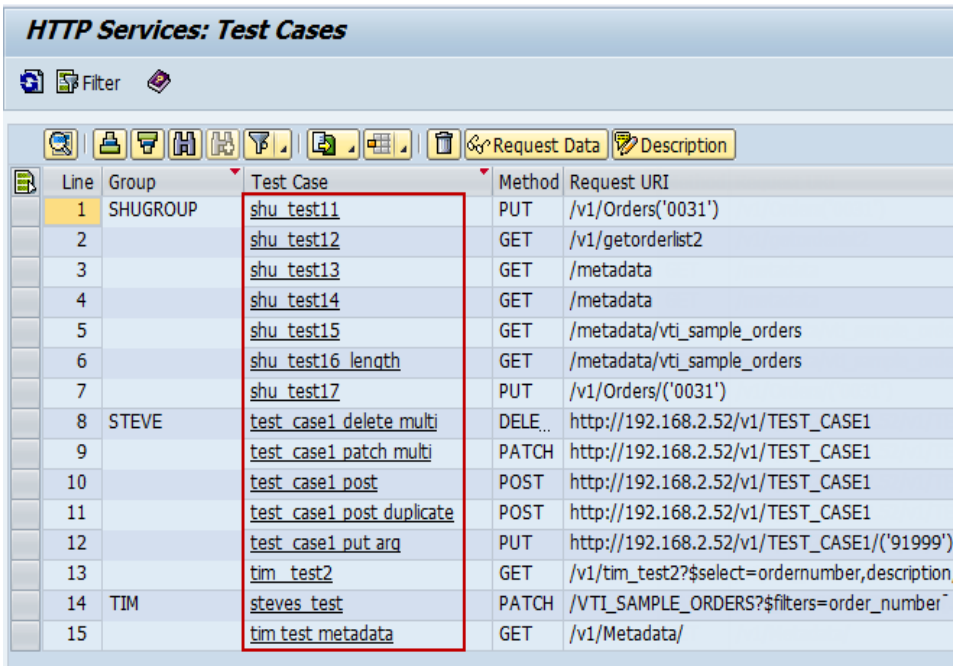


Enter the relevant values in the following fields:

Field	Description
Test Group	Identifier for a Test Group. You may collect the Test cases under a Test Group to organize and manage easily
Test Case	Identifier for a Test Case. If the test case already exists, the system prompts the user with an option to overwrite it. You may also choose an existing test case using the F4 help button

a. You can narrow down the Test Group and Test Case using wild cards.

   An example of all the test cases in the system:



HTTP Services: Test Cases

Filter | Request Data | Description

Line	Group	Test Case	Method	Request URI
1	SHUGROUP	shu_test11	PUT	/v1/Orders('0031')
2		shu_test12	GET	/v1/getorderlist2
3		shu_test13	GET	/metadata
4		shu_test14	GET	/metadata
5		shu_test15	GET	/metadata/vti_sample_orders
6		shu_test16_length	GET	/metadata/vti_sample_orders
7		shu_test17	PUT	/v1/Orders/('0031')
8	STEVE	test_case1 delete multi	DELE...	http://192.168.2.52/v1/TEST_CASE1
9		test_case1 patch multi	PATCH	http://192.168.2.52/v1/TEST_CASE1
10		test_case1 post	POST	http://192.168.2.52/v1/TEST_CASE1
11		test_case1 post duplicate	POST	http://192.168.2.52/v1/TEST_CASE1
12		test_case1 put arg	PUT	http://192.168.2.52/v1/TEST_CASE1/('91999')
13		tim_test2	GET	/v1/tim_test2?$select=ordernumber,description
14	TIM	steves_test	PATCH	/VTI_SAMPLE_ORDERS?$filters=order_number
15		tim test metadata	GET	/v1/Metadata/

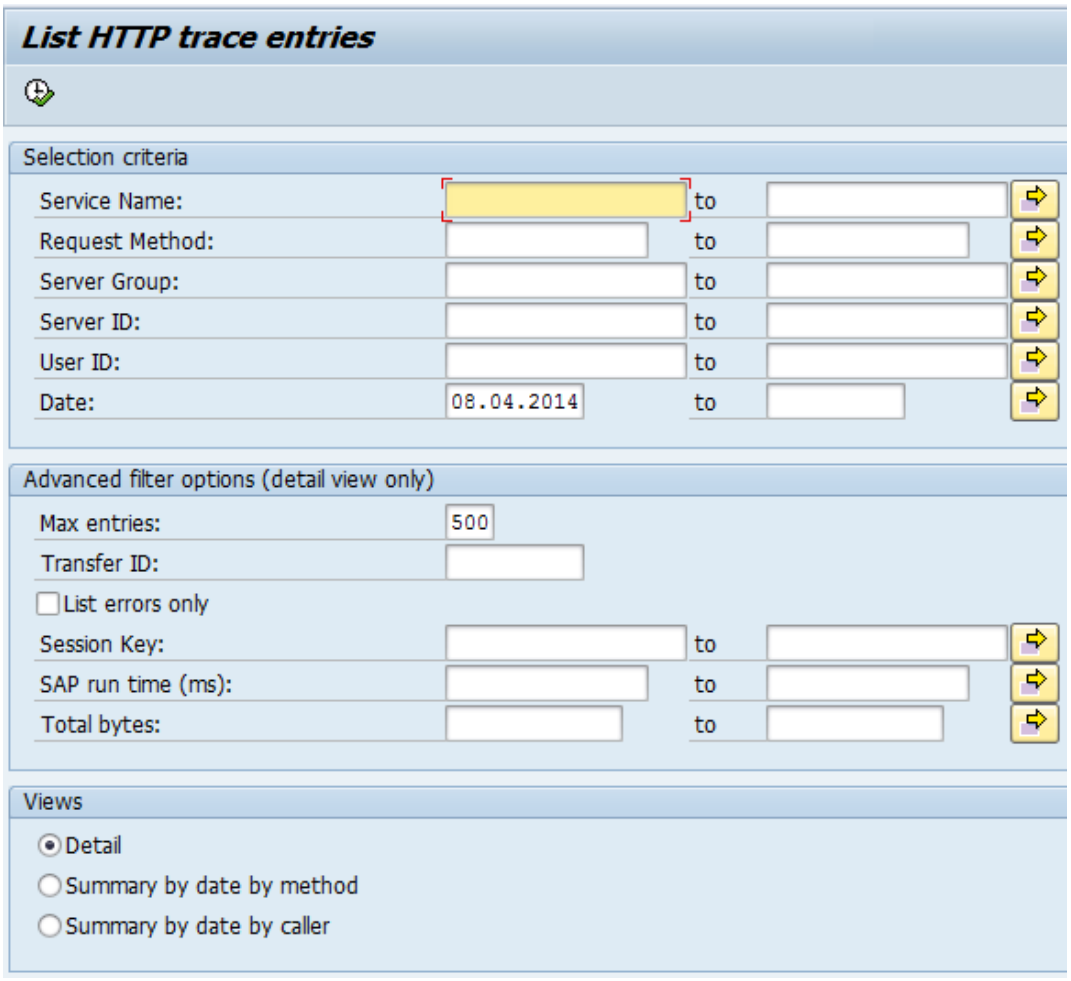
7.4.3 HTTP Services Trace Report

You may view summary reports and detail entries for traces recorded during HTTP Service calls. The tracing for HTTP Service calls is activated when you configure the request header parameter *KonySAP-SAP-Trace* to *true* and call the service.

To generate the HTTP Services Trace report, follow these steps:

1. Execute the transaction **/SKY/YVT7 (YVT7)** in SAP system.

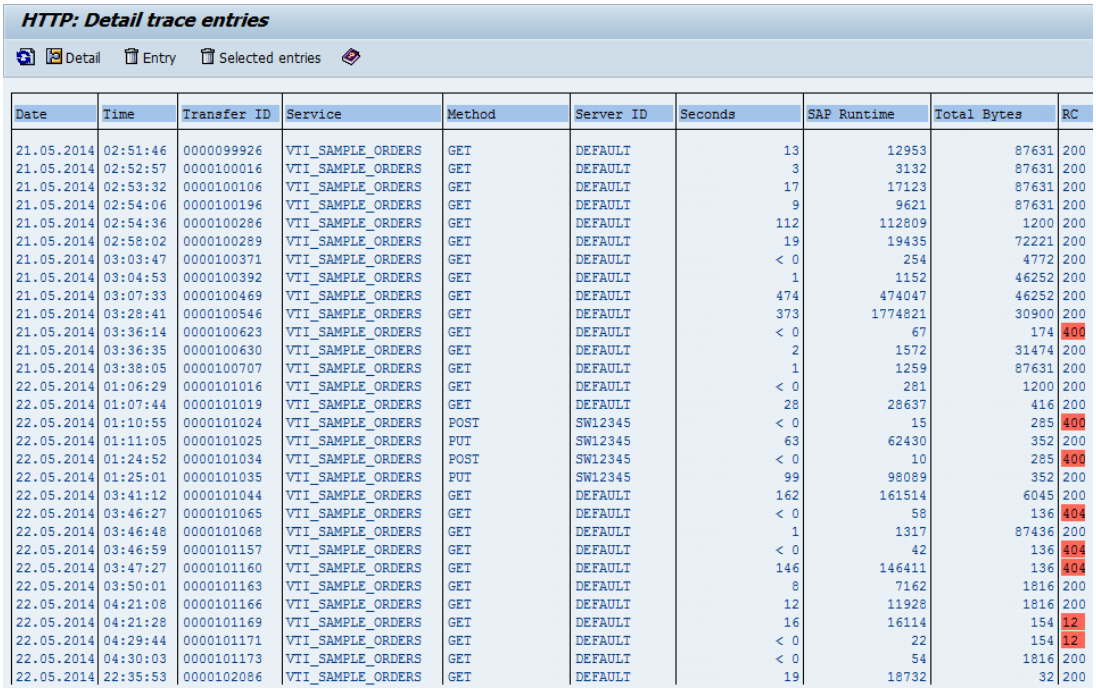
   The **List HTTP trace entries** window appears.



2. You may view the HTTP Trace Entries in the following three views:
   - Detail: Provides elaborate information about the HTTP Service Request and Response.
   - Summary by date by method: Provides summary of the HTTP service calls by date and by method.
   - Summary by date by caller: Provides summary of the HTTP service calls by date and by caller.

7.4.3.1 Detail View

a. On the **List HTTP trace entries** window, under the **Views** section, select the **Detail** option.
The **HTTP: Detail trace entries** window appears with the details.



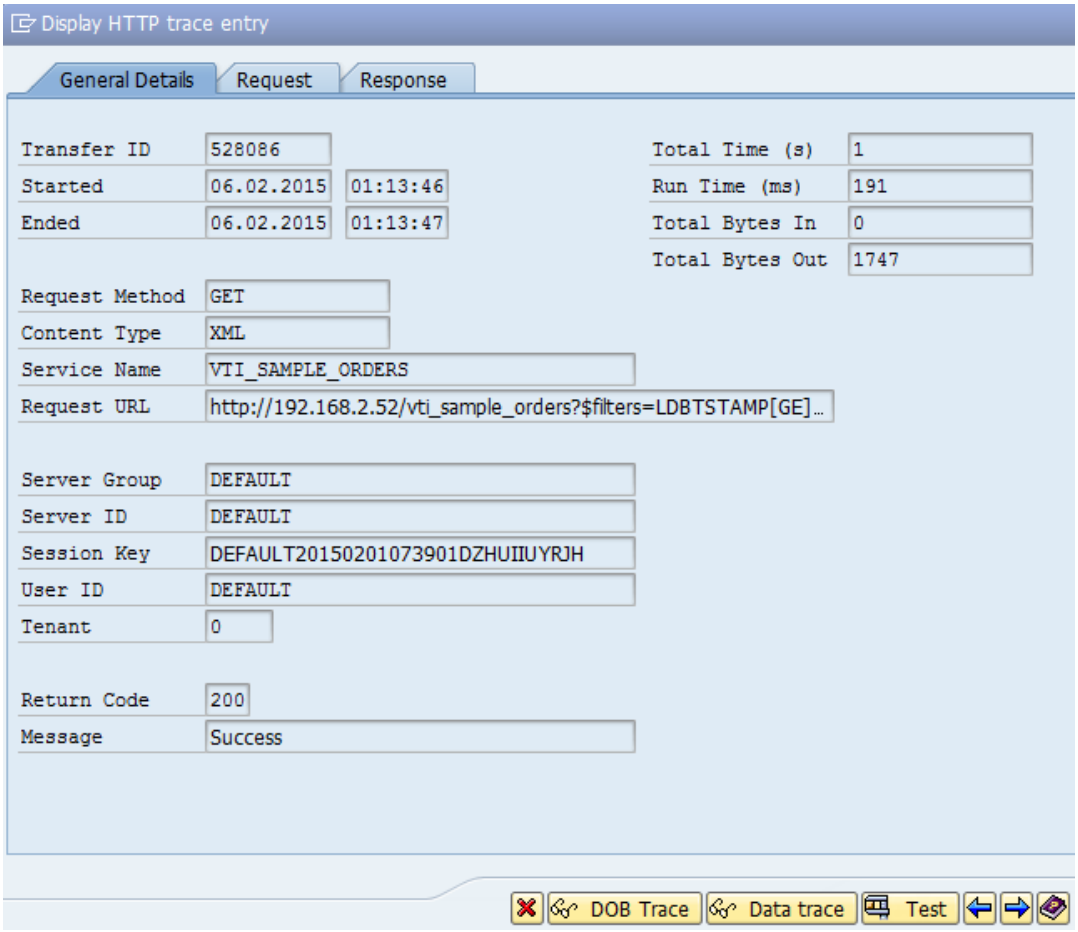
HTTP: Detail trace entries

Detail | Entry | Selected entries

Date	Time	Transfer ID	Service	Method	Server ID	Seconds	SAP Runtime	Total Bytes	RC
21.05.2014	02:51:46	0000099926	VTI_SAMPLE_ORDERS	GET	DEFAULT	13	12953	87631	200
21.05.2014	02:52:57	0000100016	VTI_SAMPLE_ORDERS	GET	DEFAULT	3	3132	87631	200
21.05.2014	02:53:32	0000100106	VTI_SAMPLE_ORDERS	GET	DEFAULT	17	17123	87631	200
21.05.2014	02:54:06	0000100196	VTI_SAMPLE_ORDERS	GET	DEFAULT	9	9621	87631	200
21.05.2014	02:54:36	0000100286	VTI_SAMPLE_ORDERS	GET	DEFAULT	112	112809	1200	200
21.05.2014	02:58:02	0000100289	VTI_SAMPLE_ORDERS	GET	DEFAULT	19	19435	72221	200
21.05.2014	03:03:47	0000100371	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	254	4772	200
21.05.2014	03:04:53	0000100392	VTI_SAMPLE_ORDERS	GET	DEFAULT	1	1152	46252	200
21.05.2014	03:07:33	0000100469	VTI_SAMPLE_ORDERS	GET	DEFAULT	474	474047	46252	200
21.05.2014	03:28:41	0000100546	VTI_SAMPLE_ORDERS	GET	DEFAULT	373	1774821	30900	200
21.05.2014	03:36:14	0000100623	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	67	174	400
21.05.2014	03:36:35	0000100630	VTI_SAMPLE_ORDERS	GET	DEFAULT	2	1572	31474	200
21.05.2014	03:38:05	0000100707	VTI_SAMPLE_ORDERS	GET	DEFAULT	1	1259	87631	200
22.05.2014	01:06:29	0000101016	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	281	1200	200
22.05.2014	01:07:44	0000101019	VTI_SAMPLE_ORDERS	GET	DEFAULT	28	28637	416	200
22.05.2014	01:10:55	0000101024	VTI_SAMPLE_ORDERS	POST	SW12345	< 0	15	285	400
22.05.2014	01:11:05	0000101025	VTI_SAMPLE_ORDERS	PUT	SW12345	63	62430	352	200
22.05.2014	01:24:52	0000101034	VTI_SAMPLE_ORDERS	POST	SW12345	< 0	10	285	400
22.05.2014	01:25:01	0000101035	VTI_SAMPLE_ORDERS	PUT	SW12345	99	98089	352	200
22.05.2014	03:41:12	0000101044	VTI_SAMPLE_ORDERS	GET	DEFAULT	162	161514	6045	200
22.05.2014	03:46:27	0000101065	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	58	136	404
22.05.2014	03:46:48	0000101068	VTI_SAMPLE_ORDERS	GET	DEFAULT	1	1317	87436	200
22.05.2014	03:46:59	0000101157	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	42	136	404
22.05.2014	03:47:27	0000101160	VTI_SAMPLE_ORDERS	GET	DEFAULT	146	146411	136	404
22.05.2014	03:50:01	0000101163	VTI_SAMPLE_ORDERS	GET	DEFAULT	8	7162	1816	200
22.05.2014	04:21:08	0000101166	VTI_SAMPLE_ORDERS	GET	DEFAULT	12	11928	1816	200
22.05.2014	04:21:28	0000101169	VTI_SAMPLE_ORDERS	GET	DEFAULT	16	16114	154	12
22.05.2014	04:29:44	0000101171	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	22	154	12
22.05.2014	04:30:03	0000101173	VTI_SAMPLE_ORDERS	GET	DEFAULT	< 0	54	1816	200
22.05.2014	22:35:53	0000102086	VTI_SAMPLE_ORDERS	GET	DEFAULT	19	18732	32	200

b. Double-click any row of the detail view.
The **Display HTTP trace entry** window appears with the **General Details**, **Request** and **Response** tabs.

Click the **General Details** tab to view the details.

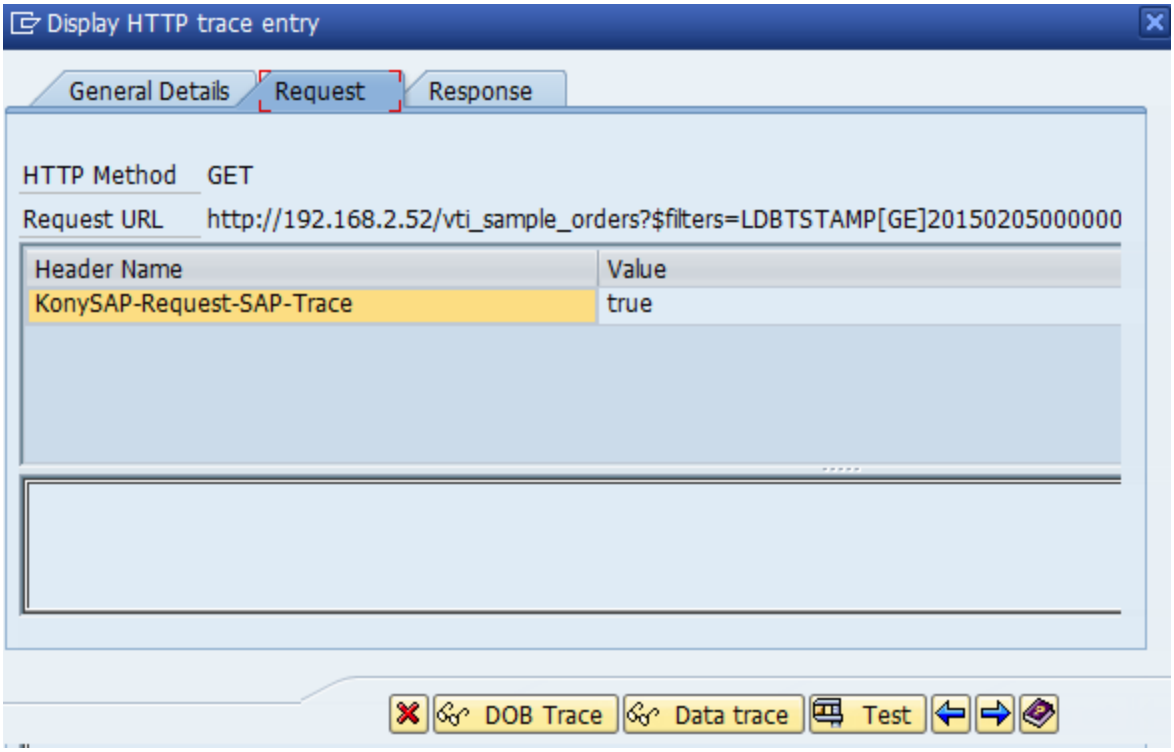


Field	Description
Transfer ID	Internal ID for the trace entry
Started	Start date and time of the HTTP service call
Ended	End date and time of the HTTP service call

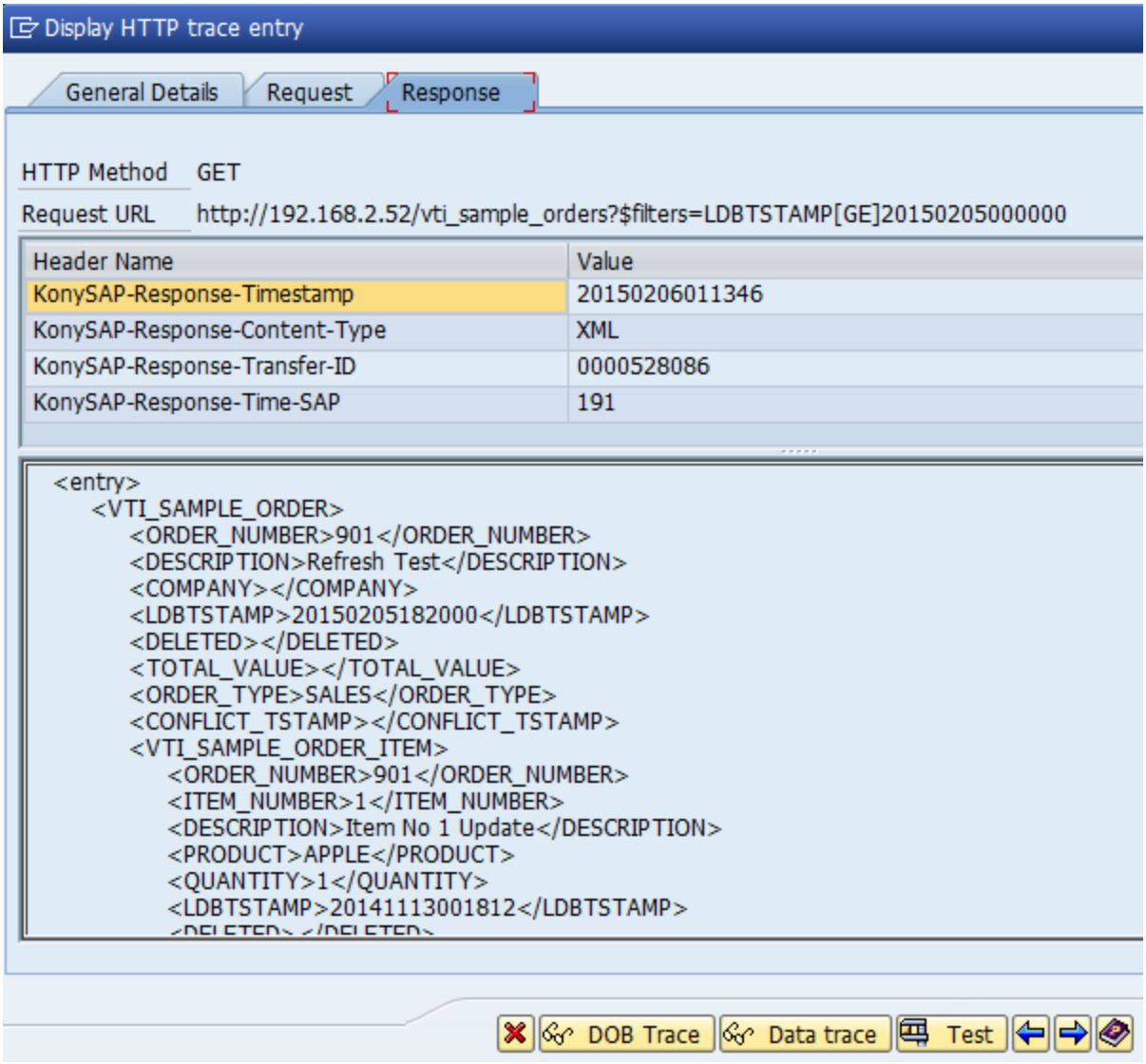
Field	Description
Request Method	HTTP request method that is called (For example, GET, PUT, POST, PATCH, and DELETE)
Service Name	Name of the HTTP service that is invoked
Request URL	Full URL of the HTTP service call
Server Group	Group name of the device that invokes the HTTP Service Call
Server ID	Server ID of the Device that invokes the HTTP service call
Session Key	Session identifier used for the HTTP service call. You require the session identifier for all services except for Login, Logoff and Ping.
User ID	User name with which the HTTP service call is made
Tenant	Tenant for the User ID
Return Code	Return code of the HTTP service call
Message	Message that pertains to the return code
Total Time (s)	Total time in seconds for the complete HTTP service call round trip

Field	Description
Run Time (ms)	Run time in milliseconds for the processing the actual HTTP service call
Total Bytes In	Size of the data (HTTP request Headers and Body) that you pass during the HTTP service call invocation
Total Bytes Out	Size of the data (HTTP response Headers and Body) that the HTTP service call processing returns

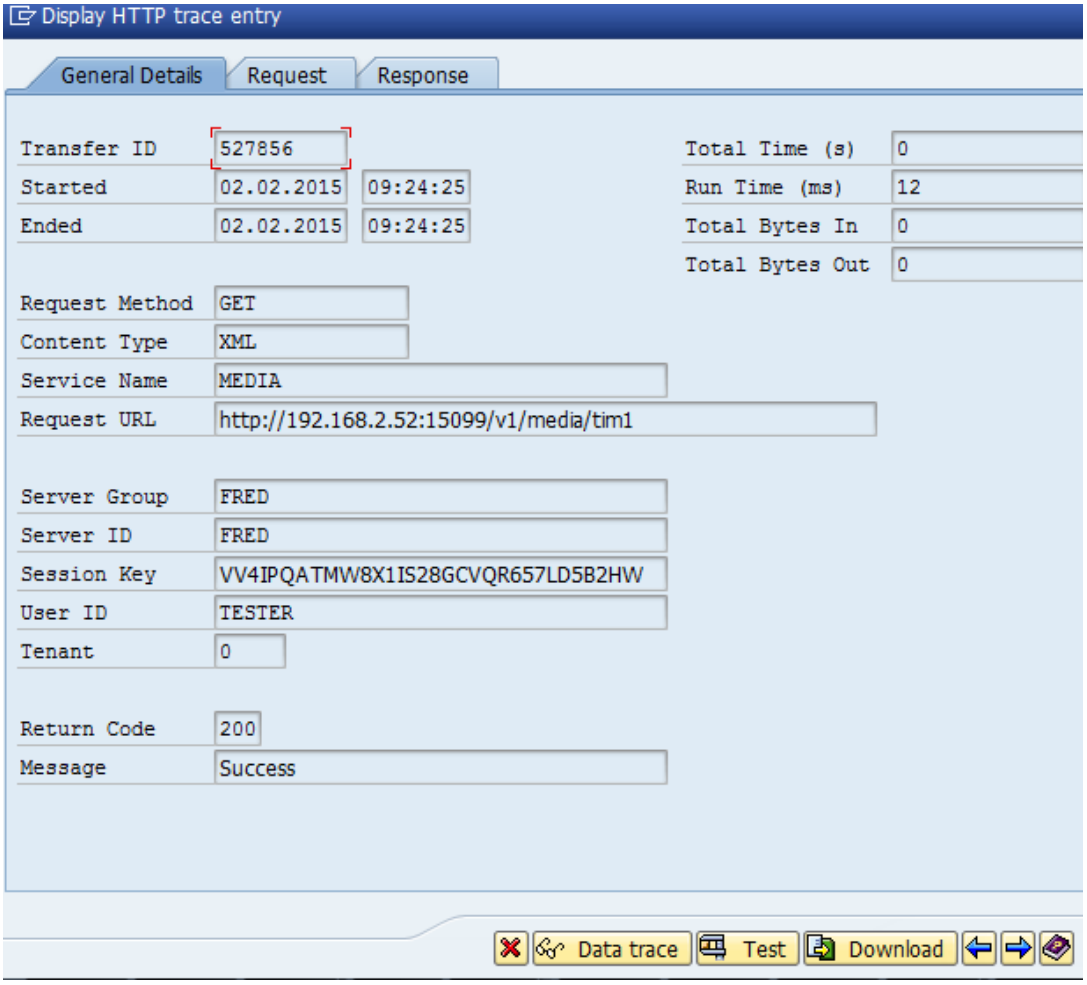
Click the **Request** tab to view the HTTP request details.



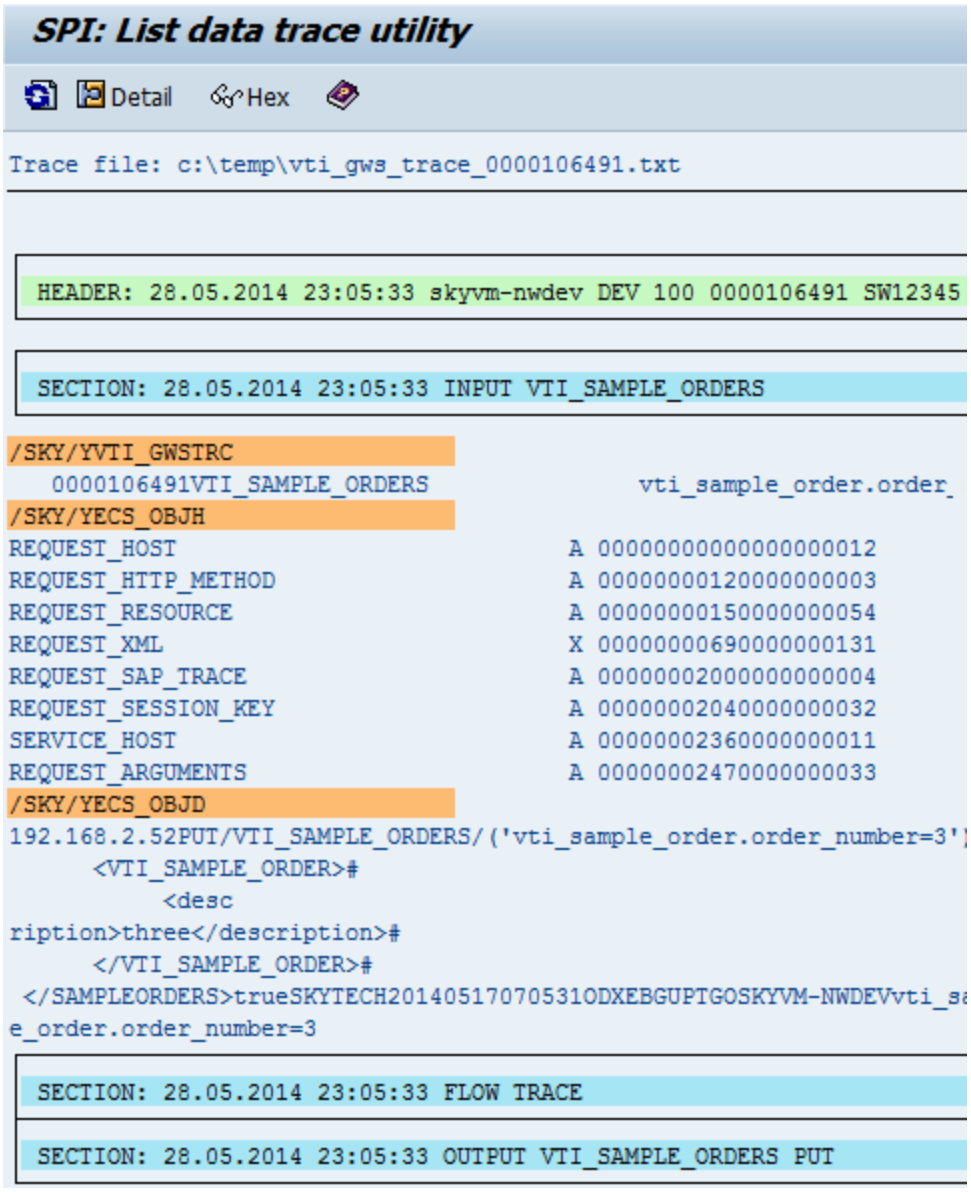
Click the **Response** tab to view the HTTP response details.



> ***Note:*** To view the media results that return a binary object in a binary file, click **Download**.



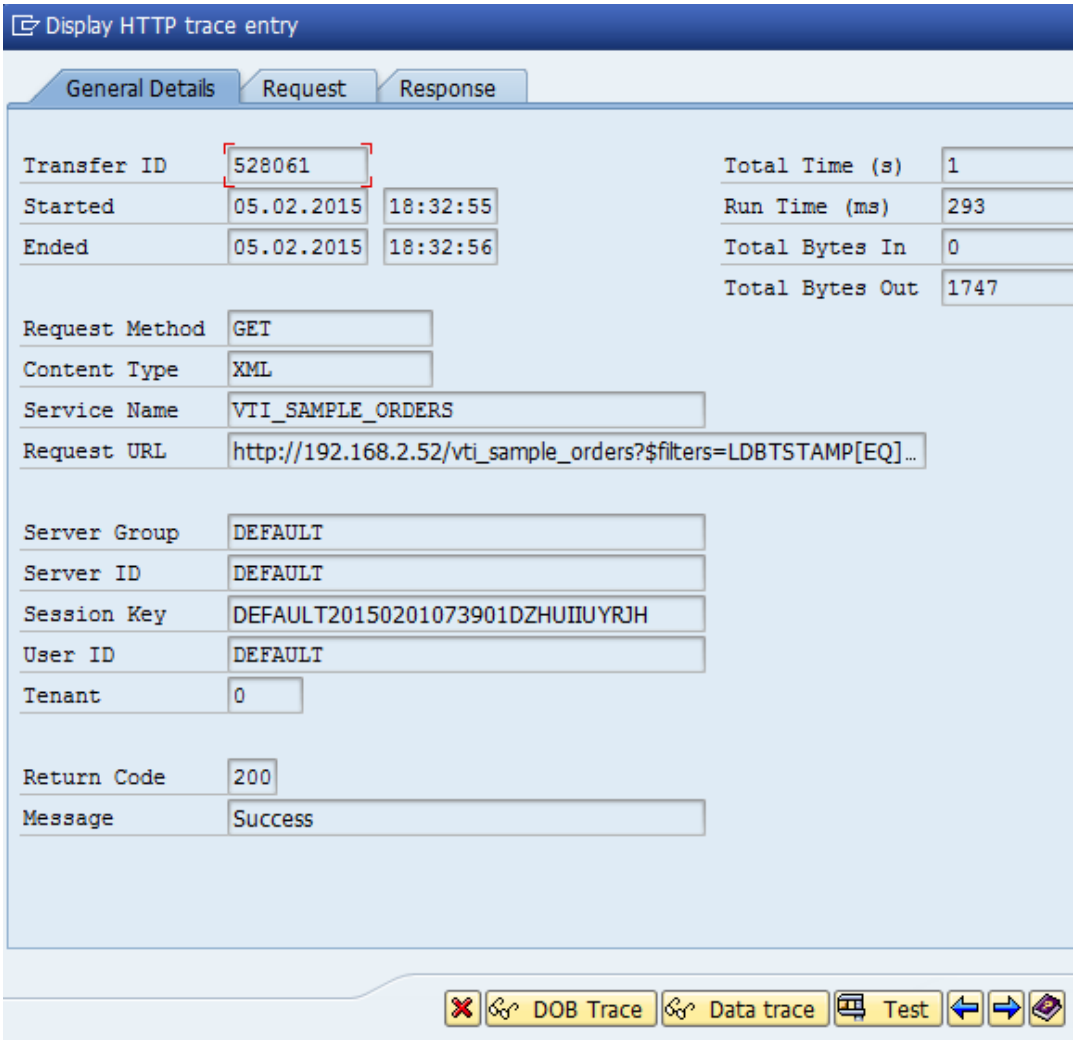
c. On each of the **General Details**, **Request** and **Response** tabs, click **Data trace** to view the recorded raw trace entries as below:



Field	Description
/SKY/YVTI_ GWSTRC	Record that contains administrative information of the Gateway HTTP trace
/SKY/YECS_ OBJH	Header record that contains the complete list of HTTP request / response attributes with their corresponding offsets and lengths in the detail record
/SKY/YECS_ OBJD	Detail record that contains the actual HTTP request / response data. You can segregate the individual entities through the offset and length information from the Header record

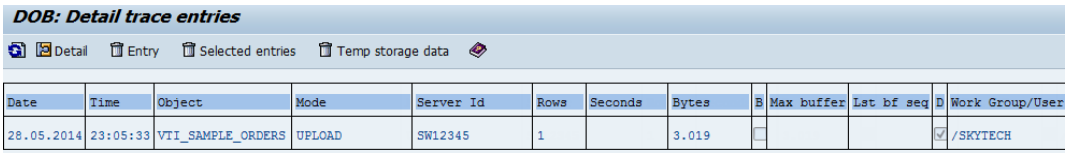
d. On each of the **General Details**, **Request** and **Response** tabs, the button **DOB trace** or **LDB Trace** may appear if the HTTP service call invoked a resource that is a data object (DOB) or and local database (LDB). Click this button to redirect to the DOB or LDB trace entries generated for this service call.

   DOB Trace:



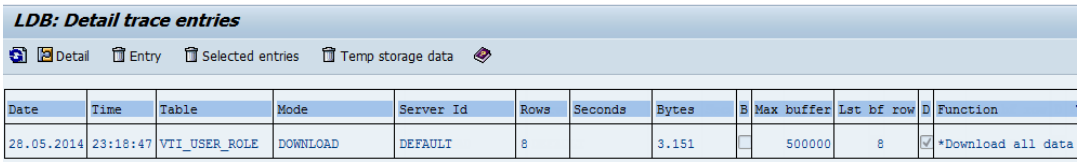
e. Click **DOB Trace**.

The **DOB: Detail trace entries** window appears.

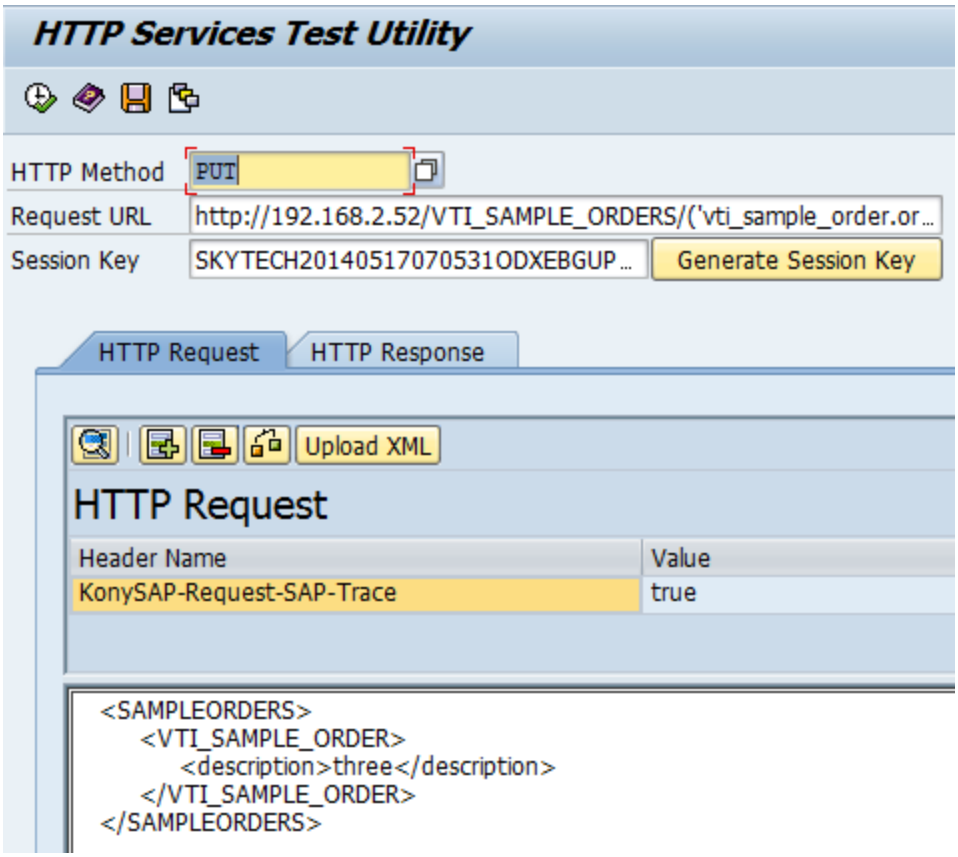


f. Click **LDB Trace**.

The **LDB: Detail trace entries** window appears.



g. On each of the **General Details**, **Request** and **Response** tabs, click **Test** to redirect to the HTTP Services Test Utility as shown below:



7.4.3.2 Summary by Date by Method View:

On the **SkyMobile: List HTTP trace entries** window, under the **Views** section, select the **Summary by date by service by method** option.

The **HTTP: Summary by date by service by method** window appears with the summary.

HTTP: Summary by date by service by method

Date	Service	Method	Total Calls	Total Errors	Total Bytes	Total Seconds	Total SAP runtime
08.04.2014	HEARTBEAT	DELETE	34	5		105	20191
08.04.2014	HEARTBEAT	GET	73			32	30997
08.04.2014	LOGOFF	DELETE	5			1	20
08.04.2014	LOGOFF	GET	7				59
08.04.2014	MEDIA	DELETE	9		1206	1	443
08.04.2014	MEDIA	GET	48	2		10	10101
08.04.2014	MEDIA	PATCH	8		1936	1	951
08.04.2014	MEDIA	POST	18	6	6334	236	234911
08.04.2014	MEDIA	PUT	20	9	7498880	1	1867
08.04.2014	METADATA	DELETE	32	32			68
08.04.2014	METADATA	GET	88	28			292
08.04.2014	PING	DELETE	67				166
08.04.2014	PING	GET	67				209
08.04.2014	PING	PATCH	67				154
08.04.2014	PING	POST	67				155
08.04.2014	PING	PUT	67				149

7.4.3.3 Summary by date by caller View:

On the **SkyMobile: List HTTP trace entries** window, under the **Views** section, select the **Summary by date by caller** option.

The **HTTP: Summary by date by caller** window appears with the summary.

HTTP:: Summary by date by caller by service

Date	Server ID	Service	Total Calls	Total Errors	Total Bytes	Total Seconds	Total SAP runtime
23.04.2014		PING	5		1561		18
23.04.2014	FRED	HEARTBEAT	11	2	326		310
23.04.2014	FRED	LOGOFF	2	1	127		8
23.04.2014	FRED	MEDIA	14	6	53048		216
23.04.2014	FRED	METADATA	5	2	12383		19
23.04.2014	FRED	PING	5		1776		25
23.04.2014	SHUTEST	METADATA	2		9206		11
23.04.2014	SHUTEST3	ORDERS	1		604		25
23.04.2014	SW12345	GETORDERLIST2	11	3	66679	210	207581
23.04.2014	SW12345	METADATA	5	1	26738		55

> ***Note:*** If you double-click any row in the **Summary by date by method** view or **Summary by date by caller** view, the **Detail** view for that row appears.

8. Mind Over Mobile

The section provides information on the Mind Over Mobile (MOM) components. MOM provides a range of functionality to assist with running and optimizing your mobile infrastructure.

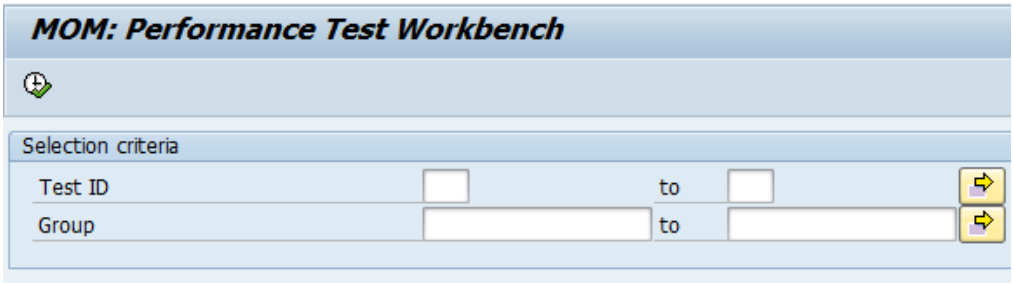
8.1 Key Topics

- Performance Test Workbench
- Performance Test Results Export Utility
- Installing the MOM Utility
- Running a Performance Test

8.2 Performance Test Workbench

The Performance Test Workbench is a comprehensive tool for the maintenance of test definitions, threads and their corresponding events. It also allows you to define instances for the test definitions. You can access this utility through the transaction **/SKY/YMOMW**.

The initial screen provides input filters for the Test ID and Group.

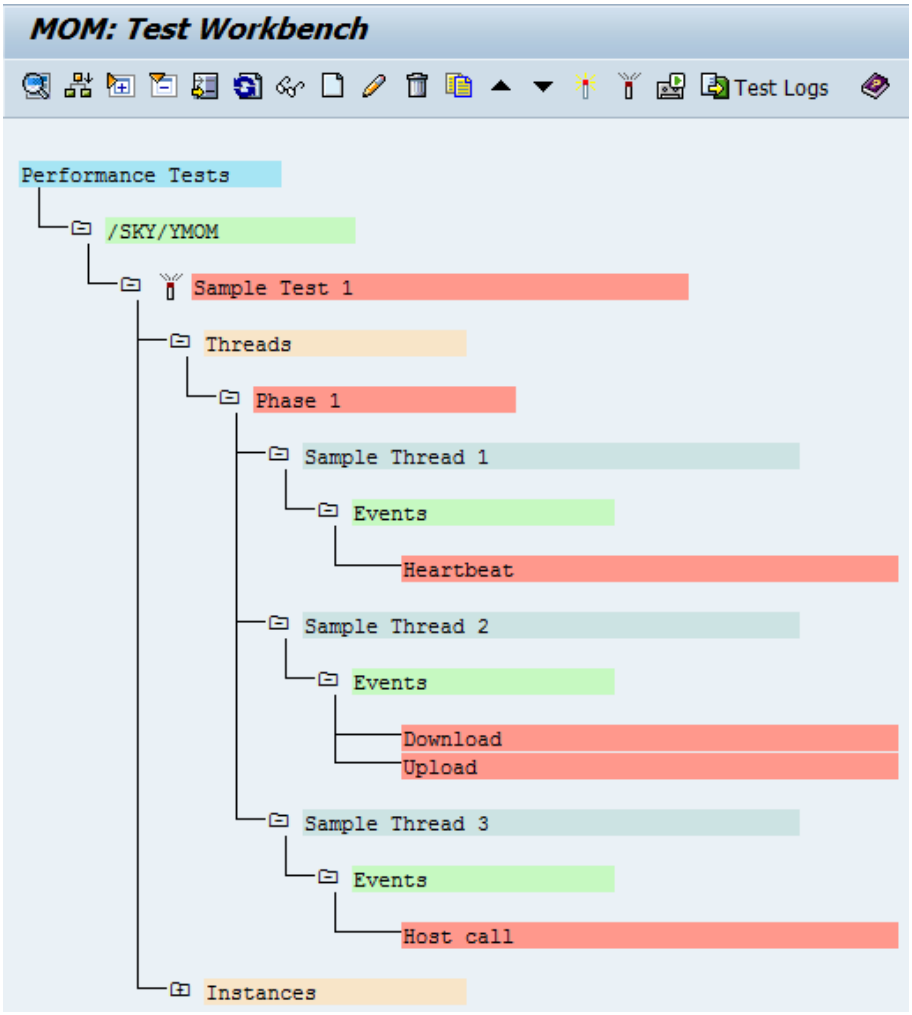


The result is a tree structure with the nodes arranged according to the following hierarchy.

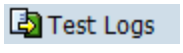
- A Test contains Threads and Instances. Tests are arranged based on the group they belong to.
- A Thread is linked to a Test and contains Events. Threads are arranged based on the Phase they belong to.

- An Event is linked to a Thread. Events within a Thread are arranged based on their Event Sequence.
- An Instance is linked to a Test and contains auto generated User IDs.

The following example is a typical hierarchy for a test definition.



Field	Field Name	Description
	Refresh Display (F5)	Refreshes / Updates Hierarchy display

Field	Field Name	Description
	Display (Ctrl +F3)	Displays details of the selected component
	Create (Shift+F2)	Creates a new component (Test / Thread / Event / Instance / User)
	Change (Shift+F4)	Edits the selected component
	Change (Shift+F5)	Deletes the selected component
	Copy (Shift+F6)	Copies the test
	Move up (Ctrl+Shift+F5)	Moves the selected event up
	Move down (Ctrl+Shift+F5)	Moves the selected event down
	Activate Definition (Ctrl +F5)	Activates the Test
	Deactivate Definition (Ctrl +F6)	Deactivates the Test
	Generate Users (Ctrl+F1)	Generates the Users for the Selected Instance or for all the instances under a Test
	Test Recorder (Ctrl+F11)	Navigates to MOM: Test Recorder tool
	Test Logs (Ctrl+F2)	Navigates to MOM: Performance Test Results Export Utility

Field	Field Name	Description
	Import Users	Imports users to the selected Instance from a CSV File

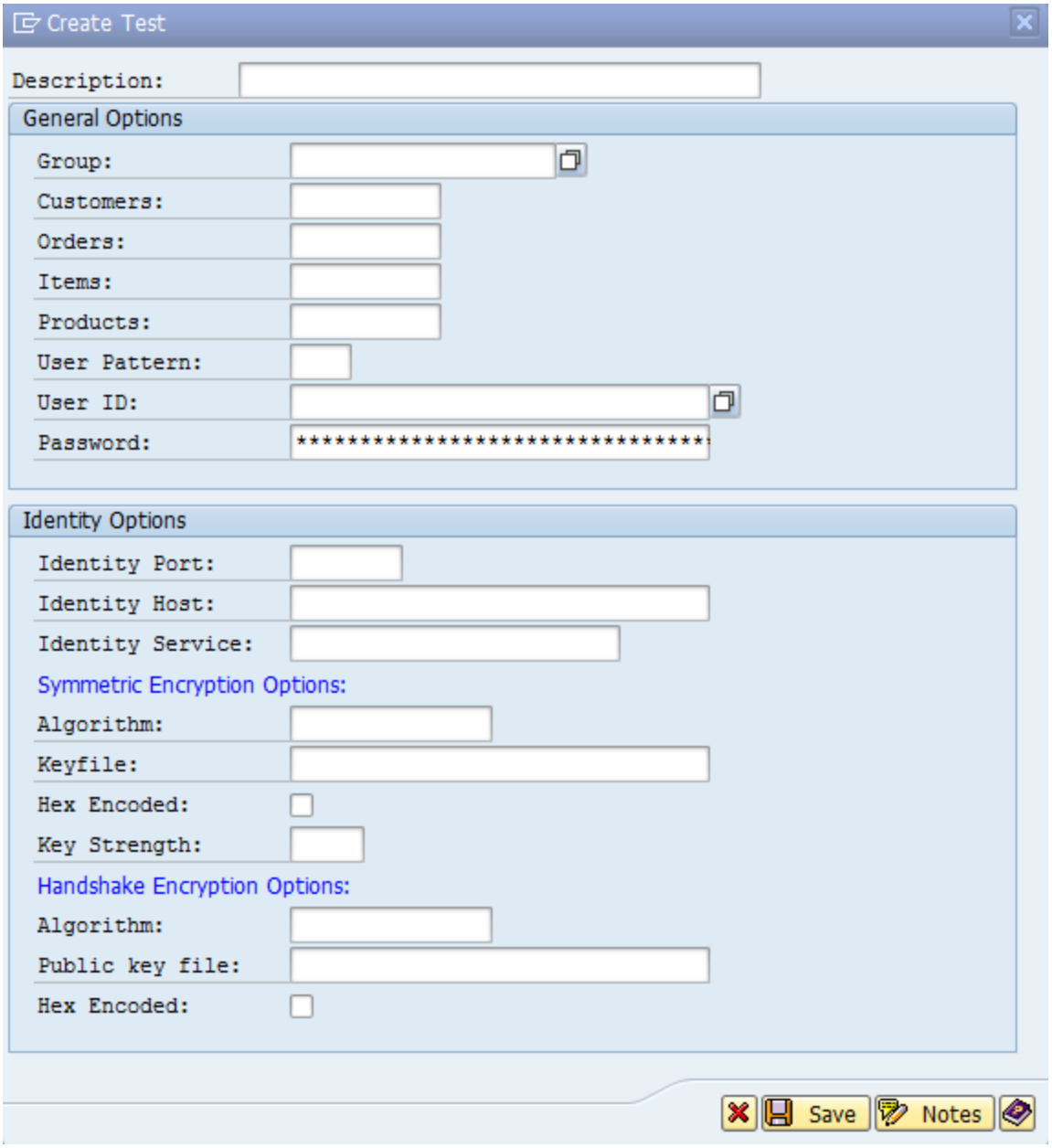
The following sections describe the maintenance of each of the above components:

- Define Test
- Define Thread
- Define Event
- Define Instance

You can export any of the components created through this utility to an external file using the Performance Test Results Export Utility.

8.2.1 Define Test

To create a test definition in the Performance Test Workbench (transaction **/SKY/YMOMW**), place the cursor on the top level node **Performance Tests** and click the **Create** button on the application toolbar.
The **Create Test** dialog box appears:



Field	Description
Description	Brief description of the test definition

Field	Description
Group	Server group to which you should assign the test. Once you save a test, it appears under the corresponding group in the hierarchy. If no group is selected, then the test appears under No Group
Customers	Number of sample customer IDs to generate for the test. These customer IDs are randomly assigned to the sample orders generated by this utility
Orders	Number of sample orders to generate for the test
Items	Number of items to generate for each order in the test
Products	Number of product IDs to generate for the test. For each order items, this utility generates a product ID is randomly assigned from this pool
User Pattern	Determines the pattern to prefix to the generated user IDs for the test
User ID	User ID to access the test
Password	Password to access the test
Identity Port	Port Number for the Identity Management For more details click here.
Identity Host	Host address for Identity Management For more details click here.
Identity Service	Service Name for Identity Management (for Example: SSO)

Field	Description
Symmetric Encryption Options: Algorithm	The algorithm to use For more details click here.
Symmetric Encryption Options: Key File	The key file to use. You need to generate this file using the Kony for SAP EKG utility and import into Kony for SAP Binary Repository. For more details click here.
Symmetric Encryption Options: Hex Encoded	Whether the key file is hex encoded or not. The key strength to use (in bits) For more details click here.
Symmetric Encryption Options: Key Strength	The key strength to use (in bits) For more details click here.
Handshake Encryption Options: Algorithm	The algorithm to use For more details click here.
Handshake Encryption Options: Public Key File	The key file to use. You need to generate this file using the Kony for SAP EKG utility and import into Kony for SAP Binary Repository For more details click here.
Handshake Encryption Options: Hex Encoded	Whether the key file is hex encoded or not. For more details click here.

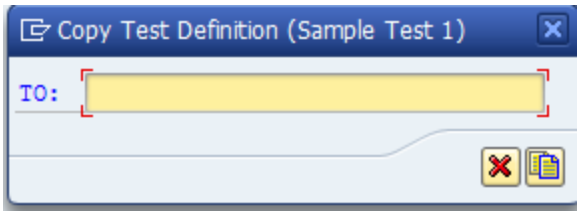
Field	Description
Notes	This button generates a pop-up box where you can maintain the long text for the test

When you save the test definition, it appears in deactivated mode. You can now create threads and instances for this test. After you completely define the test hierarchy, you can activate the test definition using the **Activate Definition** button on the application toolbar. Activating a test internally generates sample data for the customers, orders, items and products the number of which depends on the values provided earlier in your test definition.

You can change the test definition by double-clicking the test or by clicking the test and pressing the button on the application toolbar.

You can delete a test definition by clicking the test and pressing the button on the application toolbar. This also deletes any threads, events and instances that are defined under this test.

You can copy a test definition to a new test by clicking the test and pressing the **Copy** button on the application toolbar. A dialog box appears, and you can specify a new description.

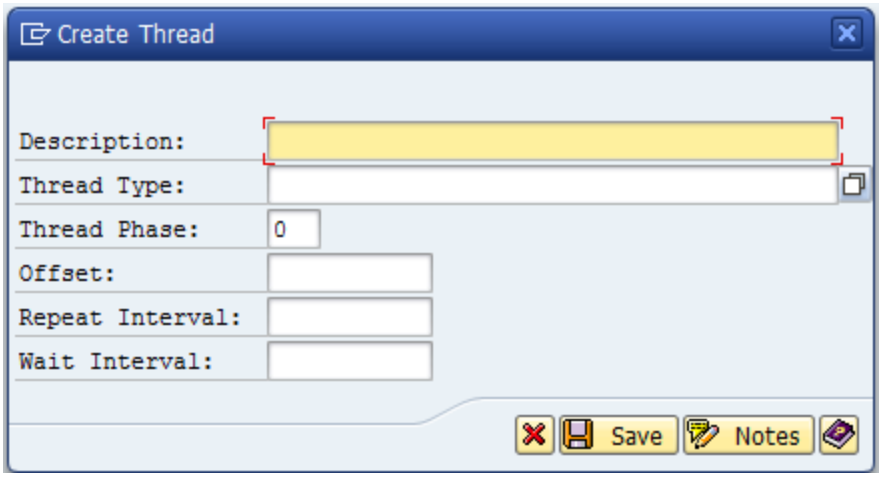


Click the **Copy** button on the above screen, to copy the complete hierarchy of the source test definition to the new test.

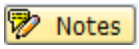
> ***Important:*** You cannot modify the activated tests. To change an existing definition, first deactivate the test.

8.2.2 Define Thread

You can create a thread definition by positioning your cursor on the **Threads** node of any deactivated test definition and clicking the Create the **Create** button on the application toolbar. The **Create Thread** dialog box appears:



Field	Description
Description	Brief description of the thread
Thread Type	Designates the type of the thread. The following thread types are currently configured in the system.
Phase	Used to group threads within a test definition. This is configure to *0* by default
Offset	This value determines the time interval after which the first event in the thread starts
Repeat Interval	For Heartbeat threads, this value determines the number of events to trigger
Wait Interval	For Heartbeat threads, this value determines the average wait time between any two events

Field	Description
	This button generates a pop-up box where you can maintain the long text for the thread

The following thread types are currently available:

Thread Type	Description
HOST	Host Thread
LOCAL	Local Thread

After a thread is successfully saved, it appears on the hierarchy under its specified phase. You can now create events for this thread.

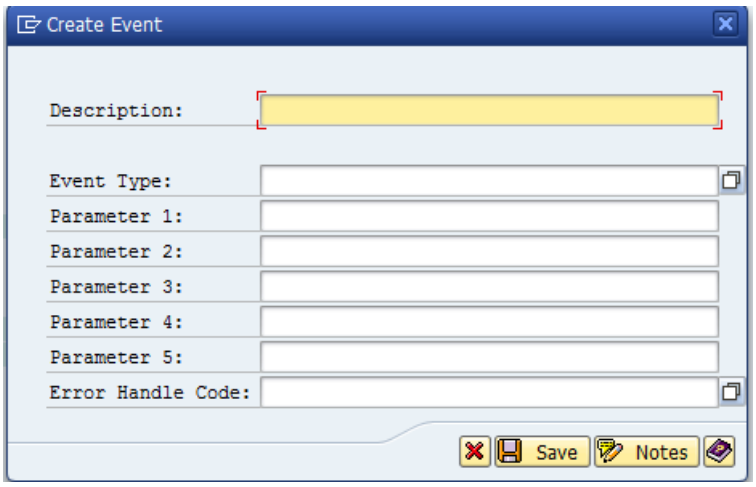
You can change the thread definition (of a deactivated test) by double-clicking the thread or by clicking the thread and clicking the **Change** button on the application toolbar.

You can delete a thread (of a deactivated test) by clicking the thread and pressing the **Delete** button on the application toolbar. This also deletes any events that are defined under this thread.

> *Important:* Specify the same phase number in related threads for easier management.

8.2.3 Define Event

To create an event definition, place the cursor on the Events node of any thread and click the **Create** button on the application toolbar.
The **Create Event** dialog box appears:



Field	Description
Description	Brief description of the event
Event Type	Designates the type of the event. The following event types are currently configured in the system.
Parameter 1..5	User parameters to pass to the Event Handler as mentioned in this table, **Event Table** row > **Parameters to pass to the Event Handler** column
Error Handle Code	Designates the default Error Handling mechanism for the event
Notes	This button generates a pop-up box where long text for the event can be maintained

Event Type	Description	Parameters to Pass to the Event Handler	
		Parameter No.	Parameters Description
01	Wait	NA	NA
02	Heartbeat	NA	NA
03	DOB Download	1 (Mandatory)	No. of Records
		2 (Optional)	Burn Time (Time taken for all the records to process)
04	DOB Refresh	1 (Mandatory)	No. of Records
		2 (Optional)	Burn Time (Time taken for all the records to process)
05	DOB Upload	1 (Mandatory)	No. of Records
		2 (Optional)	Burn Time (Time taken for all the records to process)
06	SAP Real time call	1 (Mandatory)	Wait Time
		2 (Mandatory)	Burn Time (Time taken for all the records to process)
		3 (Optional)	No. of records

After an event is successfully saved, it appears on the hierarchy under its corresponding thread.

To change the event definition (of a deactivated test) double-click an event or click the event and press the **Change** button on the application toolbar.

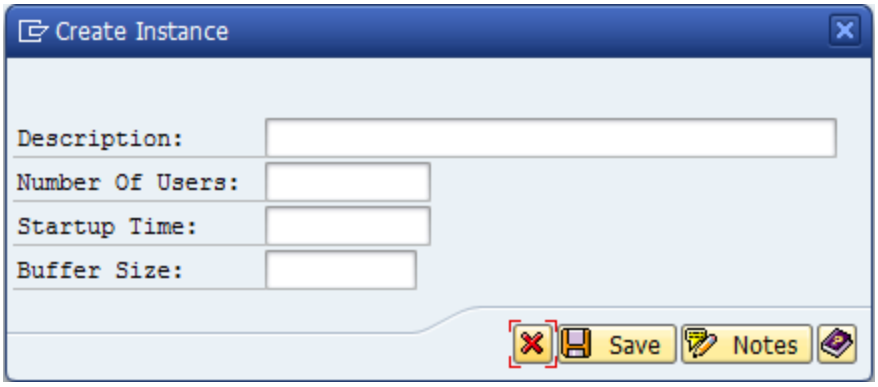
To delete an event (of a deactivated test), click the event and press the **Delete** button on the application toolbar.

The events under a thread appear, by default in the order of creation. However you can change the sequence of events within a thread by clicking any event and using the **Move up** and **Move down** buttons on the application toolbar.

> *Important:* For any thread, use event sequencing to change the order of event execution using the **Move up** and **Move down** buttons.

8.2.4 Define Instance

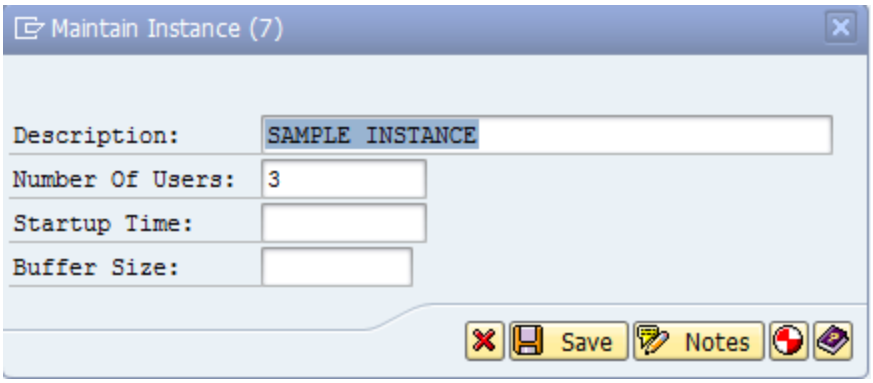
To create a test instance, place the cursor on the **Instances** node of any deactivated test definition and click the **Create** button on the application toolbar. The **Create Instance** dialog box appears:



Field	Description
Description	Brief description of the test instance
Number of Users	Number of sample user IDs to generate for the test instance

Field	Description
Startup Time	TBD
Buffer Size	TBD
Repeat Interval	For Heartbeat threads, this value determines the number of events to trigger
Notes	This button generates a pop-up box where you can maintain long text for the test instance

After an instance is successfully saved, it appears on the hierarchy under the corresponding test. You can change the instance definition (of a deactivated test) by double-clicking the instance or by clicking the instance and pressing the **Change** button on the application toolbar. In this mode, there is an additional option of generating sample user IDs based on the "Number of Users" specified in the instance.



Field	Description
	Auto generates sample user IDs based on the value specified in the Number of Users field. Each ID is a concatenation of the User Pattern, Instance ID and the number of the user in the current instance.

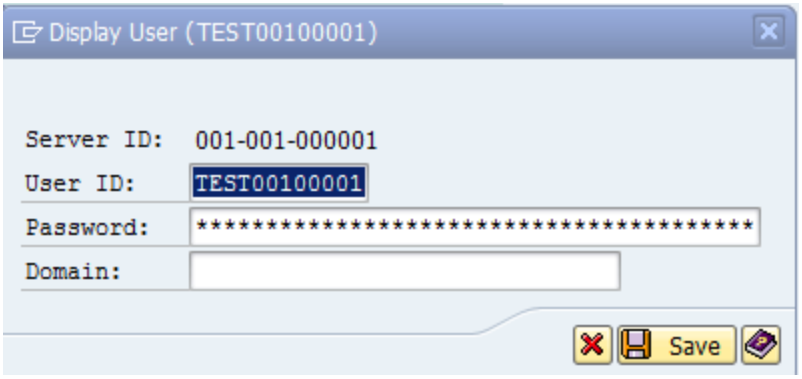
You can delete an instance (of a deactivated test) by clicking the instance and pressing the **Delete** button on the application toolbar. This also deletes the user IDs generated earlier for this instance.

You can import users for an instance from a CSV file (even for an activated test). Click the instance and click the **Import Users** button the application toolbar.

> ***Important:*** Activate the test after you save the instance.
>
> To generate the users for the instance, click the Instance and click the **Generate Users (Ctrl+F1)** button on the application toolbar.
>
> To generate the users for all the instances under a test, click the Test and click the **Generate Users (Ctrl+F1)** button on the application toolbar.

8.2.4.1 Maintain Users

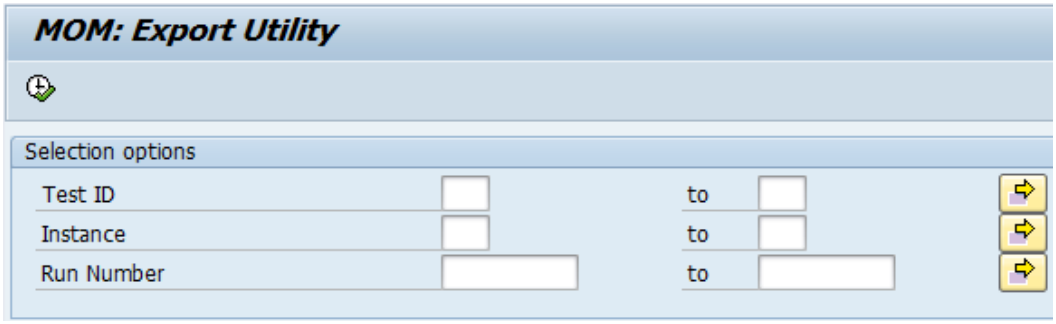
You can modify the auto generated user IDs and the passwords for the same. You can double-click any user ID in the hierarchy and maintain an alternative ID and / or a password and / or a domain.



> ***Important:*** If you change the Number of Users on an instance but forget to (re)generate the user IDs, auto generation of user IDs still happens when you activate the test.
>
> You may modify the User Details even after you activate the test.

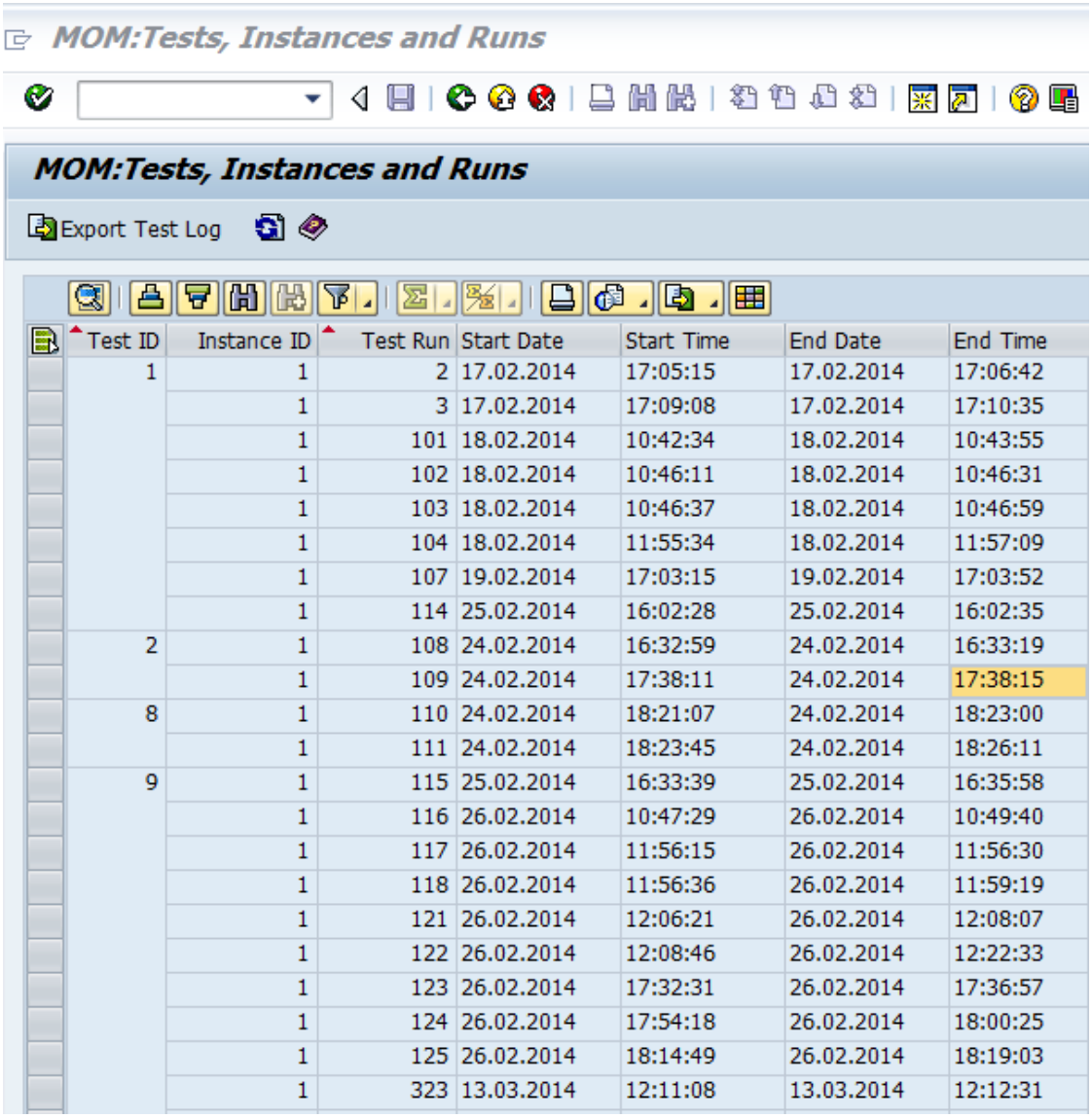
8.2.5 Performance Test Results Export Utility

The Performance Test Results Export Utility is a comprehensive tool to export test definitions, instances, test runs and their corresponding test logs, **User Load Graph Report** and **Response Time Graph Report** to an external file. You can access export test reports through transaction **/SKY/YMOME**. The initial screen provides input filters for Test ID, Instance and Run Number.



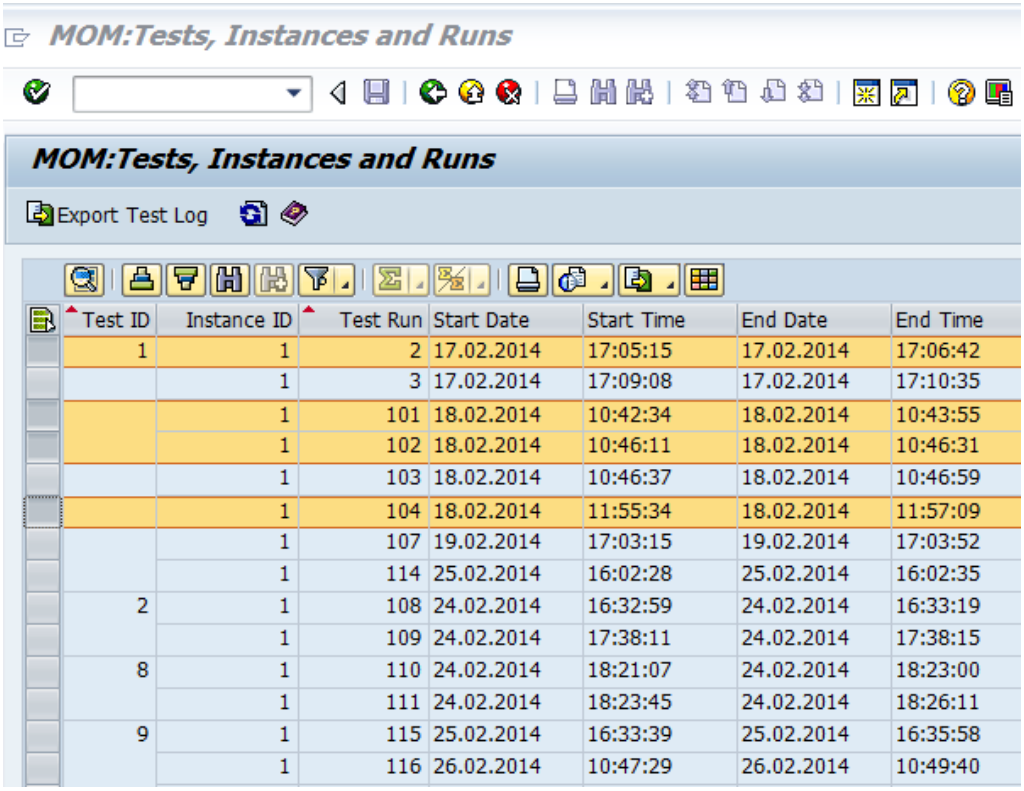
Run Summary and Log

The summary of the Test(s), its Instances and its Test Runs appear in a grid display, as in the below example:



Test ID	Instance ID	Test Run	Start Date	Start Time	End Date	End Time
1	1	2	17.02.2014	17:05:15	17.02.2014	17:06:42
	1	3	17.02.2014	17:09:08	17.02.2014	17:10:35
	1	101	18.02.2014	10:42:34	18.02.2014	10:43:55
	1	102	18.02.2014	10:46:11	18.02.2014	10:46:31
	1	103	18.02.2014	10:46:37	18.02.2014	10:46:59
	1	104	18.02.2014	11:55:34	18.02.2014	11:57:09
	1	107	19.02.2014	17:03:15	19.02.2014	17:03:52
	1	114	25.02.2014	16:02:28	25.02.2014	16:02:35
2	1	108	24.02.2014	16:32:59	24.02.2014	16:33:19
	1	109	24.02.2014	17:38:11	24.02.2014	17:38:15
8	1	110	24.02.2014	18:21:07	24.02.2014	18:23:00
	1	111	24.02.2014	18:23:45	24.02.2014	18:26:11
9	1	115	25.02.2014	16:33:39	25.02.2014	16:35:58
	1	116	26.02.2014	10:47:29	26.02.2014	10:49:40
	1	117	26.02.2014	11:56:15	26.02.2014	11:56:30
	1	118	26.02.2014	11:56:36	26.02.2014	11:59:19
	1	121	26.02.2014	12:06:21	26.02.2014	12:08:07
	1	122	26.02.2014	12:08:46	26.02.2014	12:22:33
	1	123	26.02.2014	17:32:31	26.02.2014	17:36:57
	1	124	26.02.2014	17:54:18	26.02.2014	18:00:25
	1	125	26.02.2014	18:14:49	26.02.2014	18:19:03
	1	323	13.03.2014	12:11:08	13.03.2014	12:12:31

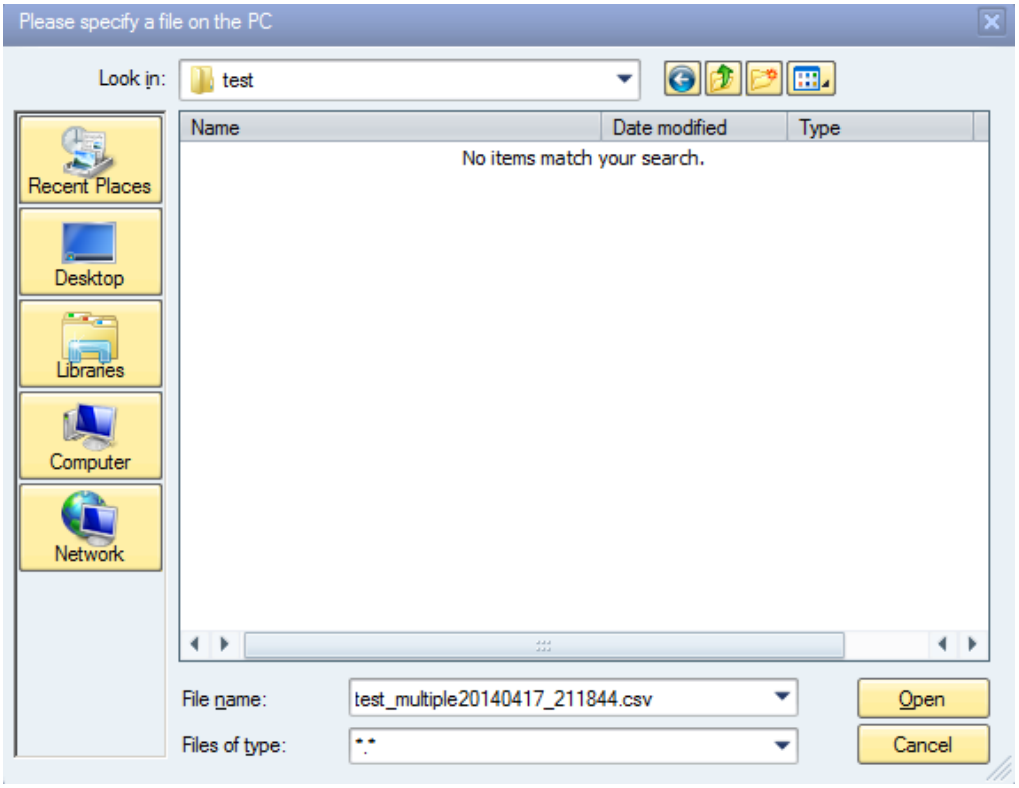
You can export the test runs to an external file by selecting multiple Tests (rows in the grid view).



Test ID	Instance ID	Test Run	Start Date	Start Time	End Date	End Time
1	1	2	17.02.2014	17:05:15	17.02.2014	17:06:42
	1	3	17.02.2014	17:09:08	17.02.2014	17:10:35
	1	101	18.02.2014	10:42:34	18.02.2014	10:43:55
	1	102	18.02.2014	10:46:11	18.02.2014	10:46:31
	1	103	18.02.2014	10:46:37	18.02.2014	10:46:59
	1	104	18.02.2014	11:55:34	18.02.2014	11:57:09
	1	107	19.02.2014	17:03:15	19.02.2014	17:03:52
	1	114	25.02.2014	16:02:28	25.02.2014	16:02:35
2	1	108	24.02.2014	16:32:59	24.02.2014	16:33:19
	1	109	24.02.2014	17:38:11	24.02.2014	17:38:15
8	1	110	24.02.2014	18:21:07	24.02.2014	18:23:00
	1	111	24.02.2014	18:23:45	24.02.2014	18:26:11
9	1	115	25.02.2014	16:33:39	25.02.2014	16:35:58
	1	116	26.02.2014	10:47:29	26.02.2014	10:49:40

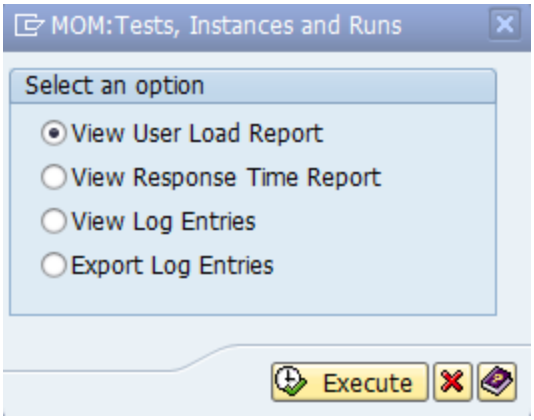
Click the **Export Test Log** button on the application toolbar.

The **Please specify a file on the PC dialog** appears.



Click **Open** to export the test runs to a MS Excel comma separated file (`.csv`).

Double-click a test run on **MOM:Tests, Instances and Runs** window.
The **MOM:Tests, Instances and Runs** dialog appears with four options:



- View User Load Report
- View Response Time Report

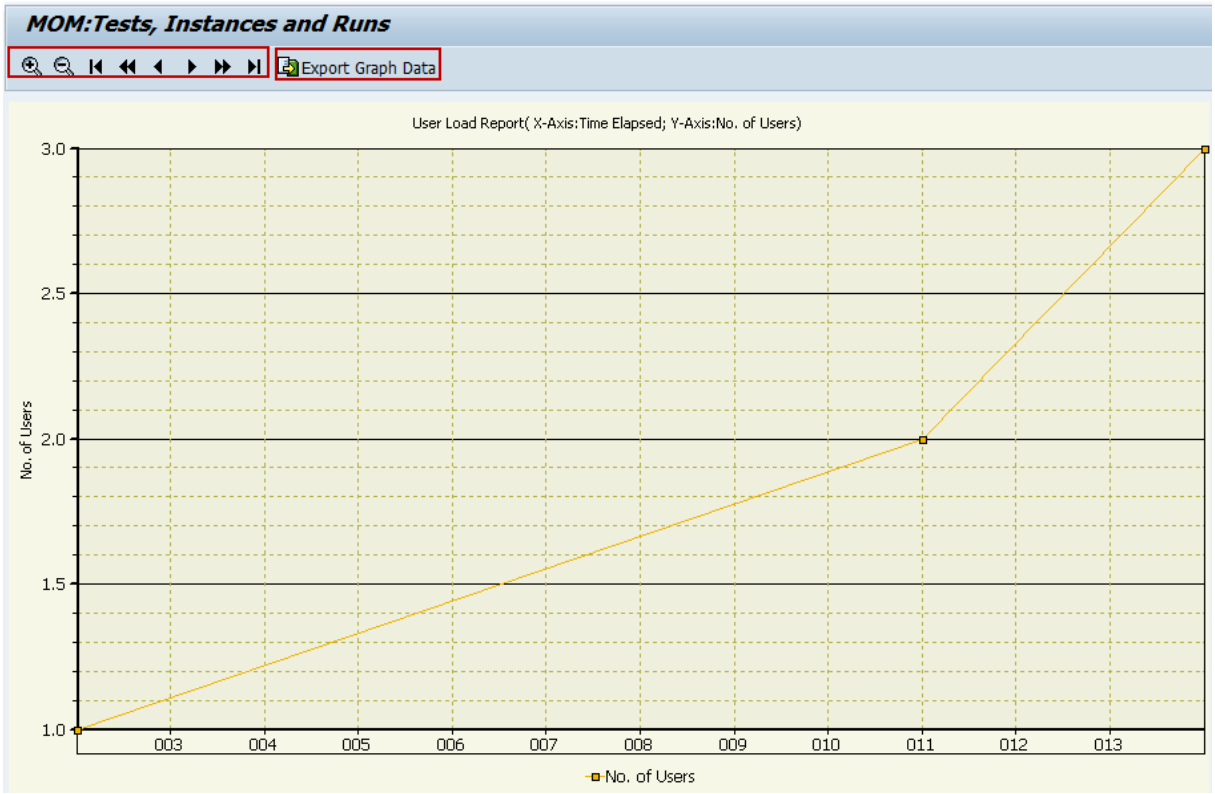
- View Log Entries
- Export Log Entries

8.2.5.1 View User Load Report

The User Load Report graph plots the number of connected users against the time elapsed. This graph depicts specific instances when a user connection fails.

To view the User Load Report graph of a particular test run,on the **MOM:Tests, Instances and Runs** dialog, select the **View User Load Report** option and click **Execute**.

The following graph appears with the **Export Graph Data** button to export the graph data to a MS Excel comma separated `.csv` file on your desktop and other buttons for the operations on the graph.



You may use the buttons for operations on the graph, when the graph data is huge and overflows to the next pages.

The following table explains the usage of the buttons:

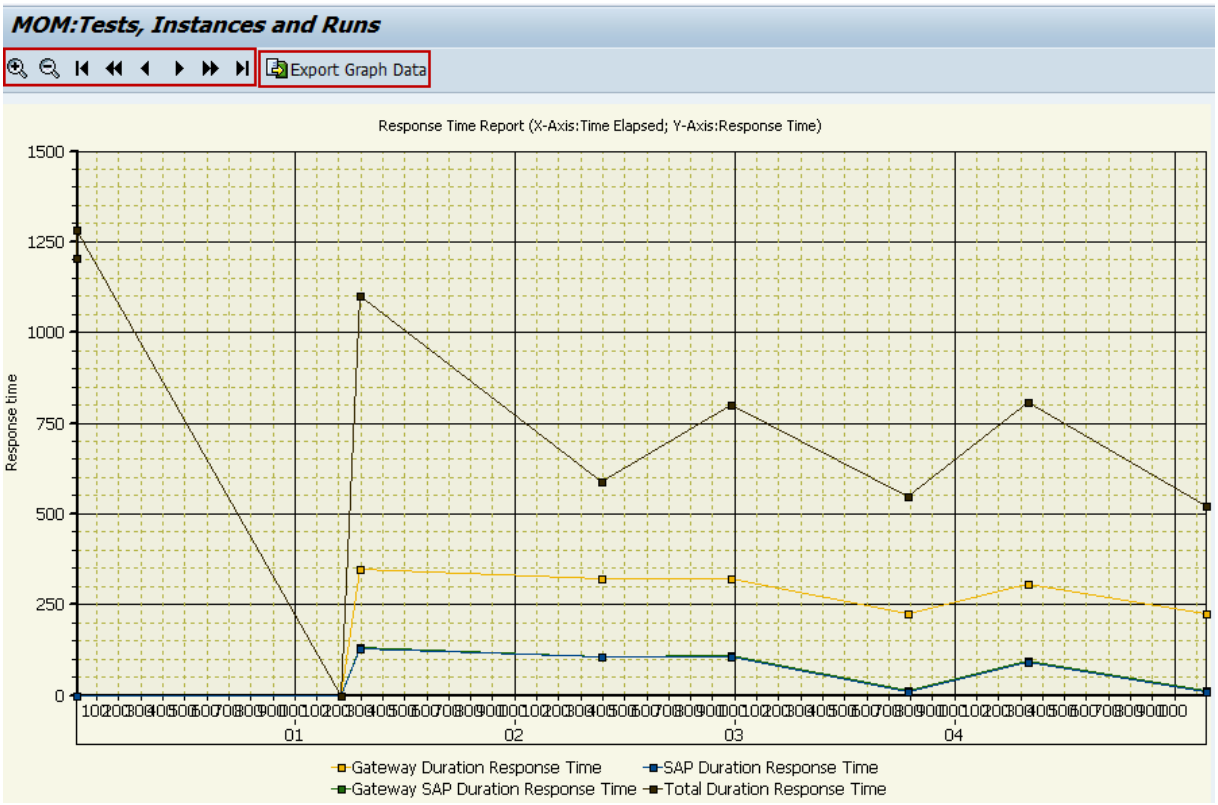
Field	Description
	Zooms in the graph
	Zooms out the graph
	Click the button to show the starting section of the graph
	Click the button to move the graph one page towards left.
	Click the button to scroll the graph towards left.
	Click the button to scroll the graph towards right.
	Click the button to scroll the graph towards right.
	Click the button to move the graph one page towards right.
	Click the button to show the end section of the graph

8.2.5.2 View Response Time Report

The Response Time Report graph plots the time elapsed against various response times like gateway duration response time, SAP duration response time, gateway SAP duration response time, and total duration response time.

To view the Response Time Report graph of a particular test run, on the **MOM:Tests, Instances and Runs** dialog, select the **View Response Time Report** option and click the **Execute**.

A graph appears with the **Export Graph Data** button on the application tool bar. The **Export Graph Data** button allows to export the graph data to a MS Excel comma separated file **.** `csv` file on your desktop. The other buttons also appear on the application tool bar to perform the operations on the graph.



You may use the buttons for operations on the graph, when the graph data is huge and overflows to the next pages.

The following table explains the usage of the buttons:

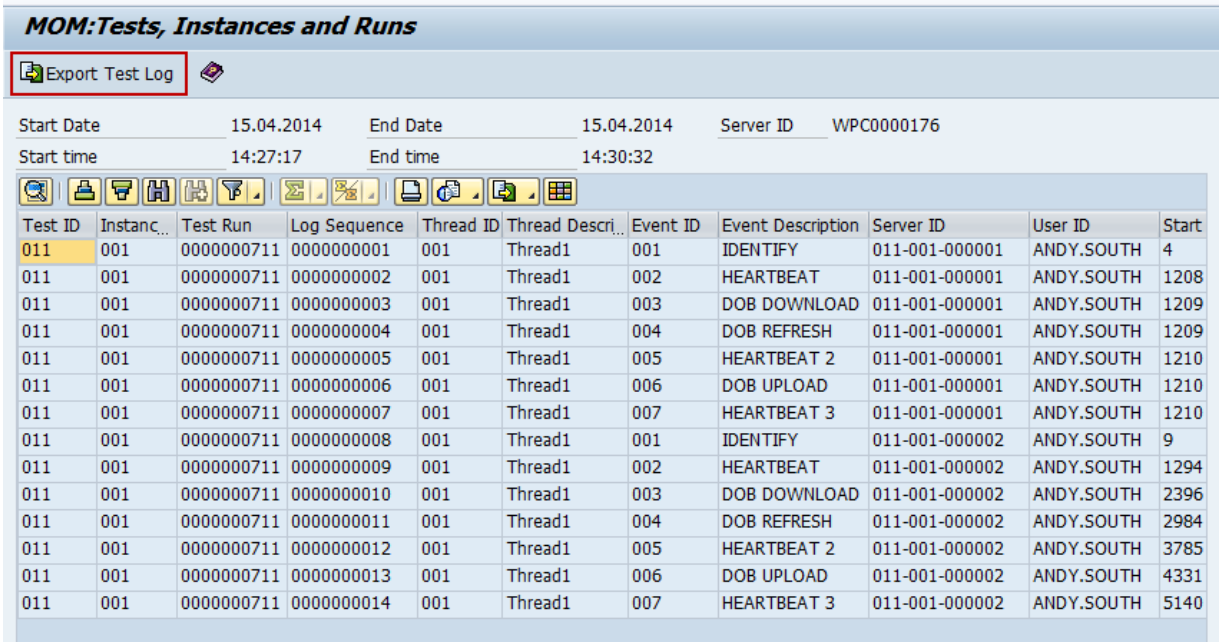
Field	Description
	Zooms in the graph
	Zooms out the graph
	Click the button to show the starting section of the graph
	Click the button to move the graph one page towards left.

Field	Description
	Click the button to scroll the graph towards left.
	Click the button to scroll the graph towards right.
	Click the button to move the graph one page towards right.
	Click the button to show the end section of the graph

8.2.5.3 View Log Entries

The Log shows data for fields Thread, Event, Server, Start Offset, Total Duration, Event Return Code and Number of entries for that particular test run.

To view the log entries of the test, on the **MOM:Tests, Instances and Runs** dialog, select the **View log entries** option and click **Execute**.
A **MOM:Tests, Instances and Runs** dialog appears.

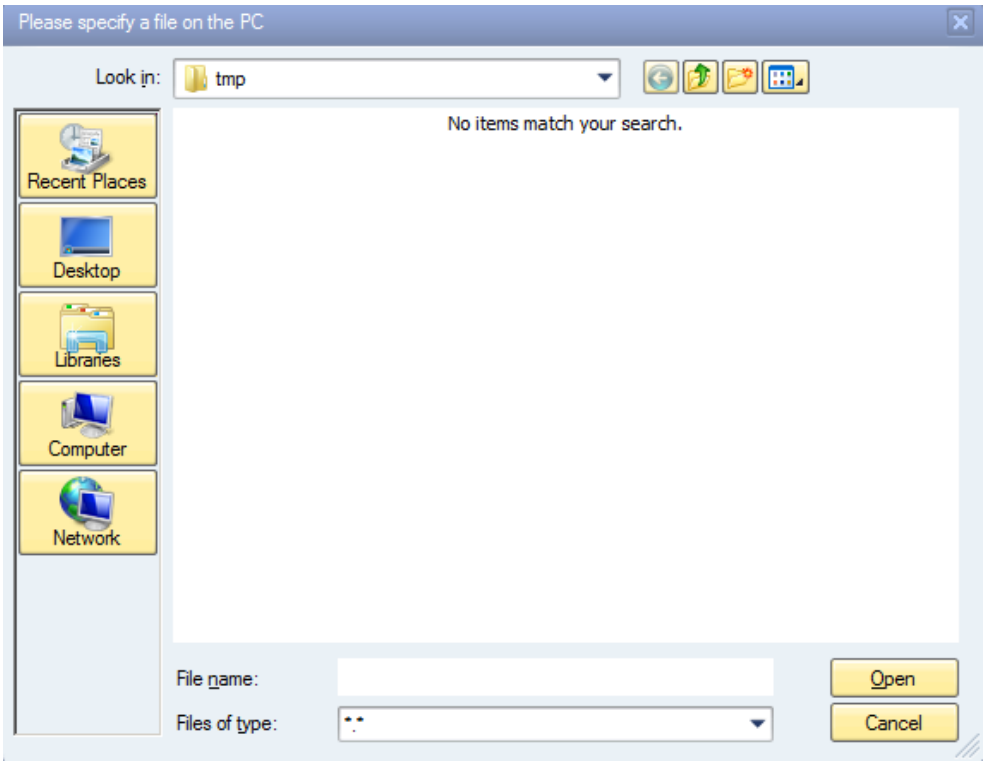


Test ID	Instanc...	Test Run	Log Sequence	Thread ID	Thread Descri...	Event ID	Event Description	Server ID	User ID	Start
011	001	0000000711	0000000001	001	Thread1	001	IDENTIFY	011-001-000001	ANDY.SOUTH	4
011	001	0000000711	0000000002	001	Thread1	002	HEARTBEAT	011-001-000001	ANDY.SOUTH	1208
011	001	0000000711	0000000003	001	Thread1	003	DOB DOWNLOAD	011-001-000001	ANDY.SOUTH	1209
011	001	0000000711	0000000004	001	Thread1	004	DOB REFRESH	011-001-000001	ANDY.SOUTH	1209
011	001	0000000711	0000000005	001	Thread1	005	HEARTBEAT 2	011-001-000001	ANDY.SOUTH	1210
011	001	0000000711	0000000006	001	Thread1	006	DOB UPLOAD	011-001-000001	ANDY.SOUTH	1210
011	001	0000000711	0000000007	001	Thread1	007	HEARTBEAT 3	011-001-000001	ANDY.SOUTH	1210
011	001	0000000711	0000000008	001	Thread1	001	IDENTIFY	011-001-000002	ANDY.SOUTH	9
011	001	0000000711	0000000009	001	Thread1	002	HEARTBEAT	011-001-000002	ANDY.SOUTH	1294
011	001	0000000711	0000000010	001	Thread1	003	DOB DOWNLOAD	011-001-000002	ANDY.SOUTH	2396
011	001	0000000711	0000000011	001	Thread1	004	DOB REFRESH	011-001-000002	ANDY.SOUTH	2984
011	001	0000000711	0000000012	001	Thread1	005	HEARTBEAT 2	011-001-000002	ANDY.SOUTH	3785
011	001	0000000711	0000000013	001	Thread1	006	DOB UPLOAD	011-001-000002	ANDY.SOUTH	4331
011	001	0000000711	0000000014	001	Thread1	007	HEARTBEAT 3	011-001-000002	ANDY.SOUTH	5140

The **Export Test Log** button also appears that allows you to export the data to a MS Excel comma separated file . `csv` file on your desktop.

8.2.5.4 Export Log Entries

To export a test log of a particular test run to your desktop, on the **MOM:Tests, Instances and Runs** dialog, select the **Export Log Entries** option and click **Execute**.
The **Please specify a file on the PC** dialog appears.

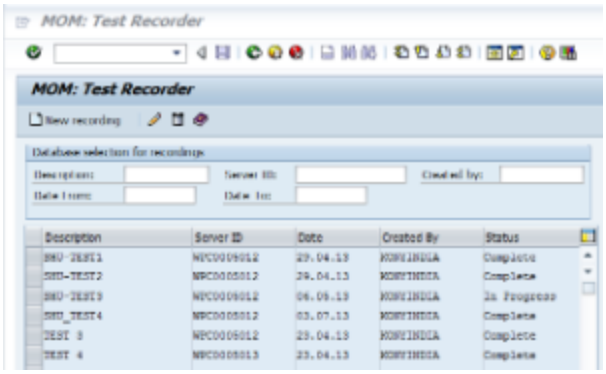


Click **Open** to export the test run to a MS Excel comma separated file . `csv` file on your desktop.
The below graph appears with an **Export Graph Data** button and other buttons to operate on the graph. The **Export Graph** button allows you to export the graph data to a `CSV` file on your desktop.

8.3 Performance Test Recorder

The Performance Test Recorder utility enables the recording of traces generated from a specified Server ID and allows the user to automatically generate a sample test definition based on the same. This test definition is then available in the MOM workbench utility where you can further customize it according to the users needs. This utility therefore, allows the creation of Test Definitions based on real life scenarios without the user having to manually determine the individual threads and events. You can access the Performance Test Recorder through the transaction /**SKY/YMOMR**.

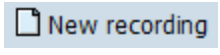
The initial screen provides filters for the Recording Description, Server ID, Recording Date Range and User.

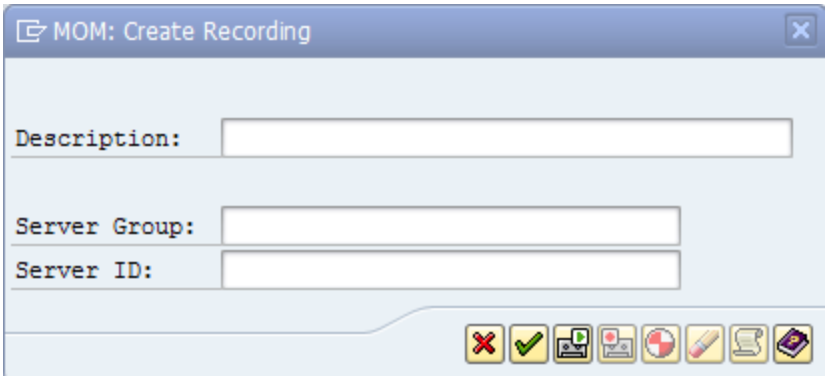


The following section describes the maintenance of recordings in further detail.

Maintain Recording

8.3.1 Maintain Recording

You can create a new recording by clicking the **New recording**  button from the initial screen of the transaction /SKY/YMOMR. The following screen appears:



Field	Description
Description	Unique description of the recording created
Server Group	Optional field specifying the server group of the device for which you can activate the trace.
Server ID	Server ID of the device for which you shall activate the trace.
	This button starts the trace recording for the specified Server ID. Once you do this, you can start executing any application from the respective device. All these actions are recorded internally as part of the trace.
	This button stops the trace recording for the specified Server ID. This button is only active when the recording is already in progress.
	This button generates a test definition for an already completed recording.
	This button clears the trace data for an already completed recording. Once this is done, you can start the recording again.
	This button displays the chronological list of events that the trace captured during the recording. The events that appear here are the basis on which the test definition is generated.

Every recording created in the Recorder is accorded a status that is visible on the initial screen of the transaction. The following statuses are assigned to a recording depending on its current stage.

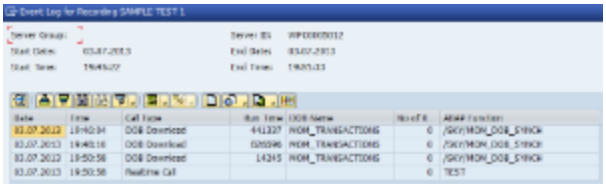
- **Pending**: This status is assigned when a recording is created but the trace is not yet started. It is also assigned when the trace of a completed recording is cleared.
- **In Progress**:This status is assigned when the recording started using  but did not yet stop. Any actions performed from the device (with the Server ID of the recording) are captured as part of the trace.
- **Complete**: This status is assigned when the recording stopped using  .

You can change an existing recording by double-clicking the corresponding line in the table or selecting the line and clicking  . Based on the status of the recording, the following actions are permitted.

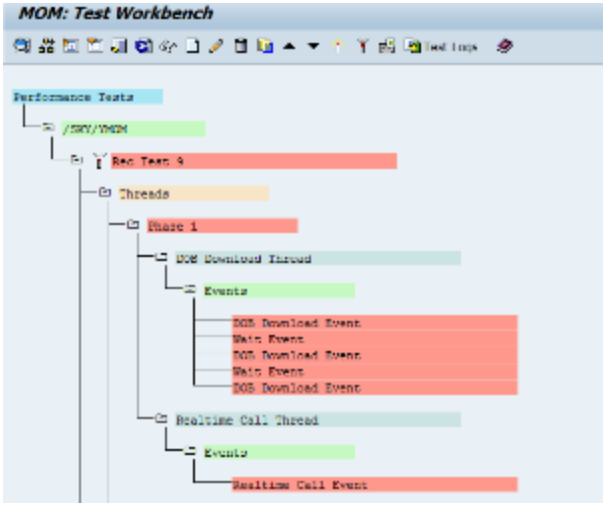
Status	Permitted Actions
Pending	Changing the Server Group and / or Server ID, Start Recording
In Progress	Stop Recording
Complete	Generate Test, Clear Recording, View Event Log

You may delete any recording that is in *Pending* or *Complete* status by selecting the recording and clicking the  button. This clears all the trace data for the recording and removes the recording from the table display on the initial screen.

Once a recording is complete, you can view the chronological log of the events recorded by clicking the  button. A sample event log:



For a completed recording, you can generate the test hierarchy by clicking the button. This prompts for a test description following which the test definition is created and control is transferred to the MOM Workbench transaction. For the above sample events, the following test definition appears:



> ***Note:*** When you are not sure about the number and type of events to create in your test definition, use the recorder first to generate a test and tweak it later according to your needs. This saves a lot of time and allows the creation of test scenarios that resemble real life use-cases.

8.4 Installing the MOM Utility

In this section, you learn how to install the Mind Over Mobile (MOM) utility into an SAP environment with Kony for SAP add-in installed. Standard SAP transport practices are used, and all objects are implemented using the SAP-certified /SKY/ namespace. There is no affect on existing SAP or customer objects.

> ***Important:*** You need to install the MOM utility separately after SkyMobile is installed, because the MOM utility is provided as a separate SAP transport.

To install the MOM utility, follow these steps:

1. Go to the Mind Over Mobile utility component from Kony Developer Portal.
2. Under the **Production Releases** tab, choose either Release 2.0 or Release 1.0, depending on your requirements.
3. Under the **Component** tab, click **PRDXXXXXX** to download the compressed Mind Over Mobile utility component.
4. Extract the above downloaded Mind Over Mobile utility component compressed file and copy the SAP transport files.
5. Check the prerequisites at the Mind Over Mobile page at Kony Developer Portal. A note appears on the **Information** tab that details the prerequisites of the required release.
6. Import the MOM utility component transport.
7. Run the check install process in each client where required.
8. Import the shortcuts transport (optional but recommended).

8.4.1 Prerequisites

The MOM utility is dependent on SkyMobile components.

For details of the specific Kony for SAP release required, refer to the Mind Over Mobile page on Kony Developer Portal.

8.4.2 MOM Utility Installation

The MOM Utility software is downloaded from Kony Developer Portal and is provided in the form of a zip file containing a standard SAP transport.

The MOM utility installation procedure is similar to that of MEAP Server Installation. Refer to the link and follow the steps to install the MOM utility software.

8.4.3 Checking MOM Installation

After the MOM add-on transport is imported into the SAP system and before you can use it, you need to run the Check Install process using the transaction **/SKY/YMOMINSTALL**.This is a generic utility that installs all the MOM Data Objects, Interface Object, Profile definitions and other meta-data that you require. The utility also checks that all the components are installed correctly and there are no *ABAP* syntax and Data Dictionary errors. You may run this utility at any time.

8.4.4 Shortcut Transaction Codes

Sky optional transport of shortcut transaction codes make prefixing transactions easier. By default, you need to prefix the transactions that you use to start the SAP add-in functionality with a namespace. With the shortcut transaction codes, for example, instead of the namespace /n/sky/ymomw, you can use **YMOMW**. This is done as a separate transport because you need to implement as non-namespace objects. There also is a remote chance that the shortcut codes may clash with existing customer transaction codes. Use the Y prefix only for Third Party utilities but customized transactions should start with Z.

To implement the shortcut transaction codes, you need an additional transport, **MOM Shortcuts** that you download from Kony Developer Portal. The procedure is the same as MEAP Server Installation SAP add-in.

> ***Important:*** Before you implement the MOM shortcut transaction codes, check for existing custom transactions that can be potentially overwritten. The check install process highlights any potential conflicts. If in doubt, run the check install again, and review the shortcut transaction code analysis.

8.5 Running a Performance Test

You can run a performance test to analyze the performance of the mobile application for a set of users. This helps to analyze and optimize the load that the system can handle as the number of users increase progressively.

This example shows how to run a Performance Test for an application using the MOM utility.

The stages involved in running a performance test are:

1. Creating a Test
2. Running the Performance Test for the Created Test
3. Reviewing the Results of the Performance Test

8.5.1 Creating a Test

You can create a performance test through one of two options:

- Recording an Application to Create a Test
- Creating a Test Manually

Recording an application automatically captures all the events triggered when the application is running. This reduces the time and number of steps to create the test.

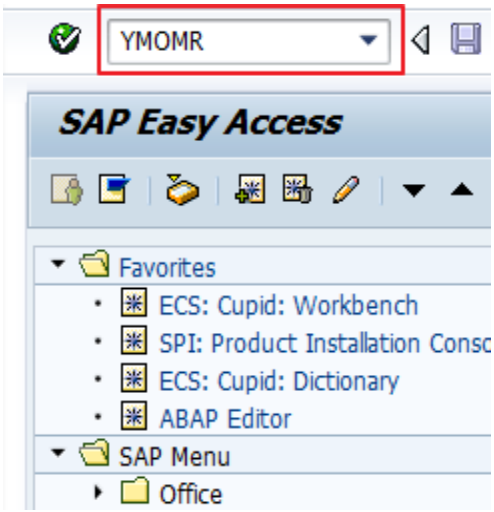
The other method is to create the test manually. You need to create all the threads, events, and instances manually.

8.5.1.1 Recording an Application to Create a Test

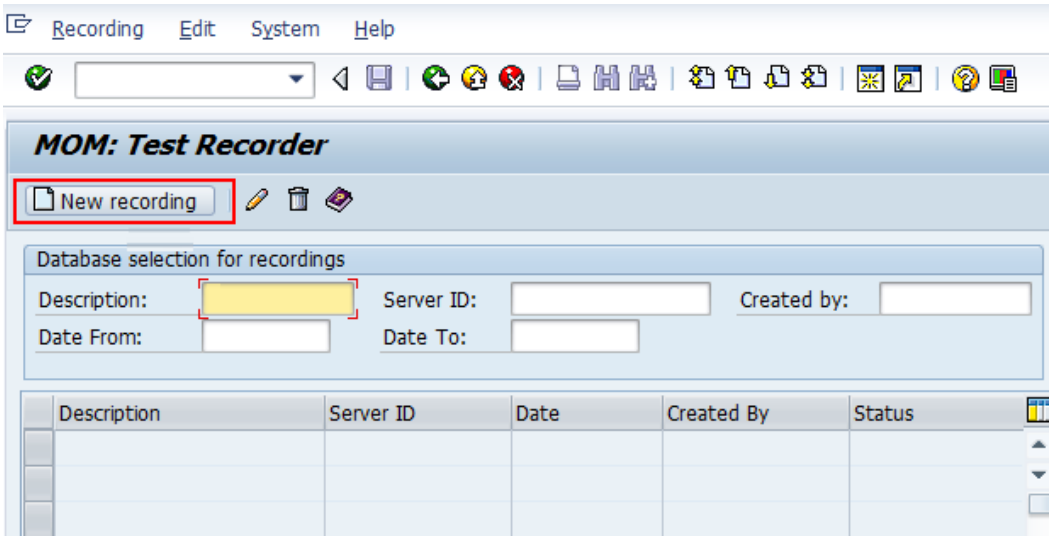
MOM Recorder is the tool that is used to record an application and then generate a test for the application. This test contains the captured events when the real application runs with the recording active.

To record an application to create a test, follow these steps:

1. To create a performance test, open the designated application on your device.
2. Run the MOM - Test Recorder by executing the transaction **YMOMR**.
   a. Log on to the SAP system on which SAP SkyMobile and MOM are installed.
   b. Run the transaction, **YMOMR**.

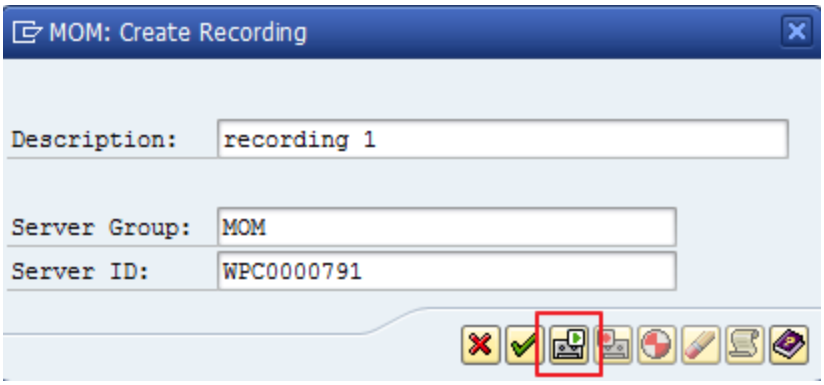


The **MOM: Test Recorder** window appears.

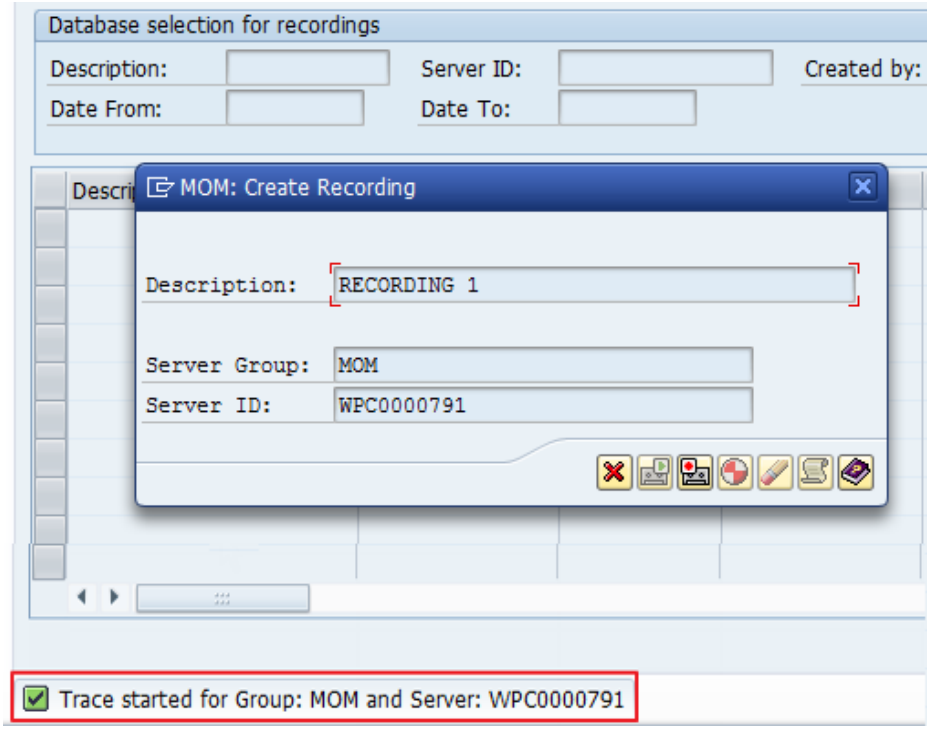


3. Click **New Recording**.
   The **MOM: Create Recording** dialog appears.

4. Enter the description for test, and type or select the relevant Server Group and the Server ID from the respective drop-down.

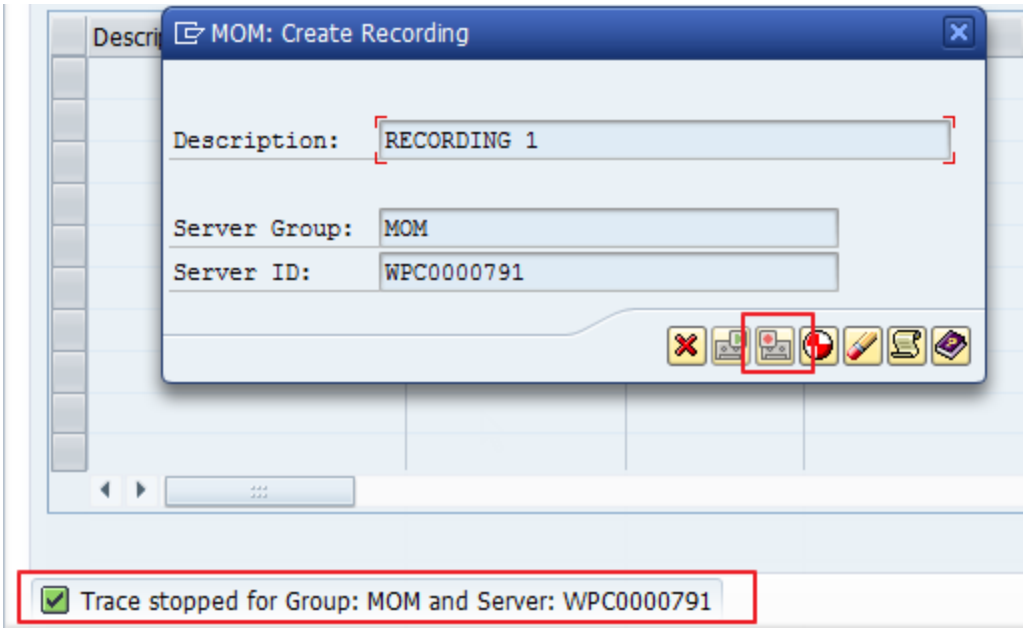


5. Enter the details for the recording and click the **Start Recording** button to start the trace (capture the events triggered while the application runs).

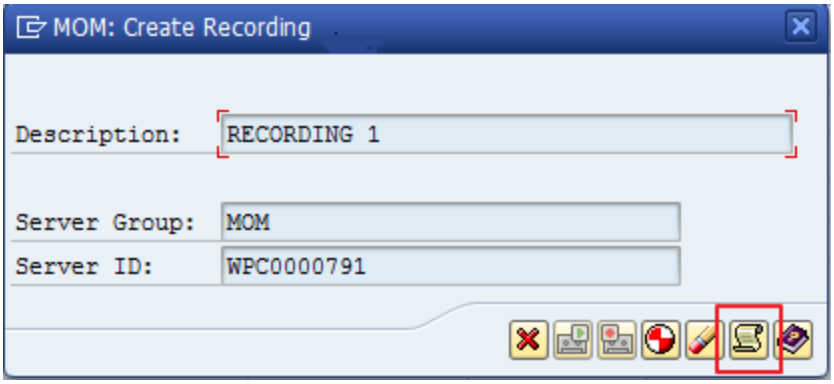


6. Run the application that was open for Performance Test on the device.

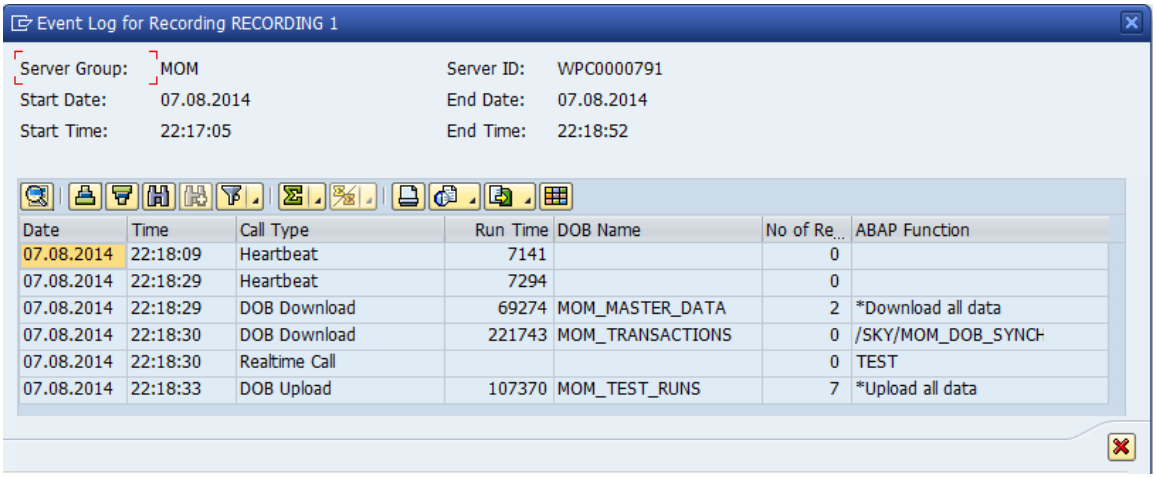
7. Click the **Stop Recording** button when the application completes.



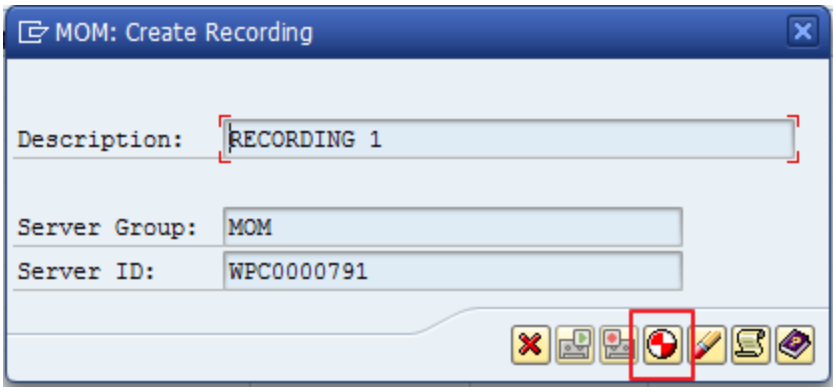
The message "Trace stopped for Group: MOM and Server: WPC0000791" appears.



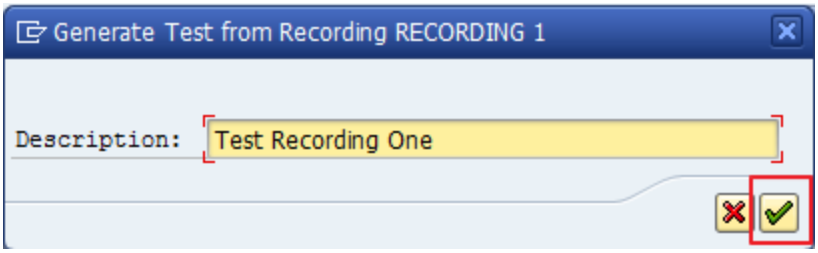
8. Click the **Event Log** button to check the events that were captured for the test.
   The **Event Log for Recording RECORDING 1** window appears with the captured events.



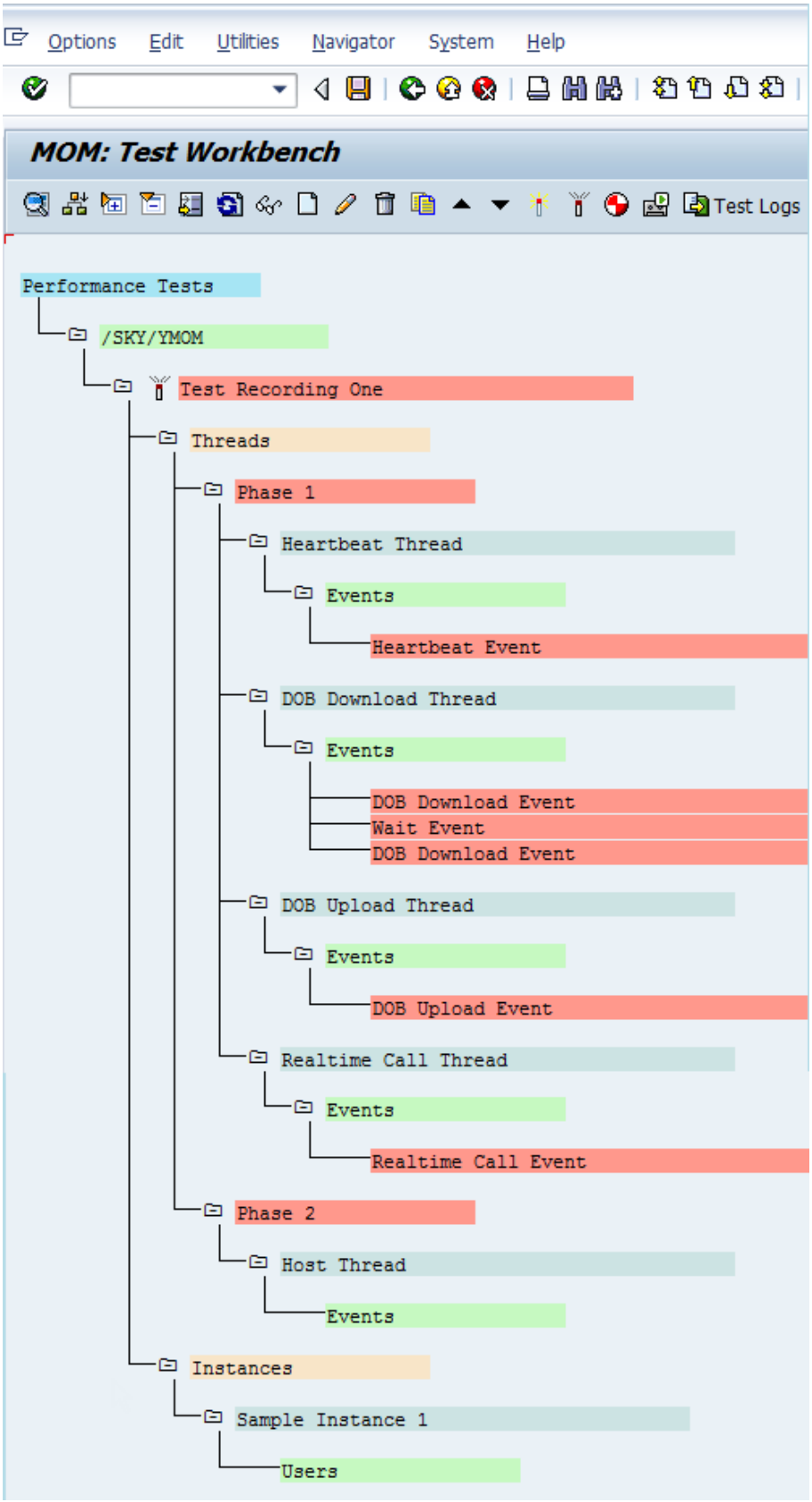
9. Click the **Generate Test** button to generate the test.



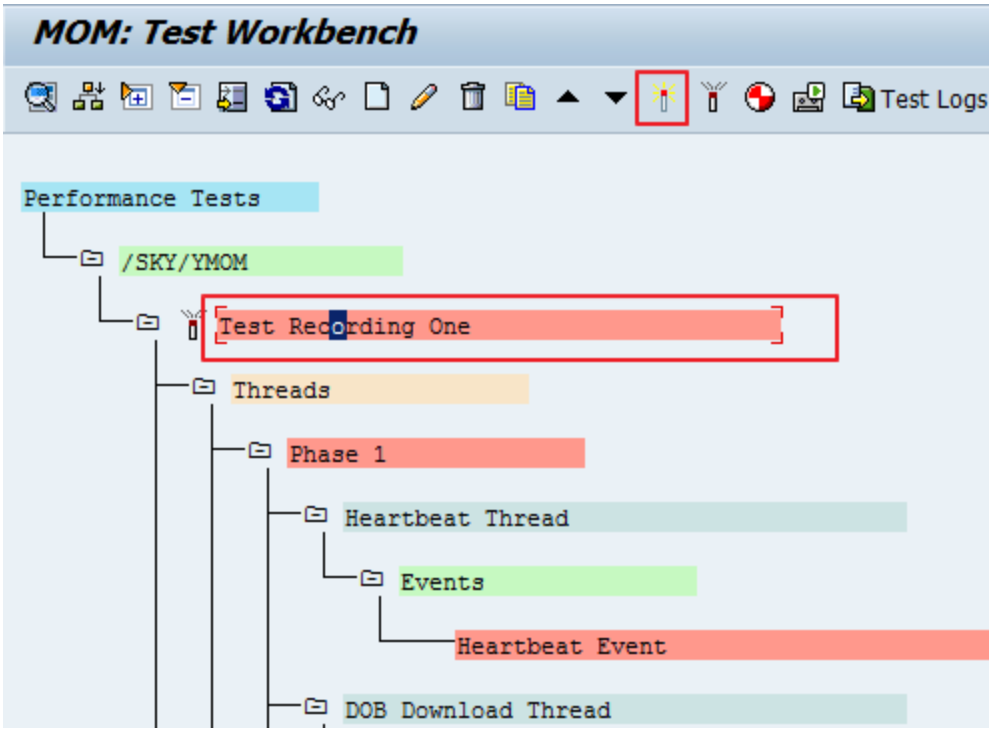
The **Generate Test from Recording RECORDING 1** window appears.



10. Enter the description of the test, and press **Enter**.
    The **MOM: Test Workbench (YMOMW)** window appears, and the test is generated in the inactive state.

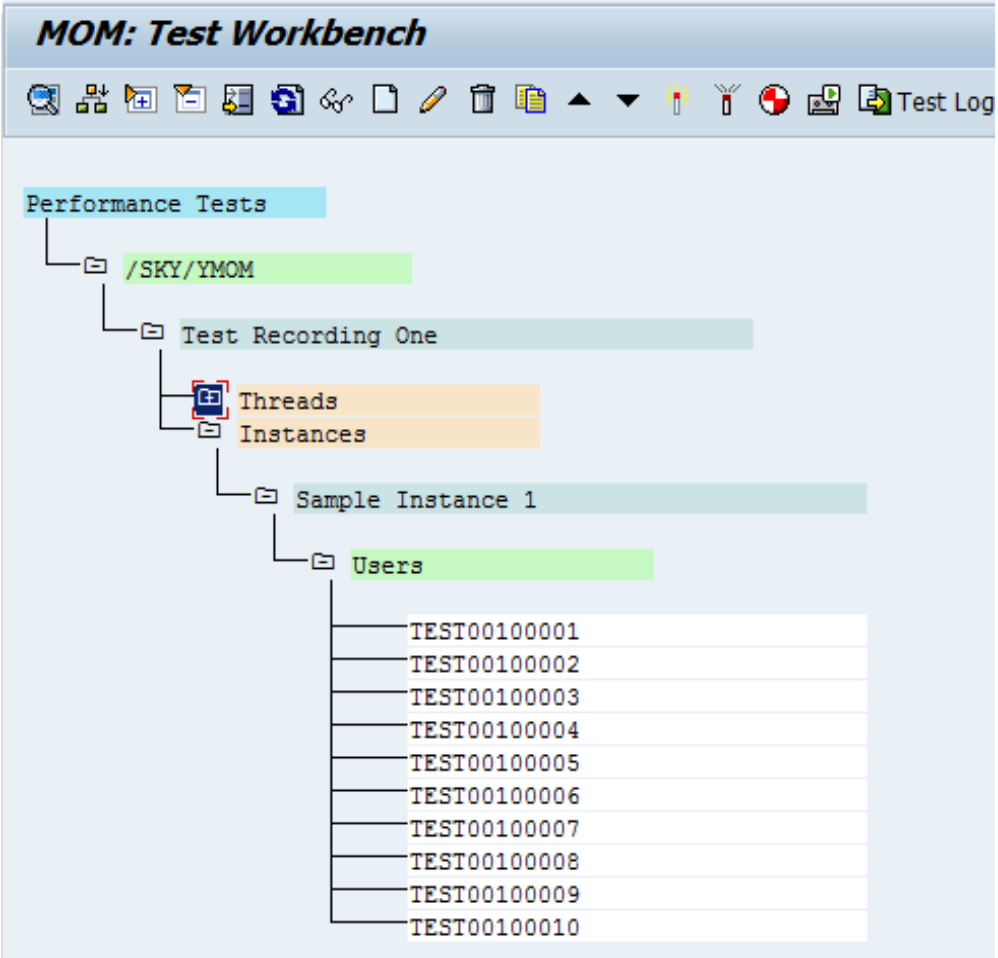


11. Place the cursor on the test, and click the **Activate** icon.



A **Confirmation** dialog appears.

12. Click **Yes**.
The test is activated, and users are generated.

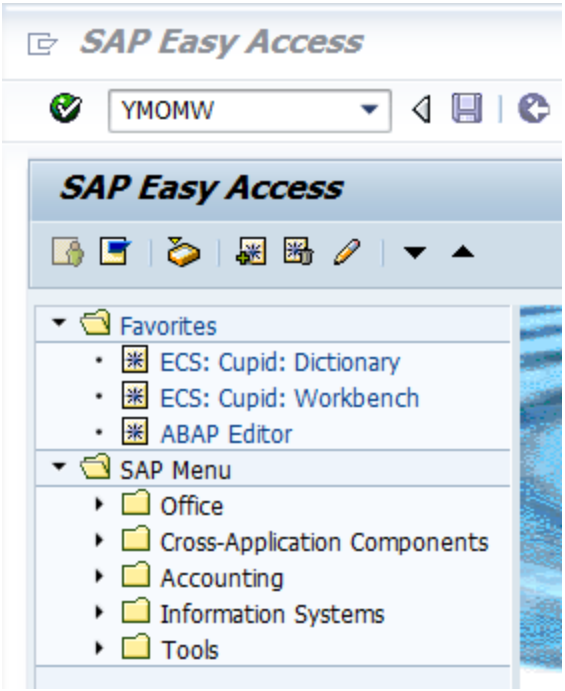


8.5.2 Creating a Test Manually

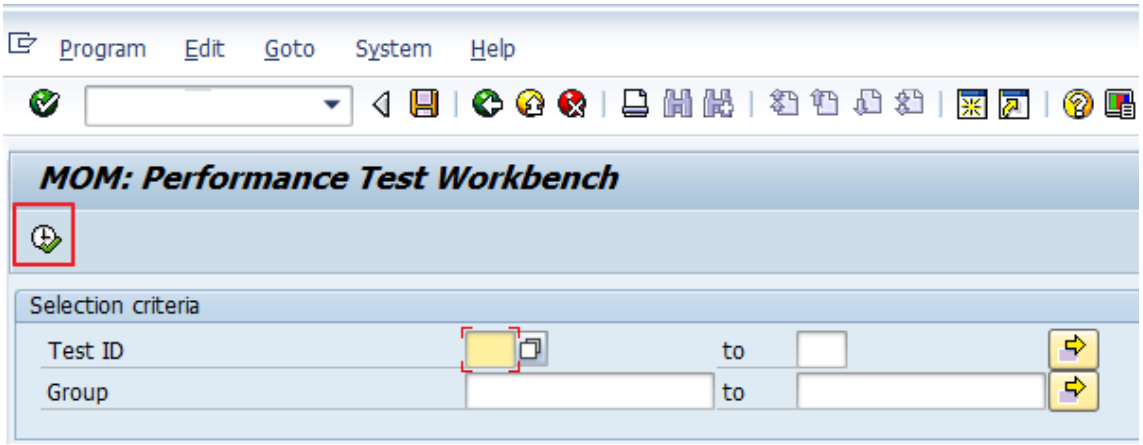
You can optionally create a customized test manually from the MOM workbench.

To create a test manually, follow these steps:

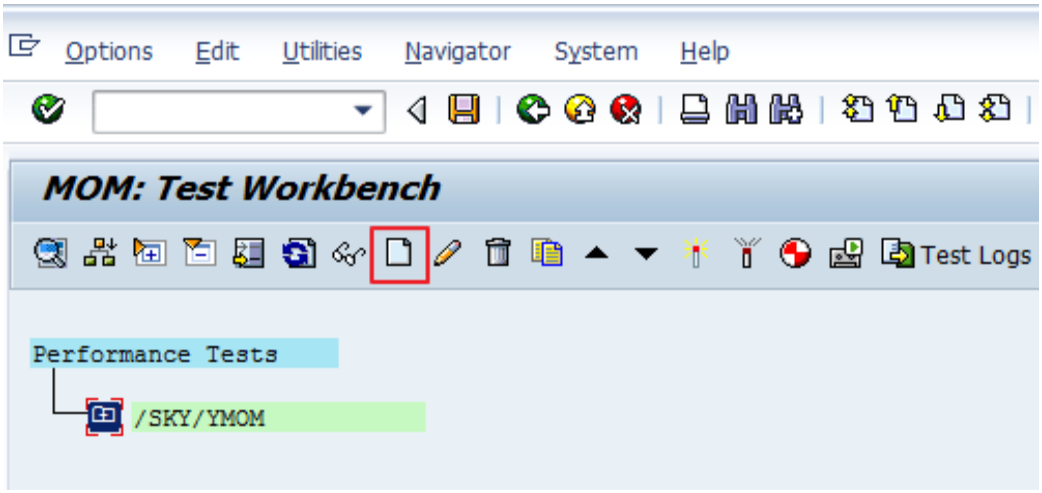
1. Log on to the SAP system where SAP SkyMobile and MOM are installed.
2. Run the transaction, **YMOMW**.



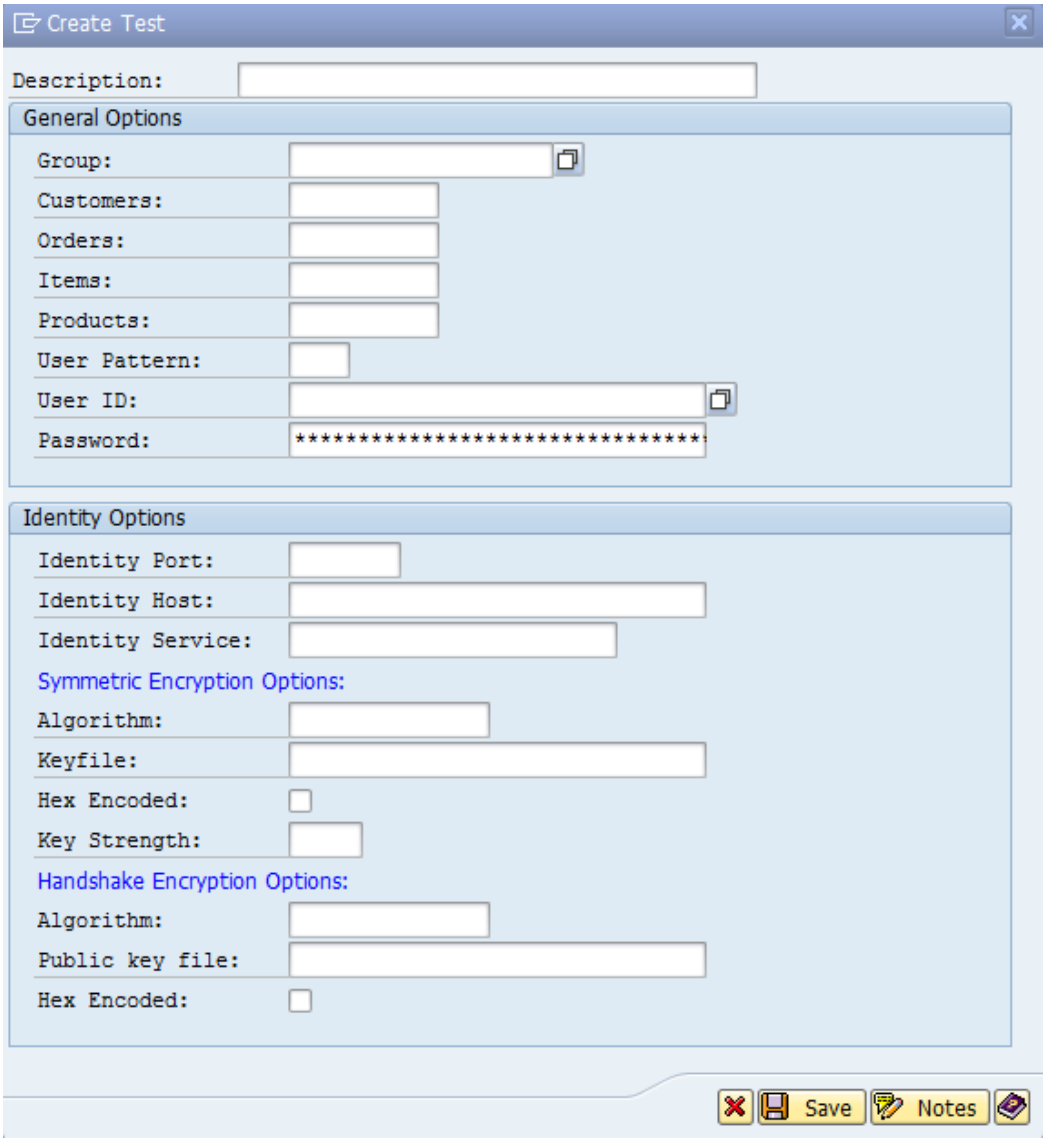
The **MOM: Performance Test Workbench** window appears.



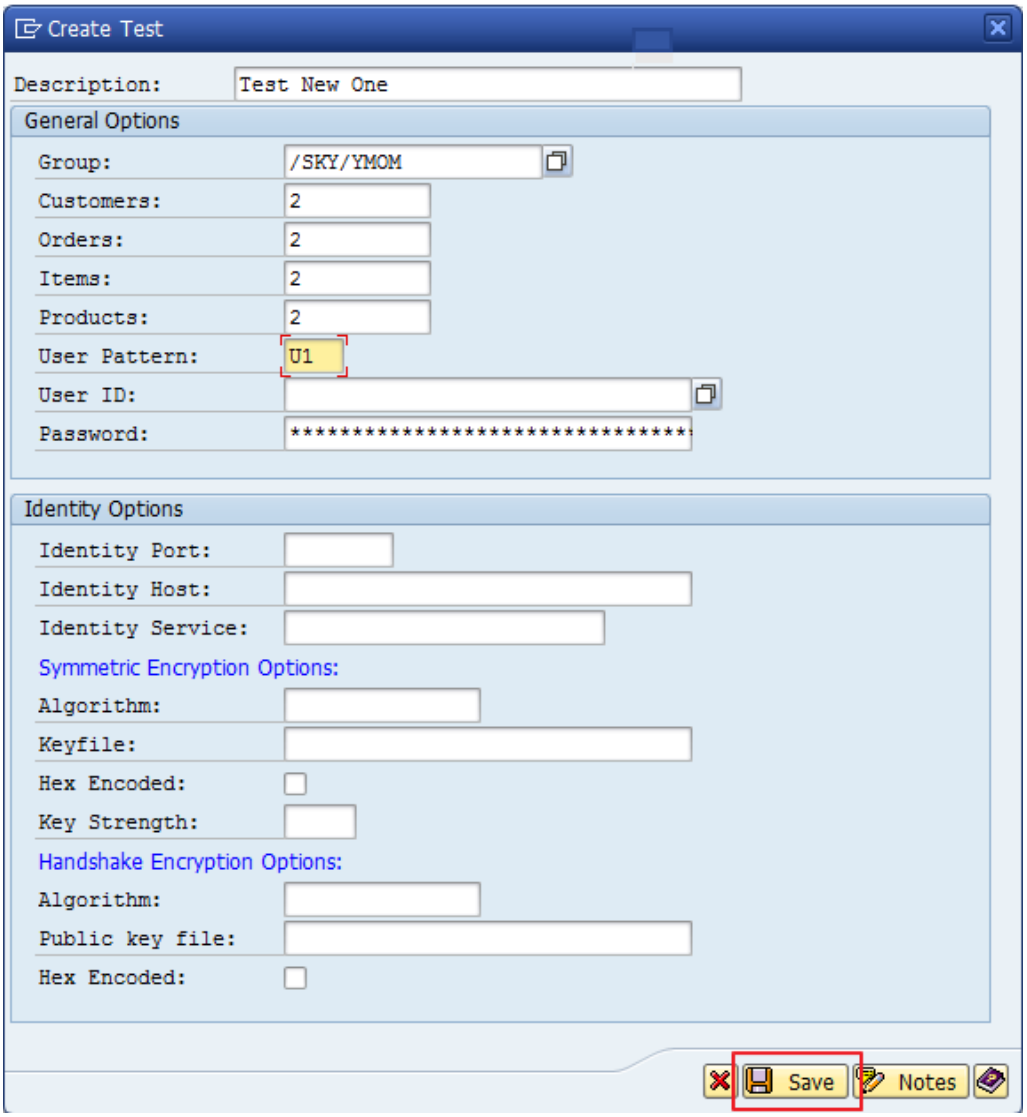
3. Click **Execute**.
   The **MOM: Test Workbench** appears with the performance tests.



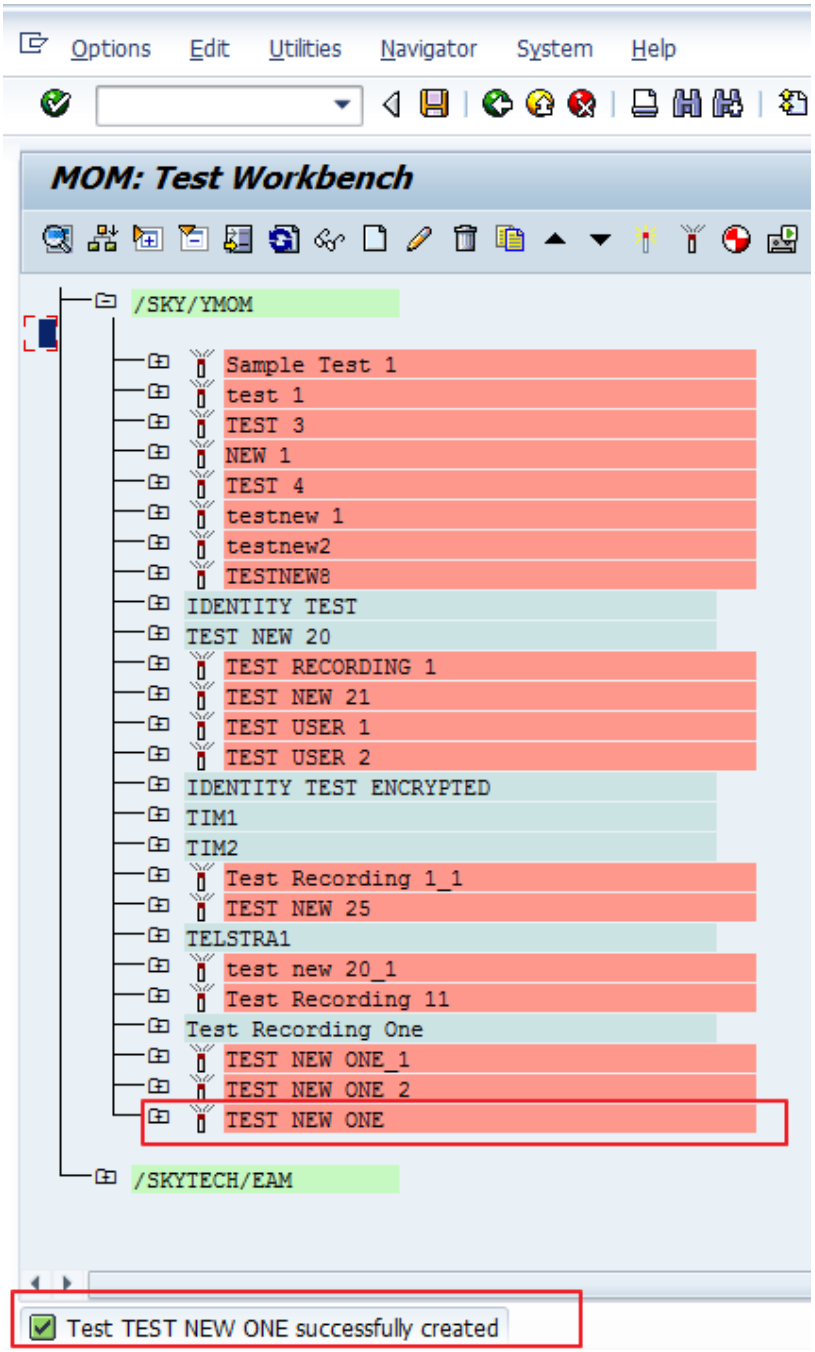
4. Click **Create**.
   The **Create Test** dialog appears.



5. Enter all the details and click **Save**.

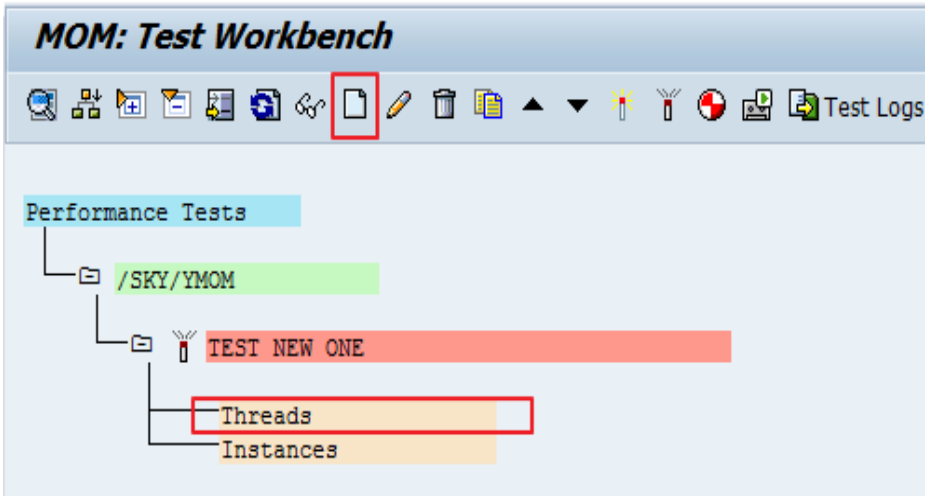


The test is created and the message "Test <TEST NEW ONE> successfully created" appears.

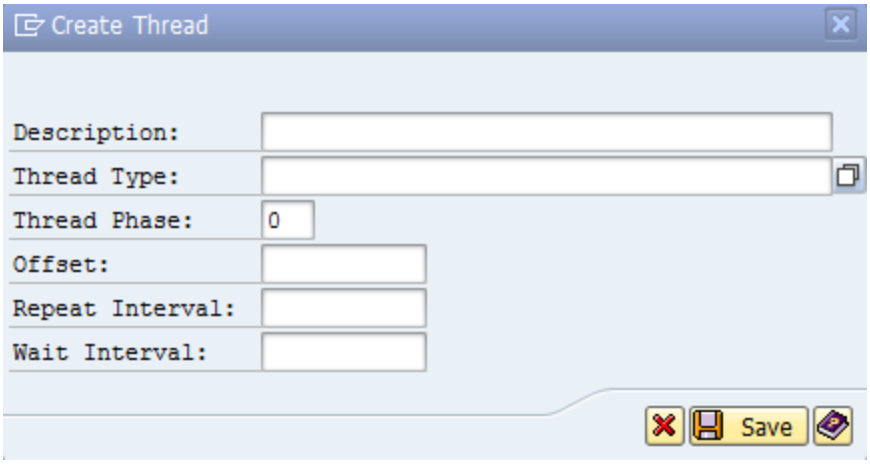


6. Create a Thread.

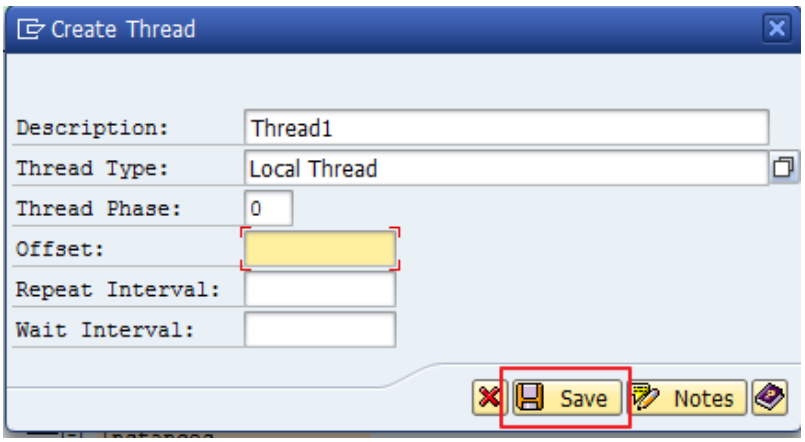
   a. Place the cursor on **Threads** and click **Create**.



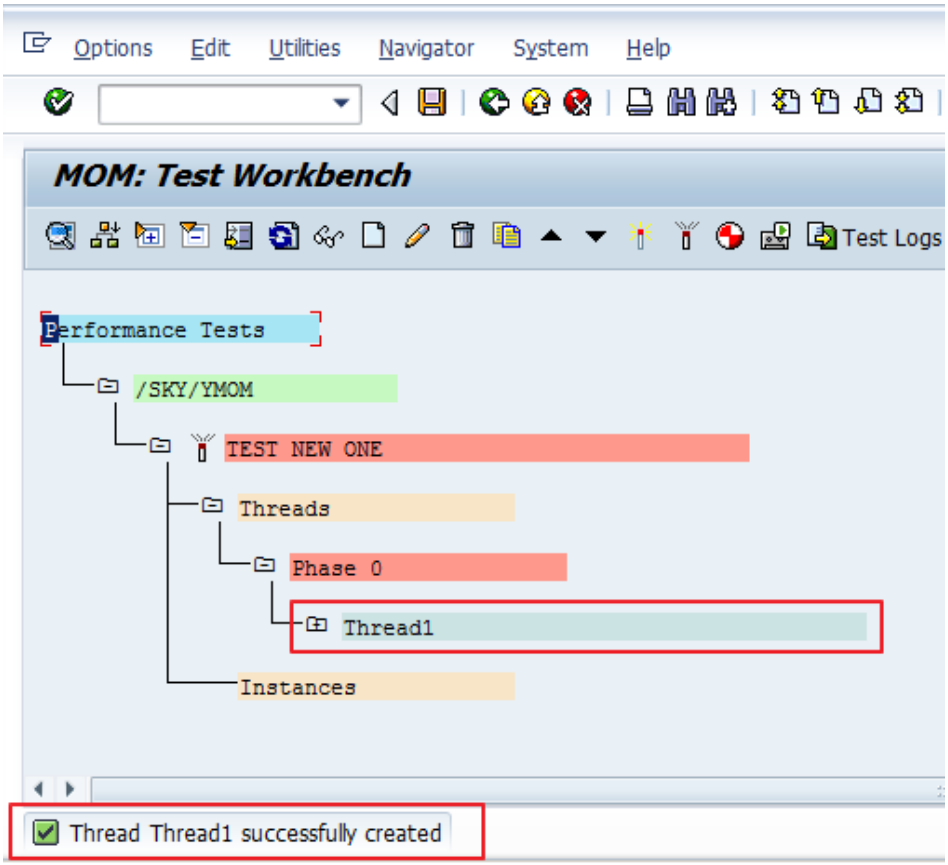
   b. The **Create Thread** dialog appears.



c. Enter the details of the thread, and click **Save**.

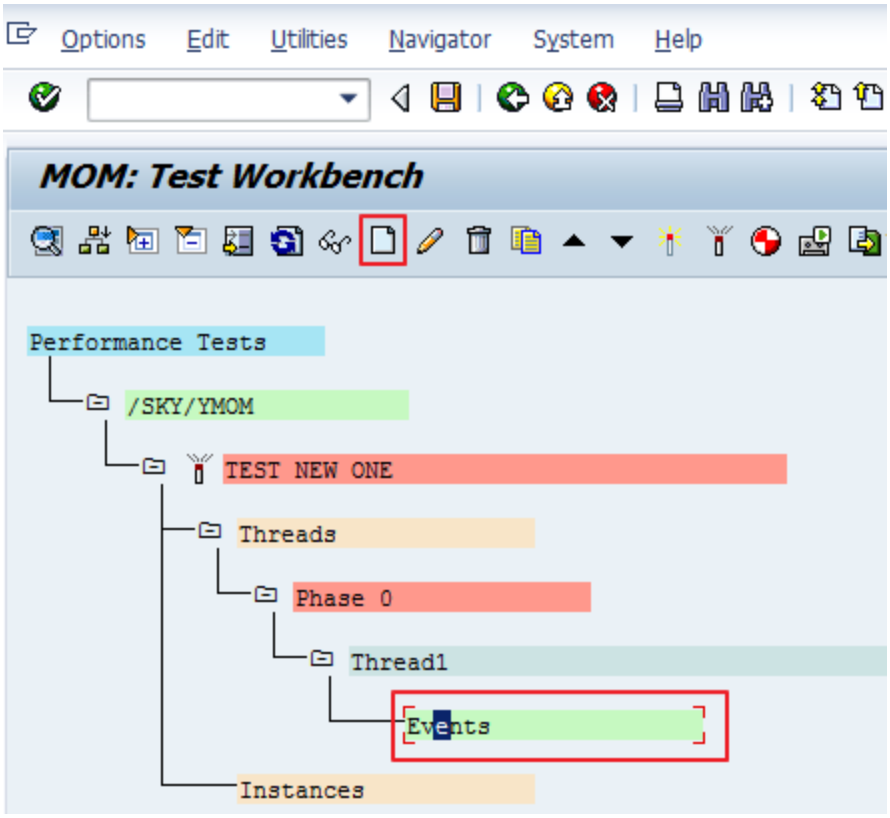


The message "Thread <Thread1> successfully created" appears.

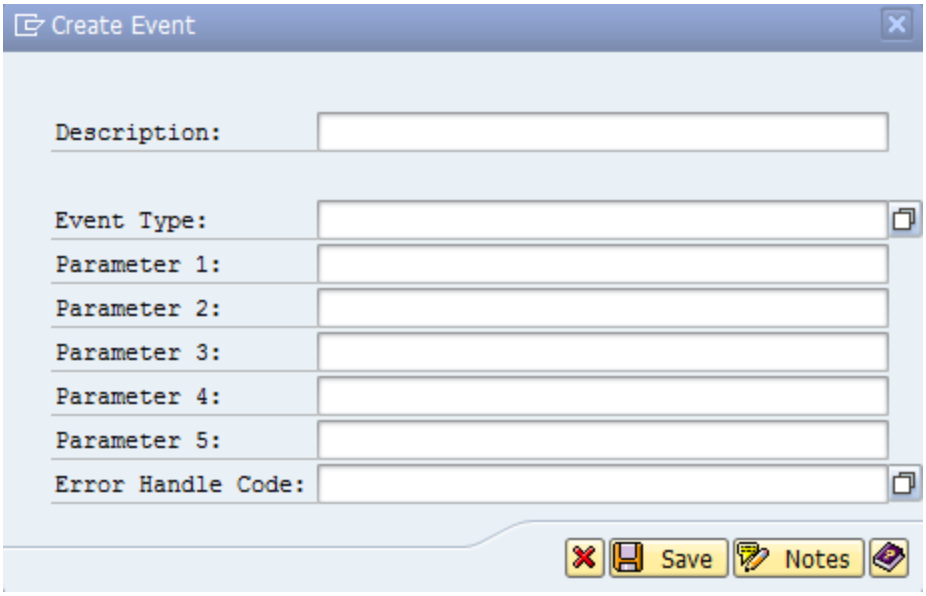


7. Create Events.

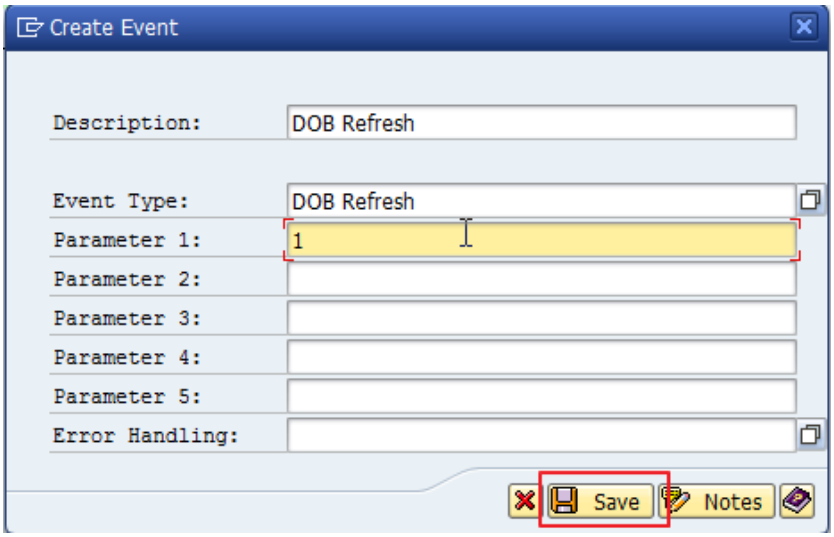
   a. Place the cursor on **Events** under **Thread1**, and click **Create**.



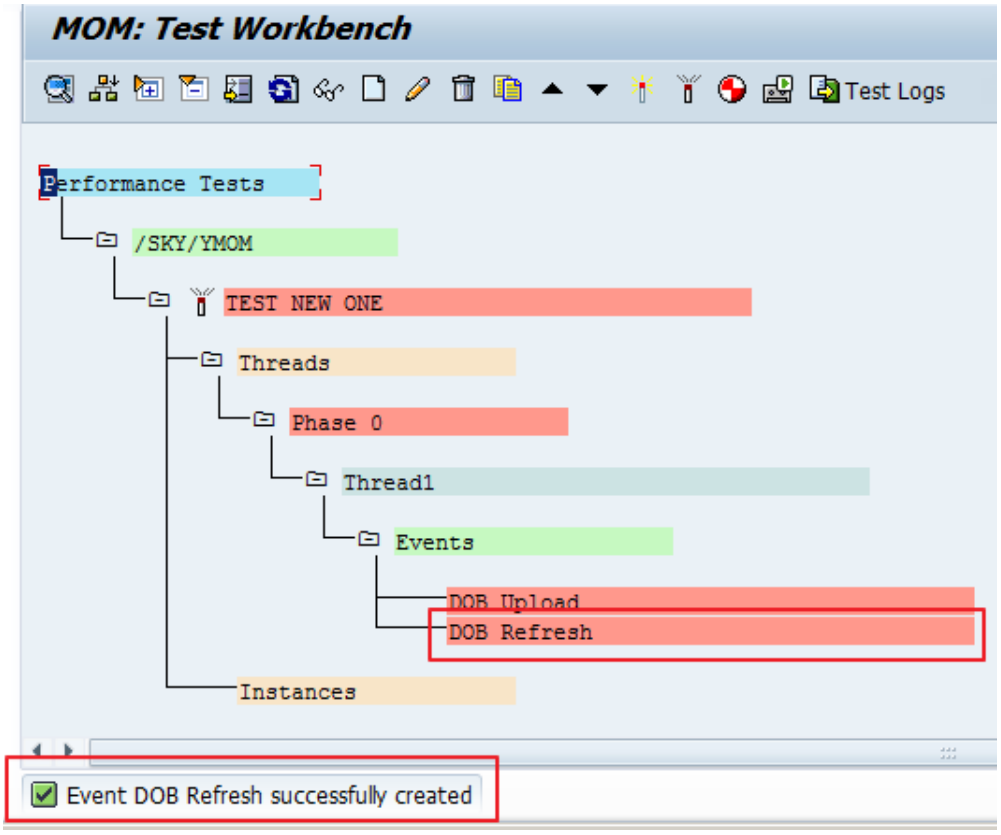
   The **Create Event** dialog appears.



b. Enter the details of the event, and click **Save**.



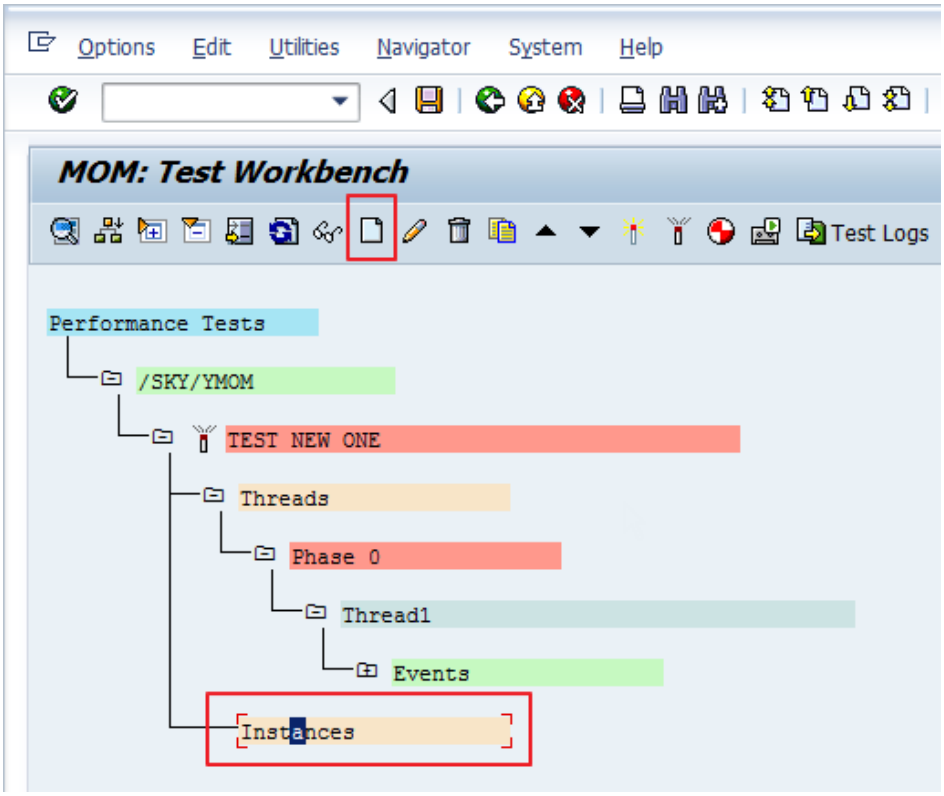
The message "Event DOB Refresh successfully created" appears.



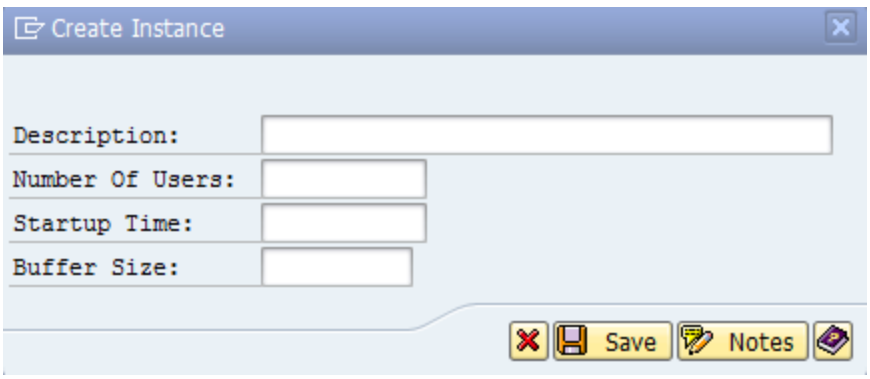
Similarly, create all the events that can be triggered when the application runs.

8. Create an Instance.

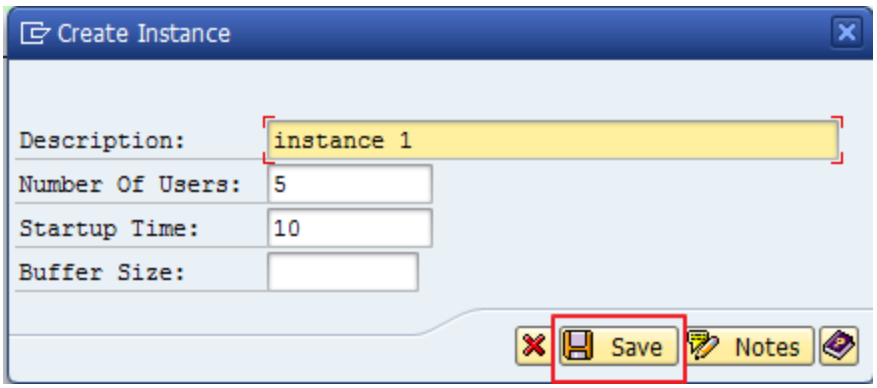
   a. Place the cursor on **Instances** under the newly created test and click **Create**.



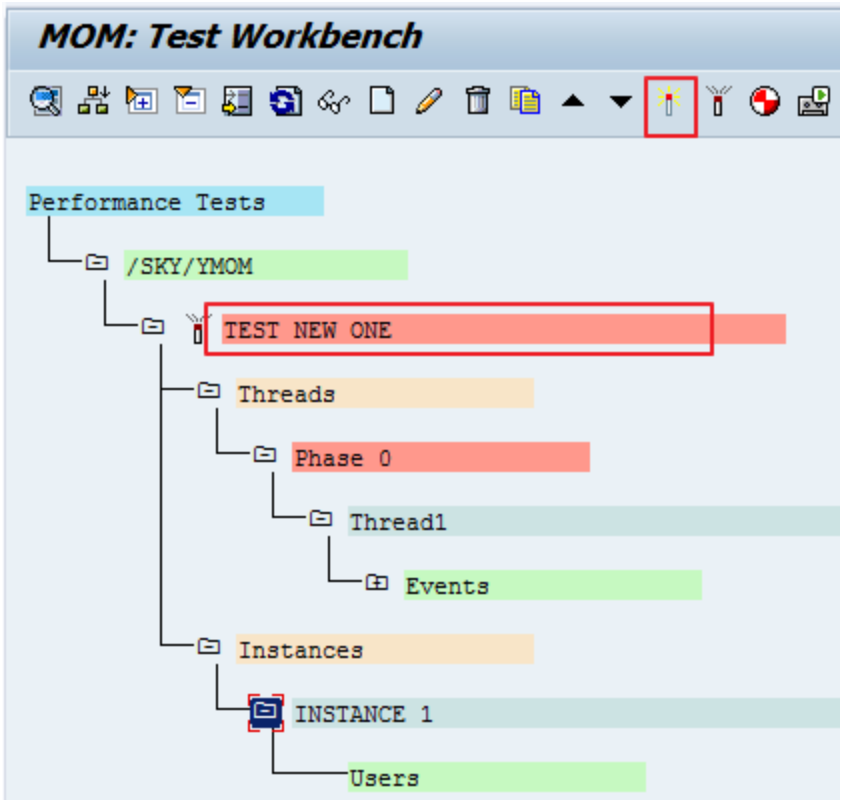
The **Create Instance** dialog appears.



b. Enter the details of the instance, and click **Save**.

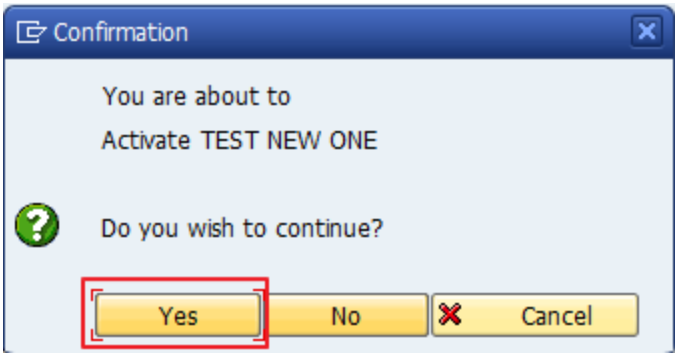


9. Place the cursor on the <Test Name> and click **Activate**.

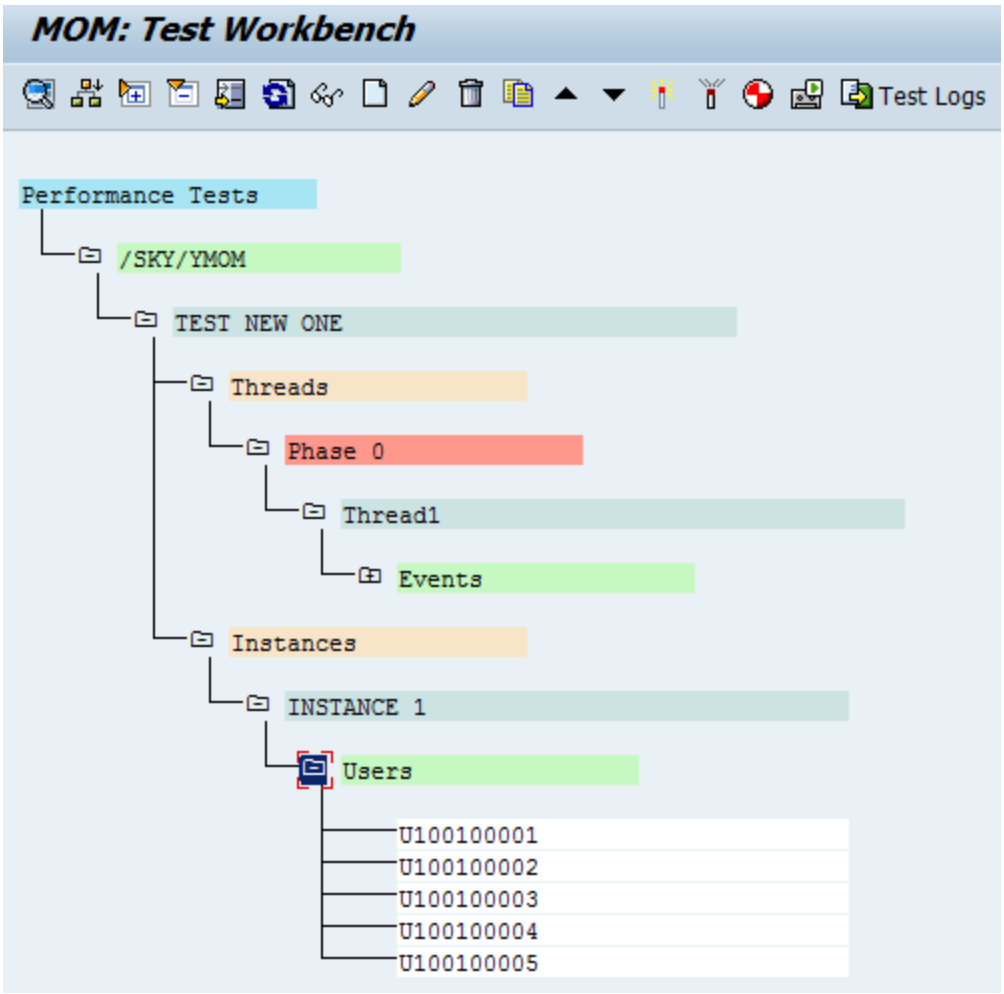


A **Confirmation** dialog appears.

10. Click **Yes**.



11. The test is activated, and users are generated under **INSTANCE 1**.

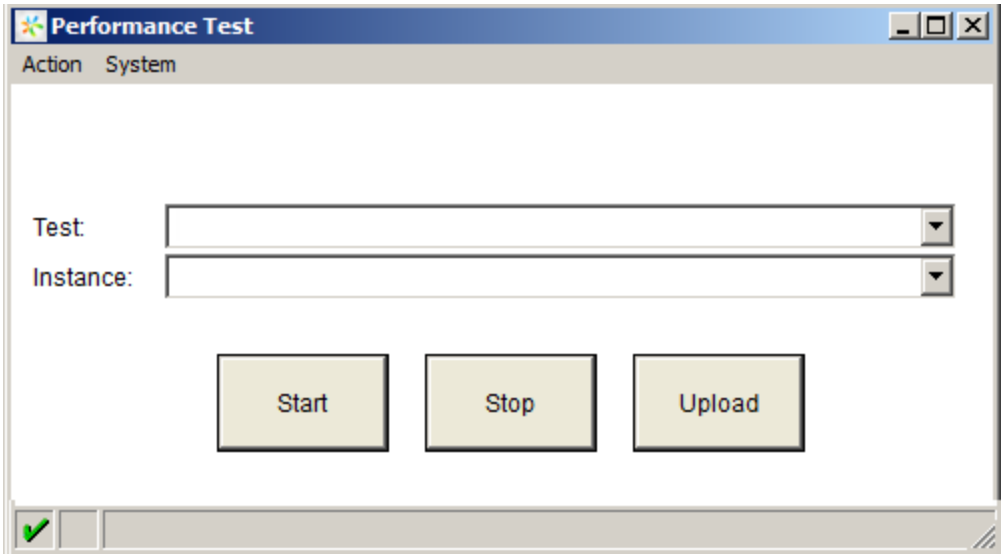


8.5.3 Running the Performance Test for the Created Test

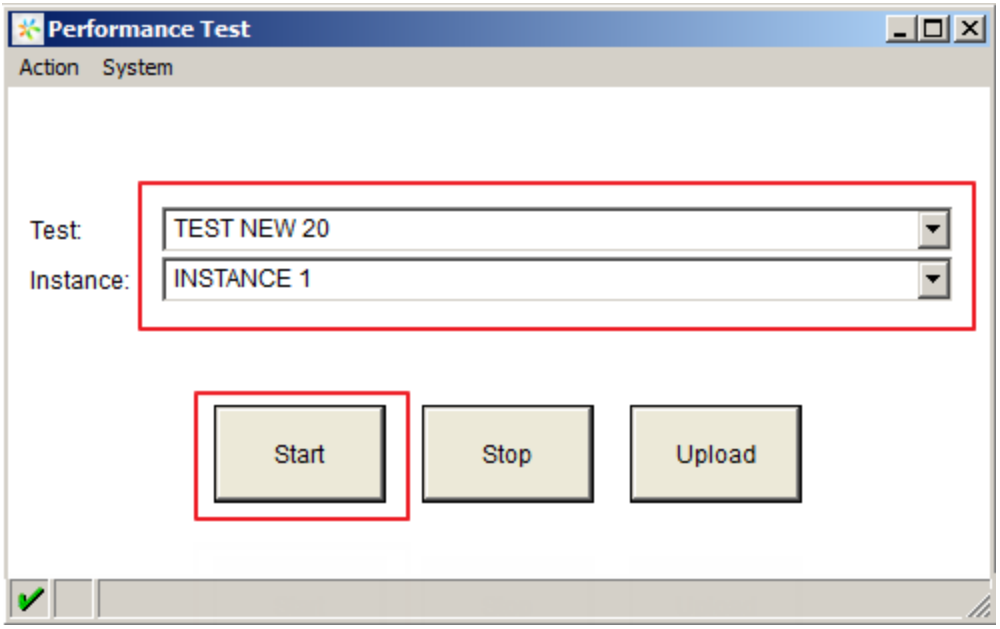
You can run a performance test for a designated set of users under Instances. The result of running this performance test is a log with all the events for all the users, User Load Report and Response Time Report graph.

To run the performance test for the test that you created, follow these steps:

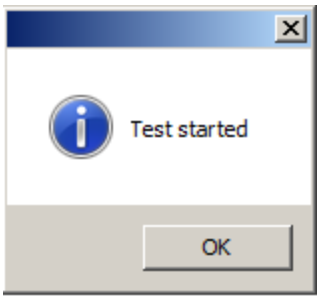
1. Access the MOM application on a Windows machine (WPC). For detail information on accessing the application on a Windows machine (WPC), refer to Thin Client Installation.



2. Select the required <Test Name> and <Instance Name> and click **Start**.



A dialog appears with the message, "Test started".



3. Click **OK**, and then click **Upload**.

8.5.4 Reviewing the Results of the Performance Test

After the performance test is successfully run, you can review the created reports:

1. Run Summary and Log
2. User Load Graph

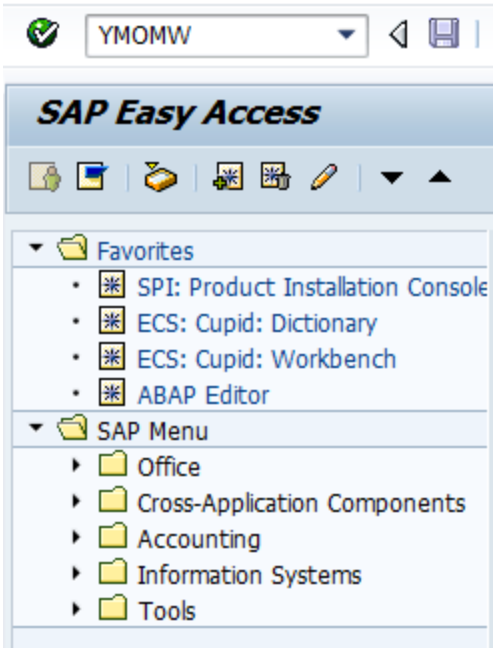
3. Response Time Graph

The following sections tell you how to access the information.

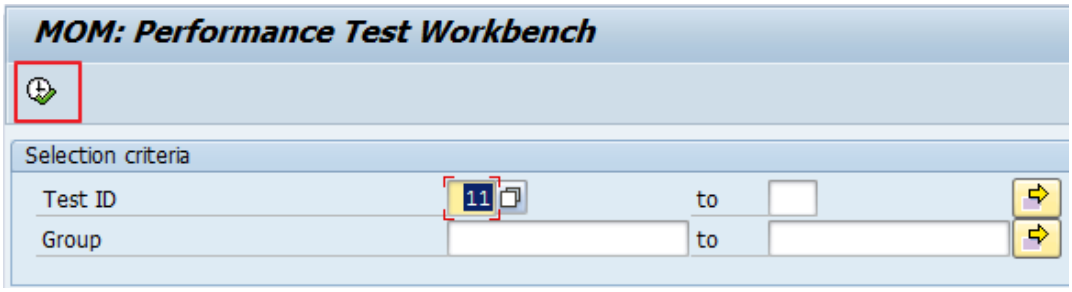
8.5.4.1 Run Summary and Log

To review the run summary, log, and graphs that are created for the performance test, follow these steps:

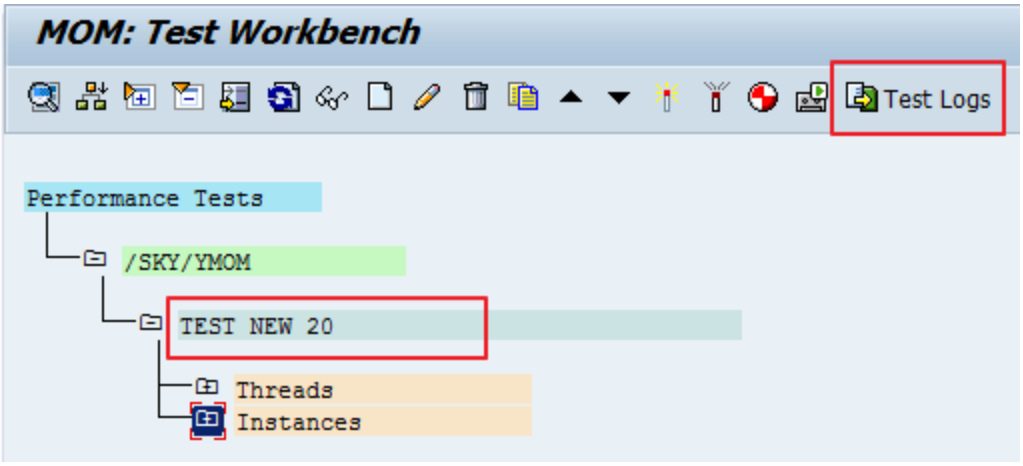
1. Go to the transaction **YMOMW**.



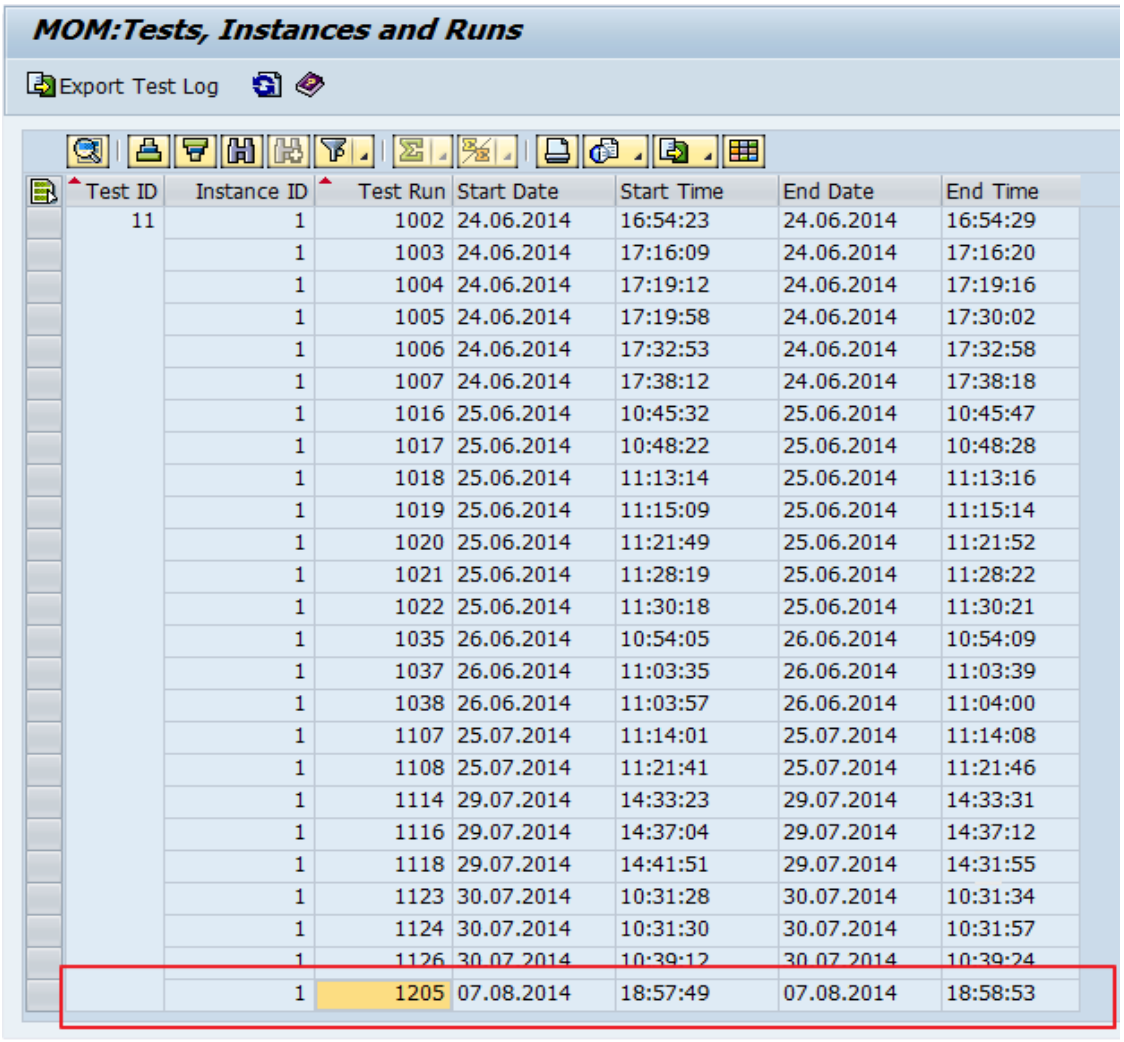
2. Enter or select the test ID from the drop-down for the test **TEST NEW 20**, and click **Execute**.



3. Place the cursor on the **TEST NEW 20** test and click **Test Logs** on the application toolbar.



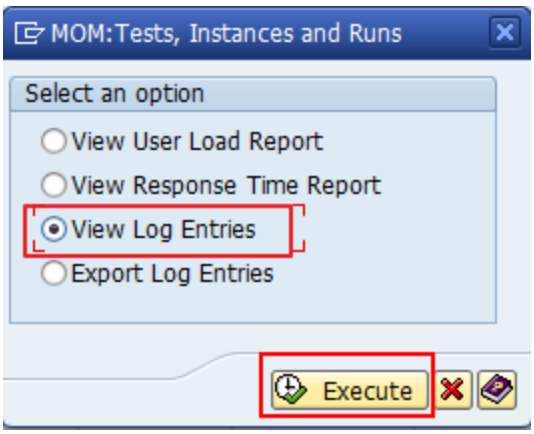
The **MOM:Tests, Instances and Runs** window appears with the test runs for the test.



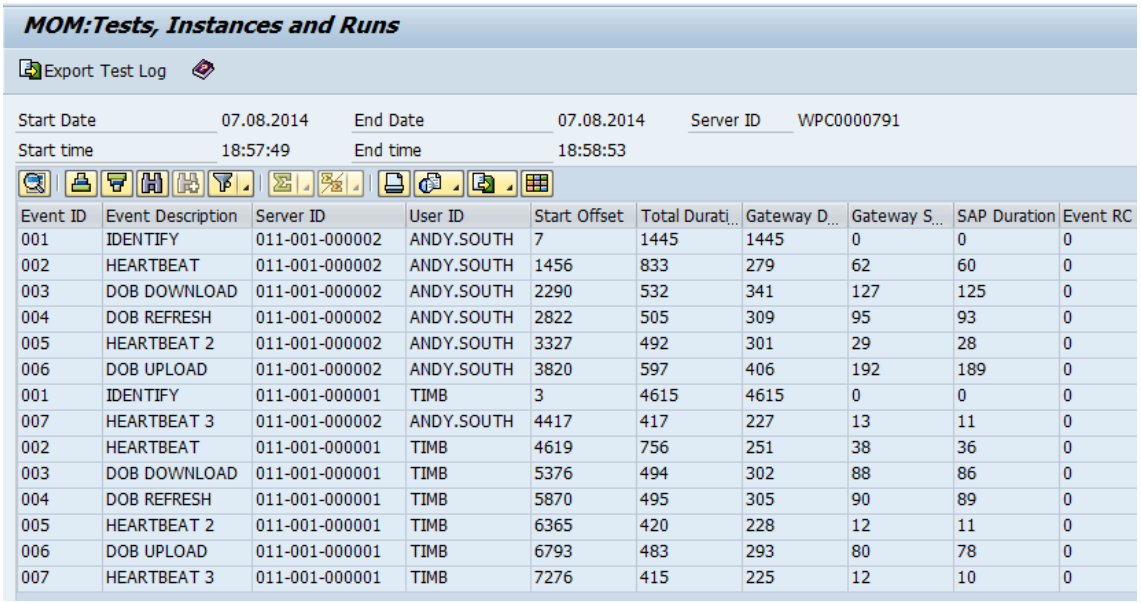
Test ID	Instance ID	Test Run	Start Date	Start Time	End Date	End Time
11	1	1002	24.06.2014	16:54:23	24.06.2014	16:54:29
	1	1003	24.06.2014	17:16:09	24.06.2014	17:16:20
	1	1004	24.06.2014	17:19:12	24.06.2014	17:19:16
	1	1005	24.06.2014	17:19:58	24.06.2014	17:30:02
	1	1006	24.06.2014	17:32:53	24.06.2014	17:32:58
	1	1007	24.06.2014	17:38:12	24.06.2014	17:38:18
	1	1016	25.06.2014	10:45:32	25.06.2014	10:45:47
	1	1017	25.06.2014	10:48:22	25.06.2014	10:48:28
	1	1018	25.06.2014	11:13:14	25.06.2014	11:13:16
	1	1019	25.06.2014	11:15:09	25.06.2014	11:15:14
	1	1020	25.06.2014	11:21:49	25.06.2014	11:21:52
	1	1021	25.06.2014	11:28:19	25.06.2014	11:28:22
	1	1022	25.06.2014	11:30:18	25.06.2014	11:30:21
	1	1035	26.06.2014	10:54:05	26.06.2014	10:54:09
	1	1037	26.06.2014	11:03:35	26.06.2014	11:03:39
	1	1038	26.06.2014	11:03:57	26.06.2014	11:04:00
	1	1107	25.07.2014	11:14:01	25.07.2014	11:14:08
	1	1108	25.07.2014	11:21:41	25.07.2014	11:21:46
	1	1114	29.07.2014	14:33:23	29.07.2014	14:33:31
	1	1116	29.07.2014	14:37:04	29.07.2014	14:37:12
	1	1118	29.07.2014	14:41:51	29.07.2014	14:31:55
	1	1123	30.07.2014	10:31:28	30.07.2014	10:31:34
	1	1124	30.07.2014	10:31:30	30.07.2014	10:31:57
	1	1126	30.07.2014	10:39:12	30.07.2014	10:39:24
	1	1205	07.08.2014	18:57:49	07.08.2014	18:58:53

4. Double-click the Test Run **1205**.
   The **MOM:Tests, Instances and Runs** dialog appears with options to generate reports.

5. Select the **View Log Entries** option and click **Execute**.



6. The log appears with the events run for all the users.



MOM:Tests, Instances and Runs

Export Test Log

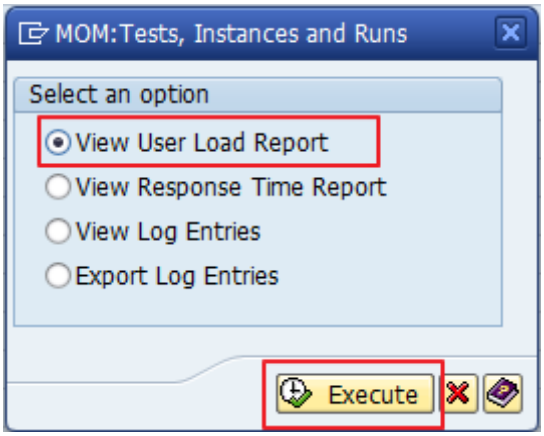
Start Date	07.08.2014	End Date	07.08.2014	Server ID	WPC0000791			
Start time	18:57:49	End time	18:58:53					

Event ID	Event Description	Server ID	User ID	Start Offset	Total Durati...	Gateway D...	Gateway S...	SAP Duration	Event RC
001	IDENTIFY	011-001-000002	ANDY.SOUTH	7	1445	1445	0	0	0
002	HEARTBEAT	011-001-000002	ANDY.SOUTH	1456	833	279	62	60	0
003	DOB DOWNLOAD	011-001-000002	ANDY.SOUTH	2290	532	341	127	125	0
004	DOB REFRESH	011-001-000002	ANDY.SOUTH	2822	505	309	95	93	0
005	HEARTBEAT 2	011-001-000002	ANDY.SOUTH	3327	492	301	29	28	0
006	DOB UPLOAD	011-001-000002	ANDY.SOUTH	3820	597	406	192	189	0
001	IDENTIFY	011-001-000001	TIMB	3	4615	4615	0	0	0
007	HEARTBEAT 3	011-001-000002	ANDY.SOUTH	4417	417	227	13	11	0
002	HEARTBEAT	011-001-000001	TIMB	4619	756	251	38	36	0
003	DOB DOWNLOAD	011-001-000001	TIMB	5376	494	302	88	86	0
004	DOB REFRESH	011-001-000001	TIMB	5870	495	305	90	89	0
005	HEARTBEAT 2	011-001-000001	TIMB	6365	420	228	12	11	0
006	DOB UPLOAD	011-001-000001	TIMB	6793	483	293	80	78	0
007	HEARTBEAT 3	011-001-000001	TIMB	7276	415	225	12	10	0

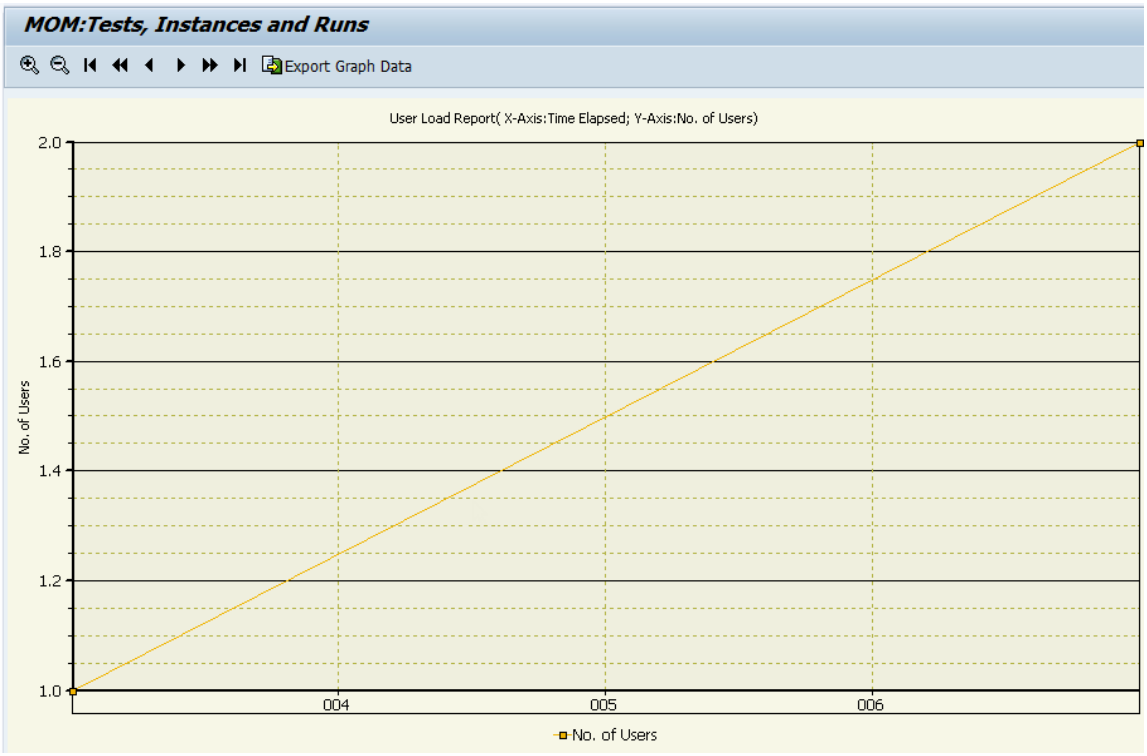
8.5.4.2 User Load Report

To review the User Load Report that is generated for the run follow these steps:

1. Select the **View User Load Report** option and click **Execute**.



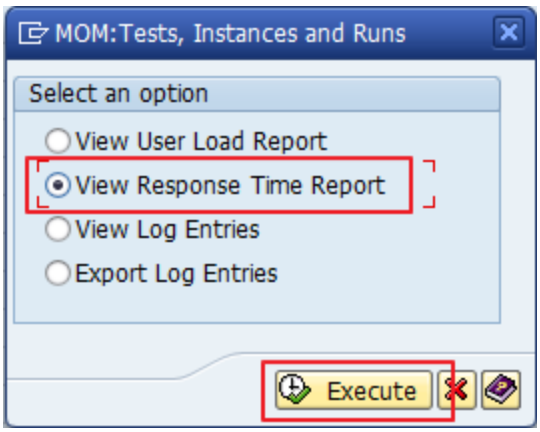
The **User Load Graph** report appears.



8.5.4.3 Response Time Report

To review the Response Time Report that is generated for the run, follow these steps:

1. Select the **View Response Time Report** option and click **Execute**.



The Response Time Graph report appears.