Agentry SAP Framework

SAP Mobile Platform 3.0

DOCUMENT ID: DC-01-0300-01
LAST REVISED: November 2013

Contents

Agentry SAP Framework

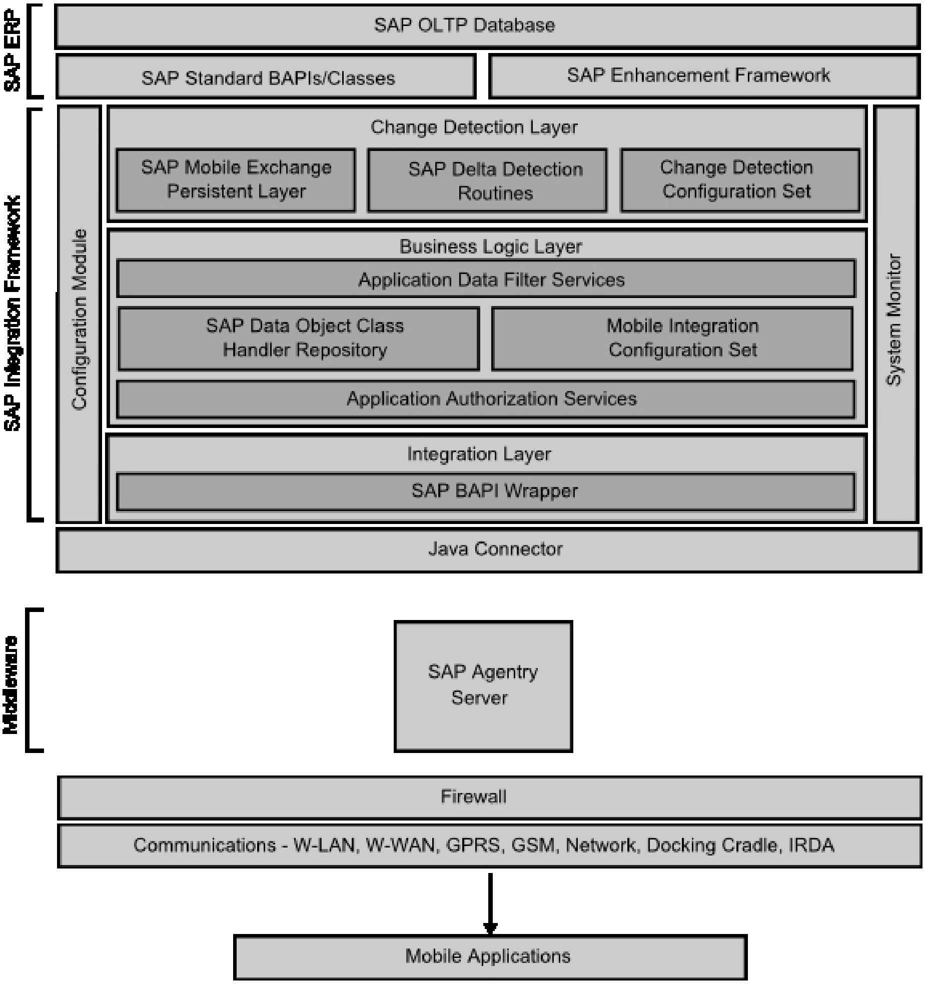
Agentry SAP Framework

The Agentry SAP Framework provides an efficient way to build mobile solutions for SAP® using SAP Mobile Platform. The framework was developed to address the following business needs:

- Establish a common mobile integration service layer and architecture for the mobile adaptation of enterprise business processes and business object data for enterprise mobile applications.
- Define a consistent integration pattern for mobile data object modeling, change detections, data distribution, delta sync calculation, and data pushes.
- Provide a framework for mobile application logging, tracing, administration, and monitoring.
- Uses a combined coding and configuration approach that enables a comprehensive and flexible application paradigm. Standard objects are configurable and extensible. Partners and customers are able to develop their own integration services using the same integration pattern.
- Uses the latest technology supported by SAP, to protect the return on investment.

The framework consists of several components shown in the following diagram.

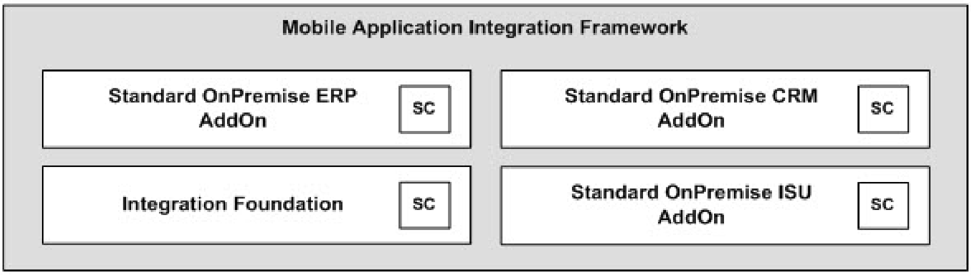
Figure 1: Agentry SAP Framework System Components


SAP ERP
SAP OLTP Database
SAP Standard BAPIs/Classes
SAP Enhancement Framework
SAP Integration Framework
Configuration Module
Change Detection Layer
SAP Mobile Exchange Persistent Layer
SAP Delta Detection Routines
Change Detection Configuration Set
Business Logic Layer
Application Data Filter Services
SAP Data Object Class Handler Repository
Mobile Integration Configuration Set
Application Authorization Services
Integration Layer
SAP BAPI Wrapper
System Monitor
Java Connector
Middleware
SAP Agentry Server
Firewall
Communications - W-LAN, W-WAN, GPRS, GSM, Network, Docking Cradle, IRDA
Mobile Applications

Software Component Layering

The Agentry SAP Framework uses a layered software component approach; dependency between each software component layer is by design and encouraged in order to provide maximum reusability. There are two categories of software components within the Agentry SAP Framework: the Foundation Add-On Component, and the Application Add-On Components, as shown in the following diagram.

Figure 2: Agentry SAP Framework System Components


Mobile Application Integration Framework
Standard OnPremise ERP AddOn
SC
Standard OnPremise CRM AddOn
SC
Integration Foundation
SC
Standard OnPremise ISU AddOn
SC

The Foundation Add-On Component (SMFND) is designed for a Netweaver ABAP-based system. It can be deployed to any SAP systems based on Netweaver AS ABAPA 7.x. The foundation component defines the standard integration patterns supported by the framework. It provides class libraries, RFC module pools, system programs, utilities, application configuration tools, and system administration tools. Application integration services are developed using the foundation component.

Application add-on components are developed using the integration patterns, services, and tools provided by the foundation component. Application add-on components provide an application level integration support for mobile applications.

A layered approach is used when developing the application add-ons. A standard application add-on is developed for each SAP Business Suite system and supports all mobile applications for the suite system. There is no mobile application-specific add-on using this approach. Each standard application add-on is designed to provide mobile integration support for all standard business processes and business objects within a specific suite system. Mobile integration services are designed and developed to be shared and reused by multiple mobile applications requiring the same business process and business object.

Currently only onPremise SAP ERP systems and SAP CRM systems are supported by the framework. Add-On SMERP supports the SAP ERP system and Add-On SMCRM supports the SAP CRM system.

The two standard application add-ons supported by the framework are SMERP Add-On for SAP ERP Systems and SMCRM Add-On for SAP CRM Systems. SMERP supports ERP-based mobile applications such as SAP Work Manager, SAP Inventory Manager, and SAP Rounds Manager. SMCRM supports CRM-based mobile applications such as SAP CRM Service Manager and SAP Sales Manager.

SAP Industry Solutions provides specialized industry specific business processes and features. To support these processes, specialized application add-ons have been developed. SMISU Add-On for SAP ERP Systems with IS-Utility is currently supported by the framework. It provides additional support for meter management integration services for SAP IS-Utility systems, in addition to the standard plant maintenance processes supported by the SMERP add-on. SMISU add-on has a dependency of SMERP add-on.

SAP Framework

The framework consists of core components. These components are logically grouped in three main layers:

- Change detection layer
- Business logic layer
- Integration layer

Within the framework, there also exists the configuration module and system monitor. The configuration module provides the interface to the integration framework to allow for the configuration and administration of the various components within it. The system monitor provides an administrative interface to monitor processing related to users, pushes, and other synchronization tasks within the framework.

Change Detection

The change detection layer of the Agentry SAP Framework contains the exchange tables and triggers used to detect and track changes made to the data in the SAP system, and stores exchange information used during synchronization between the mobile application and SAP. Included in the change detection layer are the following:

- **Mobile Exchange Persistent Layer** - Contains the exchange tables used during synchronization to compare data on the Client to data in the SAP system, and to support the synchronization of only the differences.
- **Delta Detection Routines** - Contains the triggers, created in the SAP enhancement framework, to detect changes to production data of importance to the mobile application.
- **Change Detection Configuration Set** - Configuration tools, including the interface presented in the Configuration portal, to allow for the creation, configuration, and administration of the components of the change detection layer.
- **Exchange Object** - Presented in the Configuration Panel, the exchange object encapsulates the SAP tables, exchange table, class handler, and other components involved in the change detection process. Exchange objects are utilized by the fetch process to determine what data has changed since the last transmit of the mobile device.

Business Logic Layer

The business logic layer contains the logical components to work with the exchange data objects in the change detection layer for downstream synchronization, as well as the logical components to update data from transactions in the mobile application to the SAP system. Included in the business logic layer are the following:

- **Application Data Filter Services** - Specifies the filtering of data to be retrieved from the SAP system for transmission down to the mobile application. Not all fields from a given table will necessarily be sent to the mobile application. The application data filter services allow for the specification of the specific fields from a table to be sent to the mobile application.

- **Data Object Class Handler Repository** - Contains the logic used by the mobile data objects to synchronize data. A class handler is created to retrieve data from or update data to the SAP system, making calls to the standard SAP BAPIs.
- **Application Authorization Services** - Contains the security settings specific to the mobile user group. Settings for authorization services can be applicable to users of all mobile applications synchronizing data through the Agentry SAP Framework, for a specific application, or for a specific class handler within the business logic layer.
- **Mobile Integration Configuration Set** - Configuration tools, including the interface presented in the Configuration Panel, to allow for the creation, configuration, and administration of the components of the business logic layer.
- **Mobile Data Object** - Presented in the Configuration Panel, the mobile data object encapsulates the class handler, exchange object (in the change detection layer), the application data filtering services, and other components involved in the synchronization of data. The mobile data object references the exchange object in the change detection layer to support its synchronization. It in turn is referenced by the BAPI wrappers within the integration layer.

Integration

The integration layer of the Agentry SAP Framework contains the BAPI wrappers that present the integration point to the mobile application servers. The Java steplets, complex table, and data table classes within the synchronization definitions of the mobile application call into these BAPI wrappers, through the SAP Java Connector (SAP JCo). The BAPI wrappers, in turn, call into the mobile data objects, which include the business logic for synchronization.

The architecture and design of the BAPI wrappers is intended to exclude the business logic related to the synchronization of the production data. This structure minimizes the necessity of modifying Java code within the mobile application as the result of changes to the synchronization processing configured in the integration of the Agentry SAP Framework.

Configuration Module

The configuration module is the web interface for the Agentry SAP Framework Configuration Panel. This interface provides the tools to create, configure, and administer the components of the integration of the Agentry SAP Framework.

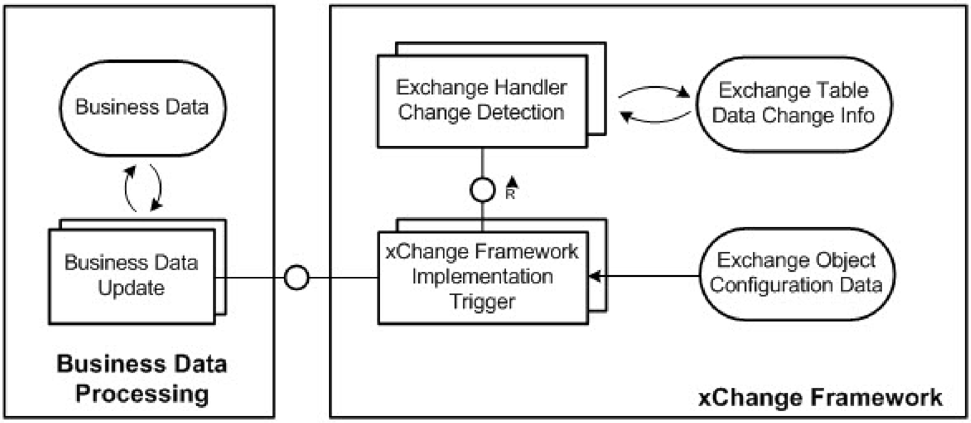
System Monitor

The system monitor is the web interface for the Agentry SAP Framework Administration and Monitoring Panel. This monitoring panel is provided to view mobile users’ synchronization activities, including fetches, transactions, and pushes.

Mobile Exchange Persistent Layer

The mobile exchange persistent layer allows the transfer of data from SAP to the mobile client as well as the transfer of data from the mobile client to SAP. This exchange process provides a consistent way to manage and capture mobile client-related master and transaction data changes within SAP. The mobile exchange persistent layer uses the SAP NetWeaver enhancement framework to implement change detection triggers.

Figure 3: System Diagram - xChange Framework


Business Data
Business Data Update
Business Data Processing
Exchange Handler Change Detection
Exchange Table Data Change Info
R
xChange Framework Implementation Trigger
Exchange Object Configuration Data
xChange Framework

Delta Detection Routine

The delta detection routines are the triggers within the enhancement framework that detect modifications to data of concern to the mobile application. When changes are made to production data in the back end system, the delta detection routines will capture the change event. If a given change is one that will affect the data in the mobile application, then the routine will update the appropriate exchange table in the mobile exchange persistent layer. Intelligence is incorporated in the delta detection routines to determine if a change is one of concern to the mobile application. If it is not, then the change is ignored by the routine.

Change Detection Configuration Set

The change detection configuration set controls how the change detection should be carried out by the mobile exchange persistent layer. Customer-defined configurations are protected through a reserved customer namespace, such as “Y*” or “Z*”. The change detection configuration set is mobile application-independent. However, it is a set of mobile application-specific change detection rules. The rules are configurable through the Agentry SAP Framework Configuration Panel.

Data Object Handler Class Repository

The data object class handler repository links BAPI wrappers with mobile data objects and encapsulates all business logic related to mobile applications. Internally standard BAPIs or custom logic can be implemented to provide optimal mobile support. The object-oriented design provides benefits such as reliability, reusability, extensibility and maintainability.

All transaction updates are performed through standard BAPIs in order to ensure data integrity. The repository also supports streamlining multiple SAP transactions or multiple BAPIs for improved business processes. The data objects within the repository are application-dependent and delivered in the relevant package based on SAP’s hierarchy, such as /SYCLO/MM or /SYCLO/PM.

The mobile data objects contained within the repository support result-set field selection, data filtering and security checks as defined by the integration rules. The mobile data objects also support monitoring and logging applications. Mobile data objects are configurable through the Agentry SAP Framework Configuration Panel. Different rules can be defined for every class handler for each mobile application.

Mobile Integration Configuration Set

The Mobile Integration Configuration Set within the business logic layer contains the user interface components presented in the Configuration Panel. The tools within this configuration set allow for the creation, configuration, and administration of the components within the business logic layer.

Application Authentication Services

Application authentication services allow you to perform additional mobile-related authentications that are not available using the standard SAP authentications and security profiles. These additional authorizations are configured using the Security Settings screen in the Configuration Panel.

BAPI Wrapper

The BAPI wrapper layer provides a consistent way to expose business logic and data from the SAP system to external mobile applications. The BAPI wrappers are decoupled from the business logic in SAP, making them easy to reconfigure or customize.

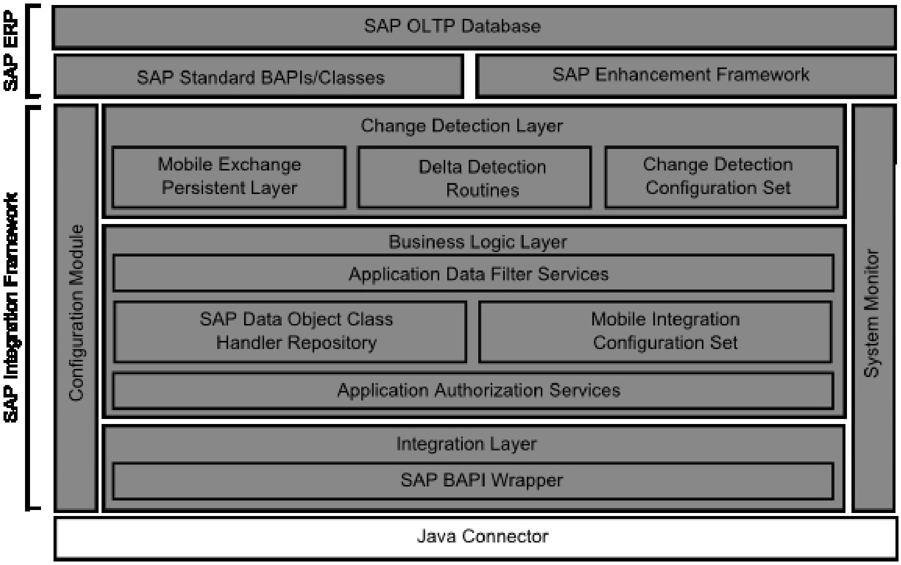
The standard BAPI wrapper library consists of a collection of custom developed BAPIs based on the SAP integration framework. BAPI wrappers improve interface consistency with standard naming conventions and development standards. The BAPI wrapper library is application-dependent and delivered in the relevant package based on SAP’s hierarchy, such as /SYCLO/MM or SYCLO/PM. BAPI wrappers set parameters for mobile data objects to post and pull data in the SAP system. When the Server calls to SAP, these calls are encapsulated in a BAPI wrapper to ensure that SAP responds in a consistent manner and performs the actions that Agentry SAP Framework is asking for.

The Configuration Panel is used to configure BAPI wrappers by assigning them to mobile data objects and handler methods (GET, CREATE, UPDATE, or DELETE), function groups, and packages within SAP.

Java Connector

The Java connector is a standard SAP Java connector. See the appropriate SAP documentation for more details.

System Diagram - Java Connector


SAP ERP
SAP Integration Framework
SAP OLTP Database
SAP Standard BAPIs/Classes
SAP Enhancement Framework
Configuration Module
Change Detection Layer
Mobile Exchange Persistent Layer
Delta Detection Routines
Change Detection Configuration Set
Business Logic Layer
Application Data Filter Services
SAP Data Object Class Handler Repository
Mobile Integration Configuration Set
Application Authorization Services
Integration Layer
SAP BAPI Wrapper
System Monitor
Java Connector

System Monitor

Administration and monitoring activities that take place behind the SAP Java connector for the Agentry SAP Framework are performed through the System Administration and Monitoring Panel. The Administration Panel is a problem-detection tool utilizing SAP’s standard application log database. It allows administrators to trace who is logging into the mobile applications, what work they’re doing, and the data being transferred between the mobile devices and the SAP system. The Administration Panel speeds problem resolution using real-time data and traceability.

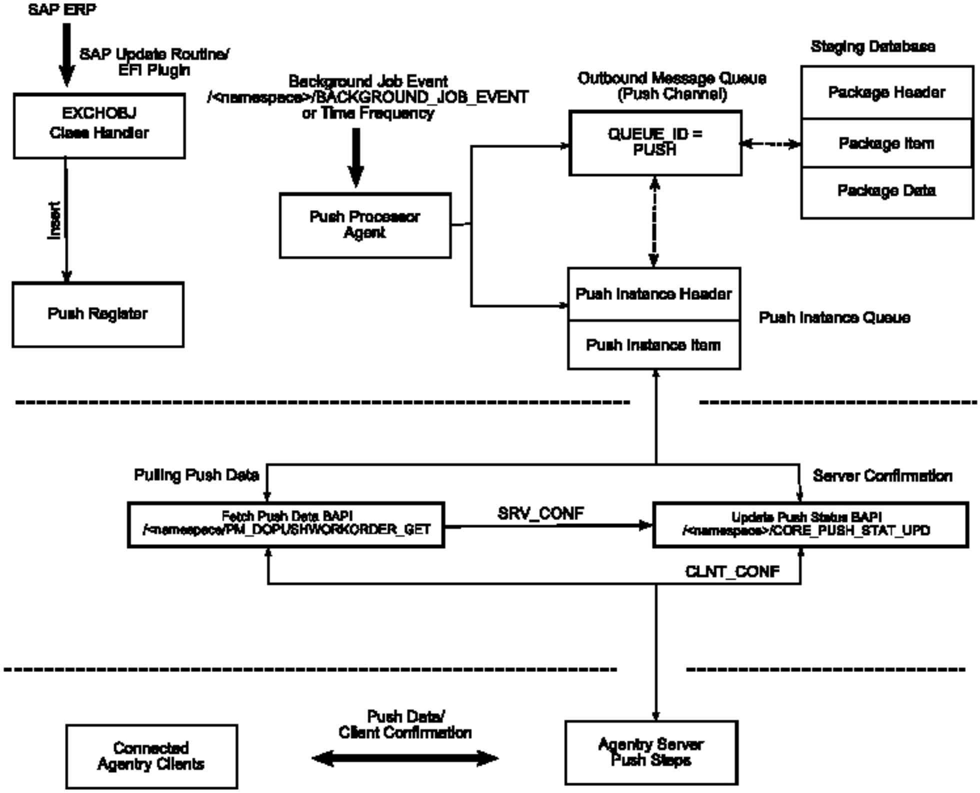
Configuration Module

The Configuration Module is the Agentry SAP Framework configuration web interface panel. The Configuration Panel simplifies modifications to existing mobile applications. It adds a layer of business logic for filtering at a system or user level. Data is mapped to the Agentry SAP Framework and pulled from SAP modules based on the user’s credentials in the SAP system. Therefore, administrators can allow remote workers to only navigate and see data relevant to their job location or function.

Working with Push Scenarios

A push scenario pushes emergency work orders to the corresponding recipients. Use the following diagram and steps to follow a push instance from generation in SAP to reception on the Client.

Push Process Flow


SAP ERP
SAP Update Routine/ EFI Plugin
EXCHOBJ Class Handler
Insert
Push Register
Background Job Event /<namespace>/BACKGROUND_JOB_EVENT or Time Frequency
Push Processor Agent
Outbound Message Queue (Push Channel)
QUEUE_ID = PUSH
Staging Database
Package Header
Package Item
Package Data
Push Instance Header
Push Instance Item
Push Instance Queue
Pulling Push Data
Server Confirmation
Fetch Push Data BAPI /<namespace/PM_DOPUSHWORKORDER_GET
SRV_CONF
Update Push Status BAPI /<namespace>/CORE_PUSH_STAT_UPD
CLNT_CONF
Connected Agentry Clients
Push Data/ Client Confirmation
Agentry Server Push Steps

1. The push exchange process initiates the push trigger based on the push conditions. Conditions are defined as filter rules in the push exchange object. For instance, `work order priority = 1` is considered an emergency work order in the base product release.
2. The work order that satisfies the push conditions inserts a record into the push register table /SYCLO/PSH01 with an object key as the work order number and a push status of NEW.

3. The event /SYCLO/BACKGROUND_JOB_EVENT is raised after the work order is saved, which triggers the background job for the push processer agent.
4. The push processer job /SYCLO/CORE_PUSH_PROC_PROG is triggered, either by using the event or the time frequency. This trigger is based on specific customer processes.
5. The push processer determines the recipients for the push work order and builds the data for each recipient as a separate instance. The instance is stored in the outbound message queue /SYCLO/PSH02 with queue ID = PUSH, using the staging database.
6. The push instance displays one of the following statuses, viewable in the push monitor in the Administration Panel:
   - NEW
   - PROCESS
   - CANCEL
   - COMPLETED
   - SRV_COMP
7. The Agentry application within the SAP Mobile Server calls the push BAPI /SYCLO/PM_DOPUSHWORKORDER_GET for every predefined time interval and checks the push queue for new items.
8. The Agentry application within the SAP Mobile Server sends the push data to the respective Clients depending on the user credentials that match the push instance.
9. Once the Client receives the push message, it sends the Client confirmation back to the Server and the Server calls the BAPI /SYCLO/CORE_PUSH_STAT_UPD to update the confirmation with status CLNT_CONF back to SAP.

Outbound Trigger Overview

An outbound trigger allows a mobile application to interface with external systems such as the Agentry application within the SAP Mobile Server from SAP. Outbound triggers can be integrated into standard mobile application processes, such as push processing. Different types of outbound triggers can be defined, such as an HTTP trigger, file trigger, Web service trigger, etc. You define an outbound trigger definition in the IMG activity.

Requirements

Outbound triggers are configured for each mobile application. Therefore, the mobile application must be defined first. The Outbound trigger handler must be developed before it can be assigned to a trigger. An outbound trigger handler should implement interface `/SYCLO/IF_CORE_OUTB_TRIGGER` and should be a subclass of `/SYCLO/CL_CORE_OTRIG_BASE`.

Agentry SAP Framework Administration Functions in SAP

All components of the Agentry SAP Framework administration in SAP, such as BAPI wrappers and mobile data objects, support logging. Activity logs generated by the Agentry SAP Framework are integrated into the standard SAP application log database.

The following are administration and monitoring functions available in SAP:

- Agentry SAP Framework Push Instance Purge Utility
- Agentry SAP Framework Data Cache Purge Utility
- Agentry SAP Framework Generic Purge Utility
- Agentry SAP Framework Exchange Table Purge Utility
- Agentry SAP Framework Subscription Queue Purge Utility
- Agentry SAP Framework Log Deletion
- Agentry SAP Framework Log Display
- Agentry SAP Framework Statistics Log Purge Utility

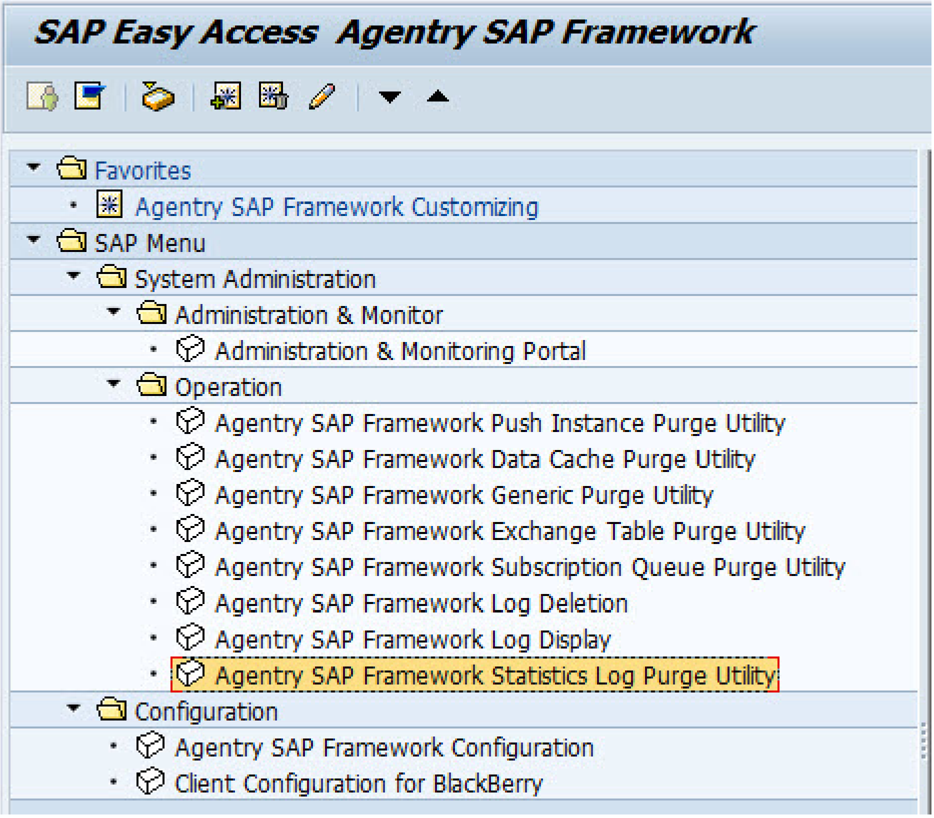
The manual process of purging data is important during configuration and modification in a development environment. Be sure to purge data after each test after reviewing it and starting the next test. This way, the data found in the logs is applicable to the newest test activities and will not cause confusion. Periodic purging of data also contributes to an optimum running environment.

Exchange table histories can be set to automatically delete after a set period of days through the Configuration Panel. All other data purges and log deletions must be done through SAP.

Accessing Administrative Functions in SAP

1. Log into SAP.
2. Type `/n/syclo/smart` into the command field and click the green checkmark to the left of the field, or press **Enter**.

   The Agentry SAP Framework Administration window displays. Expand the SAP menu tree by clicking the arrows to the left of the menu items. Expand as follows: **SAP menu > System Administration > Operation**. The available administrative functions display:


SAP Easy Access Agentry SAP Framework
Favorites
Agentry SAP Framework Customizing
SAP Menu
System Administration
Administration & Monitor
Administration & Monitoring Portal
Operation
Agentry SAP Framework Push Instance Purge Utility
Agentry SAP Framework Data Cache Purge Utility
Agentry SAP Framework Generic Purge Utility
Agentry SAP Framework Exchange Table Purge Utility
Agentry SAP Framework Subscription Queue Purge Utility
Agentry SAP Framework Log Deletion
Agentry SAP Framework Log Display
Agentry SAP Framework Statistics Log Purge Utility
Configuration
Agentry SAP Framework Configuration
Client Configuration for BlackBerry

3. Double-click on the desired administrative function to open the SAP window for that function.

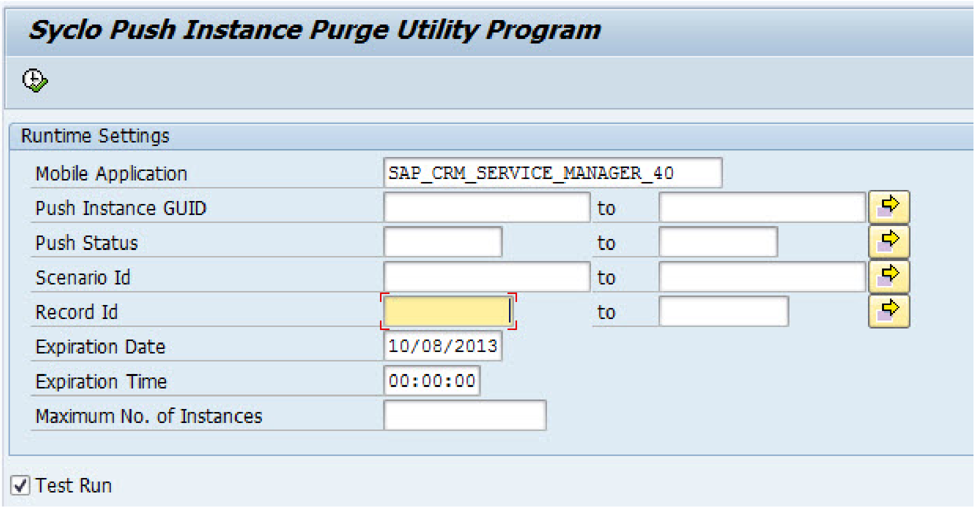
Agentry SAP Framework Push Instance Purge Utility

The push instance purge utility is used to purge exchange table information that was used during the push processing. Once the push data is pushed to Client devices, the exchange information is no longer needed and can be purged.

Note: The purge utility in SAP performs a manual purge. Automatic purges of push instances are configured in the Push Scenario Definition panel of the Configuration Panel.

When desired fields are filled, click on the clock icon in the upper left to execute the purge.

SAP Administration - Push Instance Purge Utility Program



Syclo Push Instance Purge Utility Program

Runtime Settings

Mobile Application	SAP_CRM_SERVICE_MANAGER_40		
Push Instance GUID		to	
Push Status		to	
Scenario Id		to	
Record Id		to	
Expiration Date	10/08/2013		
Expiration Time	00:00:00		
Maximum No. of Instances			

☑ Test Run

Runtime Settings

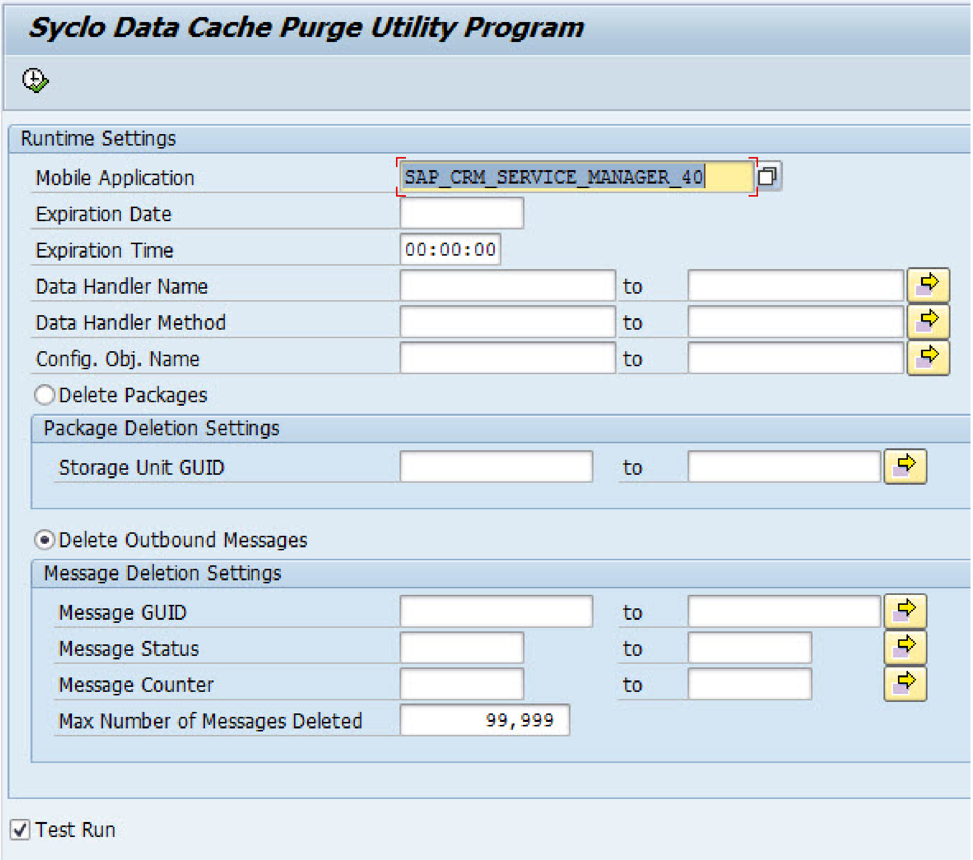
- **Mobile Application:** Click the box icon to the right of the Mobile Application field to bring up a window displaying all mobile application choices.
- **Push Instance GUID:** Use the Push Instance GUID range fields to select GUIDs contained within the push instance table in SAP to purge.
- **Push Status:** Use the Push Status range fields to select statuses contained within the push instance table in SAP to purge. By default, all statuses are included if the fields are not filled in. The push statuses are as follows:
  - NEW
  - PROCESS
  - SRV_COMP
  - COMPLETED
  - CANCEL
- **Scenario ID:** Use the Scenario ID range fields to select scenario IDs contained within the push instance table to purge.
- **Record ID:** Use the Record ID range fields to select record IDs contained within the push instance table to purge.
- **Expiration Date:** Use the Expiration Date field to choose a date when the purge utility operation will expire. The expiration date is automatically filled with the current date.
- **Expiration Time:** Use the Expiration Time field to choose a time when the purge utility operation will expire.
- **Maximum No. of Instances:** Type in the maximum number of instances to purge from the instance table.

- **Test Run:** When this box is checked, records are not purged upon execution. Rather, a list of the records selected for purging displays, in order to determine that the purge parameters are correct. Once records are purged, they cannot be recovered.

Agentry SAP Framework Data Cache Purge Utility

Use the data cache purge utility to manage and purge packages associated with messages and the outbound message queue from SAP. The outbound message queue is the only way SAP communicates with the Client device. Packages are the data attached to outbound messages. In this way, an administrator can send a message to multiple users while referencing only one data package, rather than an individual data package for each outbound message.

SAP Administration - Data Cache Purge Utility Program



Syclo Data Cache Purge Utility Program

Runtime Settings

Mobile Application	SAP_CRM_SERVICE_MANAGER_40		
Expiration Date			
Expiration Time	00:00:00		
Data Handler Name		to	
Data Handler Method		to	
Config. Obj. Name		to	

○ Delete Packages

Package Deletion Settings

Storage Unit GUID		to	

● Delete Outbound Messages

Message Deletion Settings

Message GUID		to	
Message Status		to	
Message Counter		to	
Max Number of Messages Deleted	99,999		

☑ Test Run

Runtime Settings

- **Mobile Application:** Select the desired mobile application with which the outbound messages or packages are associated.

- **Expiration Date:** Use the Expiration Date field to choose a date when the purge utility operation will expire. When used with Expiration Time, this is the maximum expiration timestamp from which to purge messages from the cache (i.e.; purge messages that are set to expire 2013.10.06 at 3 p.m. or earlier). If an expiration date / time are provided, they are converted into a timestamp. The utility will only delete those messages whose expiration, or timestamp, is less than or equal to the provided timestamp. If this timestamp is not provided, deletion will occur without respect to the messages' expiration.
- **Expiration Time:** Use the Expiration Time field to choose a time when the purge utility operation will expire. When used with Expiration Date, this is the maximum expiration timestamp from which to purge messages from the cache (i.e.; purge messages that are set to expire 2013.10.06 at 3 p.m. or earlier). If an expiration date / time are provided, they are converted into a timestamp. The utility will only delete those messages whose expiration, or timestamp, is less than or equal to the provided timestamp. If this timestamp is not provided, deletion will occur without respect to the messages' expiration.
- **Data Handler Name:** Use the Data Handler Name range fields to select data handler names contained within the data cache to purge. Data handlers are responsible for messages within packages to be purged. It limits the purge to the class handler responsible for handling the message.
- **Data Handler Method:** Use the Data Handler Method range fields to select data handler methods contained within the data cache to purge. Data handlers are responsible for messages within packages to be purged. It limits the purge to the method of a class handler responsible for handling the message.
- **Configuration Object Name:** Use the Configuration Object Name range fields to select configuration object names contained within the data cache to purge. Limits the purge to the configuration object that owns, or is the source of, the message.

Package Deletion Settings

Select the Delete Packages option to purge packages stored in SAP.

Note: If a data package is associated with a message GUID, it cannot be deleted.

- **Delete Packages:** Select this radio button if you wish to delete the packages found through the Package Deletion Settings.
- **Storage Unit GUID:** Each package is associated with a unique storage unit GUID

Message Deletion Settings

Select the Delete Outbound Messages option to purge outbound messages stored in SAP.

Note: All packages associated with outbound messages will also be deleted if the Delete Outbound Messages option is chosen.

- **Message GUID:** Each outbound message is associated with a unique message GUID
- **Message Status:** Use the following available message statuses:
  - NEW

  - SEND
  - RECEIVED
  - CANCEL
  - CONFIRMED
- **Message Status:** Use the Message status range fields to select message statuses contained within the data cache to purge.
- **Message Counter:** Use the Message Counter range fields to select message counters contained within the data cache to purge.

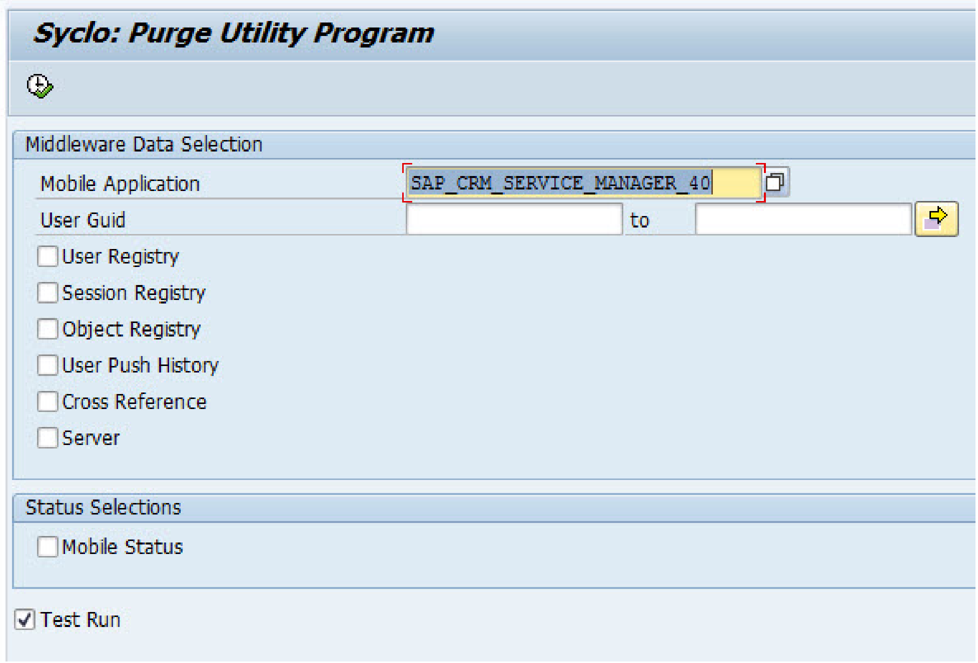
Test Run

When this box is checked, packages or messages are not purged upon execution. Rather, a list of the packages or records selected for purging appears, in order to determine that the purge parameters are correct. Once packages or records are purged, they cannot be recovered.

Agentry SAP Framework Generic Purge Utility

Use the generic purge utility to delete records in SAP pertaining to specific user IDs or middleware server records.

SAP Administration - Purge Utility Program



Syclo: Purge Utility Program

Middleware Data Selection

Mobile Application: SAP_CRM_SERVICE_MANAGER_40

User Guid: to

□ User Registry

□ Session Registry

□ Object Registry

□ User Push History

□ Cross Reference

□ Server

Status Selections

□ Mobile Status

☑ Test Run

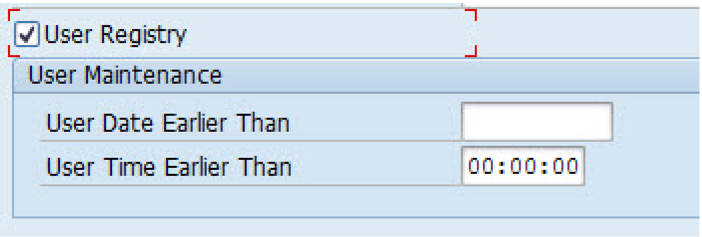
When a box in the MDW Selections field is checked, additional fields for the selection appear, allowing specific purging criteria to be set for the selection.

- **Mobile Application:** Select the desired mobile application with which the middleware server records are associated.
- **User GUID:** Use the User GUID range fields to select user GUIDs contained within the middleware server records to purge.

User Registry

When the User Registry box is checked, all records pertaining to that user ID, such as session and object records, are also purged. Each mobile user ID is a user GUID in SAP. Mobile users need a separate user GUID for each mobile application they use.

Generic Purge Utility - User Registry



☑ User Registry

User Maintenance

User Date Earlier Than

User Time Earlier Than 00:00:00

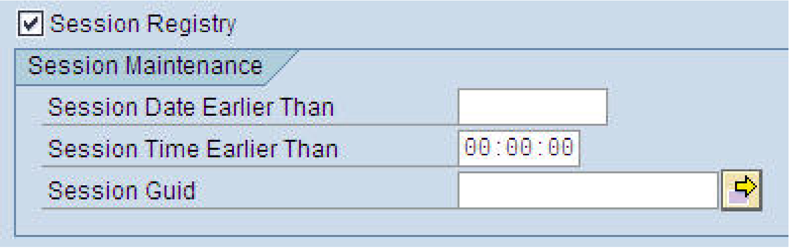
Check the User Registry box to purge user records according to the following criteria:

- **User Date Earlier Than:** Highlight the field and use the calendar to select a date to purge user records existing before the selected date.
- **User Time Earlier Than:** Highlight the field and use the time display window to select a time to purge user records existing before that time.

Session Registry

Every time a user ID connects to SAP, a session record is created. Each session record has a GUID, a start time stamp and an end time stamp.

Generic Purge Utility - Session Registry



☑ Session Registry

Session Maintenance

Session Date Earlier Than

Session Time Earlier Than 00:00:00

Session Guid

Check the Session Registry box to purge session records according to the following criteria:

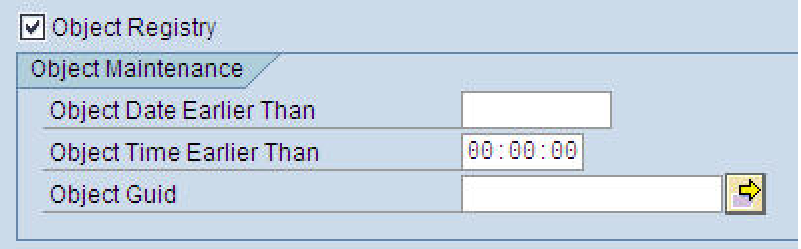
- **Session Date Earlier Than:** Highlight the field and use the calendar to select a date to purge session records created before the selected date.

- **Session Time Earlier Than:** Highlight the field and use the time display window to select a time to purge session records created before that time.
- **Session GUID:** Either type in the GUID or use the Multiple Selection icon to the right of the field to select a range of GUIDs.

Object Registry

Each object on each Client device has a unique GUID contained within the object registry table. The object registry is used by the fetch process to calculate what data is needed by each Client device based on change detections since last transmit.

Generic Purge Utility - Object Registry



☑ Object Registry

Object Maintenance

Object Date Earlier Than	
Object Time Earlier Than	00:00:00
Object Guid	

Check the Object Registry box to purge object records according to the following criteria:

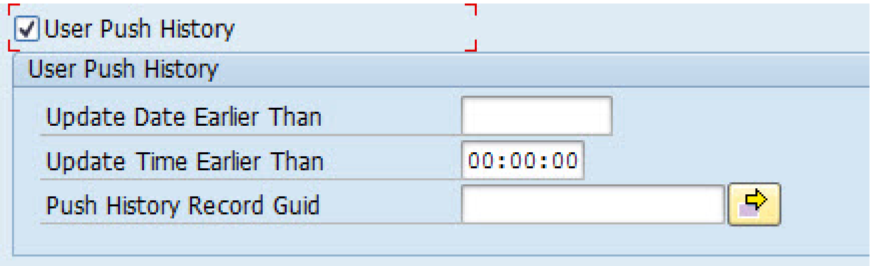
- **Object Date Earlier Than:** Highlight the field and use the calendar to select a date to purge object records created before the selected date.
- **Object Time Earlier Than:** Highlight the field and use the time display window to select a time to purge object records created before that time.
- **Object GUID:** Either type in the GUID or use the Multiple Selection icon to the right of the field to select a range of GUIDs.

User Push History

When push history is enabled, every time an object is pushed to a user it is recorded to a history table (/SYCLO/PSH05). The user push history purge utility removes old history records, or specific history records, by specifying a date/time or record GUIDs from which to purge.

Note: In order for history records to appear, the push (distribution) handler has to support the history function.

Generic Purge Utility - User Push History



☑ User Push History

User Push History

Update Date Earlier Than	
Update Time Earlier Than	00:00:00
Push History Record Guid	

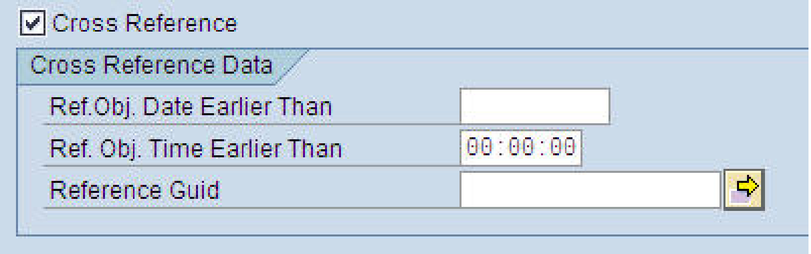
Check the User Push History box to purge object records according to the following criteria:

- **Update Date Earlier Than:** Highlight the field and use the calendar to select a date to purge user push history created before the selected date.
- **Update Time Earlier Than:** Highlight the field and use the time display window to select a time to purge user push history created before that time.
- **Push History Record GUID:** Either type in the GUID or use the Multiple Selection icon to the right of the field to select a range of GUIDs.

Cross Reference

The reference tables contained within SAP facilitate the key mapping process for Agentry applications. When a user creates a local object with a temporary ID and transmits to the system, the object is assigned an object GUID. Agentry can also break down a document into smaller chunks of information, each with its own reference GUID mapped to the object GUID.

Generic Purge Utility - Cross Reference



☑ Cross Reference

Cross Reference Data

Ref.Obj. Date Earlier Than	
Ref. Obj. Time Earlier Than	00:00:00
Reference Guid	

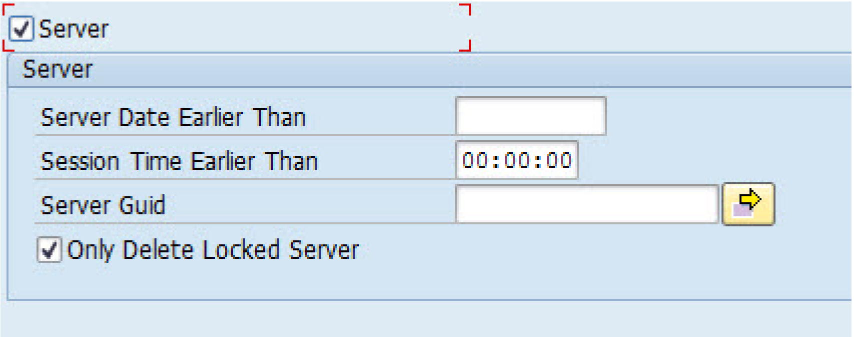
Check the Cross Reference box to purge cross references according to the following criteria:

- **Ref. Obj. Date Earlier Than:** Highlight the field and use the calendar to select a date to purge reference object records created before the selected date.
- **Ref. Obj. Time Earlier Than:** Highlight the field and use the time display window to select a time to purge reference object records created before that time.
- **Reference GUID:** Either type in the GUID or use the Multiple Selection icon to the right of the field to select a range of GUIDs.

Server

A system can contain multiple middleware servers within the system. Each server has a unique GUID associated with it.

Generic Purge Utility - Server



☑ Server

Server

Server Date Earlier Than

Session Time Earlier Than 00:00:00

Server Guid

☑ Only Delete Locked Server

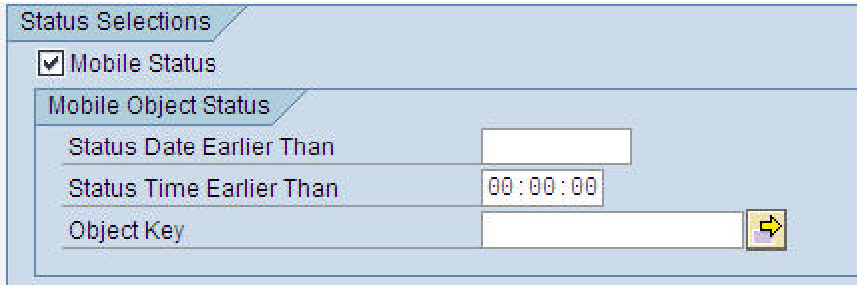
Check the Server box to purge server records according to the following criteria:

- **Server Date Earlier Than:** Highlight the field and use the calendar to select a date to purge server session records created before the selected date.
- **Session Time Earlier Than:** Highlight the field and use the time display window to select a time to purge server session records created before that time.
- **Server GUID:** Either type in the GUID or use the Multiple Selection icon to the right of the field to select a range of GUIDs.
- **Only Delete Locked Server:** Check this box if you only want to purge server records on the locked server.

Mobile Status

Whenever a technician changes the status of an object and transmits from the mobile device, the status management tables in SAP are updated. Each object has a unique object key associated with that object.

Generic Purge Utility - Status Selections



Status Selections

☑ Mobile Status

Mobile Object Status

Status Date Earlier Than

Status Time Earlier Than 00:00:00

Object Key

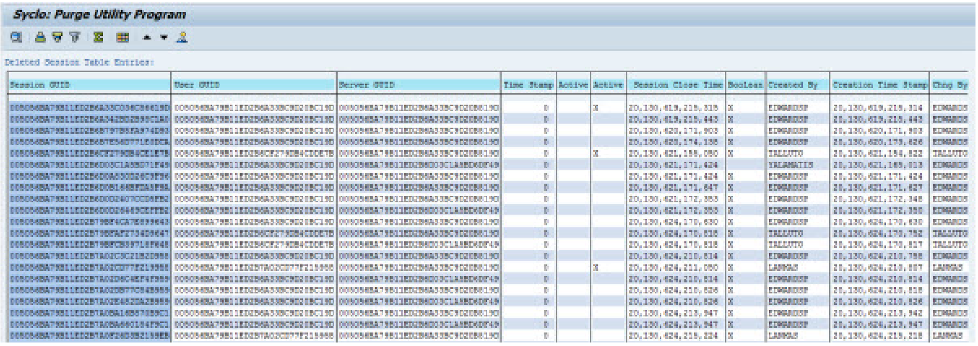
Check the Mobile Status box to purge mobile object records according to the following criteria:

- **Status Date Earlier Than:** Highlight the field and use the calendar to select a date to purge mobile status records created before the selected date.
- **Status Time Earlier Than:** Highlight the field and use the time display window to select a time to purge mobile status records created before that time.
- **Object Key:** Either type in the object key or use the Multiple Selection icon to the right of the field to select a range of object keys.

Test Run

Check the Test Run box in order to have SAP display another window that shows in table format the records that will be deleted based on the criteria selected in the Purge Utility Program screen. Once records are purged from the system, they cannot be recovered.

Generic Purge Utility - Deleted Session Table Entries Table



Sycło: Purge Utility Program

Deleted Session Table Entries:

Session GUID	User GUID	Server GUID	Time Stamp	Active	Active	Session Close Time	Boolean	Created By	Creation Time Stamp	Chng By
005056BA79B11ED2B6A33C036C86619D	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0		X	20,130,619,215,315	X	EDWARDSP	20,130,619,215,314	EDWARDS
005056BA79B11ED2B6A3425D2B95C1A0	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,619,215,443	X	EDWARDSP	20,130,619,215,442	EDWARDS
005056BA79B11ED2B6B7978BFA974D93	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,620,171,909	X	EDWARDSP	20,130,620,171,903	EDWARDS
005056BA79B11ED2B6B7E54D771E8DCA	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,620,174,138	X	EDWARDSP	20,130,620,173,626	EDWARDS
005056BA79B11ED2B6CF279DB4CE1E7B	005056BA79B11ED2B6CF279DB4CDDE7B	005056BA79B11ED2B6A33BC9D2DB819D	0		X	20,130,621,155,050	X	TALLUTO	20,130,621,154,822	TALLUTO
005056BA79B11ED2B6D03C1A5BD71F49	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,621,171,424		TALAMATIS	20,130,621,165,013	EDWARDS
005056BA79B11ED2B6D0A830D26C3F96	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,621,171,424	X	EDWARDSP	20,130,621,171,424	EDWARDS
005056BA79B11ED2B6D0B1648FDA5F9A	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,621,171,647	X	EDWARDSP	20,130,621,171,627	EDWARDS
005056BA79B11ED2B6D0D2407CCD8FB2	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,621,172,353	X	EDWARDSP	20,130,621,172,348	EDWARDS
005056BA79B11ED2B6D0D26469CEFFB2	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,621,172,353	X	EDWARDSP	20,130,621,172,350	EDWARDS
005056BA79B11ED2B798F4CA7E899643	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,170,630	X	EDWARDSP	20,130,624,170,630	EDWARDS
005056BA79B11ED2B798FAF273429647	005056BA79B11ED2B6CF279DB4CDDE7B	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,170,818	X	TALLUTO	20,130,624,170,752	TALLUTO
005056BA79B11ED2B798FCB5971EF648	005056BA79B11ED2B6CF279DB4CDDE7B	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,624,170,818	X	TALLUTO	20,130,624,170,817	TALLUTO
005056BA79B11ED2B7A02C9C21B2D998	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,210,814	X	EDWARDSP	20,130,624,210,788	EDWARDS
005056BA79B11ED2B7A02C077F219998	005056BA79B11ED2B7A02C077F219998	005056BA79B11ED2B6A33BC9D2DB819D	0		X	20,130,624,211,050	X	LAMMAS	20,130,624,210,887	LAMMAS
005056BA79B11ED2B7A02D6C4EF4F999	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,624,210,814	X	EDWARDSP	20,130,624,210,814	EDWARDS
005056BA79B11ED2B7A02DB77C848999	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,210,826	X	EDWARDSP	20,130,624,210,818	EDWARDS
005056BA79B11ED2B7A02E4820A28999	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,624,210,826	X	EDWARDSP	20,130,624,210,826	EDWARDS
005056BA79B11ED2B7A0BA14B67B99C1	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,213,947	X	EDWARDSP	20,130,624,213,942	EDWARDS
005056BA79B11ED2B7A0BA6601B4F9C1	005056BA79B11ED2B6A33BC9D2DBC19D	005056BA79B11ED2B6D03C1A5BD6DF49	0			20,130,624,213,947	X	EDWARDSP	20,130,624,213,947	EDWARDS
005056BA79B11ED2B7A0F26D3B2199EB	005056BA79B11ED2B7A02C077F219998	005056BA79B11ED2B6A33BC9D2DB819D	0			20,130,624,215,224	X	LAMMAS	20,130,624,215,218	LAMMAS

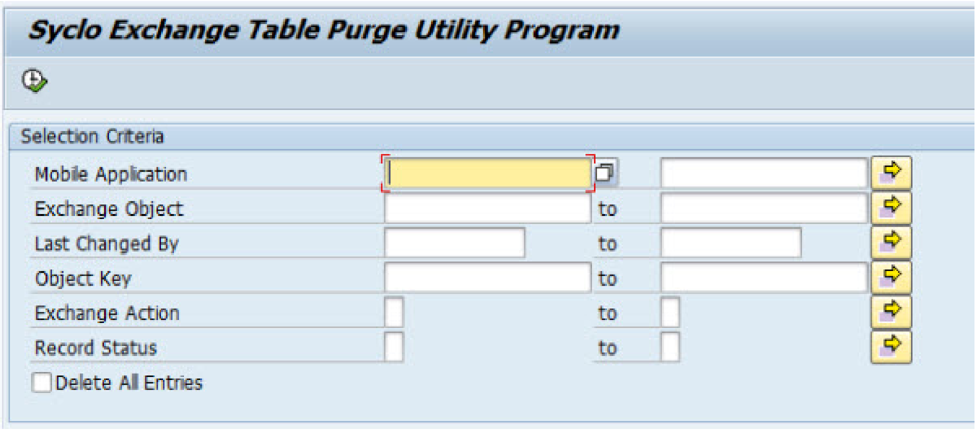
Agentry SAP Framework Exchange Table Purge Utility Program

Use the purge utility for the exchange tables to purge obsolete exchange objects from one or more mobile applications in SAP. Obsolete records are determined based on the purge frequency configured in the Agentry SAP Framework Configuration Panel.

To determine the purge frequency in the Agentry SAP Framework Configuration Panel, navigate to the Exchange Object Configuration menu item under the Mobile Application Settings section. In the tab **Technical Settings**, set the Days to Keep History field to the desired number of days and click the **Save** button.

When field selection is complete, click on the Clock icon at the top left of the screen to execute the exchange table purge.

SAP Administration - Exchange Table Purge Utility Program



Syclo Exchange Table Purge Utility Program

Selection Criteria

Mobile Application			
Exchange Object		to	
Last Changed By		to	
Object Key		to	
Exchange Action		to	
Record Status		to	

□ Delete All Entries

Selection Criteria

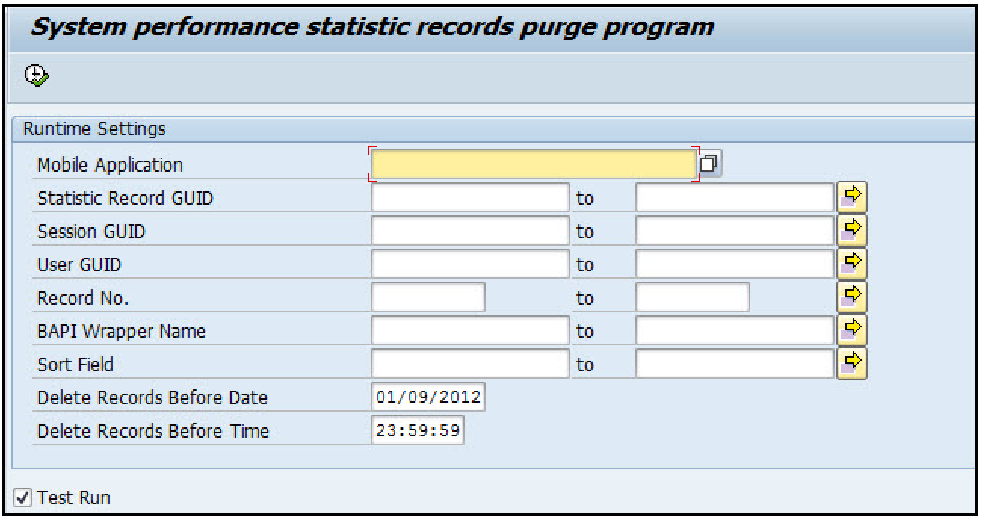
- **Mobile Application:** Use the mobile application range fields to select one or more mobile applications from which to purge objects from the exchange tables.

  Note: Selection choices may vary depending on which mobile applications are available on the system. If there is only one mobile application, this field does not apply.

- **Exchange Object:** Use the exchange object range fields to select one or more exchange objects from which to purge tables.
- **Last Changed By:** Use the range fields to select one or more user names who made changes from which to purge tables.
- **Object Key:** Key of the exchanged object. This ID is governed by how the field OBJKEY is populated in the relevant exchange table. The keys will vary from object to object.
- **Exchange Action:** Use I for Insert, **U** for Update, and/or **D** for Delete.
- **Record Status: [blank]** is used for the standard purge, **D** = Delete, and **S** = Skip.
- **Delete All Entries:** Select this checkbox to purge all obsolete records from the exchange tables, regardless of the application or exchange objects.

Agentry SAP Framework Statistics Record Purge Utility

Use the statistics record purge utility to delete statistic records generated by the SAP integration framework. Statistic records for mobile applications and individual mobile users can be purged selectively.



System performance statistic records purge program

Runtime Settings

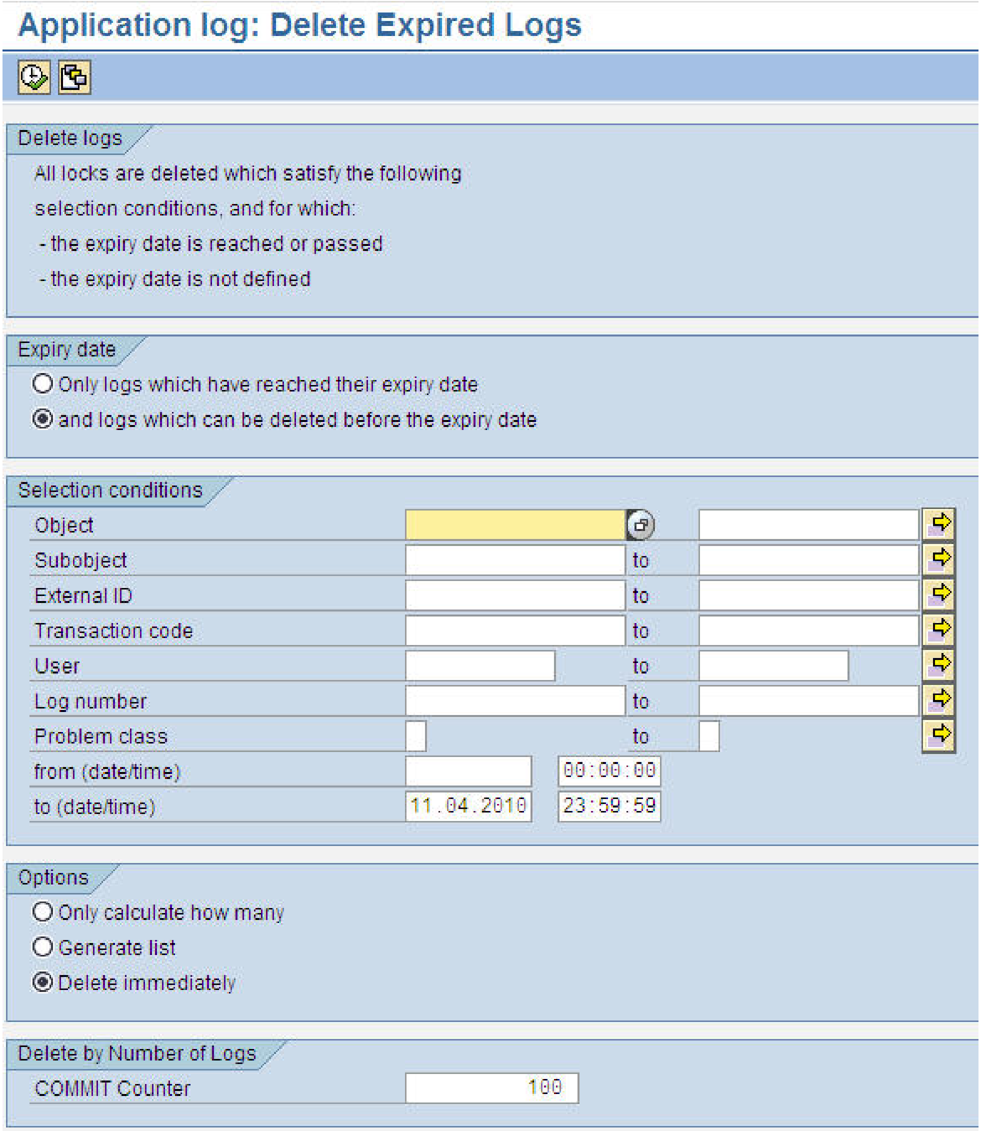
Mobile Application				
Statistic Record GUID		to		
Session GUID		to		
User GUID		to		
Record No.		to		
BAPI Wrapper Name		to		
Sort Field		to		
Delete Records Before Date	01/09/2012			
Delete Records Before Time	23:59:59			

☑ Test Run

Agentry SAP Framework Log Deletion

Use the log deletion function to delete expired logs from SAP. When desired fields are filled, click on the Clock icon to execute the deletion.

SAP Administration - Delete Expired Logs



Application log: Delete Expired Logs

Delete logs

All locks are deleted which satisfy the following selection conditions, and for which:

- the expiry date is reached or passed
- the expiry date is not defined

Expiry date

○ Only logs which have reached their expiry date

● and logs which can be deleted before the expiry date

Selection conditions

Object			
Subobject		to	
External ID		to	
Transaction code		to	
User		to	
Log number		to	
Problem class		to	
from (date/time)		00:00:00	
to (date/time)	11.04.2010	23:59:59	

Options

○ Only calculate how many

○ Generate list

● Delete immediately

Delete by Number of Logs

COMMIT Counter 100

Expiry Date

Click the first radio button to only delete logs which have reached their expiry date. Click the second radio button to delete logs that have reached their expiry date as well as logs that can be deleted before their expiry date has passed.

Selection Conditions

- **Object:** Select the desired object by either typing it in the field or clicking on the selection icon. The object is usually set to **/syclo/**.
- **Subobject:** Select from the following subobjects if the main object is `/syclo/`:
  - **ADMIN -** Administration portal logs
  - **BAPI -** Client application logs
  - **CONFIG -** Configuration portal logs
  - **DEFAULT -** All logs not covered through the rest of the subobjects
  - **EXCHANGE -** Exchange and transaction process logs
- **External ID:** Each log entry has an external ID, depending on what the log is for.
- **Transaction Code:** Not used
- **User:** SAP user ID that created the logs
- **Log Number:** Each log entry has a log number associated with it. If the log number or range of numbers is known, enter them here.
- **Problem Class:** The problem class of the logs
- **from (date/time):** Select the beginning start date and time of the logs to delete by clicking in the fields and using the calendar and the time window display to choose the correct date and time.
- **to (date/time):** Select the end date and time of the logs to delete by clicking in the fields and using the calendar and the time window display to choose the correct date and time.

Options

- **Only calculate how many:** Click this radio button to generate a popup window with the number of table logs that fit the criteria for deletion. Selecting this option does not result in any log deletion.
- **Generate list:** Click this radio button to generate a list of which table logs fit the field criteria for deletion. This list opens a different SAP window with multiple options for working within the list, including individual selection of table logs to delete. Selecting this option does not result in any log deletion.
- **Delete immediately:** Click this radio button to delete all table logs immediately. The system will still confirm the deletion of logs in a popup window before permanently deleting them.

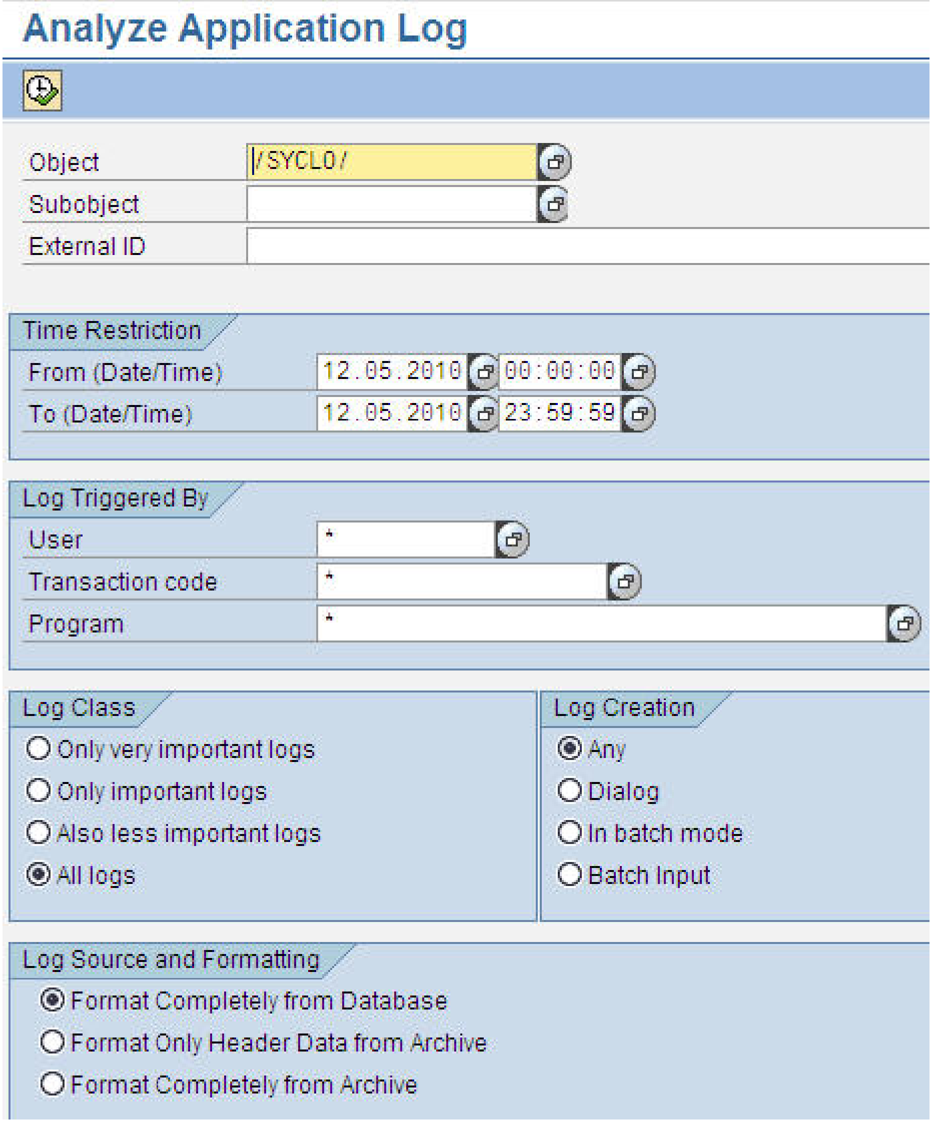
Delete by Number of Logs

- **COMMIT Counter:** Type in the number of logs desired to delete.

Agentry SAP Framework Log Display

Use the log display utility to view activity logs based on the criteria selected on the main screen. When all desired criterion are selected, click on the Clock icon at the top of the screen to execute the request and display the desired logs.

SAP Administration - Analyze Application Log



Analyze Application Log

Object: /SYCLO/

Subobject:

External ID:

Time Restriction

From (Date/Time): 12.05.2010 00:00:00

To (Date/Time): 12.05.2010 23:59:59

Log Triggered By

User: *

Transaction code: *

Program: *

Log Class

○ Only very important logs

○ Only important logs

○ Also less important logs

● All logs

Log Creation

● Any

○ Dialog

○ In batch mode

○ Batch Input

Log Source and Formatting

● Format Completely from Database

○ Format Only Header Data from Archive

○ Format Completely from Archive

- **Object:** Select the desired object by either typing it in the field or clicking on the selection icon to the right of the field. The object is usually set to `/syclo/` to view logs.
- **Subobject:** Select from the following subobjects if the main object is `/syclo/`:
  - **ADMIN -** Administration portal logs
  - **BAPI -** Client application logs
  - **CONFIG -** Configuration portal logs

  - **DEFAULT -** All logs not covered through the rest of the subobjects
  - **EXCHANGE -** Exchange and transaction process logs
- **External ID:** Type the external ID into the field.

Time Restriction

- **From (Date/Time):** Click on the white square icons to the right of the date and time fields to select a start date and time of the beginning of the logs chosen for display. The date and time are automatically set for the current date at 00:00:00 hours.
- **To (Date/Time):** Click on the white square icons to the right of the date and time fields to select an end date and time of the final logs chosen for display. The date and time are automatically set for the current date at 23:59:59 hours.

Log Triggered By

- **User:** SAP user ID
- **Transaction Code:** Standard SAP transaction codes
- **Program:** Standard SAP programs

Log Class

Select the appropriate standard SAP log class. Log classifications are based on the implementation by the developer.

Log Creation

Select the appropriate log creation setting. These are standard SAP settings based on the implementation by the developer.

Log Source and Formatting

Select the appropriate log source and formatting setting. These are standard SAP settings based on the implementation by the developer.

Enable SAP Solution Manager to Diagnose Agentry Issues

You can use the SAP Solution Manager to diagnose issues with the Agentry SAP Framework.

To enable this **SAP Solution Manager 7 EhP 1** must be installed.

- OSS Note 1371097 Diagnostics Setup for Agentry Servers
- End-To-End Root Cause Analysis System Landscape Setup Guide
  available here:

```
https://service.sap.com/~sapidb/011000358700000074392009E
```

Reporting Issues Using SAP Service Marketplace

You can report issues with the Agentry SAP Framework using the SAP Service Marketplace. Customer issues entered into the SAP Service Marketplace are automatically sent to SAP's Technical Support team.

Java Development for SAP

In order to modify the communications between the SAP Agentry Server and the SAP system, it is necessary to implement a Java development environment. This section contains some information to aid the developer or implementor in creating this environment.

Set Up the SAP Java Project in Eclipse

Once the Java IDE has been installed, you must set up the development or build project that will allow you to modify, compile, and debug the Java portion of the SAP application.

The following items must be a part of your Java development project:

- The Java source files for the SAP application
- The jCo.jar file
- The Agentry-level *.class files, provided with the SAP Agentry Server in the sub-directory `Java\Syclo\Agentry`. All of the *.class files in this folder should be part of the project.
- Optionally, the junit.jar file, provided with the Eclipse Java IDE.

- Retrieve the Java source files for the SAP application by contacting SAP's Customer Support. Included in these files will be the Java source files for all of the Java classes used by the SAP. These include the steplet classes used in the Java step definitions, and the classes used by the data tables and complex tables of the application. When creating the project in your Java IDE, the Java source files from SAP should be imported to the development project in the IDE.
- The Java files provided by SAP are the source files for your project. The *.class files must also be a part of the project's build path. These are the Java files that make up the Agentry Java API. They are installed with the SAP and can be referenced from the Server's installation location, or they can be copied to a location convenient to the IDE.
- The jCo.jar file for the SAP Java Connector toolkit must be a part of the build path for the project. This can be referenced in the SAP Agentry Server location, or at some other location convenient to the Java IDE. If the Editor is installed to a separate host system from that of the SAP Agentry Server, the JCo.jar file must be extracted from the Java Connector ZIP file. It can be placed in any convenient location. Be sure to note this location as it will be needed in the Java project created in the IDE.
- SAP recommends that you include this junit.jar file in your build path if you are using the Eclipse IDE. The junit.jar file is provided with Eclipse and is a library of Java classes used for debugging purposes, several of which are implemented in the SAP Java classes. If this

is not included in your project, you will receive warnings related to these classes at build time. The junit.jar file can be found in the Eclipse install directory at:

```
C:\eclipse\plugins\org.junit_4.2.8
```

Java Architecture

The Java back end uses classes of type SAPObject to represent data objects sent to and from the client. Most of these classes are POJOS (Plain Old Java Objects), meaning they store data in fields, provide accessory/mutator methods, and know how to construct themselves when told to by other code.

SAPObjects

SAPObjects can be composed of other SAPObjects stored as arrays of SAPObjects (e.g., Notifications have NotificationItems, etc). Since fetch BAPIs will now bring down child objects as well, the logic will populate the children using a single BAPI call without the need for read steps.

StepHandler

Stephandler classes provide the calling interface to classes subclassing the Agentry Java API (steplets, data tables, complex tables). Methods in StepHandler classes should be static. StepHandler methods also provide an interface for JUnit test suites.

All of this metadata is encapsulated in the SAPObject class rather than in an external file.

BAPI

The BAPI class encapsulates all of the BAPI processing needed by the Java back end. It is abstract because there are specific kinds of BAPIs:

- **FetchBAPI:** FetchBAPIs are the type of BAPIs that create SAPObjects from the exchange process. The SAPObjects are then parsed by Agentry for use on the client.
- **DataTableBAPI**
- **ComplexTableBAPI**
- **TransactionBAPI:** TransactionBAPIs take an Agentry transaction on the client and turn it into one or more JCO.Tables, then run the BAPI passing the tables in order to update SAP.

The BAPI objects that the code will use will be subclasses of FetchBAPI, TransactionBAPI, DataTableBAPI or ComplexTableBAPI. Developers need to write these BAPI classes to call the BAPIs to do the things that the “application” needs to do. A developer-written BAPI is a specific ABAP function call for a particular application (i.e., fetch all work orders for a user, send up a locally added notification, etc.).

Note that there is not necessarily a one-to-one correspondence between BAPI classes and ABAP function names (e.g. `/SYCLO/MM_DOPHYSINVDOC_GET`).

BAPI classes do the following:

- Create themselves out of the JCO.Repository when necessary. This is just before they are about to be called, but a developer might make these poolable/cacheable.
- Set input values in JCO.Table parameters, such as search ranges in the case of FetchBAPIs or transaction data in the case of TransactionBAPIs.
- Are executed when called (Get the results or post the data to SAP).
- Report exceptions and errors back to calling code. This includes reading return tables and parsing for error messages when necessary.

Data Flow

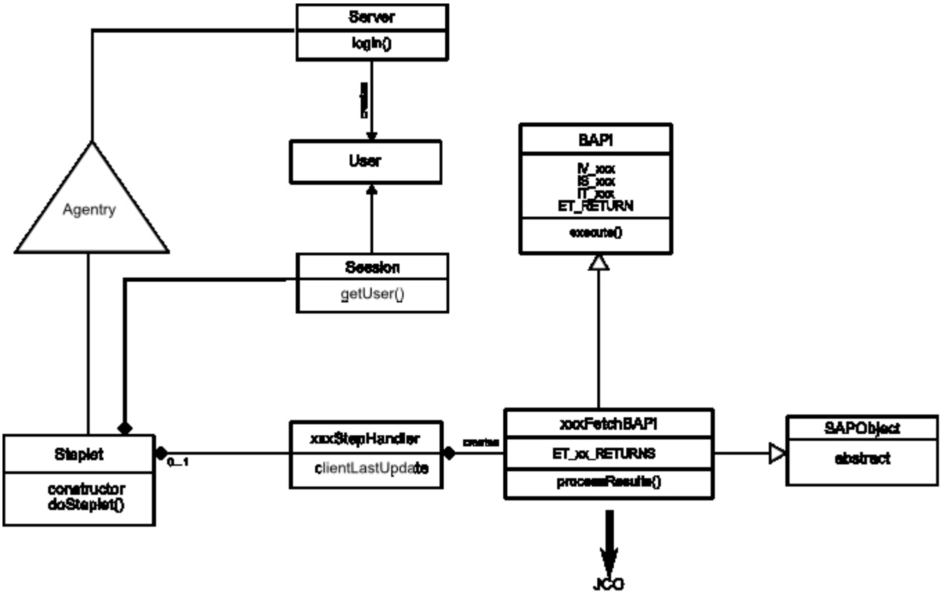
The synchronization processes described in the following sections illustrate the default processes as provided by SAP with the application. During implementation, these standard processes may be configured to support implementation-specific needs.

Data Flow - Fetch

A fetch defines how the SAP Agentry Server synchronizes data for a target object collection. This object collection must be a top-level collection within the module. A fetch is made up of steps that retrieve the data for the collection from the back end system. These steps are grouped into three categories within the Fetch definition: Client Exchange Steps, Server Exchange Steps, and Removal Steps. A fetch may also include properties to store data captured from the user and validation rules for those property values.

The following diagram and steps depict what happens when the Agentry Client must load or reload a fetch.

Figure 4: Data Flow - Fetch



1. A Server exchange steplet defined in a fetch calls the steplet `doSteplet()` method to fetch objects for the user.
2. The `doSteplet()` method calls the appropriate method in the appropriate `xxxStepHandler` class.
3. The `xxxStepHandler` method instantiates the necessary appropriate `xxxFetchBAPI` object, passing the `User` object and `clientLastUpdate` parameter to it.
4. The `xxxFetchBAPI` constructor retrieves the JCo function object from the repository, using the connection on `User`.
5. `xxxFetchBAPI` class sets BAPI import parameters `IV_xxx` and/or `IS_xxx`.
6. `xxxFetchBAPI` constructor adds `IT_xxx` records to BAPI import tables for search criteria or other input parameters.
7. `xxxStepHandler` method calls `processResults()` method from `xxxFetchBAPI`.
8. `xxxFetchBAPI processResults()` calls `execute()` method on BAPI and checks for exceptions.
9. `xxxFetchBAPI processResults()` reads appropriate `ET_xx_RETURNS` table for error messages.
10. `xxxFetchBAPI processResults()` iterates over `ET_xx_RETURNS` table and reads records from the table.

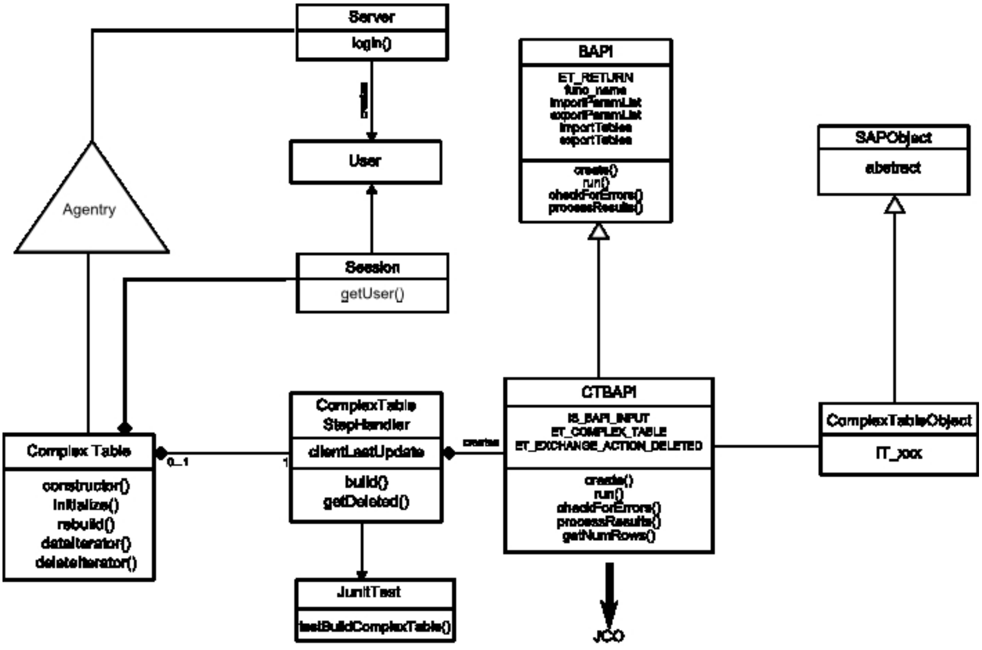
11. For each record, `xxxFetchBAPI processResults()` calls appropriate constructor in the appropriate `SAPObject` subtype.
12. The `SAPObject` subtype constructor maps JCo record column names to field names.
13. `xxxFetchBAPI processResults()` collects these `SAPObjects` in an array and passes it back to `xxxStepHandler`.
14. `xxxStephandler` passes `SAPObjects` array back to steplet `doSteplet()`.
15. Steplet `doSteplet()` stores `SAPObjects` array in `_returnData`.
16. Agentry application within the SAP Mobile Server parses `_returnData` and sends object collection up to Client.

Data Flow - Complex Table

The complex table definition defines a table of records containing multiple fields stored on the SAP Client in a structured and searchable format. A complex table can contain large amounts of data with records numbering in the thousands. Included in the complex table are the fields for its records and indexes on fields to provide search functionality and structure to the overall data in the table. The complex table definition also defines how its data is synchronized.

The following diagram and steps depict what happens when the Agentry Client must load or reload a complex table.

Figure 5: Data Flow - Complex Table



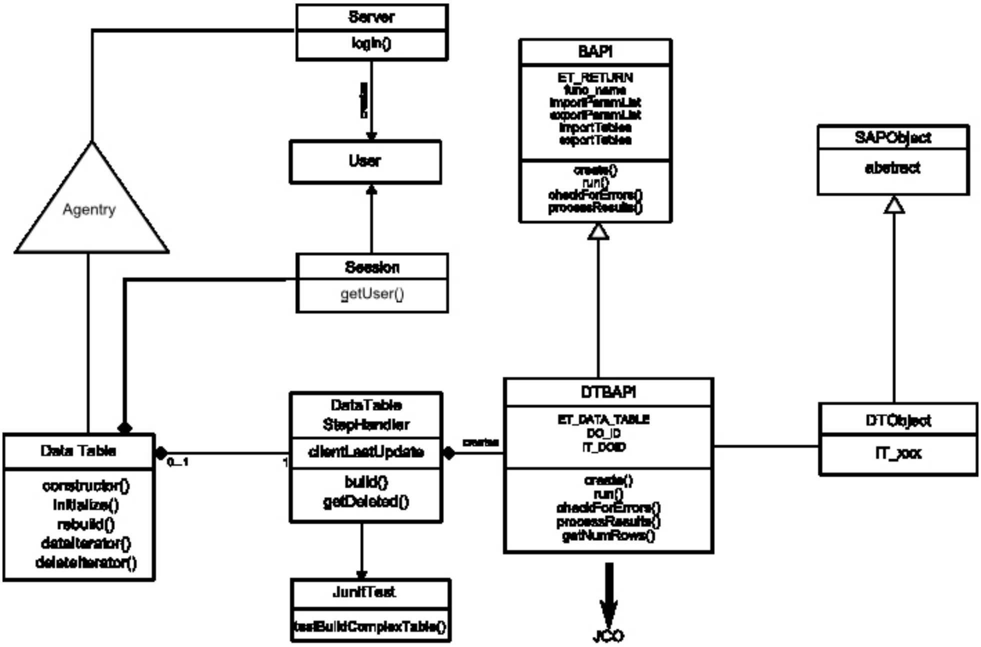
1. The complex table's `initialize()` method is called.
2. The `initialize()` method calls a `ComplexTableStepHandler build()` static method passing the `User` object.
3. The `ComplexTableStepHandler build()` method constructs the necessary `CTBAPI` class, passing the `User` object and `clientLastUpdate` parameter to it.
4. The `CTBAPI` constructor retrieves the JCo function object from the repository, using the connection on `User`.
5. `CTBAPI` sets BAPI import parameter `IS_BAPI_INPUT`.
6. `CTBAPI` adds `IT_xxx` record(s) to `ComplexTableObject` import tables for search criteria or other input parameters.
7. `ComplexTableStepHandler build()` method calls `getNumRows()` method in `CTBAPI`.
8. `CTBAPI getNumRows()` method calls the `execute()` function and checks for exceptions.
9. `CTBAPI getNumRows()` method reads `ET_RETURN` table in the `BAPI` class for error messages.
10. `CTBAPI getNumRows()` method iterates over `ET_COMPLEX_TABLE` and reads records from the table.
11. For each record, `CTBAPI getNumRows()` method calls the appropriate constructor in the appropriate `SAPObject` subtype.
12. The `SAPObject` subtype constructor maps the JCo record column names to field names.
13. `CTBAPI getNumRows()` method collects the `SAPObjects` in `ComplexTableIterator` and passes them back to `ComplexTableStepHandler`.
14. `ET_EXCHANGE_ACTION_DELETED` is read and follows the same steps as *10 - 13*. These `SAPObjects` populate another `ComplexTableIterator`.
15. `ComplexTableHandler` passes the iterators back to `ComplexTable dataIterator()` and `deleteIterator().`
16. `ComplexTable initialize()` stores the iterators in `ComplextTableIterator` and returns them to Agentry in `dataIterator()` and `deleteIterator`.
17. Agentry application defined in the SAP Mobile Server parses these iterators and sends table rows up to Client.

Data Flow - Standard Data Table

The complex table selection property type is used to store a selection made by the user from a complex table. The value stored in a complex table selection property is the key field of the selected record within the complex table. The data type of this value will be a string, integral number, or decimal number, based on the data type of the key field.

The following diagram and steps depict what happens when the Agentry Client must load or reload a data table.

Figure 6: Data Flow - Standard Data Table



1. The data table's `initialize()` method is called.
2. The `initialize()` method calls a `DataTableStepHandler build()` static method passing the `User` object.
3. The `DataTableStepHandler build()` method constructs the necessary `DTBAPI` class, passing the `User` object, `clientLastUpdate` parameter, and table name to it.
4. The `DTBAPI` constructor retrieves the JCo function object for `/SYCLO/CORE_DT_GET` from the repository, using the connection on `User`.
5. `DTBAPI` sets the table name in the table `IT_DOID`, in the field `DO_ID` in the `BAPI` import parameters.
6. `DTBAPI` adds `IT_xxx` record(s) to `DTObject` import tables for search criteria or other input parameters.
7. `DataTableStepHandler build()` method calls `getNumRows()` method in `DTBAPI`.
8. `DTBAPI getNumRows()` calls the `execute()` function and checks for exceptions.
9. `DTBAPI getNumRows()` reads `ET_RETURN` table in the `BAPI` class for error messages.

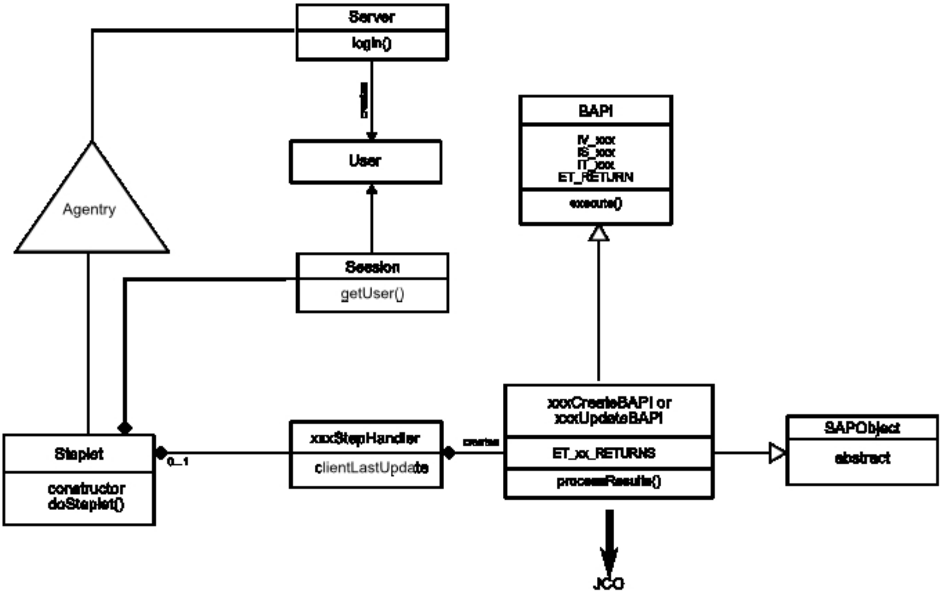
10. `CTBAPI getNumRows()` method iterates over `ET_DATA_TABLE` and reads records from the table.
11. For each record, `DTBAPI getRows()` method calls the appropriate constructor in the appropriate `SAPObject` subtype.
12. The `SAPObject` subtype constructor maps the JCo record column names to field names.
13. `DTBAPI getNumRows()` method collects the `SAPObjects` in `DataTable` and passes them back to `DataTableStepHandler`.
14. DataTableStepHandler build() method passes the DataTableObject back to the initialize() method and is then stored in the appropriate field.
15. Agentry application within the SAP Mobile Server parses these iterators and sends the table rows to the client.

Data Flow - Transaction

The transaction definition defines data to be captured on the SAP Client. As a part of its definition, the transaction includes a target object type, data values to be captured, client-side data validation, and updating its data to the back end system by the SAP Agentry Server during synchronization. Transactions can add new object instances, edit an existing object, delete an object, or modify a complex table or data table record. Each of these behaviors is exhibited by a different transaction type, selected during the creation of the transaction.

The following diagram and steps depict what happens when the Agentry client must load or reload a transaction.

Figure 7: Data Flow - Transaction

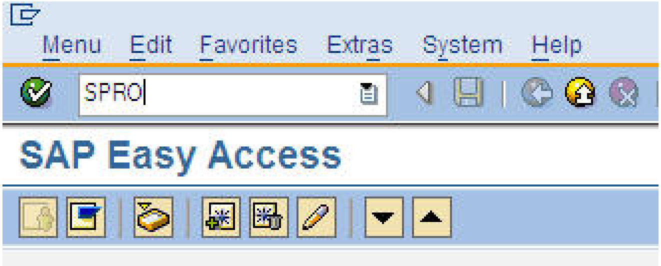


1. A Server exchange steplet defined in a Transaction calls Steplet doSteplet() method to begin the transaction.
2. Steplet's doSteplet() calls appropriate method in the xxxStepHandler class to create, update, or delete objects in SAP.
3. xxxStephandler method constructs the necessary SAPObject subtype by passing the User to the SAPObject's constructor.
4. xxxStepHandler method instantiates the necessary xxxCreateBAPI or xxxUpdateBAPI object, passing the SAPObject to constructor.
5. The xxxCreateBAPI or xxxUpdateBAPI constructor maps the SAPObject's fields to the necessary JCO record columns.
6. xxxCreateBAPI or xxxUpdateBAPI constructor retrieves JCO function object from the repository, using the connection on User.
7. xxxCreateBAPI or xxxUpdateBAPI constructor sets BAPI import parameters IV_xxx and/or IS_xxx.
8. xxxCreateBAPI or xxxUpdateBAPI constructor adds IT_xxx records to BAPI import tables.
9. xxxStepHandler method calls add(), update() or delete() method of xxxCreateBAPI or xxxUpdateBAPI.
10. xxxCreateBAPI or xxxUpdateBAPI calls the execute() method on BAPI and checks for exceptions.

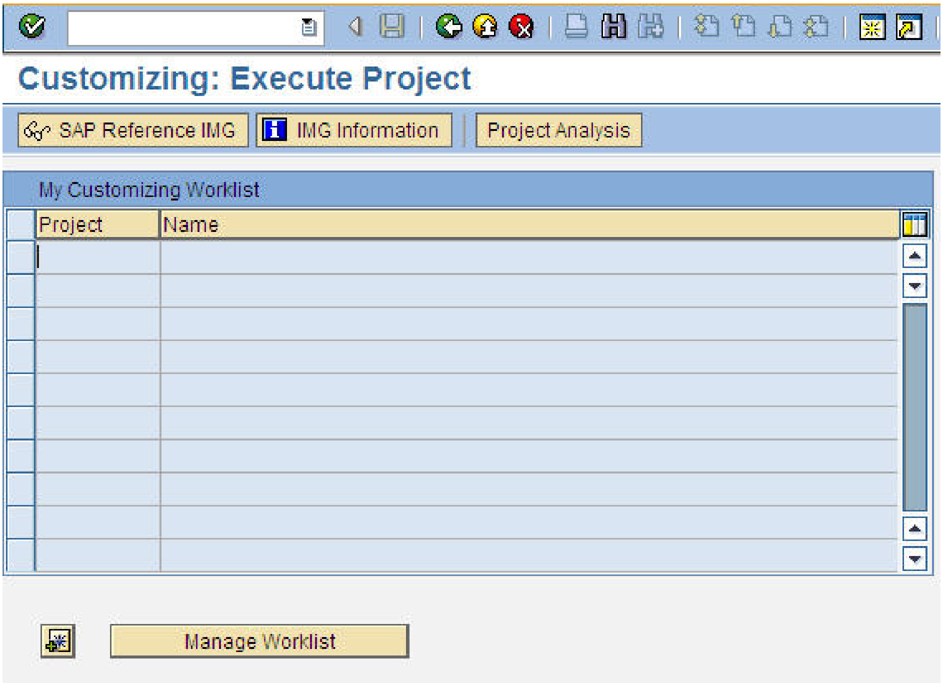
11. xxxCreateBAPI or xxxUpdateBAPI reads ET_RETURN table for error messages.
12. If successful, control passes back to xxxStepHandler, Steplet, and Agentry.

Accessing the Agentry SAP Framework Configuration Panel

1. Log into SAP.
2. Type the command `SPRO` into the command box and click the green check mark or click **Enter**.

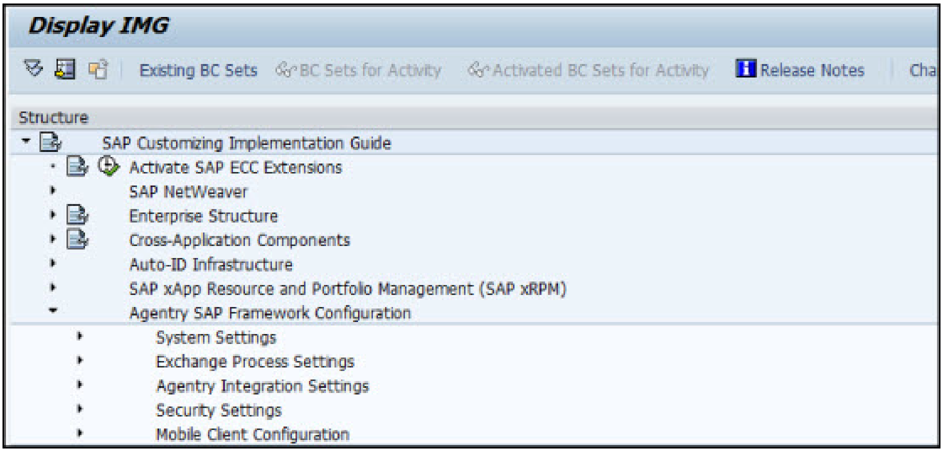


The Customizing: Execute Project screen displays.



3. Click the **SAP Reference IMG** button.

   The Display IMG screen displays.



Display IMG

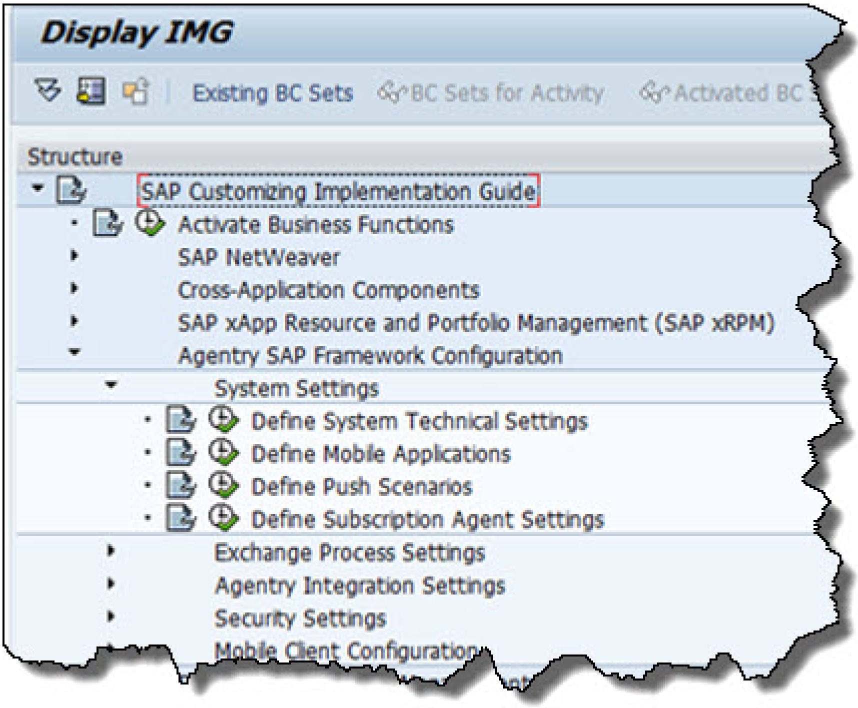
Existing BC Sets | BC Sets for Activity | Activated BC Sets for Activity | Release Notes | Cha

Structure

- SAP Customizing Implementation Guide
  - Activate SAP ECC Extensions
  - SAP NetWeaver
  - Enterprise Structure
  - Cross-Application Components
  - Auto-ID Infrastructure
  - SAP xApp Resource and Portfolio Management (SAP xRPM)
  - Agentry SAP Framework Configuration
    - System Settings
    - Exchange Process Settings
    - Agentry Integration Settings
    - Security Settings
    - Mobile Client Configuration

4. Expand the Agentry SAP Framework Configuration line by clicking on the arrow to the left of the line.

Agentry SAP Framework submenus display. Expand any one of the submenus to display specific configuration functions and click the clock icon to open the Agentry SAP Framework ConfigPanel.



Display IMG

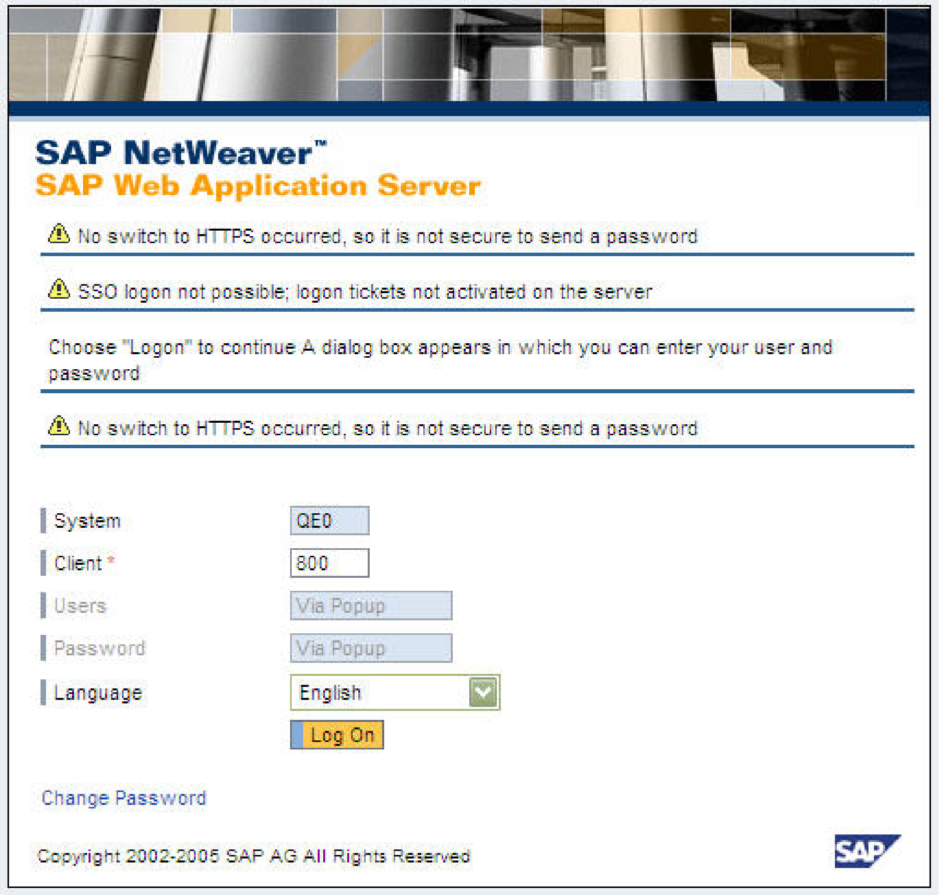
Existing BC Sets | BC Sets for Activity | Activated BC

Structure

- SAP Customizing Implementation Guide
  - Activate Business Functions
  - SAP NetWeaver
  - Cross-Application Components
  - SAP xApp Resource and Portfolio Management (SAP xRPM)
  - Agentry SAP Framework Configuration
    - System Settings
      - Define System Technical Settings
      - Define Mobile Applications
      - Define Push Scenarios
      - Define Subscription Agent Settings
    - Exchange Process Settings
    - Agentry Integration Settings
    - Security Settings
    - Mobile Client Configuration

Note: Clicking the paper icon displays a screen with a brief description of the specific configuration function.

The SAP NetWeaver Web Application Server log on screen opens in a browser window.



SAP NetWeaver™
SAP Web Application Server

⚠ No switch to HTTPS occurred, so it is not secure to send a password

⚠ SSO logon not possible; logon tickets not activated on the server

Choose "Logon" to continue A dialog box appears in which you can enter your user and password

⚠ No switch to HTTPS occurred, so it is not secure to send a password

System	QE0
Client *	800
Users	Via Popup
Password	Via Popup
Language	English

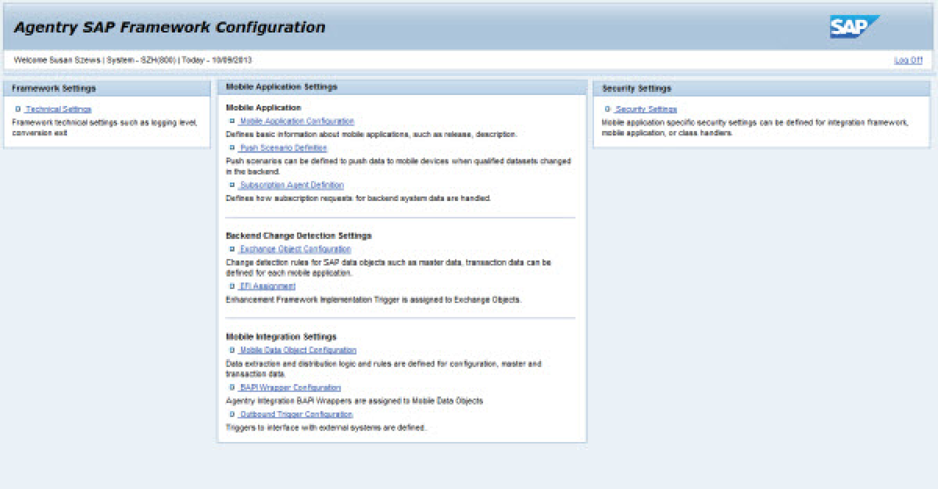
Copyright 2002-2005 SAP AG All Rights Reserved

Note: Depending on your configuration, the log on screen may be different than what is shown.

5. Expand any one of the submenus to display specific configuration functions and click the clock icon to open the Agentry SAP Framework Configuration panel.
6. Fill in any necessary fields for your specific implementation and click **Log On**.

   The Agentry SAP Framework Configuration portal opens in the browser window. The window that opens corresponds to the submenu line item chosen in SAP.
7. Click the ConfigPanel Home link at the top of the screen to navigate to the main configuration page.

The Agentry SAP Framework Configuration Panel home page displays.



Agentry SAP Framework Configuration

Welcome Susan Szews | System - SZH(800) | Today - 10/09/2013

Log Off

Framework Settings

- Technical Settings

Framework technical settings such as logging level, conversion exit

Mobile Application Settings

Mobile Application

- Mobile Application Configuration

Defines basic information about mobile applications, such as release, description.

- Push Scenario Definition

Push scenarios can be defined to push data to mobile devices when qualified datasets changed in the backend.

- Subscription Agent Definition

Defines how subscription requests for backend system data are handled.

Backend Change Detection Settings

- Exchange Object Configuration

Change detection rules for SAP data objects such as master data, transaction data can be defined for each mobile application.

- EFI Assignment

Enhancement Framework Implementation Trigger is assigned to Exchange Objects.

Mobile Integration Settings

- Mobile Data Object Configuration

Data extraction and distribution logic and rules are defined for configuration, master and transaction data.

- BAPI Wrapper Configuration

Agentry Integration BAPI Wrappers are assigned to Mobile Data Objects

- Outbound Trigger Configuration

Triggers to interface with external systems are defined.

Security Settings

- Security Settings

Mobile application specific security settings can be defined for integration framework, mobile application, or class handlers.

Standard Operations in the Configuration Panel

While each mobile application configuration is unique, there are certain standard buttons and options available to perform the configurations.

Filtering by Mobile Application

If more than one mobile application is available on the same system, you can use the filter function to only view items for a specific application. The filter option is found on the main portal home page, as well as any other page where multiple application items could be displayed.

To filter by application, click the arrow to the right of the Mobile Application Filter field and select the appropriate mobile application. To remove the selection and view all items for all mobile applications on the system, click in the field again and select the asterisk (*****) symbol.

Creating, Copying, Deleting, and Changing Items

There are four standard actions available to configure different components and items within your mobile application setup.

- **Create:** Creates a new item. All modifiable fields are empty.
- **Copy:** Copies the item that was highlighted and creates a new item. All modifiable fields are filled in with the information from the existing item and are available for changes prior to saving.
- **Delete:** Deletes the highlighted item.
- **Change:** Allows changes to be made to the highlighted item in the modifiable fields.

Saving or Cancelling Changes to an Item

Once the Create, Copy, or Change button is clicked, the Save and Cancel buttons appear. After making any changes to the configuration, click **Save** to save the changes or **Cancel** to discard the changes.

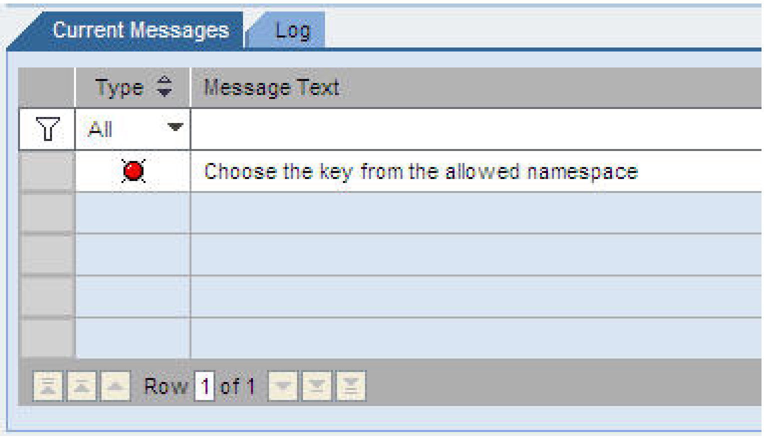
Note: If the Save and Cancel buttons are active, the ConfigPanel Home main menu link is not available. You must either save your changes or cancel out of the changes in order to return to the main Configuration portal page.

Message List

Certain actions can generate system messages. These messages can be error messages or informational messages. If you perform an action that prompts a system message, a message bar appears above the main panel with a brief description of the message.

Click the **Show List** button to display the detailed view of the message list.

Figure 8: Configuration Panel - Detailed Message Display



	Type	Message Text
	All	
	●	Choose the key from the allowed namespace

Agentry SAP Framework Configuration Panel Overview

All configuration activities for Agentry SAP Framework are performed using the Configuration screen.

Configuration changes made through the Configuration portal have significant impact to the behavior of the framework component and mobile applications. Changes should be made in the development environment and fully tested before they are migrated to the rest of the SAP system landscape.

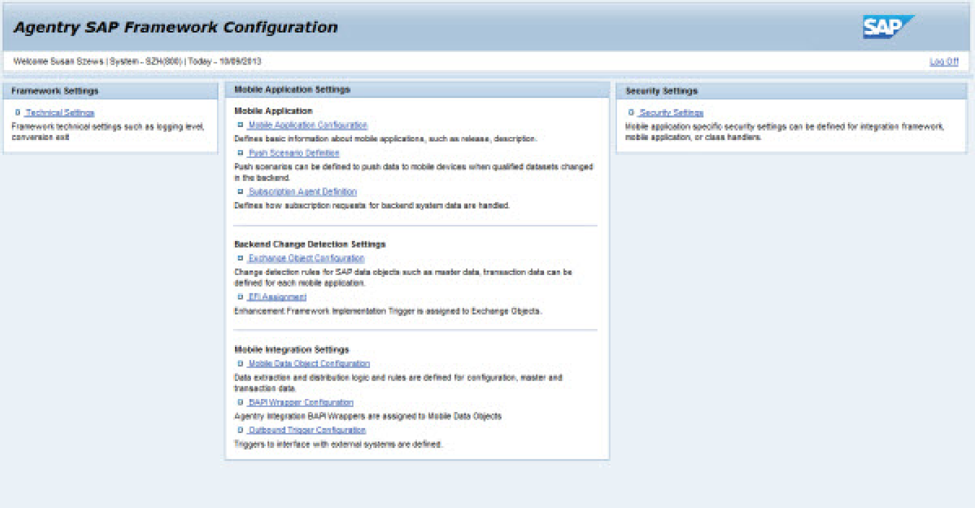
The following user authorizations are required in order to work with the configuration portal:

- *Authorization object* - S_ICF

- *Authorization field* - ICF_FIELD - SERVICE
- *Authorization field* - ICF_VALUE - SYCLOADM
- *Authorization object* - S_TCODE
- *Authorization field* - TCD - /SYCLO/CONFIGPANEL

If you create additional security roles through the Security Settings panel in the ConfigPanel, you must incorporate them into the system as well.

Figure 9: Agentry SAP Framework Configuration Panel Main Screen



Agentry SAP Framework Configuration

Welcome Susan Szews | System - SZH(800) | Today - 10/09/2013

Log Off

Framework Settings

- Technical Settings

Framework technical settings such as logging level, conversion exit

Mobile Application Settings

Mobile Application

- Mobile Application Configuration

Defines basic information about mobile applications, such as release, description.

- Push Scenario Definition

Push scenarios can be defined to push data to mobile devices when qualified datasets changed in the backend.

- Subscription Agent Definition

Defines how subscription requests for backend system data are handled.

Backend Change Detection Settings

- Exchange Object Configuration

Change detection rules for SAP data objects such as master data, transaction data can be defined for each mobile application.

- EFI Assignment

Enhancement Framework Implementation Trigger is assigned to Exchange Objects.

Mobile Integration Settings

- Mobile Data Object Configuration

Data extraction and distribution logic and rules are defined for configuration, master and transaction data.

- BAPI Wrapper Configuration

Agentry Integration BAPI Wrappers are assigned to Mobile Data Objects

- Outbound Trigger Configuration

Triggers to interface with external systems are defined.

Security Settings

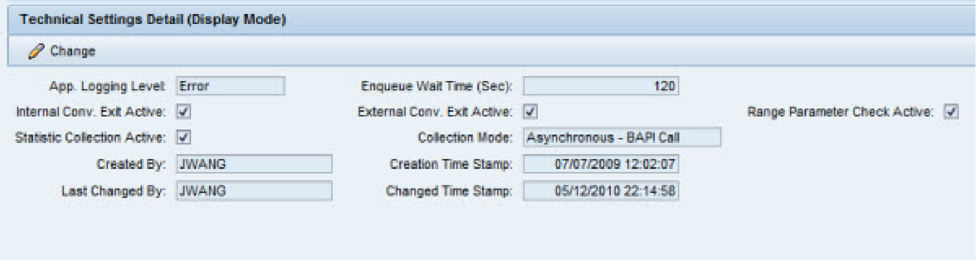
- Security Settings

Mobile application specific security settings can be defined for integration framework, mobile application, or class handlers.

Configuration Panel - Technical Settings

Framework technical settings affect all components of the framework, including mobile data objects, BAPI wrappers, and exchange objects.

Figure 10: Agentry SAP Framework Configuration Panel Technical Settings Screen



Technical Settings Detail (Display Mode)

Change

App. Logging Level: Error

Internal Conv. Exit Active: ☑

Statistic Collection Active: ☑

Created By: JWANG

Last Changed By: JWANG

Enqueue Wait Time (Sec): 120

External Conv. Exit Active: ☑

Collection Mode: Asynchronous - BAPI Call

Creation Time Stamp: 07/07/2009 12:02:07

Changed Time Stamp: 05/12/2010 22:14:58

Range Parameter Check Active: ☑

Configuration Panel - Mobile Application Settings

Mobile application settings are used to define and configure how the mobile application, such as SAP, functions. There are three areas used to configure the mobile application:

- **Mobile Application Configuration** - Defines basic information about mobile applications, such as release and descriptions
- **Push Scenario Definition** - Defines Push Scenarios to push data to mobile devices when qualified data sets change in the back end
- **Subscription Agent Definition** - Defines how subscription requests for back end system data are handled

See the applicable sections in this manual for further information.

Configuration Panel - Back End Change Detection Settings

Back end change detection settings are used to define and configure how the mobile application, such as SAP, communicates with SAP and the object tables contained within SAP. There are two areas used to configure the back end change detection:

- **Exchange Object Configuration** - Change detection rules for SAP data objects, such as master data and transaction data, defined for each mobile application
- **EFI Assignment** - Enhancement Framework Implementation trigger assigned to Exchange Objects

Note: You must create tables and objects in SAP and Agentry before you can create or configure information in the Configuration Panel.

See the applicable sections in this manual for further information.

Configuration Panel - Mobile Integration Settings

Mobile integration settings are used to link BAPI wrappers with mobile data objects and encapsulate the business logic related to the mobile application.

There are three areas used to configure mobile integration:

- **Mobile Data Object Configuration** - Data extraction and distribution logic and rules defined for configuration of master data and transaction data
- **BAPI Wrapper Configuration** - Agentry Integration BAPI Wrappers assigned to Mobile Data Objects
- **Outbound Trigger Configuration** - Triggers to interface with external systems

See these sections in the manual for further information.

Configuration Panel - Security Settings

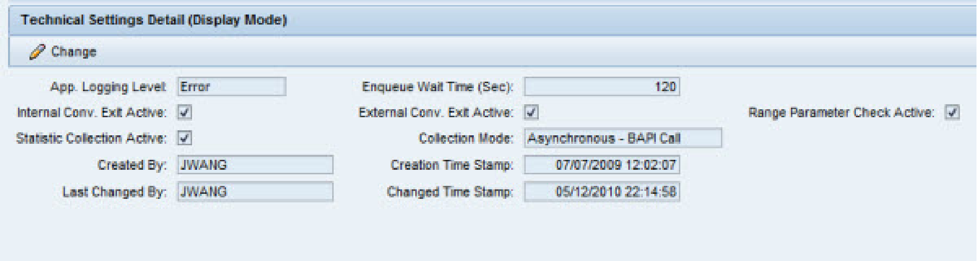
Use the Security Settings page in the Configuration Panel to set mobile application security parameters at the following levels:

- **System**- Security at this level is application-independent and applies to all components built on the SAP integration framework
- **Product** - Security at this level is at the SAP application level
- **Class Handler** - Security at this level is specific to a data object class handler

All security checks are carried out by the SAP integration framework at runtime.

Technical Settings

Use the Technical Settings Detail screen in the Agentry SAP Framework Configuration Panel to change the settings for the application logs viewable in SAP. Here, you can change framework technical settings such as the logging level and conversion exit.



Technical Settings Detail (Display Mode)

App. Logging Level:	Error	Enqueue Wait Time (Sec):	120		
Internal Conv. Exit Active:	☑	External Conv. Exit Active:	☑	Range Parameter Check Active:	☑
Statistic Collection Active:	☑	Collection Mode:	Asynchronous - BAPI Call		
Created By:	JWANG	Creation Time Stamp:	07/07/2009 12:02:07		
Last Changed By:	JWANG	Changed Time Stamp:	05/12/2010 22:14:58		

Application Logging Level

Defines the logging level for all framework components. Logging entries are recorded in the SAP application log database under the object **/syclo/**. The logging levels are:

- No logging
- Abort
- Error
- Warning
- Info
- Debug
- Trace

Enqueue Wait Time (Sec)

If an SAP object is locked and inaccessible during an update by a mobile device, this parameter controls the number of seconds the underlying component should continue to attempt to access the locked object in intervals of 1 second. If accessing the locked object is still unsuccessful after the wait time, the update process is aborted.

Internal Conversion Exit Active

When checked, the framework runtime data manager performs standard SAP external-to-internal format conversion exit for all inbound BAPI parameters. This option is enabled by default. This setting should only be changed by the application developer, as it will have direct impact to the result of the mobile application.

External Conversion Exit Active
When enabled, the framework runtime data manager performs standard SAP internal-to-external format conversion exit for all outbound BAPI parameters. This option is enabled by default. This setting should only be changed by the application developer, as it will have direct impact to the result of the mobile application.

Range Parameter Check Active
When enabled, the framework runtime data manager will perform checks on all SAP range parameters of inbound BAPI parameters. The SAP range parameter has the structure of SIGN, OPTION, LOW and HIGH. Check routine will set the SIGN value to ‘I’ and the OPTION value to ‘EQ’ if not specified. This option is enabled by default. This setting should only be changed by the application developer, as it will have direct impact to the result of the mobile application.

Collection Mode
Collection mode determines how system statistic records are written to the database. Two modes are supported currently: synchronously and asynchronously. When Synchronously is selected, the statistics record is written to the database in real-time during BAPI calls. However, selecting this option incurs a performance penalty. Selecting Asynchronously means that statistics are collected in-memory and written asynchronously to the database at the end of the BAPI call.

Statistic Collection Active
When enabled, the framework records all runtime statistics associated with the BAPI calls between the middleware server and SAP. This collection provides data for the KPI statistics collections found in the Administration portal. This setting should only be changed by the application developer, as it will have direct impact to the result of the mobile application.

Created By, Creation Time Stamp, Last Changed By, Changed Time Stamp
The user ID and time stamps are automatically logged when a record is created or changed.

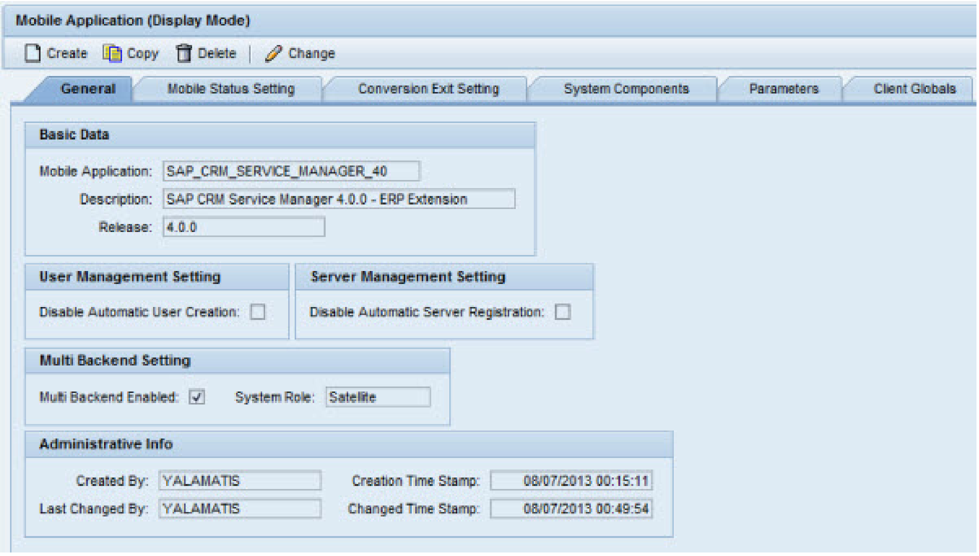
Mobile Application Configuration

Use the Mobile Application Configuration page to set general settings for the entire SAP mobile application.

Mobile Application - General

Use the General tab to create or change basic information about a mobile application.

Mobile Application - General Settings



Basic Data

Mobile Application: SAP_CRM_SERVICE_MANAGER_40

Description: SAP CRM Service Manager 4.0.0 - ERP Extension

Release: 4.0.0

User Management Setting

Disable Automatic User Creation: □

Server Management Setting

Disable Automatic Server Registration: □

Multi Backend Setting

Multi Backend Enabled: ☑ System Role: Satellite

Administrative Info

Created By: YALAMATIS Creation Time Stamp: 08/07/2013 00:15:11

Last Changed By: YALAMATIS Changed Time Stamp: 08/07/2013 00:49:54

Basic Data

- **Mobile Application:** The name of the mobile application, limited to 40 characters. This is a required field.
- **Description:** A brief, easy to understand description of the mobile application, limited to 60 characters. This is a required field.
- **Release:** The release number of the mobile application

User Management Setting

Disable Automatic User Creation: When checked, a new user GUID is not automatically created when a new mobile client is detected in the system. The system administrator must manually create and maintain mobile users through the Administration portal.

Server Management Setting

Disable Automatic Server Registration: When checked, a new server GUID is not automatically created when a new server is detected in the system. The system administrator must manually create and maintain servers through the Administration portal.

Multi Back End Setting

- **Multi Back End Enabled:** When checked, enables a specific mobile application to connect to multiple SAP systems, consisting of one host server and one or more satellite servers
- **System Role:** Drop-down menu listing Host or Satellite -
  - A host system is the connection between SAP and the Agentry application in the SAP Mobile Server. The host server provides the logic to the Agentry application and

functions as a bridge to the satellite server(s). There can only be one host server per system.

- Satellite servers communicate with SAP through the host server.

In order to complete multi-back end configuration, the host and satellite servers must be configured in the System Components tab. See the *Mobile Application - System Components* section for more details.

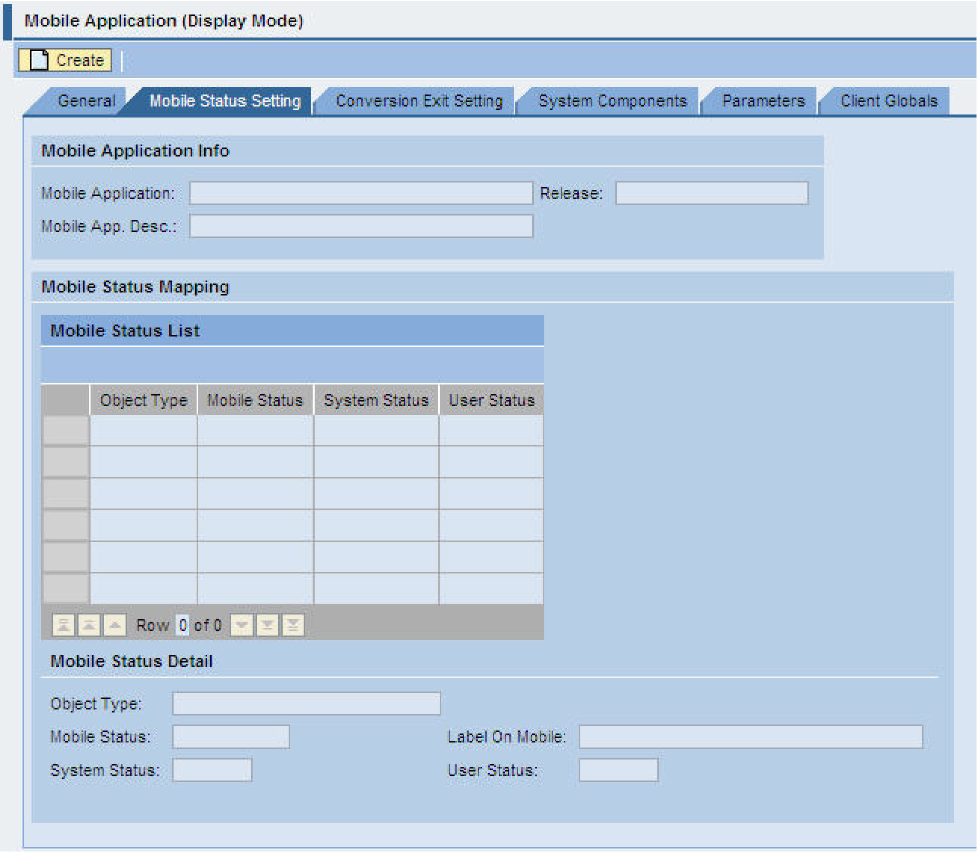
Administrative Info

- **Created By:** SAP user ID of the person who created the mobile data object
- **Creation Time Stamp:** Date and time of the creation of the mobile data object
- **Last Changed By:** SAP user ID of the person who last changed the mobile data object
- **Changed Time Stamp:** Date and time of the change to the mobile data object

Mobile Application - Mobile Status Setting

Use the Mobile Status Setting tab to map the available mobile statuses that a mobile data object (MDO) supports on the client side. If a user status also exists for the same MDO, you can link it to the mobile status and the system status through this tab.

Mobile Application - Mobile Status Setting



Mobile Application Info

- **Mobile Application:** (Read Only) The name of the mobile application
- **Mobile Application Description:** (Read Only) A brief, easy to understand description of the mobile application
- **Release:** (Read Only) The release number of the mobile application

Mobile Status Mapping

- **Create button:** Click **Create** to create a new mobile status detail. Fill in the fields in the Mobile Status Detail section to automatically fill in the fields in this table.
- **Delete button:** Click **Delete** to delete an existing mobile status detail. To delete a mobile status detail, click the rectangle to the left of the Object Type column in the row you want to delete and click **Delete**.
- **Object Type:** (Read Only) Object type from the Mobile Status Detail section
- **Mobile Status:** (Read Only) Mobile status from the Mobile Status Detail section

- **System Status:** (Read Only) System status from the Mobile Status Detail section
- **User Status:** (Read Only) User status from the Mobile Status Detail section

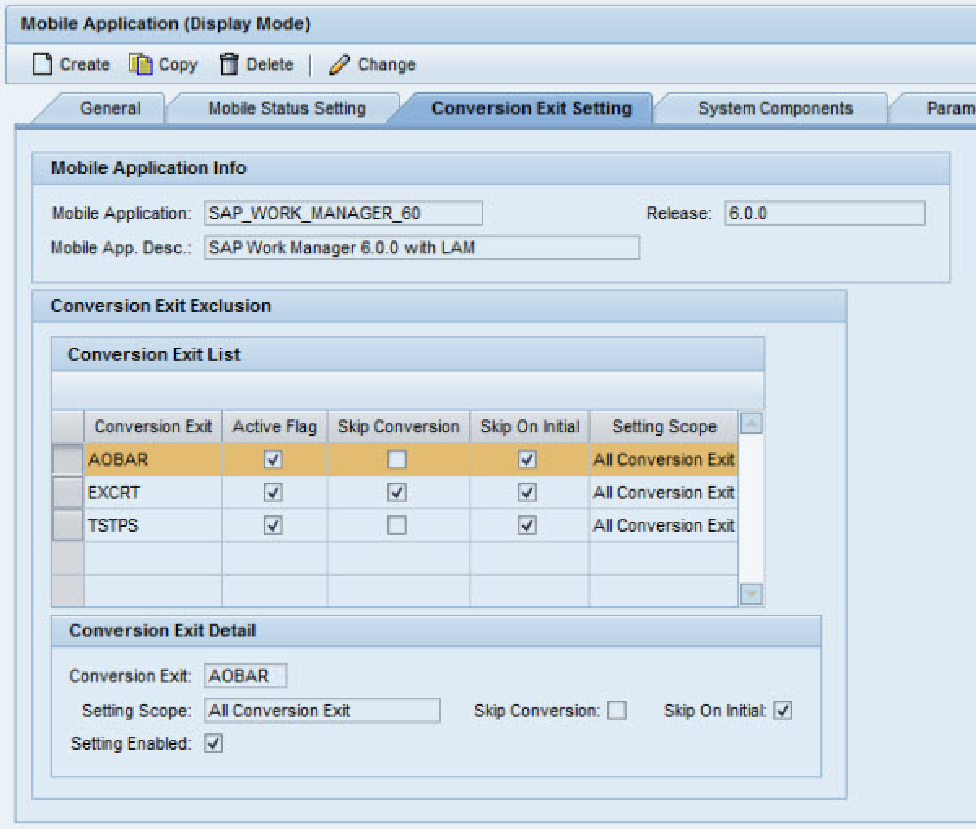
Mobile Status Detail

- **Object Type:** The specific object in a mobile application, i.e., NOTIFICATION
- **Mobile Status:** Status defined by the mobile application
- **Label on Mobile:** Not used
- **System Status:** Standard SAP status code
- **User Status:** SAP user status code as defined in SAP

Mobile Application - Conversion Exit Setting

Use the Conversion Exit Setting tab to list the SAP conversion exits to exclude during runtime by the framework.

Mobile Application - Conversion Exit Setting



Mobile Application Info

Mobile Application: SAP_WORK_MANAGER_60

Release: 6.0.0

Mobile App. Desc.: SAP Work Manager 6.0.0 with LAM

Conversion Exit Exclusion

Conversion Exit List

Conversion Exit	Active Flag	Skip Conversion	Skip On Initial	Setting Scope
AOBAR	☑	□	☑	All Conversion Exit
EXCRT	☑	☑	☑	All Conversion Exit
TSTPS	☑	□	☑	All Conversion Exit

Conversion Exit Detail

Conversion Exit: AOBAR

Setting Scope: All Conversion Exit

Skip Conversion: □

Skip On Initial: ☑

Setting Enabled: ☑

Mobile Application Info

- **Mobile Application:** (Read Only) The name of the mobile application
- **Mobile App(lication) Desc(ription):** (Read Only) A brief, easy to understand description of the mobile application
- **Release:** (Read Only) The release number of the mobile application

Conversion Exit List

- **Add button:** Click **Add** to create a new conversion exit detail. Fill in the fields in the Conversion Exit Detail section to automatically fill in the fields in this table.
- **Delete button:** Click **Delete** to delete an existing conversion exit detail. To delete a conversion exit detail, click the rectangle to the left of the Conversion Exit column in the row you want to delete and click **Delete**.
- **Conversion Exit:** (Read Only) Conversion exit from the Conversion Exit Detail section.
- **Active Flag (column 1):** (Read Only) When checked, the Skip Conversion box is checked in the Conversion Exit Detail section
- **Skip Conversion (column 2):** (Read Only) When checked, the Skip on Initial box is checked in the Conversion Exit Detail section
- **Skip On Initial (column 3):** (Read Only) When checked, the Setting Scope box is checked in the Conversion Exit Detail section
- **Setting Scope:** (Read Only) Conversion exit scope from the Conversion Exit Detail section

Conversion Exit Detail

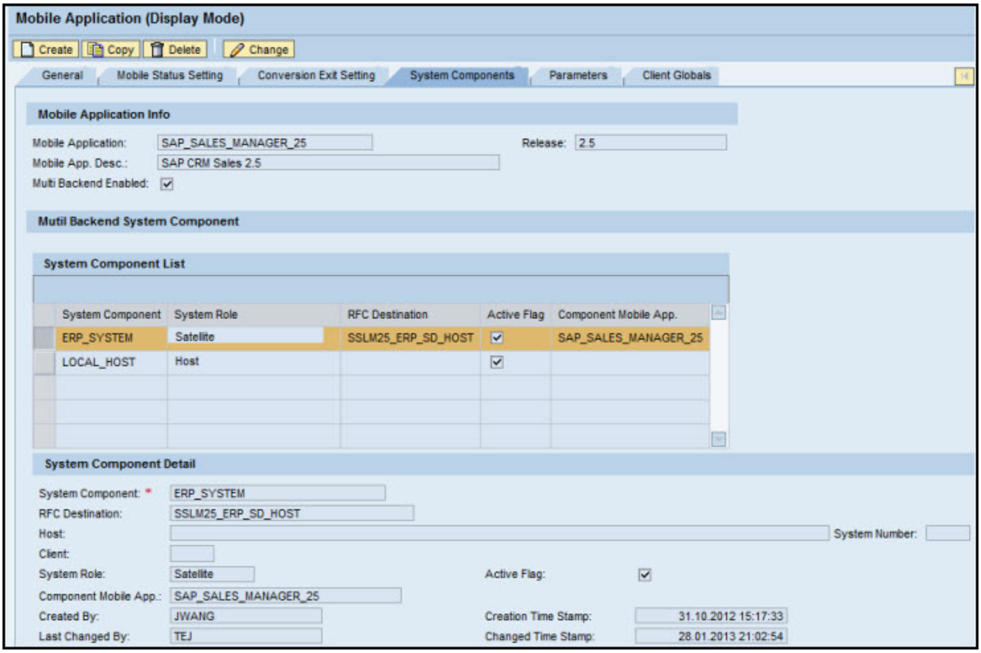
- **Conversion Exit:** Name of the conversion exit as found in SAP
- **Setting Scope:** Choose from the drop down menu choices:
  - All Conversion Exit - Both input and output conversion exit routines are excluded
  - Input Conversion Exit - Conversion routines are exited when data is sent to SAP
  - Output Conversion Exit - Conversion routines are exited when data is pulled out of SAP
- **Skip Conversion:** When checked, the conversion routine is always exited. When you check this box, it automatically checks the box in the first Active Flag column in the table above.
- **Skip on Initial:** When checked, the conversion routine is only excluded when the initial field does not contain a value. If the initial field contains any value, the conversion routine runs. When you check this box, it automatically checks the box in the second Active Flag column in the table above.
- **Setting Enabled:** When checked, the settings configured for the conversion exit are enabled and the exit is active. When you check this box, it automatically checks the box in the third Active Flag column in the table above.

Mobile Application - System Components

Use this tab to define system components in a multi-back end system. Configuration in this tab is not necessary if the application does not require a multi-back end system.

Note: You must check the Multi Back End Enabled box in the General tab of the Mobile Application Configuration pane in order for system component configuration to function.

Mobile Application - System Components



Mobile Application Info

Mobile Application:	SAP_SALES_MANAGER_25	Release:	2.5
Mobile App. Desc.:	SAP CRM Sales 2.5		
Multi Backend Enabled:	☑		

Mutil Backend System Component

System Component List

	System Component	System Role	RFC Destination	Active Flag	Component Mobile App.
	ERP_SYSTEM	Satelite	SSLM25_ERP_SD_HOST	☑	SAP_SALES_MANAGER_25
	LOCAL_HOST	Host		☑	

System Component Detail

System Component: *	ERP_SYSTEM		
RFC Destination:	SSLM25_ERP_SD_HOST		
Host:		System Number:	
Client:			
System Role:	Satellite	Active Flag:	☑
Component Mobile App.:	SAP_SALES_MANAGER_25		
Created By:	JWANG	Creation Time Stamp:	31.10.2012 15:17:33
Last Changed By:	TEJ	Changed Time Stamp:	28.01.2013 21:02:54

Mobile Application Info

- **Mobile Application:** (Read Only) The name of the mobile application
- **Release:** (Read Only) The release number of the mobile application
- **Mobile App(lication) Desc(ription):** (Read Only) A brief, easy to understand description of the mobile application
- **Multi Back End Enabled:** (Read Only) When checked, the multi-backend was activated in the General tab

System Component List

- **Add button:** Click **Add** to create a new system component detail. Fill in the fields in the System Component Detail section to automatically fill in the fields in this table.

- **Delete button:** Click **Delete** to delete an existing system component detail. To delete a system component detail, click the rectangle to the left of the System Component column in the row you wish to delete and click **Delete**.
- **System Component:** (Read Only) System component from the System Component Detail section
- **System Role:** (Read Only) System role from the System Component Detail section
- **RFC Destination:** (Read Only) RFC destination from the System Component Detail section
- **Active Flag:** (Read Only) When checked, the Active Flag box is checked in the System Component Detail section
- **Component Mobile App**: In multi-back end scenarios, when different back end names are used, this is the application name that can virtually tie all applications together

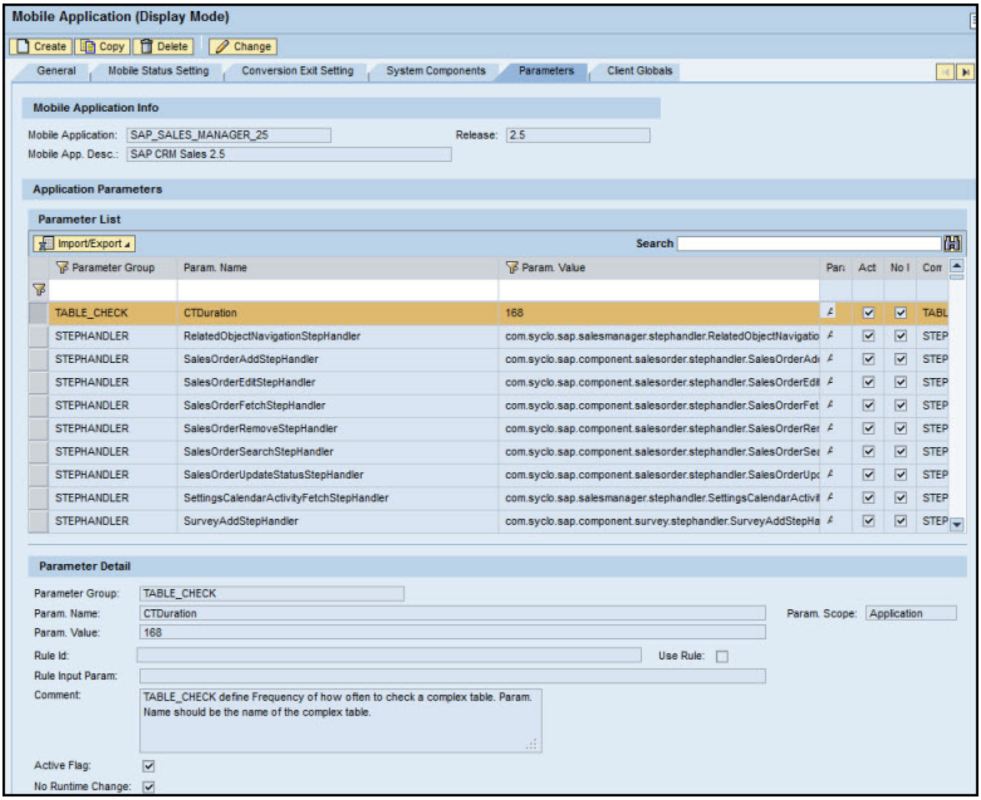
System Component Detail

- **System Component:** Descriptive name of the component. This is a required field.
- **RFC Destination:** Must be defined in SAP prior to configuration in Agentry SAP Framework. Use transaction code SM59 in SAP to create or change the RFC destination.
- **Host:** (Read Only) Identifying host name, defined in SAP
- **System Number:** (Read Only) Identifying server number, defined in SAP
- **Client:** (Read Only) Number of the client that the system component connects to, defined in SAP
- **System Role:** Determines if the system component is a host or a satellite. There can only be one host per multi-back end system.
- **Active Flag:** When checked, the system component is activated in the multi-back end system
- **Component Mobile App:** Common application name for multi-back end systems
- **Created By:** SAP user ID of the person who created the mobile data object
- **Creation Time Stamp:** Date and time of the creation of the mobile data object
- **Last Changed By:** SAP user ID of the person who last changed the mobile data object
- **Changed Time Stamp:** Date and time of the change to the mobile data object

Mobile Application - Parameters

Use this tab to define system parameters.

Mobile Application - Parameters



Mobile Application Info

Mobile Application: SAP_SALES_MANAGER_25

Release: 2.5

Mobile App. Desc.: SAP CRM Sales 2.5

Application Parameters

Parameter List

	Parameter Group	Param. Name	Param. Value	Par:	Act	No I	Con
	TABLE_CHECK	CTDuration	168	A	☑	☑	TABL
	STEPHANDLER	RelatedObjectNavigationStepHandler	com.syclo.sap.salesmanager.stephandler.RelatedObjectNavigatio	A	☑	☑	STEP
	STEPHANDLER	SalesOrderAddStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderAd	A	☑	☑	STEP
	STEPHANDLER	SalesOrderEditStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderEdi	A	☑	☑	STEP
	STEPHANDLER	SalesOrderFetchStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderFet	A	☑	☑	STEP
	STEPHANDLER	SalesOrderRemoveStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderRer	A	☑	☑	STEP
	STEPHANDLER	SalesOrderSearchStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderSe	A	☑	☑	STEP
	STEPHANDLER	SalesOrderUpdateStatusStepHandler	com.syclo.sap.component.salesorder.stephandler.SalesOrderUp	A	☑	☑	STEP
	STEPHANDLER	SettingsCalendarActivityFetchStepHandler	com.syclo.sap.salesmanager.stephandler.SettingsCalendarActivi	A	☑	☑	STEP
	STEPHANDLER	SurveyAddStepHandler	com.syclo.sap.component.survey.stephandler.SurveyAddStepHa	A	☑	☑	STEP

Parameter Detail

Parameter Group: TABLE_CHECK

Param. Name: CTDuration

Param. Scope: Application

Param. Value: 168

Rule Id:

Use Rule: □

Rule Input Param:

Comment: TABLE_CHECK define Frequency of how often to check a complex table. Param. Name should be the name of the complex table.

Active Flag: ☑

No Runtime Change: ☑

Mobile Application Info

- **Mobile Application:** (Read Only) The name of the mobile application
- **Release:** (Read Only) The release number of the mobile application
- **Mobile Application Description:** (Read Only) A brief, easy to understand description of the mobile application

Parameter List

Note: The columns in the Parameter List table are read-only. Use the Parameter Detail section to make any additions or edits to the table.

- **Add button:** Click **Add** to create a new parameter detail. Fill in the fields in the Parameter Detail section to automatically fill in the fields in this table.
- **Delete button:** Click **Delete** to delete an existing parameter detail. To delete a parameter detail, click the rectangle to the left of the Parameter Group column in the row you want to delete and click **Delete**.
- **Parameter Group:** Parameter group from the Parameter Detail section

- **Param. Name:** Parameter name from the Parameter Detail section
- **Param. Value:** Parameter detail from the Parameter Detail section
- **Param. Scope:** Parameter scope from the Parameter Detail section
- **Active Flag:** When checked, the Active Flag box is checked in the Parameter Detail section.
- **No Runtime Change:** When checked, the No Runtime Change box is checked in the Parameter Detail section
- **Comment:** Comments from the Parameter Detail section

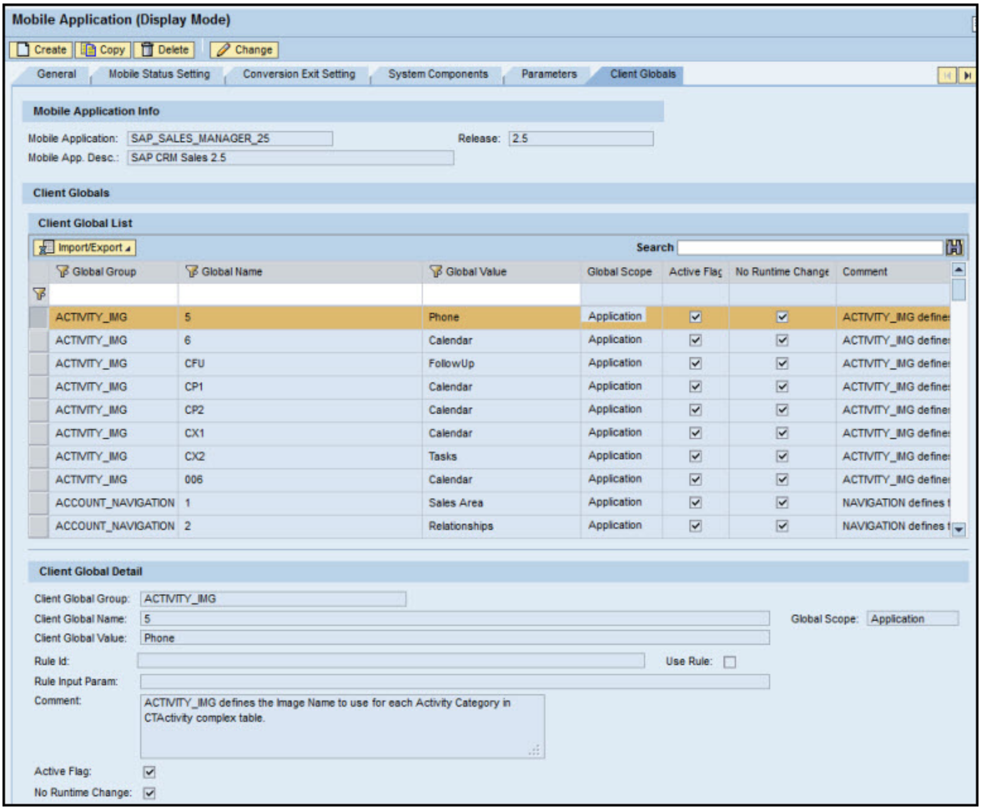
Parameter Detail

- **Parameter Group:** The group to which the parameter belongs. Groups are a means of organizing parameters. References to a parameter include both the group name and the parameter name.
- **Param. Name:** The unique name of the parameter
- **Param. Scope:** The scope of the parameter value. There are two options:
  - Mobile Application: Value for all users of the application
  - Mobile User: Value that can be overridden for individual users. To override a user's parameter value, see the Administration & Monitoring portal information on parameters.
- **Param. Value:** The currently configured value of the parameter. References to this parameter will return this value
- **Rule Id:** If enabled, this is the rule to be used at run time
- **Use Rule:** When checked, you can define a rule to be used at run time
- **Rule Input Param:** If the specified rule has optional parameters, define them here
- **Active Flag:** When checked, the parameter is used by the mobile application. Inactive parameters are not available to the mobile application.
- **No Runtime Change:** When checked, the value of the parameter cannot be overridden. The configured value is always the value. If not checked, parameter values can be overridden at runtime through synchronization processing.
- **Comment:** Any comments applicable to the parameter that describe its purpose or value. This has no effect on the parameter's behavior and is provided for reference purposes only.

Mobile Application - Client Globals

Use this tab to define client globals.

Mobile Application - Client Globals



Mobile Application (Display Mode)

Mobile Application Info

Mobile Application:	SAP_SALES_MANAGER_25	Release:	2.5
Mobile App. Desc.:	SAP CRM Sales 2.5		

Client Globals

Client Global List

Global Group	Global Name	Global Value	Global Scope	Active Flag	No Runtime Change	Comment
ACTIVITY_IMG	5	Phone	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	6	Calendar	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	CFU	FollowUp	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	CP1	Calendar	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	CP2	Calendar	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	CX1	Calendar	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	CX2	Tasks	Application	☑	☑	ACTIVITY_IMG define
ACTIVITY_IMG	006	Calendar	Application	☑	☑	ACTIVITY_IMG define
ACCOUNT_NAVIGATION	1	Sales Area	Application	☑	☑	NAVIGATION defines t
ACCOUNT_NAVIGATION	2	Relationships	Application	☑	☑	NAVIGATION defines t

Client Global Detail

Client Global Group:	ACTIVITY_IMG		
Client Global Name:	5	Global Scope:	Application
Client Global Value:	Phone		
Rule Id:		Use Rule:	□
Rule Input Param:			
Comment:	ACTIVITY_IMG defines the Image Name to use for each Activity Category in CTActivity complex table.		
Active Flag:	☑		
No Runtime Change:	☑		

Mobile Application Info

- **Mobile Application:** (Read Only) The name of the mobile application
- **Release:** (Read Only) The release number of the mobile application
- **Mobile Application Description:** (Read Only) A brief, easy to understand description of the mobile application

Client Global List

Note: The columns in the Client Global List table are read only. Use the Client Global Detail section to make any additions or edits to the table.

- **Add button:** Click **Add** to create a new global. Fill in the fields in the Client Global Detail section to automatically fill in the fields in this table.
- **Delete button:** Click **Delete** to delete an existing global. To delete a global, press the rectangle to the left of the Global Group column in the row you wish to delete and click **Delete**.
- **Global Group:** Global group from the Client Global Detail section

- **Global Name:** Client global group name from the Client Global Detail section
- **Global Value:** Client global value from the Client Global Detail section
- **Global Scope:** Global scope from the Client Global Detail section
- **Active Flag:** When checked, the Active Flag box is checked in the Client Global Detail section
- **No Runtime Change:** When checked, the No Runtime Change box is checked in the Client Global Detail section
- **Comment:** Comments from the Client Global Detail section

Client Global Detail

- **Client Global Group:** The group to which the global belongs. Groups are a means of organizing globals. References to a global include both the group name and the global name.
- **Client Global Name:** The unique name for the global
- **Global Scope:** The scope of the global value. There are two options:
  - Mobile Application: Value for all users of the application
  - Mobile User: Value that can be overridden for individual users.
- **Client Global Value:** The currently configured value of the global. References to the global return this value.
- **Rule ID:** Name, or ID, of the ABAP or class
- **Use Rule:** When checked, the rule listed in the Rule ID field is active. If this value is active, then the Client Global Value field is not used.
- **Rule Input Param:** Parameters to use with the rule. Examples include a key value pair, a user parameter, or a table.
- **Comment:** Displays any comments added to the global to describe its purpose or current value. This has no effect on the global’s behavior and is provided for reference purposes only.
- **Active Flag:** When checked, the client global is activated in the system. Inactive globals are not available to the mobile application.
- **No Runtime Change:** When checked, the value of the global cannot be overridden. The configured value in the Configuration portal will always be the value. Globals without this setting can be overridden at runtime through synchronization processing.

Push Scenario Definition

A push scenario is the data that the SAP Agentry Server can push to mobile application users. For example, an Emergency Service order is pushed down to the SAP Client of a single user or multiple users. The Push Scenario Definition panel provides an interface to configure what data is pushed and the triggers that initiate a push.

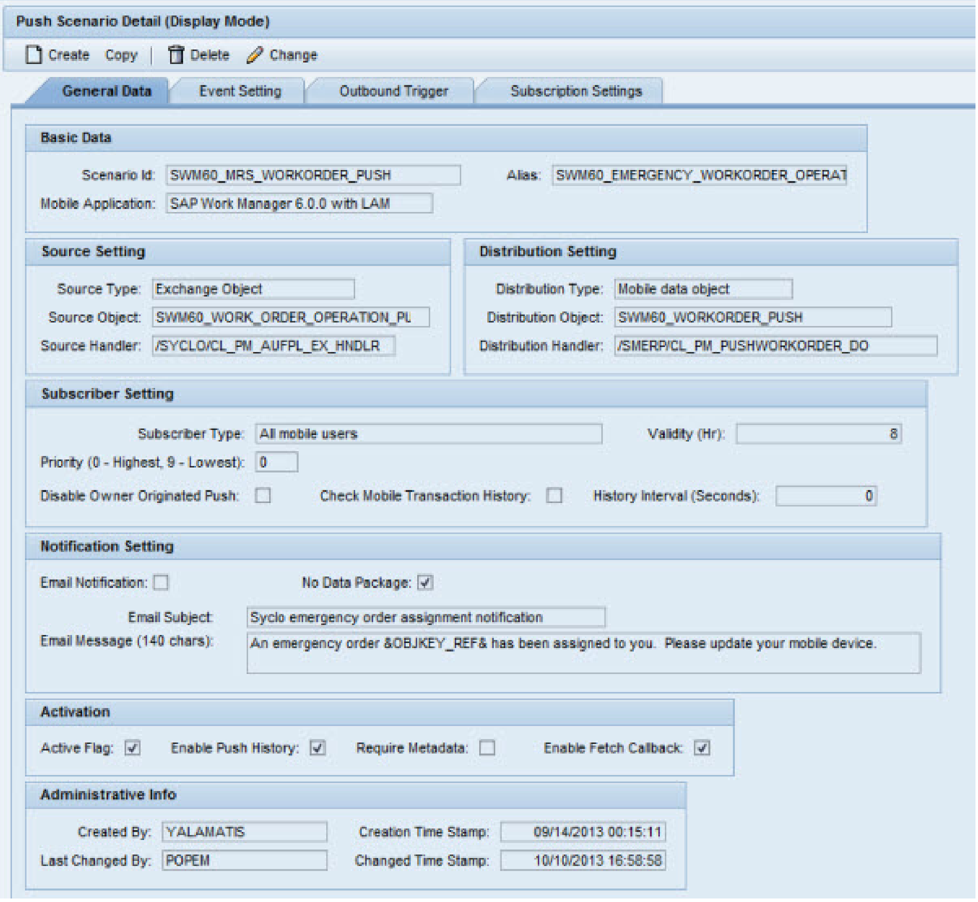
Pushing data from SAP to the mobile client using the ABAP Add-On push framework consists of two key steps:

1. Push relevant data change is detected in SAP and a push instance is registered with the push registry.
2. The new push instance entry in the push registry is processed by a system program (push processor). The data content to be pushed and the recipients for the push instance are determined. For each recipient, a message is generated in his/her out box of the outbound message queue, and waits for pick up.

Push Scenario Definition - General Data

Use the General Data tab of the Push Scenario Detail screen to modify data for a push scenario. You can define source, subscriber, notification, and activation settings.

Push Scenario Definition - General Data Tab



Push Scenario Detail (Display Mode)

Basic Data

Scenario Id: SWM60_MRS_WORKORDER_PUSH
Alias: SWM60_EMERGENCY_WORKORDER_OPERAT
Mobile Application: SAP Work Manager 6.0.0 with LAM

Source Setting

Source Type: Exchange Object
Source Object: SWM60_WORK_ORDER_OPERATION_PL
Source Handler: /SYCLO/CL_PM_AUFPL_EX_HNDLR

Distribution Setting

Distribution Type: Mobile data object
Distribution Object: SWM60_WORKORDER_PUSH
Distribution Handler: /SMERP/CL_PM_PUSHWORKORDER_DO

Subscriber Setting

Subscriber Type: All mobile users
Validity (Hr): 8
Priority (0 - Highest, 9 - Lowest): 0
Disable Owner Originated Push: □
Check Mobile Transaction History: □
History Interval (Seconds): 0

Notification Setting

Email Notification: □
No Data Package: ☑
Email Subject: Syclo emergency order assignment notification
Email Message (140 chars): An emergency order &OBJKEY_REF& has been assigned to you. Please update your mobile device.

Activation

Active Flag: ☑
Enable Push History: ☑
Require Metadata: □
Enable Fetch Callback: ☑

Administrative Info

Created By: YALAMATIS
Creation Time Stamp: 09/14/2013 00:15:11
Last Changed By: POPEM
Changed Time Stamp: 10/10/2013 16:58:58

Basic Data

- **Scenario ID:** Name of the push scenario
- **Alias:** Alias of the push scenario
- **Mobile Application:** Application to which the push scenario belongs

Source Setting

- **Source Type:** Type of source object associated with the push scenario
- **Source Object:** Drop-down list containing the available source objects for the push scenario

  Note: In order for an exchange object to be listed in the drop-down menu, the Push Relevant box must be checked in the Push Settings tab of the Push Scenario Definition screen.

- **Source Handler:** Class handler associated with the source object for the push scenario. This is a non-editable field.

Distribution Setting

- **Distribution Type:** Type of distributed object associated with the push scenario
- **Distribution Object:** Name of the specific object associated with the push scenario, chosen by a drop-down list
- **Distribution Handler:** Name of the class handler from the class repository that is responsible for updating the exchange table. This field is automatically filled when choosing the source object and is not editable.

Subscriber Setting

- **Subscriber Type:** Drop-down list to choose if the push is sent to all users, users with active connections, or users defined in a scenario subscriber list.
- **Validity (Hr):** Amount of time, in hours, of the validity of the data to be pushed to clients. When the time limit has passed, the data within the push scenario is no longer valid and will not be pushed to any more mobile applications.
- **Priority:** The priority assigned to the push, with the default set to 0. The higher the priority setting, the higher the push is in the push queue. For instance, a push priority set to 0 is processed before a different push with a priority set to 5. For push instances with the same set priority or default priority, the pushes are processed in the order in which they were created.

Notification Setting

- **Email Notification:** When checked, sends an email to all users affected by the push scenario.

Note: User set up for email notification must be performed in the System Administration and Monitoring portal in the Administration - User Management panel before email notification is performed.

- **No Data Package:** When checked, the data package, or push information, is not sent to the mobile device when the email notification is sent. A user must connect to the system and perform a regular fetch in order to retrieve the push information. In this way, users do not receive outdated push information if they are seldom actively connected to the system.
- **Email Subject:** Subject, or header, of the email message
- **Email Message (140 chars):** Body of the email message, limited to 140 characters. The limit is to support short messages to websites such as Twitter.

  A special variable, **&OBJKEY&**, is available for use in the body of the email message. This variable is substituted for the actual object identifier content during runtime and presented on the mobile Client.

Activation

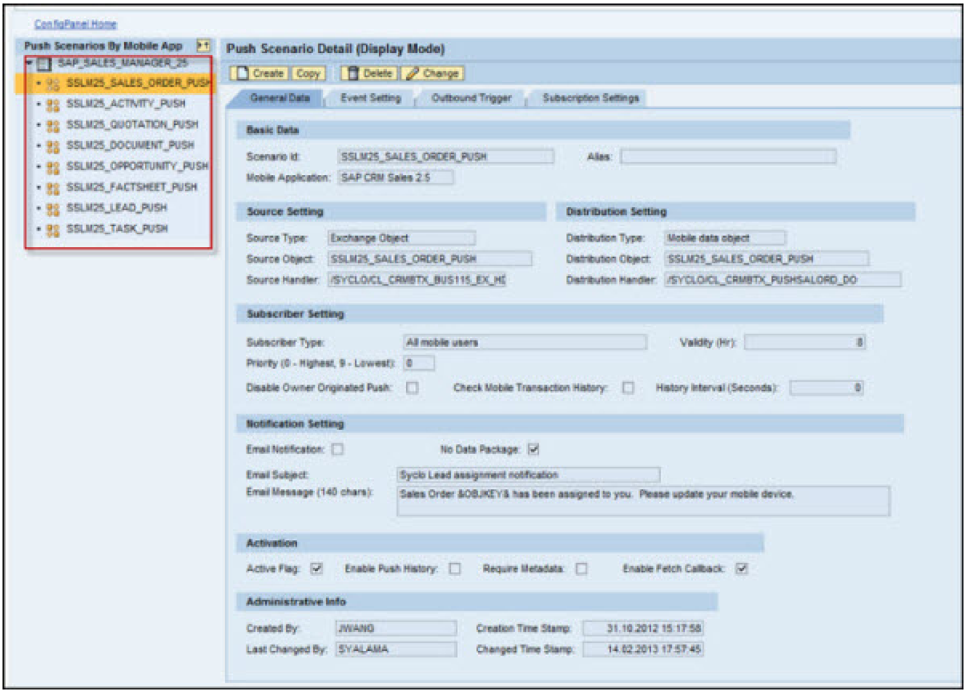
- **Active Flag:** When checked, the push scenario is in an active state. If unchecked, the push scenario is not performed.
- **Enable Push History:** When checked, the push history table in SAP is populated. The push history table contains a list of users and the object keys pushed to those users.
- **Require Metadata:** The metadata table is only populated for the push transaction when this option is checked. When unchecked, the default, the metadata table is not populated, saving Server resources.
- **Enable Fetch Callback**: Fetch callback is used to augment data, in order to make it a two-step process. If implemented, an extra callback routine is invoked when Agentry is retrieving push messages for SAP. Note that you must enable the logic on the back end as well, before fetch callback will be possible. When information changes in SAP, a push is triggered and the distribution agent writes persisting information into the queue. This information is written to the database, which historically has created overhead to the system. The fetch callback avoids the overhead by not performing calculations in the queue; rather, it waits for Agentry to get the push. At that point, calculations for the fetch occur. In this way, the overhead of writing to the database is eliminated. An advantage to implementing fetch callback is that the copy is always fresh, as it is on demand. This eliminates obsolete copies. However, the disadvantage of using fetch callback is that calculations occur every time Agentry demands it.

Administrative Info

- **Created By:** SAP user ID of the person who created the push scenario
- **Creation Time Stamp:** Date and time of the creation of the push scenario
- **Last Changed By:** SAP user ID of the person who last changed the push scenario
- **Changed Time Stamp:** Date and time of the change to the push scenario

Example

The following sample screen shows that a push is enabled for all the transaction types defined in the list. For example, if a sales order is created and/or updated in the SAP back end, the changes for the transaction are pushed to the mobile client to the appropriate user without any transmit or trigger from the mobile client.



Push Scenario Detail (Display Mode)

Basic Data

Scenario Id: SSLM25_SALES_ORDER_PUSH
Alias:
Mobile Application: SAP CRM Sales 2.5

Source Setting

Source Type: Exchange Object
Source Object: SSLM25_SALES_ORDER_PUSH
Source Handler: /SYCLO/CL_CRMBTX_BUS115_EX_HE

Distribution Setting

Distribution Type: Mobile data object
Distribution Object: SSLM25_SALES_ORDER_PUSH
Distribution Handler: /SYCLO/CL_CRMBTX_PUSHSALORD_DO

Subscriber Setting

Subscriber Type: All mobile users
Validity (Hr): 8
Priority (0 - Highest, 9 - Lowest): 0
Disable Owner Originated Push: □
Check Mobile Transaction History: □
History Interval (Seconds): 0

Notification Setting

Email Notification: □
No Data Package: ☑
Email Subject: Syclo Lead assignment notification
Email Message (140 chars): Sales Order &OBJKEY& has been assigned to you. Please update your mobile device.

Activation

Active Flag: ☑
Enable Push History: □
Require Metadata: □
Enable Fetch Callback: ☑

Administrative Info

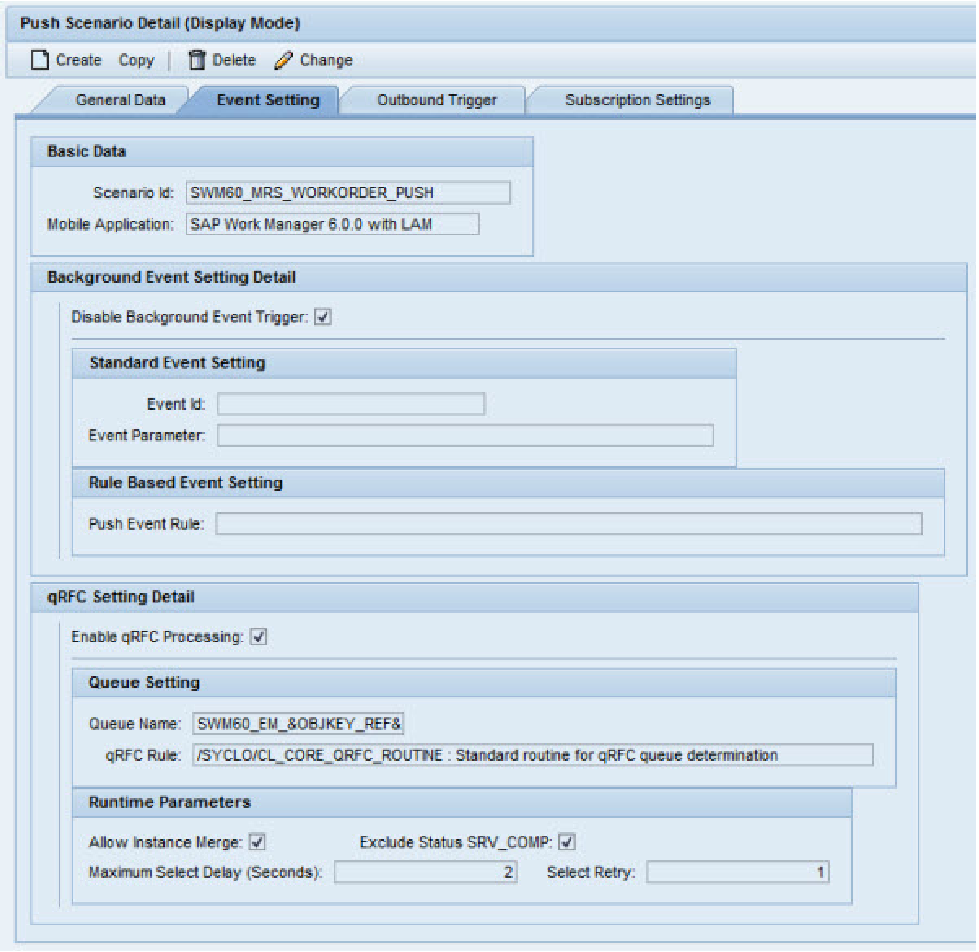
Created By: JWANG
Creation Time Stamp: 31.10.2012 15:17:58
Last Changed By: SYALAMA
Changed Time Stamp: 14.02.2013 17:57:45

Push Scenario Definition - Event Setting

Use the Event Setting tab of the Push Scenario Detail screen to control how the push processing is triggered for specific events when data is changed in SAP and triggers a push. Event settings can be set using variables or by assigning specific events or queues to initiate the push process.

You can initiate push processing either via an SAP background event or via an SAP qRFC. Each of these options is described in the appropriate section following the “Push Scenario Definition - Event Setting Tab” screen shot.

Push Scenario Definition - Event Setting Tab



Basic Data

Scenario Id: SWM60_MRS_WORKORDER_PUSH

Mobile Application: SAP Work Manager 6.0.0 with LAM

Background Event Setting Detail

Disable Background Event Trigger: ☑

Standard Event Setting

Event Id:

Event Parameter:

Rule Based Event Setting

Push Event Rule:

qRFC Setting Detail

Enable qRFC Processing: ☑

Queue Setting

Queue Name: SWM60_EM_&OBJKEY_REF&

qRFC Rule: /SYCLO/CL_CORE_QRFC_ROUTINE : Standard routine for qRFC queue determination

Runtime Parameters

Allow Instance Merge: ☑

Exclude Status SRV_COMP: ☑

Maximum Select Delay (Seconds): 2

Select Retry: 1

Basic Data

- **Scenario ID:** Name of the push scenario
- **Mobile Application:** Application to which the push scenario belongs

Background Event Setting Detail

Complete this section to enable push processing via an SAP background event.

If you enable this option, an SAP background event is raised after a push instance is registered with the push registry. The event can subsequently start an SAP background job that is subscribed to the background event. The started SAP background job can process new push instances in the push registry.

- **Disable Background Event Trigger:** Uncheck this box when using the Background Event Setting Detail fields. When checked, the push process is triggered by the information configured here, rather than by the background event and no background event is raised after the push instance is registered with the push registry.

- **Event ID:** Either a static or variable event ID to be raised. For information on variables, hover over the field to read the tool tip that appears. Event IDs are used to determine when to raise the event.
- **Event Parameter:** A free-text field to configure parameters associated with the Event ID when raised. Parameters can be static or use variables. For information on variables, hover over the field to read the tool tip that appears. Typical parameters can be push scenario IDs, etc.
- **Push Event Rule:** This is a special ABAP class routine that returns the Event ID and event parameters programmatically. A standard routine is provided, but a custom routine can be developed by the customer if there is a business need.

Special variables can be used when defining the Event ID. Special variables are substituted at runtime. If special variables are used, the push event rule must be defined in order for the substitution to occur. Special variables include the following:

- &OBJKEY&
- &OBJKEY_REF&
- &MOBILE_APP&
- &INSTANCE_GUID&
- &SCENARIO_ID&
- &SOURCE_OBJECT&
- &SOURCE_TYPE&
- &DATUM&
- &UZEIT&
- &UNAME&
- &HOST&

Monitoring SAP background event trigger by push instances

To monitor the SAP push, use the following tools:

- Go to the **Administration & Monitoring Portal -> Monitoring -> Push Instance Monitor** and search for push instances using proper search filters. Verify the Event ID and the event parameter of the push instance.
- Use the transaction code **SM37** to search and verify that SAP background jobs are being triggered properly by the Event ID and the event parameters raised during the push process.

qRFC Setting Detail

Complete this section to enable push processing via SAP qRFC.

If you enable this option, after a push instance is registered with the push registry, a new entry is added to the SAP qRFC queue for the specific push instance. The qRFC queue entry is processed automatically by the SAP system. When the qRFC queue is processed, the specific push instance is processed by the push processor.

- **Enable qRFC Processing:** When checked, enables initiating the push process through the qRFC queue.
- **Queue Name:** Either a static or a variable qRFC queue name.
- **qRFC Rule:** This is a special ABAP class routine that returns the qRFC queue name programmatically. A standard routine is provided, but a custom routine can be developed by the customer if there is a business need.
- **Allow Instance Merge:** When checked, if multiple processes are triggered on the same SAP object, the instances are merged to save processing time.
- **Exclude Status SRV_COMP:** When checked, push instances with SRV_COMP status are not reprocessed.
- **Maximum Select Delay:** This is the number of seconds to wait before reading from the push registry DB table. This is only needed for slow systems. This is typically set to 1 or 0. If the system takes longer to write to the database, set to higher than 1.
- **Select Retry:** Number of times to retry the select from DB table is nothing is found. This is only needed for slow systems.

Special variables can be used when defining the queue name. Special variables are substituted at runtime. If special variables are used, the push event rule must be defined in order for the substitution to occur. Special variables include the following:

- &OBJKEY&
- &OBJKEY_REF&
- &MOBILE_APP&
- &INSTANCE_GUID&
- &SCENARIO_ID&
- &SOURCE_OBJECT&
- &SOURCE_TYPE&
- &DATUM&
- &UZEIT&
- &UNAME&
- &HOST&

Monitoring push instance processing via SAP qRFC

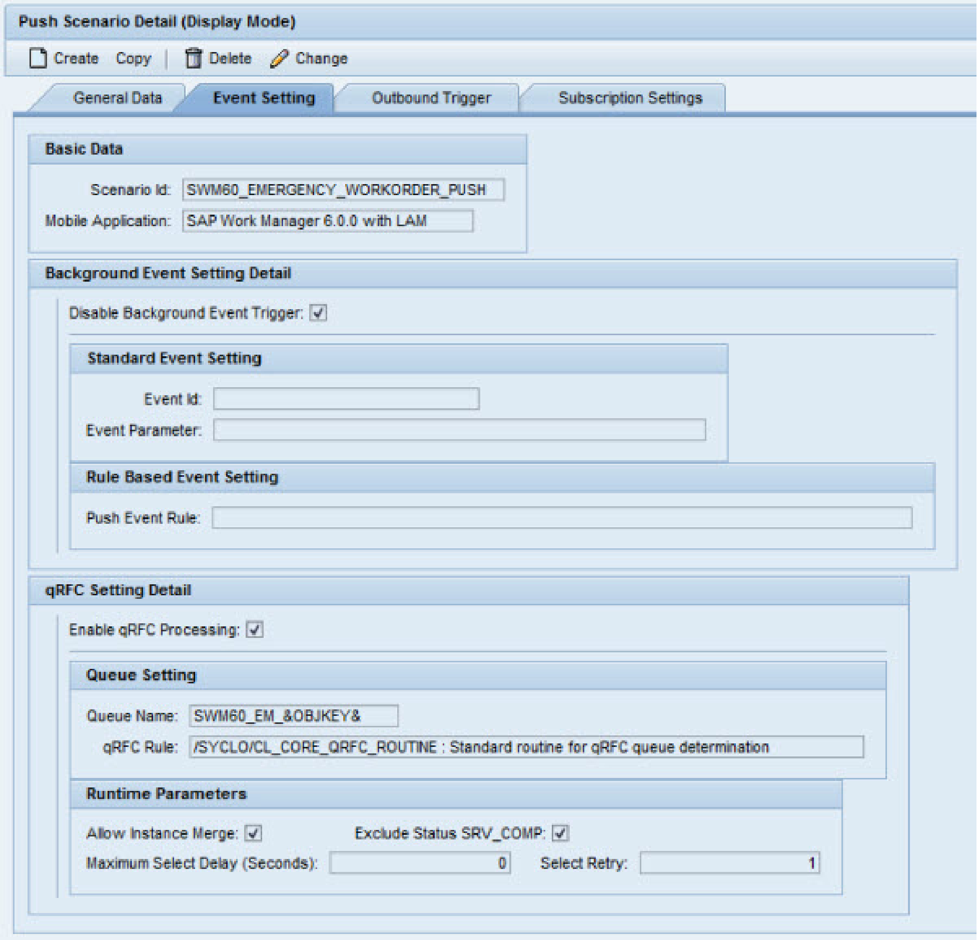
To monitor the SAP push, use the following tools:

- Go to the **Administration & Monitoring Portal -> Monitoring -> Push Instance Monitor** and search for push instances using proper search filters. Verify the qRFC queue name of the push instance.

- Use the transaction code **SMQ1** to search and verify that there are not outstanding entries waiting in the qRFC queue.

Example

The following sample screen shows that a push is enabled via the SAP qRFC queue.



Push Scenario Detail (Display Mode)

Basic Data

Scenario Id: SWM60_EMERGENCY_WORKORDER_PUSH

Mobile Application: SAP Work Manager 6.0.0 with LAM

Background Event Setting Detail

Disable Background Event Trigger: ☑

Standard Event Setting

Event Id:

Event Parameter:

Rule Based Event Setting

Push Event Rule:

qRFC Setting Detail

Enable qRFC Processing: ☑

Queue Setting

Queue Name: SWM60_EM_&OBJKEY&

qRFC Rule: /SYCLO/CL_CORE_QRFC_ROUTINE : Standard routine for qRFC queue determination

Runtime Parameters

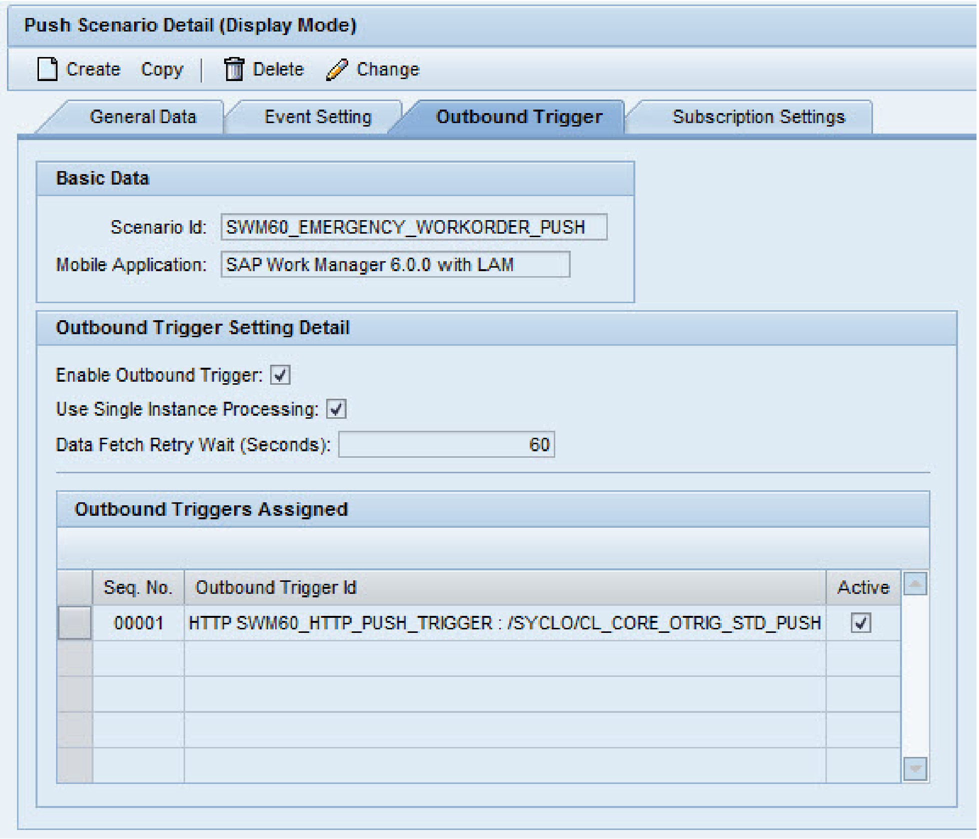
Allow Instance Merge: ☑ Exclude Status SRV_COMP: ☑

Maximum Select Delay (Seconds): 0 Select Retry: 1

Push Scenario Definition - Outbound Trigger

Use the Outbound Trigger tab of the Push Scenario Detail screen to assign one or more outbound triggers to the push scenario. After push generation, the outbound trigger should be enabled to notify the Agentry application in the SAP Mobile Server.

Push Scenario Definition - Outbound Trigger Tab



Basic Data

Scenario Id: SWM60_EMERGENCY_WORKORDER_PUSH

Mobile Application: SAP Work Manager 6.0.0 with LAM

Outbound Trigger Setting Detail

Enable Outbound Trigger: ☑

Use Single Instance Processing: ☑

Data Fetch Retry Wait (Seconds): 60

Outbound Triggers Assigned

Seq. No.	Outbound Trigger Id	Active
00001	HTTP SWM60_HTTP_PUSH_TRIGGER : /SYCLO/CL_CORE_OTRIG_STD_PUSH	☑

Basic Data

- **Scenario ID:** Name of the push scenario
- **Mobile Application:** Application to which the push scenario belongs

Outbound Trigger Setting Detail

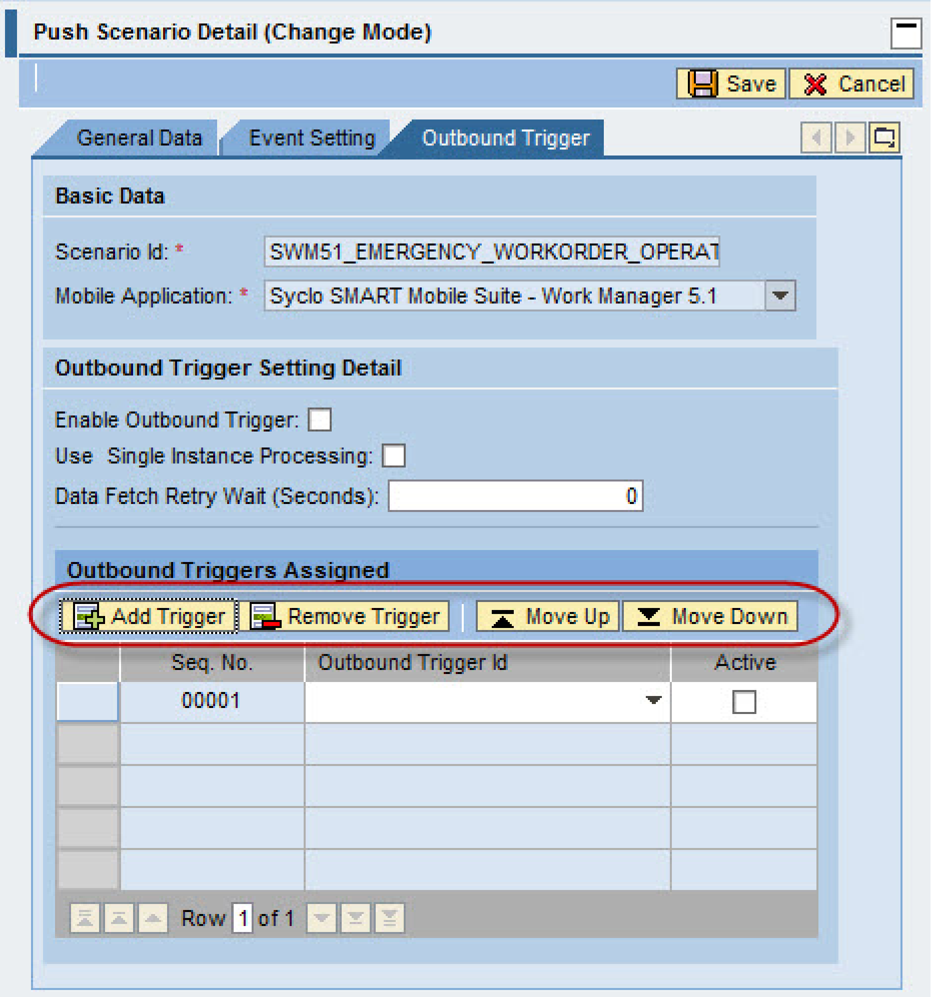
- **Enable Outbound Trigger:** When checked, allows all “active” outbound triggers to process.
- **Use Single Instance Processing:** When checked, the system sends each outbound trigger action as a separate XML file. This is usually reserved for test mode. In most instances, you will want to leave this option unchecked to initiate batch processing rather than single instance processing.
- **Data Fetch Retry Wait (Seconds):** Setting a data fetch retry wait reduces the hits to the Server in cases of Server miscommunication.

Outbound Trigger Setting Detail

This table shows all outbound triggers assigned to this push scenario with the following three fields:

- **Seq. No.:** When there are multiple triggers for the push scenario, the sequence number defines the order in which the triggers are processed.
- **Outbound Trigger ID:** The identification number of the outbound trigger
- **Active:** Indicates whether or not the outbound trigger is active. Note: you must use the **Enable Outbound Trigger** checkbox to enable processing for active triggers.

Push Scenario Definition - Outbound Trigger Tab in Change Mode



Push Scenario Detail (Change Mode)

Basic Data

Scenario Id: * SWM51_EMERGENCY_WORKORDER_OPERAT

Mobile Application: * Syclo SMART Mobile Suite - Work Manager 5.1

Outbound Trigger Setting Detail

Enable Outbound Trigger: □

Use Single Instance Processing: □

Data Fetch Retry Wait (Seconds): 0

Outbound Triggers Assigned

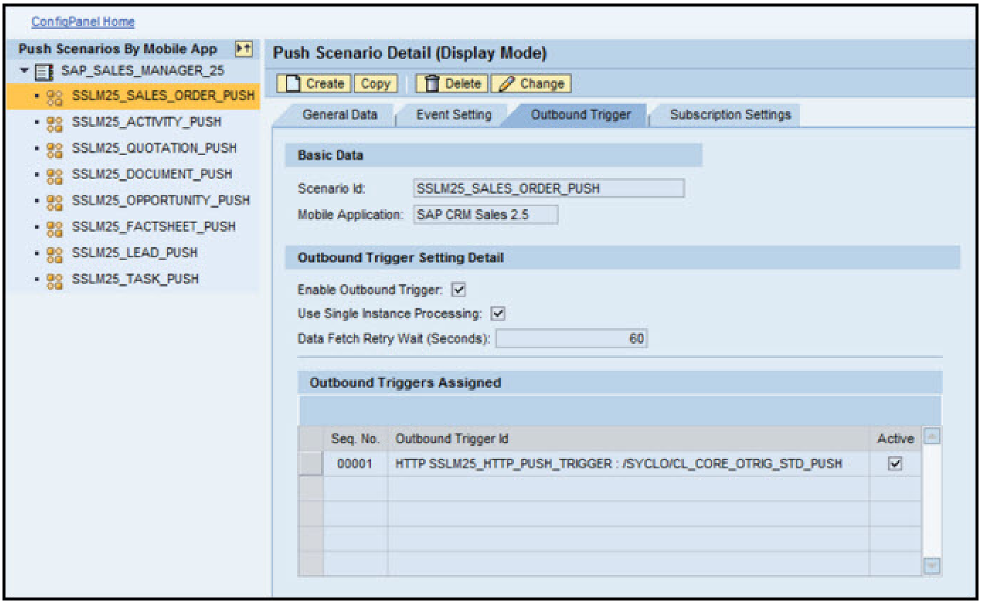
Seq. No.	Outbound Trigger Id	Active
00001		□

Row 1 of 1

When in Change Mode, use the action buttons that display to add or remove triggers to the push scenario or to change the sequence of the selected triggers.

Example

The following sample screen shows that an outbound trigger is enabled for Sales Order. For example, if a sales order is created and/or updated in the SAP back end, notifications of the changes for the transaction are sent to the Agentry application.



Push Scenarios By Mobile App

- SAP_SALES_MANAGER_25
  - SSLM25_SALES_ORDER_PUSH
  - SSLM25_ACTIVITY_PUSH
  - SSLM25_QUOTATION_PUSH
  - SSLM25_DOCUMENT_PUSH
  - SSLM25_OPPORTUNITY_PUSH
  - SSLM25_FACTSHEET_PUSH
  - SSLM25_LEAD_PUSH
  - SSLM25_TASK_PUSH

Push Scenario Detail (Display Mode)

Basic Data

Scenario Id: SSLM25_SALES_ORDER_PUSH

Mobile Application: SAP CRM Sales 2.5

Outbound Trigger Setting Detail

Enable Outbound Trigger: ☑

Use Single Instance Processing: ☑

Data Fetch Retry Wait (Seconds): 60

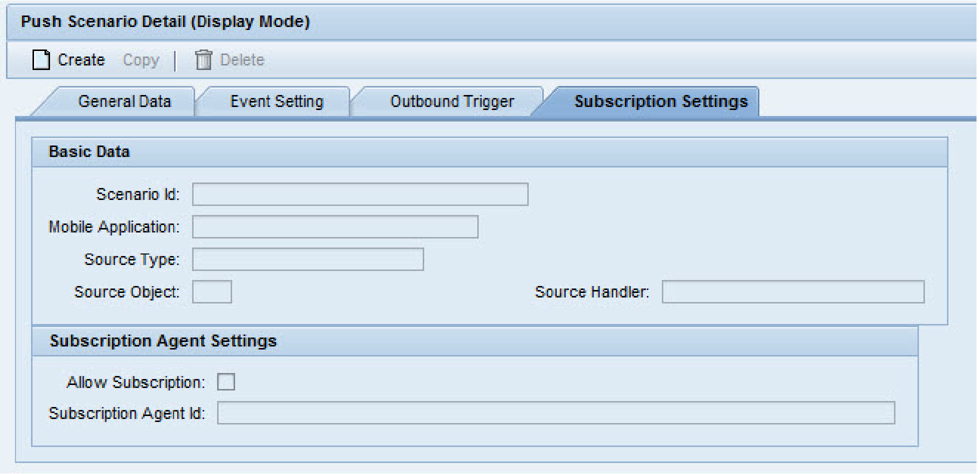
Outbound Triggers Assigned

Seq. No.	Outbound Trigger Id	Active
00001	HTTP SSLM25_HTTP_PUSH_TRIGGER : /SYCLO/CL_CORE_OTRIG_STD_PUSH	☑

Push Scenario Definitions - Subscription Settings

Use this tab to define the push scenarios allowed through subscription. Enable the subscription settings when the push is based on a subscription-based push or receives OnDemand requests from the mobile client.

Subscription Settings



Once you select one of the available push scenarios from the list in the left panel, the Basic Data section is automatically populated.

Subscription Agent Settings

To allow subscriptions to the push scenario, check the **Allow Subscription** check box and enter a **Subscription Agent ID**.

Note: For subscriptions to work, you must also select the Source Type **Client On Demand Request**, which appropriately changes the Subscriber Type. You must also complete the information on the Subscription Agent Definition screen.

The subscription agent is a simple Yes/No gatekeeper. It either accepts or does not accept the push request based on logic in the back end, or based on quota requests. If the subscription agent approves the push request, it generates a push instance. The subscription agent then puts a subscription request into the Subscription Request table. The table dictates which user requests which request key. After this, an instance is generated as an OnDemand request, with a reference to the Subscription Request table. The push is then processed as a normal push, through the push channel, rather than through the fetch channel.

OnDemand Push

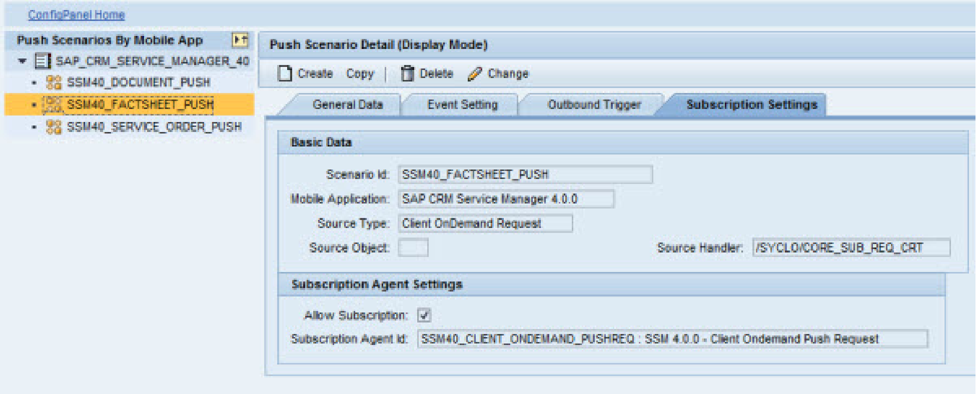
With an OnDemand push, the client receives information through the push channel, rather than the fetch channel. In this way, the user can still work on the device, even while the push is adding data to the device. As an example, a user can request a PDF document to be added to the device, and requests it. An OnDemand push is used to retrieve that PDF from the back end and add it to the device through the push channel. Use the **Source Type** field to configure which type of push is required for the chosen subscription setting.

Subscription-Based Push

A subscription-based push uses the same concept as an OnDemand push. When there is a change in the back end, the push will figure out who subscribes to the data being pushed and then push that data out to the users, to their client devices. Use the **Source Type** field to configure which type of push is required for the chosen subscription setting.

Example

The following sample screen shows that OnDemand push is enabled for FactSheet generation push. To allow subscriptions to the push scenario, check the **Allow Subscriptions** check box and enter a Subscription Agent ID.



Basic Data

Scenario Id: SSM40_FACTSHEET_PUSH

Mobile Application: SAP CRM Service Manager 4.0.0

Source Type: Client OnDemand Request

Source Object:

Source Handler: /SYCLO/CORE_SUB_REQ_CRT

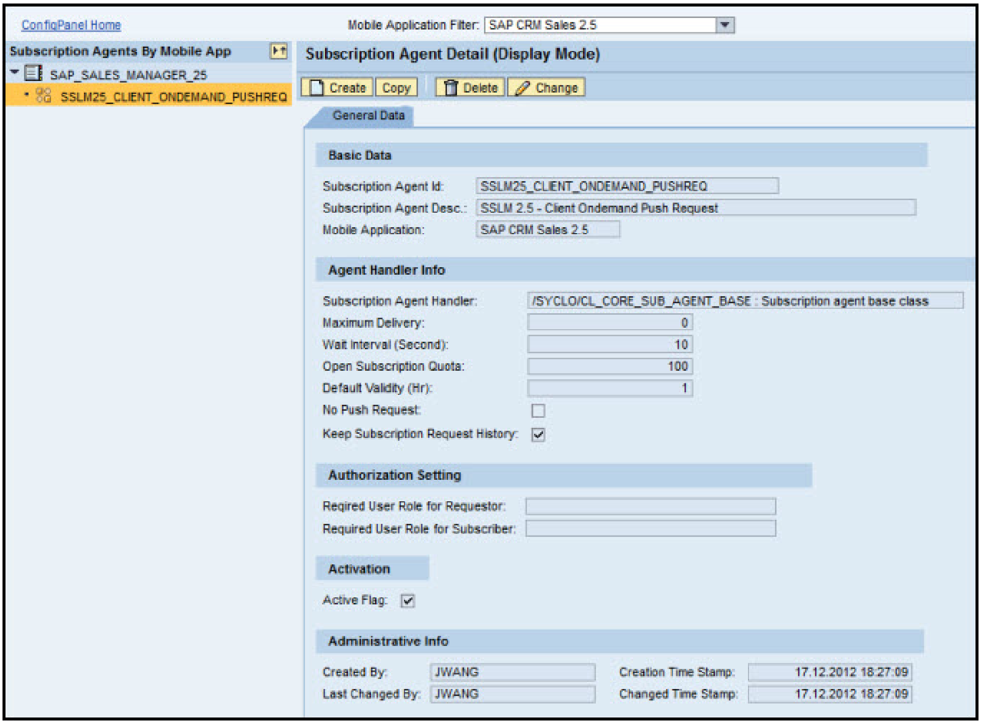
Subscription Agent Settings

Allow Subscription: ☑

Subscription Agent Id: SSM40_CLIENT_ONDEMAND_PUSHREQ : SSM 4.0.0 - Client Ondemand Push Request

Subscription Agent Definition

The Subscription Agent Definition page allows you to define how subscription requests for back end System data are handled. On Demand subscriptions allow you to define push options. For instance, some accounts or activities have attachments which are not automatically pushed down. If subscribed, however, a Sales Manager can get those attachments On Demand, based on the settings defined on this page.



Mobile Application Filter: SAP CRM Sales 2.5

Subscription Agents By Mobile App

- SAP_SALES_MANAGER_25
  - SSLM25_CLIENT_ONDEMAND_PUSHREQ

Subscription Agent Detail (Display Mode)

Basic Data

Subscription Agent Id: SSLM25_CLIENT_ONDEMAND_PUSHREQ

Subscription Agent Desc.: SSLM 2.5 - Client Ondemand Push Request

Mobile Application: SAP CRM Sales 2.5

Agent Handler Info

Subscription Agent Handler: /SYCLO/CL_CORE_SUB_AGENT_BASE : Subscription agent base class

Maximum Delivery: 0

Wait Interval (Second): 10

Open Subscription Quota: 100

Default Validity (Hr): 1

No Push Request: □

Keep Subscription Request History: ☑

Authorization Setting

Reqired User Role for Requestor:

Required User Role for Subscriber:

Activation

Active Flag: ☑

Administrative Info

Created By: JWANG

Last Changed By: JWANG

Creation Time Stamp: 17.12.2012 18:27:09

Changed Time Stamp: 17.12.2012 18:27:09

Basic Data

- Subscription Agent ID: the name of the on demand push requirement subscription
- **Subscription Agent Description:** text description of the agent ID
- **Mobile Application:** the most current valid application version

Agent Handler Info

- **Subscription Agent Handler:** the handler name and location for this subscription agent
- **Maximum Delivery:** maximum number of items that will be delivered through the On Demand request
- **Wait Interval (Second):** this is the time, in seconds, to wait between pushes
- **Open Subscription Quota:** Enter a non-zero value to limit the number of open subscription requests in the system. If this field has a value, no new subscription requests are accepted once this quota is reached.
- **Default Validity (Hr):** the amount of time, in hours, that the On Demand request is valid
- **No Push Request:** check this box to indicate Client On Demand Push Requests are not allowed
- **Keep Subscription Request History:** check this box to keep a history log of all subscription requests

Authorization Setting

- **Required User Role for Requestor:** To require requestors to have a specific role, enter the role here
- **Required User Role for Subscribers:** To require subscribers to have a specific role, enter the role here

Activation

- **Active Flag:** check this box to indicate that the On Demand Push Request is active

Next Steps

In order for the Subscription Agent ID to work, you must do the following:

1. Go to the Push Scenario Detail screen and on the General Data tab, select the Source Type **Client On Demand Request**. This changes the subscriber type.
2. On the Push Scenario Detail screen, go to the Subscription Settings tab and select **Allow Subscription**.

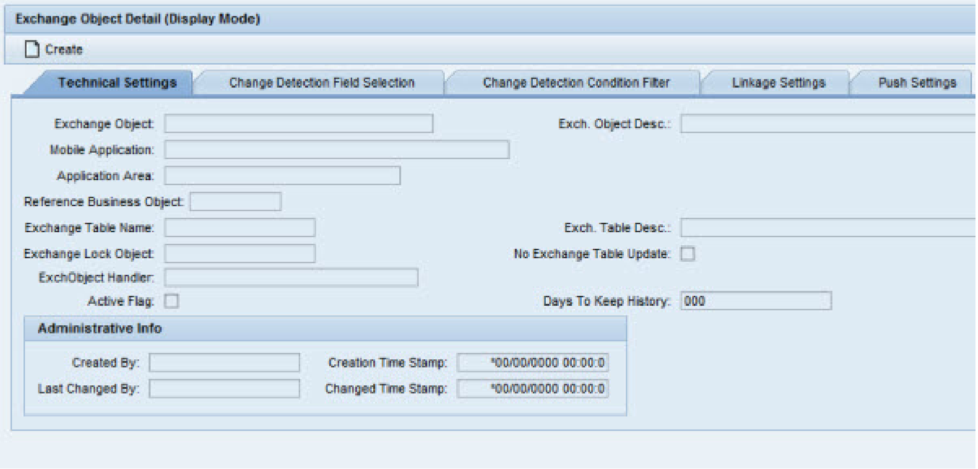
Exchange Object Configuration

The exchange object defines what in the exchange table is to be updated in the exchange persistent layer, what class handler should be called to update the exchange table, and what fields are related to the change detection. Use the Configuration Panel to specify which changes are relevant to the mobile application and what conditions must be satisfied for an update action to be triggered.

Exchange Object - Technical Settings

Use the Technical Settings tab to configure basic settings for an exchange object.

Exchange Object - Technical Settings Tab


Exchange Object Detail (Display Mode)
Create
Technical Settings
Change Detection Field Selection
Change Detection Condition Filter
Linkage Settings
Push Settings

Exchange Object:

Exch. Object Desc.:

Mobile Application:

Application Area:

Reference Business Object:

Exchange Table Name:

Exch. Table Desc.:

Exchange Lock Object:

No Exchange Table Update: □

ExchObject Handler:

Active Flag: □

Days To Keep History: 000

Administrative Info

Created By:

Creation Time Stamp: '00/00/0000 00:00:0

Last Changed By:

Changed Time Stamp: '00/00/0000 00:00:0

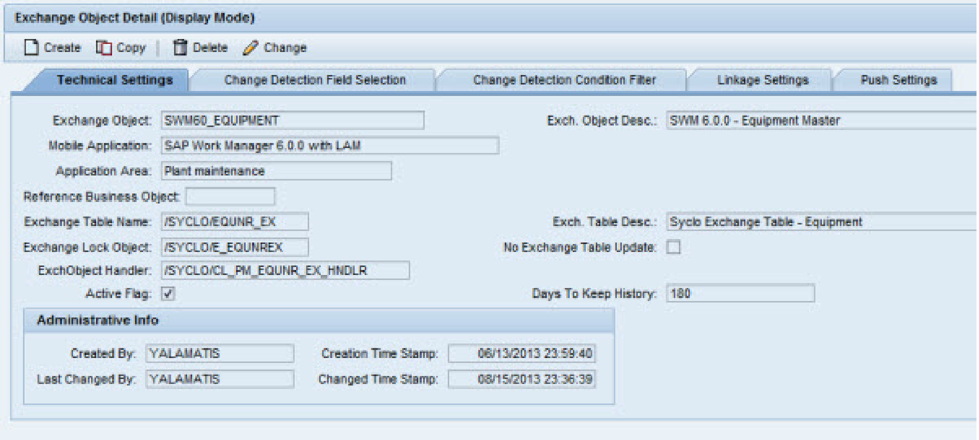
- **Exchange Object:** ID of the exchange object, limited to 40 characters
- **Exchange Object Description:** Brief description of the exchange object, limited to 60 characters
- **Mobile Application:** Specific mobile application to which the exchange object belongs using a drop-down selection field
- **Application Area:** Classifies the exchange object based on standard SAP application areas using a drop-down selection field
- **Reference Business Object:** Standard SAP business object
- **Exchange Table Name:** Name of the table stored in SAP that contains the technical data
- **Exchange Table Description:** Brief description of the exchange table
- **Exchange Lock Object:** SAP lock object used when updating the exchange table
- **No Exchange Table Update:** When checked, the record is not written to the exchange table in SAP when the record is changed.
- **Exchange Object Handler:** Name of class handler from the repository that is responsible for updating the exchange table
- **Active Flag:** When checked the exchange object is in an active state. If unchecked, the exchange object performs no actions.
- **Days to Keep History:** Number of days the historical data should be kept in the exchange table.

Administrative Info

- **Created By:** SAP user ID of the person who created the exchange object
- **Creation Time Stamp:** Date and time of the creation of the exchange object
- **Last Changed By:** SAP user ID of the person who last changed the exchange object
- **Changed Time Stamp:** Date and time of the change to the exchange object

Example

The following screen sample shows that the exchange process is enabled for Equipment. Any changes for Equipment master data will be recorded in the exchange table and transmitted to the Client during the next transmit.



Exchange Object Detail (Display Mode)

Exchange Object: SWM60_EQUIPMENT
Exch. Object Desc.: SWM 6.0.0 - Equipment Master
Mobile Application: SAP Work Manager 6.0.0 with LAM
Application Area: Plant maintenance
Reference Business Object:
Exchange Table Name: /SYCLO/EQUNR_EX
Exch. Table Desc.: Syclo Exchange Table - Equipment
Exchange Lock Object: /SYCLO/E_EQUNREX
No Exchange Table Update: □
ExchObject Handler: /SYCLO/CL_PM_EQUNR_EX_HNDLR
Active Flag: ☑
Days To Keep History: 180

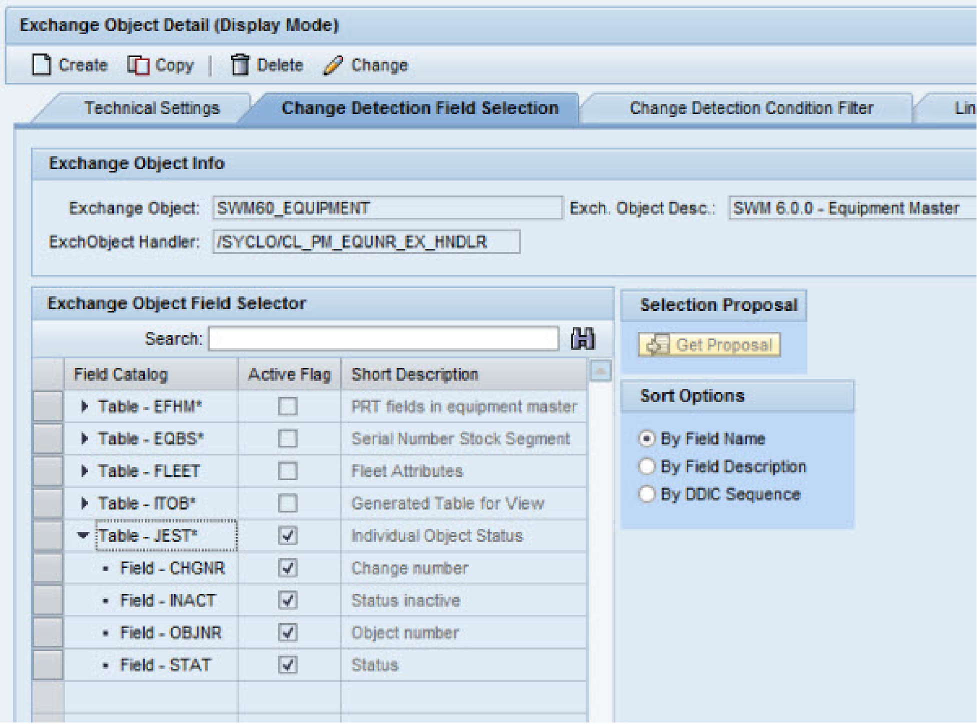
Administrative Info

Created By: YALAMATIS
Creation Time Stamp: 06/13/2013 23:59:40
Last Changed By: YALAMATIS
Changed Time Stamp: 08/15/2013 23:36:39

Exchange Object - Change Detection Field Selection

The Change Detection Field Selection tab provides the ability to optimize the change detection process for mobile applications. If a value change is detected for any fields within the group, the object identifier is written to the exchange table, indicating that a change has been made. If the Active Flag is not checked for a field, any value changes made to that field will not be detected and recorded to SAP during the exchange process. By default, all fields are initially checked.

Figure 11: Exchange Object - Change Detection Field Selection Tab


Exchange Object Detail (Display Mode)
Create
Copy
Delete
Change
Technical Settings
Change Detection Field Selection
Change Detection Condition Filter
Lin
Get Proposal

Exchange Object Info

Exchange Object: SWM60_EQUIPMENT
Exch. Object Desc.: SWM 6.0.0 - Equipment Master
ExchObject Handler: /SYCLO/CL_PM_EQUNR_EX_HNDLR

Exchange Object Field Selector

Search:

Field Catalog	Active Flag	Short Description
▸ Table - EFHM*	□	PRT fields in equipment master
▸ Table - EQBS*	□	Serial Number Stock Segment
▸ Table - FLEET	□	Fleet Attributes
▸ Table - ITOB*	□	Generated Table for View
▾ Table - JEST*	☑	Individual Object Status
• Field - CHGNR	☑	Change number
• Field - INACT	☑	Status inactive
• Field - OBJNR	☑	Object number
• Field - STAT	☑	Status

Selection Proposal

Sort Options

● By Field Name
○ By Field Description
○ By DDIC Sequence

Exchange Object by Application Area

The Exchange Object by Application tree lists all application areas and the exchange objects linked to each application area. Expand the tree by clicking on the arrows to the left of the application area to display the exchange objects associated with it.

Exchange Object Info

- **Exchange Object:** ID of the exchange object. This is a non-editable field.
- **Exchange Object Description:** Brief description of the exchange object. This is a non-editable field.
- **Exchange Object Handler:** Name of class handler from the repository that is responsible for updating the exchange table. This is a non-editable field.

Exchange Object Field Selector

- **Field Catalog:** All fields that can be detected by the class handler when changes are made, grouped by the technical table name of the SAP business object. This is a non-editable field.
- **Active Flag:** When checked, either the table or a field within a table is active. Any value change to the selected field will be detected by the class handler.

Note: Checking the Active Flag box on a Table row selects all fields within the table.

- **Short Description:** Brief description of the table or the field within the table. This is a non-editable field.

Selection Proposal

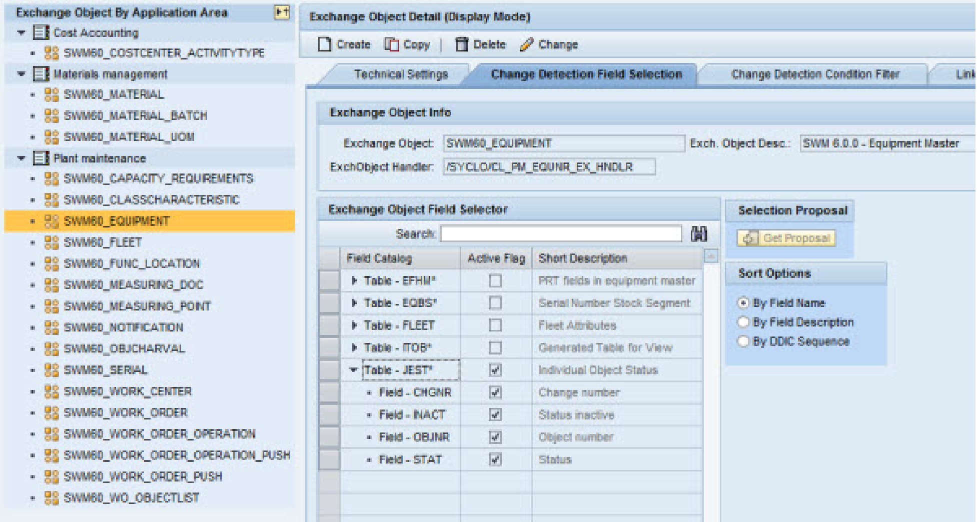
In a typical installation, it is not desirable to have all fields in all exchange tables checked as active for change detection. Rather, only the fields that are active on the mobile data object that are brought down to the mobile device should also be active in the exchange object.

Based on mobile data object usage in the mobile application, the Selection Proposal will examine the active flags that are checked for an exchange object's table fields and provide recommendations to the administrator on which fields in the exchange object should be checked or unchecked.

Example

The properties for the enabled exchange object Equipment that should be captured and recorded in the exchange table are defined on the Change Detection Condition Filter tab.

The properties for account general data that triggers the exchange are defined on this tab, as shown in the following sample.


Exchange Object By Application Area
Exchange Object Detail (Display Mode)
Create
Copy
Delete
Change
Technical Settings
Change Detection Field Selection
Change Detection Condition Filter
Link
Exchange Object Info
Exchange Object Field Selector
Search:
Selection Proposal
Get Proposal
Sort Options

- Cost Accounting
  - SWM60_COSTCENTER_ACTIVITYTYPE
- Materials management
  - SWM60_MATERIAL
  - SWM60_MATERIAL_BATCH
  - SWM60_MATERIAL_UOM
- Plant maintenance
  - SWM60_CAPACITY_REQUIREMENTS
  - SWM60_CLASSCHARACTERISTIC
  - SWM60_EQUIPMENT
  - SWM60_FLEET
  - SWM60_FUNC_LOCATION
  - SWM60_MEASURING_DOC
  - SWM60_MEASURING_POINT
  - SWM60_NOTIFICATION
  - SWM60_OBJCHARVAL
  - SWM60_SERIAL
  - SWM60_WORK_CENTER
  - SWM60_WORK_ORDER
  - SWM60_WORK_ORDER_OPERATION
  - SWM60_WORK_ORDER_OPERATION_PUSH
  - SWM60_WORK_ORDER_PUSH
  - SWM60_WO_OBJECTLIST

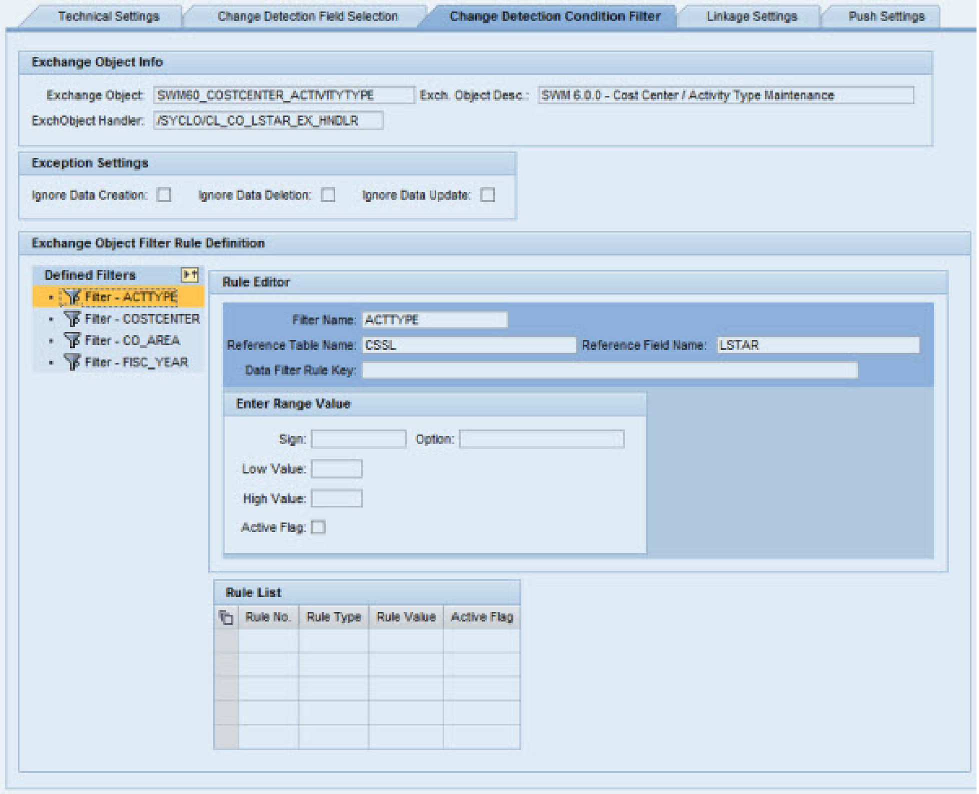
Exchange Object: SWM60_EQUIPMENT Exch. Object Desc.: SWM 6.0.0 - Equipment Master
ExchObject Handler: /SYCLO/CL_PM_EQUNR_EX_HNDLR

Field Catalog	Active Flag	Short Description
▸ Table - EFHM*	□	PRT fields in equipment master
▸ Table - EQBS*	□	Serial Number Stock Segment
▸ Table - FLEET	□	Fleet Attributes
▸ Table - ITOB*	□	Generated Table for View
▾ Table - JEST*	☑	Individual Object Status
• Field - CHGNR	☑	Change number
• Field - INACT	☑	Status inactive
• Field - OBJNR	☑	Object number
• Field - STAT	☑	Status

● By Field Name
○ By Field Description
○ By DDIC Sequence

Exchange Object - Change Detection Condition Filter

The Change Detection Condition Filter tab provides the ability to restrict change detection based on data content. For exchange handlers to support this feature, you must define data filter conditions for which the underlying SAP business object must qualify before the change detection process is triggered. The condition is defined at the table field level and is in the SAP range table format.


Technical Settings
Change Detection Field Selection
Change Detection Condition Filter
Linkage Settings
Push Settings

Exchange Object Info

Exchange Object: SWM60_COSTCENTER_ACTIVITYTYPE
Exch. Object Desc.: SWM 6.0.0 - Cost Center / Activity Type Maintenance
ExchObject Handler: /SYCLO/CL_CO_LSTAR_EX_HNDLR

Exception Settings

Ignore Data Creation: □ Ignore Data Deletion: □ Ignore Data Update: □

Exchange Object Filter Rule Definition

Defined Filters
- Filter - ACTTYPE
- Filter - COSTCENTER
- Filter - CO_AREA
- Filter - FISC_YEAR

Rule Editor

Filter Name: ACTTYPE
Reference Table Name: CSSL
Reference Field Name: LSTAR
Data Filter Rule Key:

Enter Range Value

Sign:
Option:
Low Value:
High Value:
Active Flag: □

Rule List

Rule No.	Rule Type	Rule Value	Active Flag

Exchange Table Object Info

- **Exchange Object:** ID of the exchange object. This is a non-editable field.
- **Exchange Object Description:** Brief description of the exchange object. This is a non-editable field.
- **Exchange Object Handler:** Name of class handler from the repository that is responsible for updating the exchange table. This is a non-editable field.

Exception Settings

- **Ignore Data Creation:** When checked, new records/data created are not processed to the exchange table
- **Ignore Data Deletion:** When checked, deleted records/data are not processed to the exchange table
- **Ignore Data Update:** When checked, updated records/data are not processed to the exchange table

Defined Filters

The Defined Filters box lists all data filters supported by the class handlers.

Rule Editor

- **Filter Name:** Name of the filter as defined in the class handler method. This information is defined by the class handler developer and is not editable.
- **Reference Table Name:** Technical name of the SAP database table where this filter is applied. This information is defined by the class handler developer and is not editable.
- **Reference Field Name:** Technical name of the SAP database table field where this filter is applied. This information is defined by the class handler developer and is not editable.
- **Data Filter Rule Key:** Internal technical key used by the framework at runtime

Enter Range Value

- **Sign:** Value for the SAP Range table column SIGN
- **Option:** Value for the SAP Range table column OPTION
- **Low Value:** Value for the SAP Range table column LOW
- **High Value:** Value for the SAP Range table column HIGH
- **Active Flag:** When checked, the rule is active

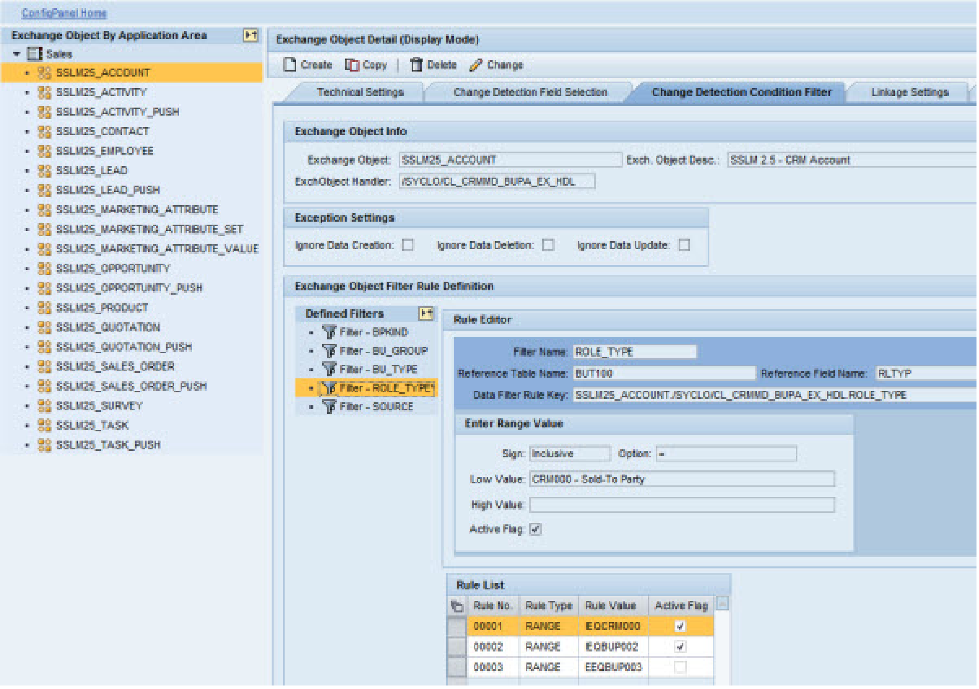
Rule List

The Rule List table displays a list of rules that have been defined.

- **Rule No.:** Number of the rule that is defined, in chronological order
- **Rule Type:** Rule type, automatically assigned by the rule type selected in the DOF Rule Type field
- **Rule Value:** Internal rule value saved by SAP
- **Active Flag:** When checked, the rule is active

Example

The following sample screen shows that any exchange detected for the Account will be considered only if the Account is maintained in one of the roles defined in the ROLE_TYPE filter criteria.



Exchange Object: SSLM25_ACCOUNT

Exch. Object Desc.: SSLM 2.5 - CRM Account

ExchObject Handler: /SYCLO/CL_CRMMD_BUPA_EX_HDL

Ignore Data Creation: □ Ignore Data Deletion: □ Ignore Data Update: □

Filter Name: ROLE_TYPE

Reference Table Name: BUT100 Reference Field Name: RLTYP

Data Filter Rule Key: SSLM25_ACCOUNT./SYCLO/CL_CRMMD_BUPA_EX_HDL.ROLE_TYPE

Sign: Inclusive Option: =

Low Value: CRM000 - Sold-To Party

High Value:

Active Flag: ☑

Rule List

Rule No.	Rule Type	Rule Value	Active Flag
00001	RANGE	IEQCRM000	☑
00002	RANGE	IEQBUP002	☑
00003	RANGE	EEQBUP003	□

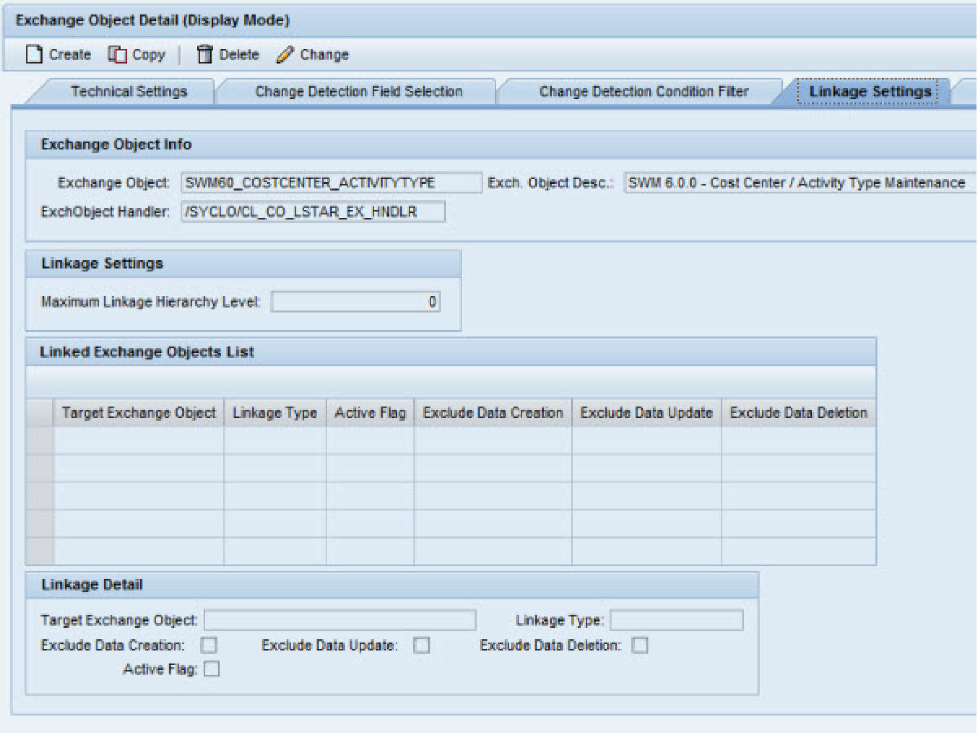
The properties for the enabled exchange object Account that should be captured and recorded in the exchange table are listed in this tab.

The properties for account general data that triggers the exchange are defined on the Change Detection Field Selection tab.

Exchange Object - Linkage Settings

The Linkage Settings tab allows the exchange objects that are linked together to communicate with each other. The communication is one-directional, with the exchange object sending information to the object(s) listed in the Linked Exchange Objects List. When there is a value change to the exchange object, that value change information is passed on to the linked exchange objects. The linked exchange objects then go through additional processes related to the value change.

Exchange Object - Linkage Settings


Exchange Object Detail (Display Mode)
Create
Copy
Delete
Change
Technical Settings
Change Detection Field Selection
Change Detection Condition Filter
Linkage Settings

Exchange Object Info

Exchange Object: SWM60_COSTCENTER_ACTIVITYTYPE

Exch. Object Desc.: SWM 6.0.0 - Cost Center / Activity Type Maintenance

ExchObject Handler: /SYCLO/CL_CO_LSTAR_EX_HNDLR

Linkage Settings

Maximum Linkage Hierarchy Level: 0

Linked Exchange Objects List

Target Exchange Object	Linkage Type	Active Flag	Exclude Data Creation	Exclude Data Update	Exclude Data Deletion

Linkage Detail

Target Exchange Object:

Linkage Type:

Exclude Data Creation: □ Exclude Data Update: □ Exclude Data Deletion: □

Active Flag: □

Exchange Object Info

- **Exchange Object:** ID of the exchange object. This is a non-editable field.
- **Exchange Object Description:** Brief description of the exchange object. This is a non-editable field.
- **Exchange Object Handler:** Name of class handler from the repository that is responsible for updating the exchange table. This is a non-editable field.

Linkage Settings

With the Linkage Hierarchy, you have the ability to go ‘n’ levels deep with linked objects. Any node changes triggers changes to the lower-level nodes linked to the parent node. These relationships are defined in the Linked Exchange Objects list.

For example:

```
Measuring Point
    Functional Location
                        Work Order
```

If the Measuring Point data changes, then the Functional Location and the Work Order will change as well.

- **Maximum Linkage Hierarchy Level:** the maximum number of levels allowed to be set for linkage in the hierarchy

Linked Exchange Objects List

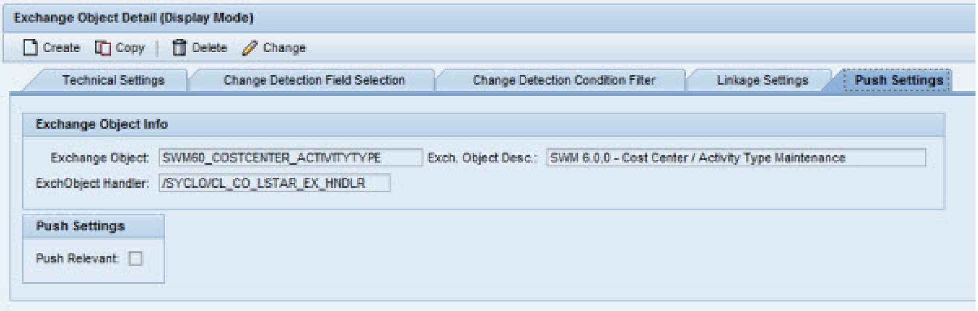
- **Add Linkage button:** Use this button to add a new linked exchange object. Click the **Add Linkage** button and use the fields in the Linkage Detail section to add information.
- **Delete Linkage button:** Use this button to delete a linkage. Highlight the row you wish to delete by clicking on the rectangle to the left of the Target Exchange Object cell and click the **Delete Linkage** button.
- **Target Exchange Object:** Displays the target exchange object selected in the Linkage Detail section.
- **Linkage Type:** Displays either an ‘A’ for asynchronous or an ‘S’ for synchronous, selected in the Linkage Detail section.
- **Active Flag:** When checked in the Linkage Detail section, the linkage between exchange objects is active.

Linkage Detail

- **Target Exchange Object:** The exchange objects that are linked to the exchange object listed in the Exchange Object Info section.
- **Linkage Type:** Currently, Synchronous is the only option available. When a value change occurs to the exchange object, notification to the linked exchange object is performed in real-time.
- **Exclude Data Creation / Update / Deletion:** The linkage for a source exchange object to a target exchange may be limited by the action done on the source object. The possible actions are Create, Update, and Delete. By checking any of these three ‘exclude’ boxes, the linkage is not triggered for that action.
- **Active Flag:** When checked, the linkage between the exchange object and the target exchange object is active.

Exchange Object - Push Settings

If the exchange object will be part of a push instance, it must be configured in the Push Settings tab before the object can appear in the Push Scenario definition Source Object drop-down menu.



Exchange Object Detail (Display Mode)

Exchange Object Info

Exchange Object: SWM60_COSTCENTER_ACTIVITYTYPE
Exch. Object Desc.: SWM 6.0.0 - Cost Center / Activity Type Maintenance
ExchObject Handler: /SYCLO/CL_CO_LSTAR_EX_HNDLR

Push Settings

Push Relevant: □

Exchange Object Info

- **Exchange Object:** ID of the exchange object. This is a non-editable field.
- **Exchange Object Description:** Brief description of the exchange object. This is a non-editable field.
- **Exchange Object Handler:** Name of class handler from the repository that is responsible for updating the exchange table. This is a non-editable field.

Push Settings

Push Relevant: When checked, the exchange object is listed as a selection in the Source Object drop-down list in the Push Scenario Definition screen.

EFI Assignment

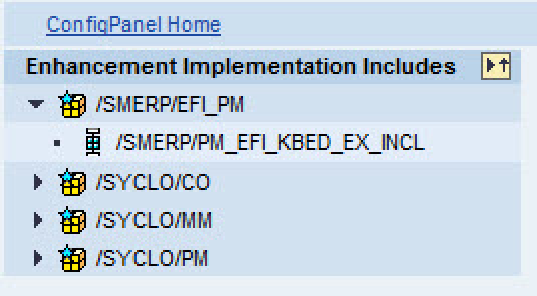
Enhancement Framework Implementation (EFI) source code plug-ins are implemented by the Agentry SAP Framework for each business object where change detection must be implemented. The source code plug-in is provided as an ABAP include file. Each exchange object is assigned to a plug-in to handle the actual change detection process. EFIs are typically available across multiple mobile applications running on the same system.

EFIs collect before and after images of data in an SAP object that has been created, modified, or deleted. The EFI then hands those images to the exchange object, which continues with the data processing. Therefore, the EFIs must be linked to the appropriate exchange objects.

Enhancement Implementation Includes

The Enhancement Implementation Includes list is a tree of the include file list in the package. Click on the arrow to the left of the first item to expand the list.

Enhancement Implementation Includes Tree



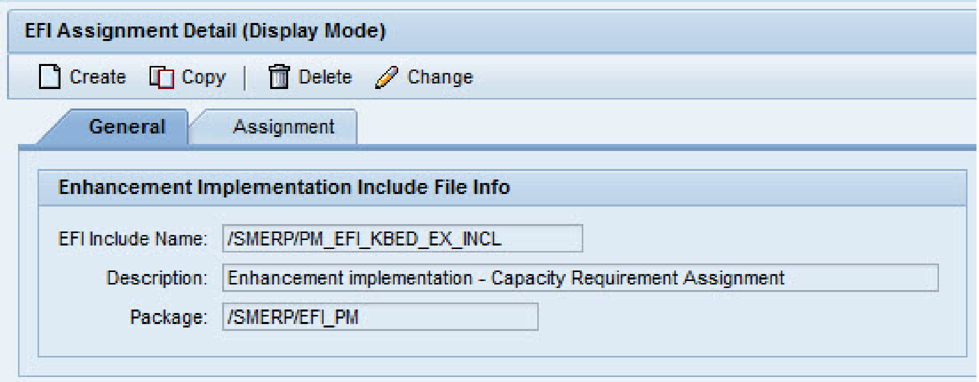
Enhancement Implementation Includes

- /SMERP/EFI_PM
  - /SMERP/PM_EFI_KBED_EX_INCL
- /SYCLO/CO
- /SYCLO/MM
- /SYCLO/PM

EFI Assignment - General Settings

Use the General tab to modify the general settings for a chosen EFI file.

EFI Assignment Detail - General Settings



EFI Assignment Detail (Display Mode)

Enhancement Implementation Include File Info

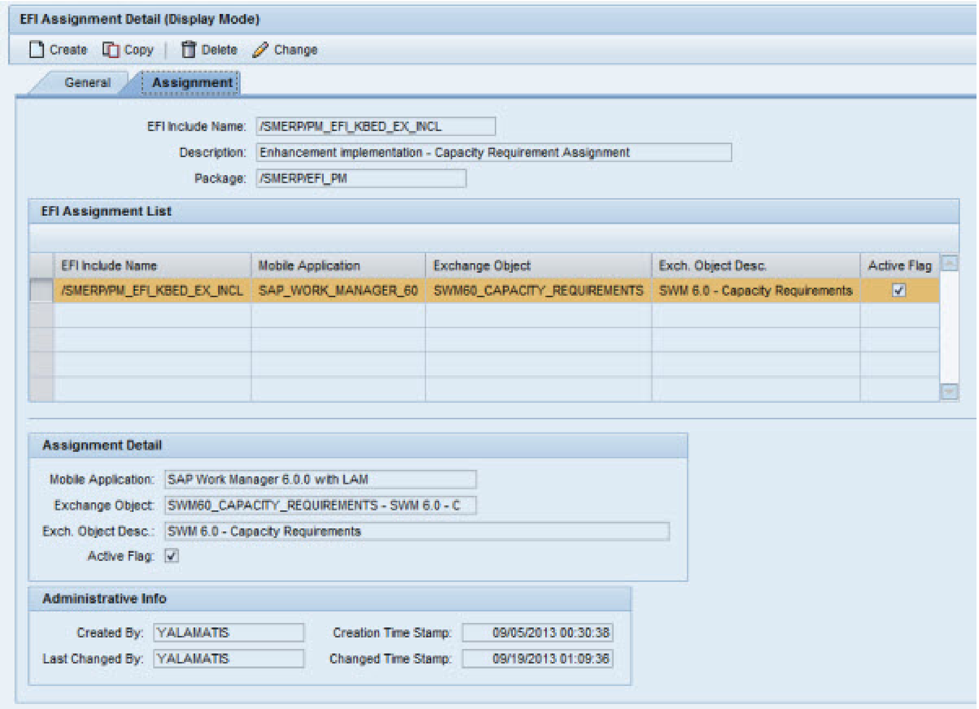
EFI Include Name:	/SMERP/PM_EFI_KBED_EX_INCL
Description:	Enhancement implementation - Capacity Requirement Assignment
Package:	/SMERP/EFI_PM

- **EFI Include Name:** Source code plug-in file name.
- **Description:** Short description of the EFI. This is not an editable field and is automatically filled in when the EFI Include Name is selected.
- **Package:** Package where the EFI is located. This is not an editable field and is automatically filled in when the EFI Include Name is selected.

EFI Assignment - Assignment Settings

Use the Assignment tab to modify the EFI assignments.

EFI Assignment - Assignment Settings



EFI Assignment Detail (Display Mode)

EFI Include Name: /SMERP/PM_EFI_KBED_EX_INCL

Description: Enhancement implementation - Capacity Requirement Assignment

Package: /SMERP/EFI_PM

EFI Assignment List

EFI Include Name	Mobile Application	Exchange Object	Exch. Object Desc.	Active Flag
/SMERP/PM_EFI_KBED_EX_INCL	SAP_WORK_MANAGER_60	SWM60_CAPACITY_REQUIREMENTS	SWM 6.0 - Capacity Requirements	☑

Assignment Detail

Mobile Application: SAP Work Manager 6.0.0 with LAM

Exchange Object: SWM60_CAPACITY_REQUIREMENTS - SWM 6.0 - C

Exch. Object Desc.: SWM 6.0 - Capacity Requirements

Active Flag: ☑

Administrative Info

Created By: YALAMATIS

Creation Time Stamp: 09/05/2013 00:30:38

Last Changed By: YALAMATIS

Changed Time Stamp: 09/19/2013 01:09:36

EFI Info

The fields in this section are taken from information in the General tab and are not editable.

- **EFI Include Name:** Source code plug-in include file name
- **Description:** Description of the EFI include file name
- **Package:** SAP package to which the include file belongs

EFI Assignment List

EFI Assignment table: Table that displays which plug-ins are assigned to a specific include file. All column information is replicated in the Assignment Detail section directly below the table.

To highlight an individual row, click on the grey square to the left of the EFI Include Name column in that row.

Assignment Detail

Information in this section will change depending on which row is highlighted in the EFI Assignment List section table.

- **Mobile Application:** The specific mobile application and its release number. This field is non-editable.
- **Exchange Object:** Exchange object to which the EFI include file is assigned.

- **Exch. Object Desc:** A brief, easy to understand description of the exchange object, limited to 60 characters.
- **Active Flag:** When checked, the exchange object is in an active state. If unchecked, the EFI is not linked to the assigned mobile data object.

Administrative Info

- **Created By:** SAP user ID of the person who created the EFI assignments
- **Creation Time Stamp:** Date and time of the creation of the EFI assignments
- **Last Changed By:** SAP user ID of the person who last changed the EFI assignments
- **Changed Time Stamp:** Date and time of the change to the EFI assignments

Mobile Data Object Configuration

A mobile data object represents a mobile semantic view of data and activity combination for an SAP business object. Mobile data objects are data repositories in the namespace that can get, create, update, and delete information in SAP. They encapsulate the business logic of mobile applications by defining transactions, data structures, and business rules.

There are three types of mobile data objects:

- **DT - Data Table:** A simple representation of SAP business objects KEY and VALUE.
- **CT - Complex Table:** A two-dimensional representation of a business object with a single table of multiple columns.
- **DO - Standard Data Object:** A multi-dimensional representation of a business object with multiple tables representing different subsets of the business object.

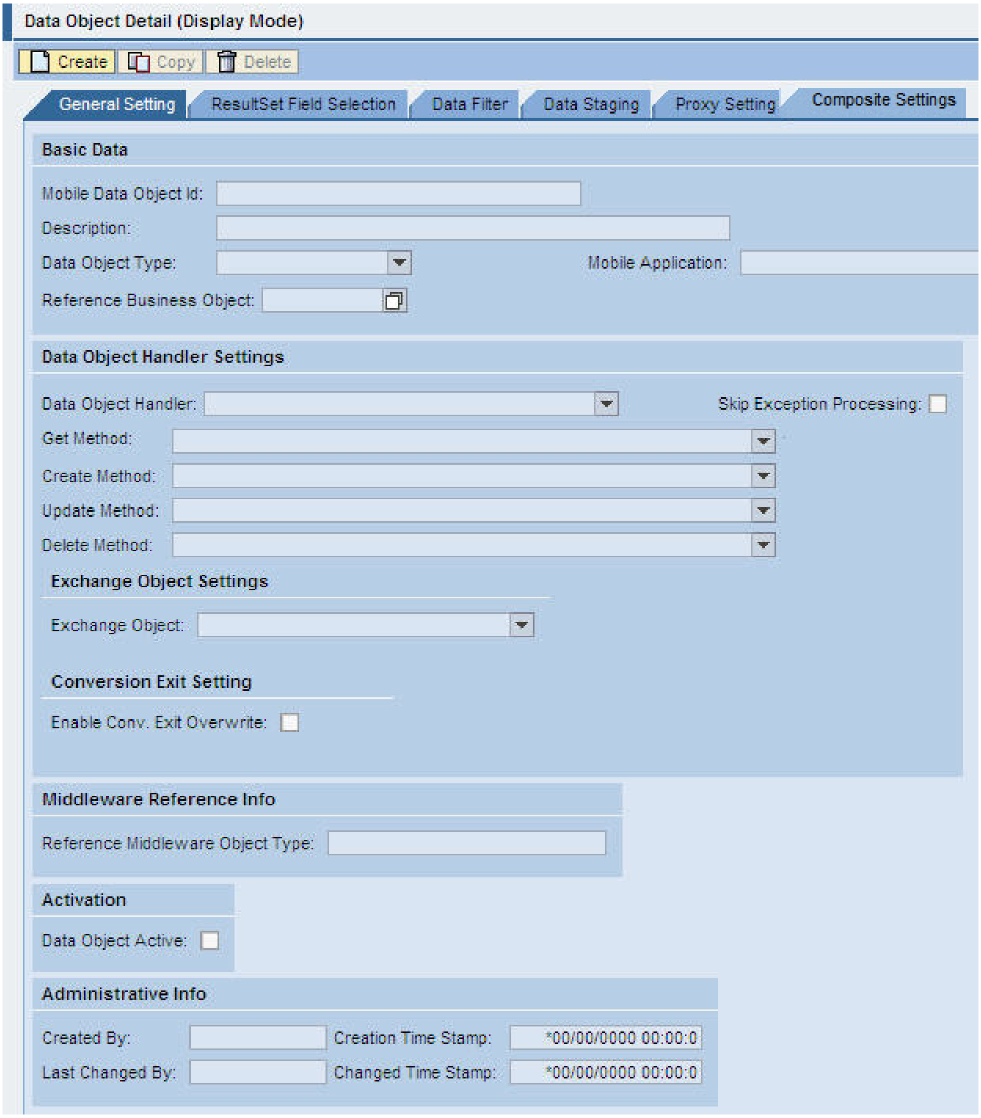
Warning! Mobile data objects should be created, copied, or changed only by the mobile application developer or system administrator.

The configuration portal is used to easily modify mobile data object properties such as object type, class handlers, data filters, and other settings. For example, instead of modifying BAPIs to change what information is retrieved from SAP and pushed out to mobile devices, administrators can use the Configuration portal to modify mobile data objects and set up data filter rules.

Mobile Data Object - General Settings

Use the General Setting tab to modify the general settings for a chosen mobile data object.

Mobile Data Object - General Setting Tab



Basic Data

The Basic Data section provides general information about the specific mobile data object. This information is used in multiple panels in the Configuration portal.

- **Mobile Data Object ID:** The name of the mobile data object, limited to 40 characters. This is a required field.
- **Description:** A brief, easy to understand description of the mobile data object, limited to 60 characters. This is a required field.

- **Data Object Type:** A drop-down list of the three mobile data object types:
  - Data Table - A simple representation of the SAP business objects KEY and VALUE
  - Complex Table - A two-dimensional representation of the business object with a single table of multiple columns
  - Standard Data Object - A multi-dimensional representation of a complete business object. It can have multiple tables representing different subsets of the business objects.
- **Mobile Application:** The name of the mobile application. Choose the mobile application from the drop-down list.
- **Reference Business Object:** The SAP business object for which the mobile data object is being created

Data Object Handler Settings

The Data Object Handler Settings section is used to configure the methods of the mobile data object.

- **Data Object Handler:** Name of the ABAP OO class handler from SAP's class repository. The ABAP OO class handler is developed by the application developer with predefined business logic and scope to perform fetch, create, delete, or update activities for an SAP business object.
- **Get Method:** Method defined in the class handler that fetches data for the underlying SAP business object. This is an optional field.
- **Create Method:** Method defined in the class handler that creates data for the underlying SAP business object. This is an optional field.
- **Update Method:** Method defined in the class handler that updates data for the underlying SAP business object. This is an optional field.
- **Delete Method:** Method defined in the class handler that deletes data for the underlying SAP business object. This is an optional field.
- **Skip Exception Processing:** When checked, if an exception occurs during mobile data object handling, the exception processing step is not invoked.

Exchange Object Settings

The Exchange Object Settings section allows the mobile data object to be associated with an exchange object. The exchange object is configured through the Exchange Object Configuration panel in the Configuration portal.

- **Exchange Object:** The name of the exchange object as defined in the SAP mobile exchange persistent layer. Specify this field by choosing an exchange object from the drop-down menu if the selected class handler should use the mobile exchange persistent layer to determine data exchanges to the mobile application.
- **Enable Conv. Exit Overwrite:** When checked, you will be able to define specific internal and external conversion settings in the Framework Technical Settings panel.

Middleware Reference Info

If the middleware is specified, the mobile data object will perform the standard exchange process as well as perform a lookup in the client object register table to determine what information the client contains. If data has been removed from the client that still exists in the SAP table, the data is re-added to the client during the transmit.

- **Reference Middleware Object Type:** Middleware objects are set through the Administration portal. This is an optional field.

Activation

Use this checkbox to enable or disable a mobile data object in the application without deleting the mobile data object.

Data Object Active: When checked, the mobile data object is in an active state. If unchecked, the mobile data object performs no actions.

Administrative Info

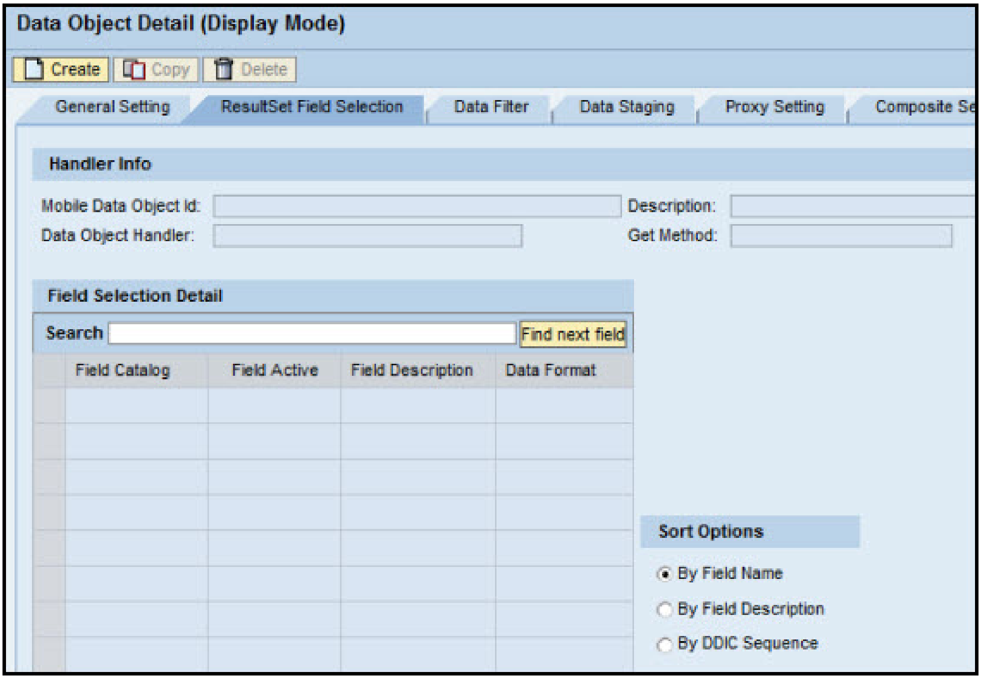
This section is used to easily determine which SAP user created or modified a specific mobile data object in the system.

- **Created By:** SAP user ID of the person who created the mobile data object
- **Creation Time Stamp:** Date and time of the creation of the mobile data object
- **Last Changed By:** SAP user ID of the person who last changed the mobile data object
- **Changed Time Stamp:** Date and time of the change to the mobile data object

Mobile Data Object - ResultSet Field Selection

When a field selector function is enabled for a class handler, the option to select fields for the GET method to populate is available. The class handler is designed to be mobile application-neutral. It can typically supply more data than the mobile application needs. Therefore, in order to preserve system performance, you can customize field usage settings to only retrieve required data for the mobile application. This ability prevents the need to develop a new class handler for each mobile application.

Mobile Data Object - ResultSet Field Selection Tab



● By Field Name
○ By Field Description
○ By DDIC Sequence

Handler Info

- **Mobile Data Object ID:** Name of the mobile data object
- **Description:** Description of the mobile data object, limited to 60 characters. This is a required field.
- **Data Object Handler:** Name of the ABAP OO class handler from SAP's class repository. The ABAP OO class handler is developed by the application developer with predefined business logic and scope to perform fetch, create, delete, or update activities for an SAP business object.
- **Get Method:** Name of the GET method set in the General Setting tab.

Field Selection Detail

- **Field Catalog column:** Lists all the fields that can be returned by the class handler method, grouped in the order of the name of the class handler method, SAP table name, and field name. To display all information in this column, click on the arrows to the left of a name in order to expand the row.
- **Field Active column:** When checked, the data for the selected field is returned by the class handler method.
- **Field Description column:** Description of the specific field in the SAP table.
- **Data Format column:** How data is presented

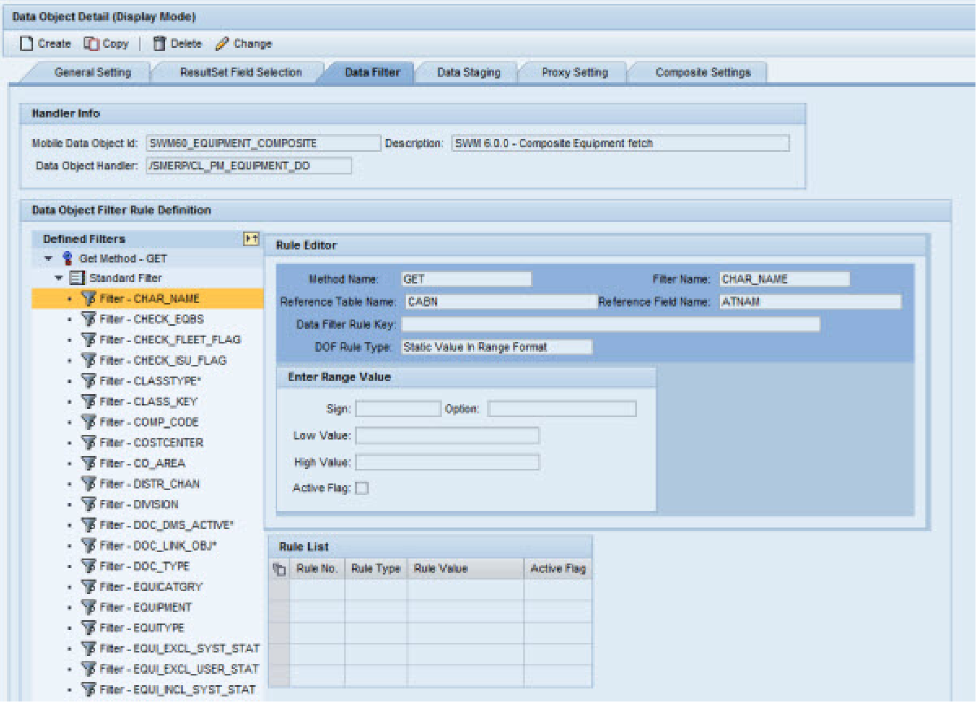
- **Sort Options:** When there is a large amount of information presented, use the sort options to find the information required easily. When a different radio button is selected, the rows are collapsed and must be expanded again to display the new field sorting.

Mobile Data Object - Data Filter

When a data filter function is enabled for a class handler, the option exists to define various types of filter rules to control what data can be viewed by the mobile application based on a customer's business process. In an SAP environment, each user is assigned a role-based profile with authorization restrictions for what data is viewed and which activities performed.

For example, a user who works for a specific plant should not be able to view data for another plant. Data filter rules allow you to restrict data access for mobile applications. Data filters can be user-dependent or applied to the entire mobile application.

Mobile Data Object - Data Filter Tab



Handler Info

Mobile Data Object Id: SWM60_EQUIPMENT_COMPOSITE
Description: SWM 6.0.0 - Composite Equipment fetch
Data Object Handler: /SMERP/CL_PM_EQUIPMENT_DO

Data Object Filter Rule Definition

Defined Filters

- Get Method - GET
  - Standard Filter
    - Filter - CHAR_NAME
    - Filter - CHECK_EQBS
    - Filter - CHECK_FLEET_FLAG
    - Filter - CHECK_ISU_FLAG
    - Filter - CLASSTYPE*
    - Filter - CLASS_KEY
    - Filter - COMP_CODE
    - Filter - COSTCENTER
    - Filter - CO_AREA
    - Filter - DISTR_CHAN
    - Filter - DIVISION
    - Filter - DOC_DMS_ACTIVE*
    - Filter - DOC_LINK_OBJ*
    - Filter - DOC_TYPE
    - Filter - EQUICATGRY
    - Filter - EQUIPMENT
    - Filter - EQUITYPE
    - Filter - EQUI_EXCL_SYST_STAT
    - Filter - EQUI_EXCL_USER_STAT
    - Filter - EQUI_INCL_SYST_STAT

Rule Editor

Method Name: GET
Filter Name: CHAR_NAME
Reference Table Name: CABN
Reference Field Name: ATNAM
Data Filter Rule Key:
DOF Rule Type: Static Value In Range Format

Enter Range Value

Sign:
Option:
Low Value:
High Value:
Active Flag: □

Rule List

Rule No.	Rule Type	Rule Value	Active Flag

Handler Info

- **Mobile Data Object ID:** Name of the mobile data object
- **Description:** Description of the mobile data object, limited to 60 characters. This is a required field.
- **Data Object Handler:** Name of the ABAP OO class handler from SAP's class repository. The ABAP OO class handler is developed by the application developer with predefined

business logic and scope to perform fetch, create, delete, or update activities for an SAP business object.

Defined Filters

The Defined Filters tree lists all data filters supported by the class handlers defined in the Data Object Handler Settings field in the General Setting tab. To expand the tree, click on the arrows to the left of the class handler methods to display the filters associated with the methods.

Rule Editor

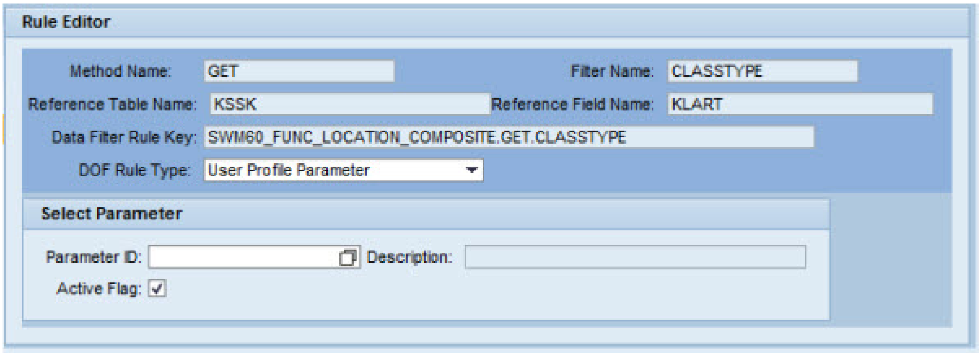
- **Method Name:** Name of the class handler method where the data filter is defined. The data filter function is only supported for the GET method.
- **Filter Name:** Name of the filter as defined in the class handler method. This information is defined by the class handler developer and is not editable.
- **Reference Table Name:** Technical name of the SAP database table where this filter is applied. This information is defined by the class handler developer and is not editable.
- **Reference Field Name:** Technical name of the SAP database table field where this filter is applied. This information is defined by the class handler developer and is not editable.
- **Data Filter Rule Key:** Internal technical key used by the framework at runtime
- **DOF Rule Type:** Type of rule

There are three different types of rules:

- User Profile Parameter
- Static Value in Range Format
- Syclo Filter Class Handler
- Runtime Session Data

The settings are dependent on the specific type of rule set for the filter.

Data Filter - User Profile Parameter Rule



Rule Editor

Method Name: GET

Filter Name: CLASSTYPE

Reference Table Name: KSSK

Reference Field Name: KLART

Data Filter Rule Key: SWM60_FUNC_LOCATION_COMPOSITE.GET.CLASSTYPE

DOF Rule Type: User Profile Parameter

Select Parameter

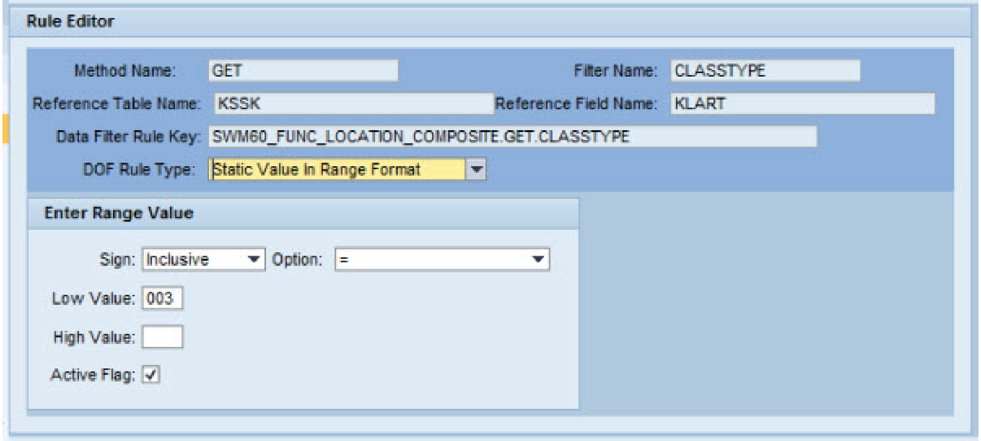
Parameter ID:

Description:

Active Flag: ☑

- **Parameter ID:** Memory parameter ID as defined in SAP and specified in the user profile. Click on the icon to the right of the field box to perform a search on all available parameter IDs.
- **Description:** Description of the memory parameter ID. This is not an editable field and is automatically filled in when the parameter ID is selected.
- **Active Flag:** When checked, the rule is active.

Data Filter - Static Value in Range Format



Rule Editor

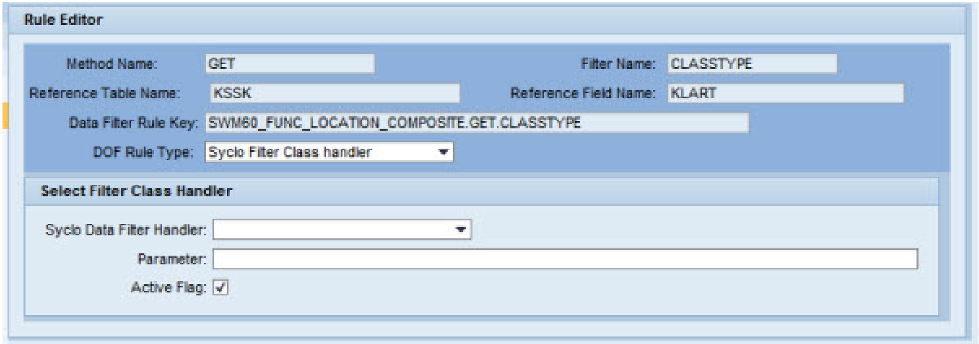
Method Name: GET
Filter Name: CLASSTYPE
Reference Table Name: KSSK
Reference Field Name: KLART
Data Filter Rule Key: SWM60_FUNC_LOCATION_COMPOSITE.GET.CLASSTYPE
DOF Rule Type: Static Value In Range Format

Enter Range Value

Sign: Inclusive
Option: =
Low Value: 003
High Value:
Active Flag: ☑

- **Sign:** Value for the SAP Range table column SIGN
- **Option:** Value for the SAP Range table column OPTION
- **Low Value:** Value for the SAP Range table column LOW
- **High Value:** Value for the SAP Range table column HIGH
- **Active Flag:** When checked, the rule is active

Data Filter - Syclo Filter Class Handler



Rule Editor

Method Name: GET
Filter Name: CLASSTYPE
Reference Table Name: KSSK
Reference Field Name: KLART
Data Filter Rule Key: SWM60_FUNC_LOCATION_COMPOSITE.GET.CLASSTYPE
DOF Rule Type: Syclo Filter Class handler

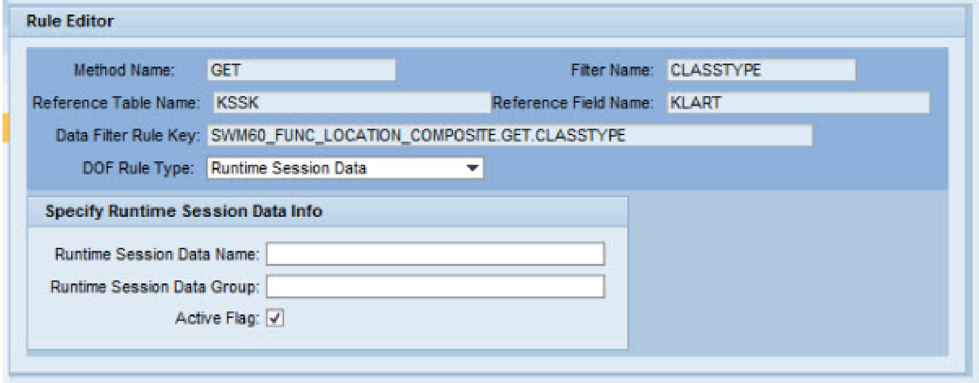
Select Filter Class Handler

Syclo Data Filter Handler:
Parameter:
Active Flag: ☑

- **Syclo Data Filter Handler:** Name of the handler class as defined in the system.

- **Parameter:** Additional processing information that is passed to the class handler. The parameters are entered as free-text, and the syntax of the parameter stream is defined by the developer.
- **Active Flag:** When checked, the rule is active.

Data Filter - Runtime Session Data



Rule Editor

Method Name: GET

Filter Name: CLASSTYPE

Reference Table Name: KSSK

Reference Field Name: KLART

Data Filter Rule Key: SWM60_FUNC_LOCATION_COMPOSITE.GET.CLASSTYPE

DOF Rule Type: Runtime Session Data

Specify Runtime Session Data Info

Runtime Session Data Name:

Runtime Session Data Group:

Active Flag: ☑

- **Runtime Session Data Name:**
- **Runtime Session Data Group:**
- **Active Flag:** When checked, the rule is active.

Rule List

The Rule List table displays a list of rules that have been defined.

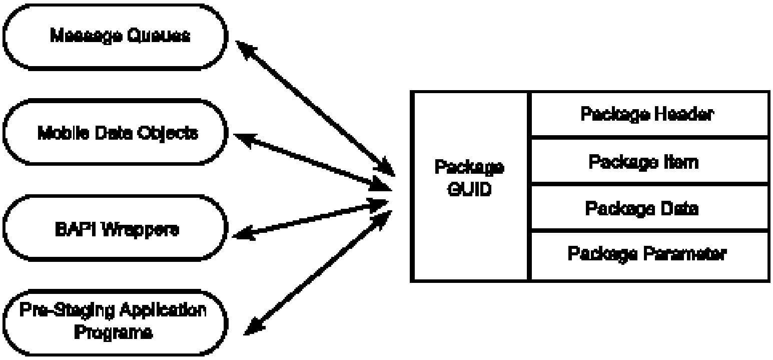
- **Rule No.:** Number of the rule that is defined, in chronological order
- **Rule Type:** Rule type, automatically assigned by the rule type selected in the DOF Rule Type field
- **Rule Value:** Internal rule value saved by SAP

Mobile Data Object - Data Staging

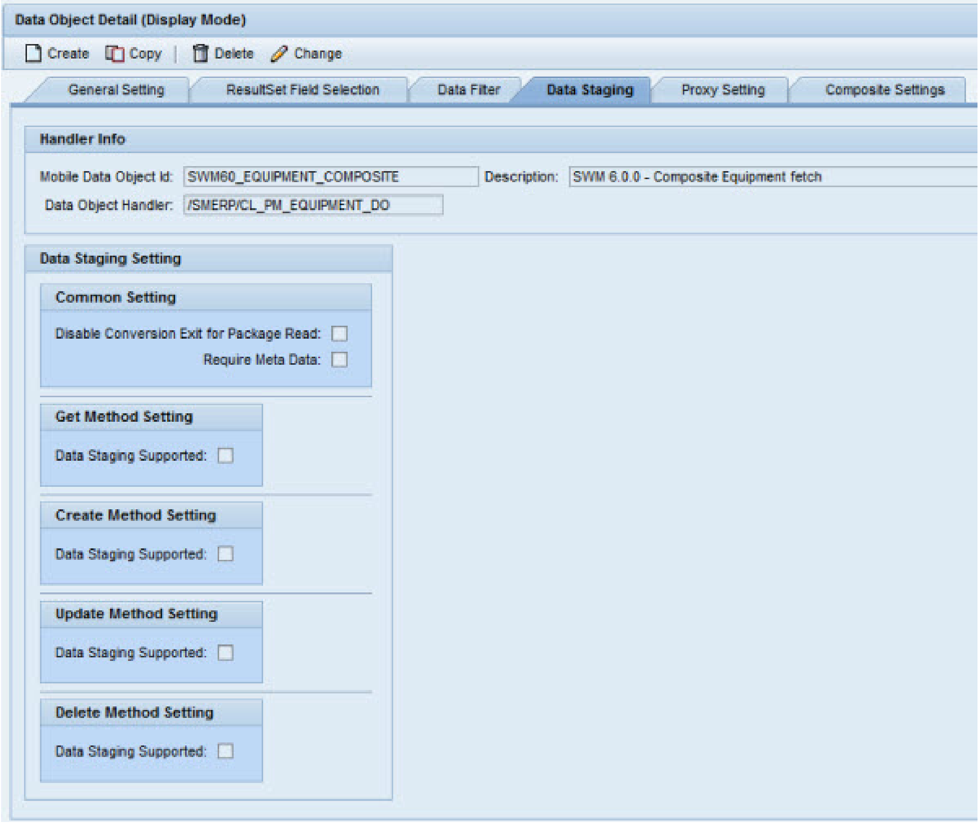
If an application processes a large amount of objects, data staging of the objects can assist with processing times. If an object is configured for data staging, the data within the object is stored as a package and is split into packets. The data can contain metadata and tagging for easy lifecycle management and data lookup. Standard APIs are provided for package management.

The following figure illustrates a general-purpose data staging model.

Figure 12: Mobile Data Object - Data Staging Model



Mobile Data Object - Data Staging Tab



Data Object Detail (Display Mode)

Handler Info

Mobile Data Object Id: SWM60_EQUIPMENT_COMPOSITE

Description: SWM 6.0.0 - Composite Equipment fetch

Data Object Handler: /SMERP/CL_PM_EQUIPMENT_DO

Data Staging Setting

Common Setting

Disable Conversion Exit for Package Read: □

Require Meta Data: □

Get Method Setting

Data Staging Supported: □

Create Method Setting

Data Staging Supported: □

Update Method Setting

Data Staging Supported: □

Delete Method Setting

Data Staging Supported: □

Handler Info

- **Mobile Data Object ID:** Name of the mobile data object

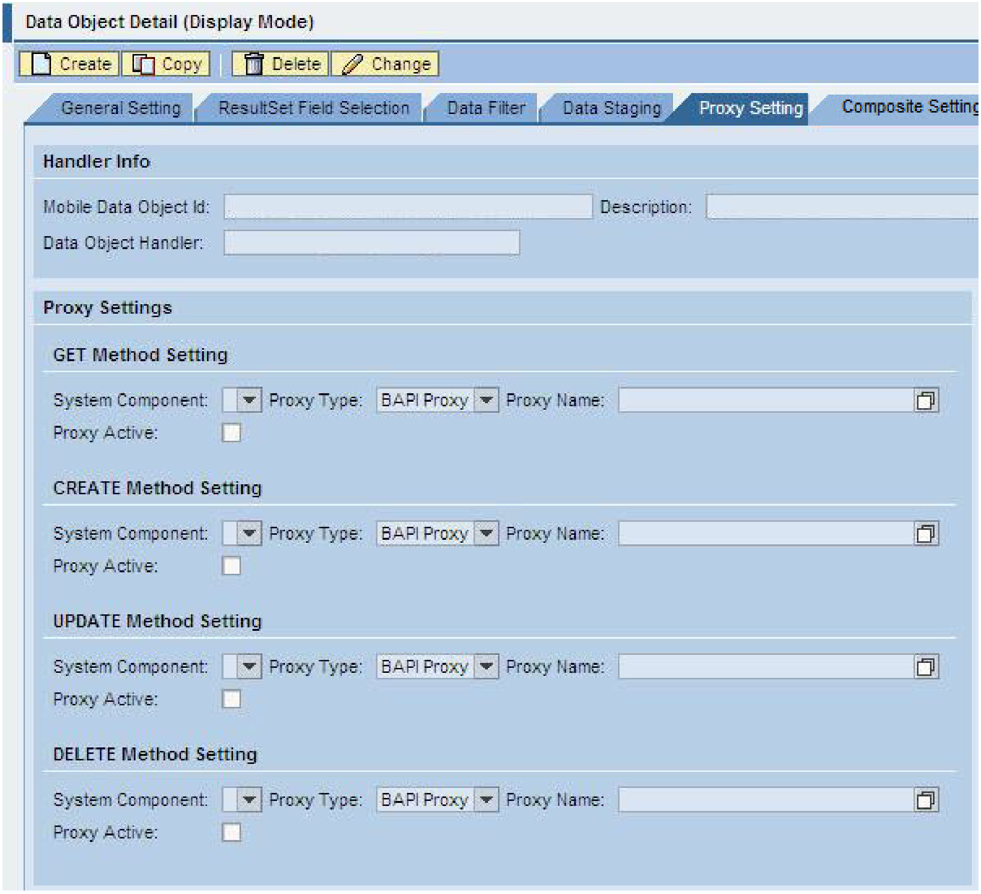
- **Description:** Description of the mobile data object, limited to 60 characters. This is a required field.
- **Data Object Handler:** Name of the ABAP OO class handler from SAP's class repository. The ABAP OO class handler is developed by the application developer with predefined business logic and scope to perform fetch, create, delete, or update activities for an SAP business object.

Data Staging Setting

- **Disable Conversion Exit for Package Read:**
- **Require Metadata:**
- **Get / Create / Update / Delete Method Setting:** When a checkbox is marked for a specific method, data staging is active for that method.

Mobile Data Object - Proxy Setting

Figure 13: Mobile Data Object - Proxy Setting Tab



Handler Info

Mobile Data Object Id:
Description:
Data Object Handler:

Proxy Settings

GET Method Setting

System Component: Proxy Type: BAPI Proxy Proxy Name:
Proxy Active: □

CREATE Method Setting

System Component: Proxy Type: BAPI Proxy Proxy Name:
Proxy Active: □

UPDATE Method Setting

System Component: Proxy Type: BAPI Proxy Proxy Name:
Proxy Active: □

DELETE Method Setting

System Component: Proxy Type: BAPI Proxy Proxy Name:
Proxy Active: □

Handler Info

- **Mobile Data Object ID:** Name of the mobile data object
- **Description:** Description of the mobile data object, limited to 60 characters. This is a required field.
- **Data Object Handler:** Name of the ABAP OO class handler from SAP's class repository. The ABAP OO class handler is developed by the application developer with predefined business logic and scope to perform fetch, create, delete, or update activities for an SAP business object.

Proxy Settings

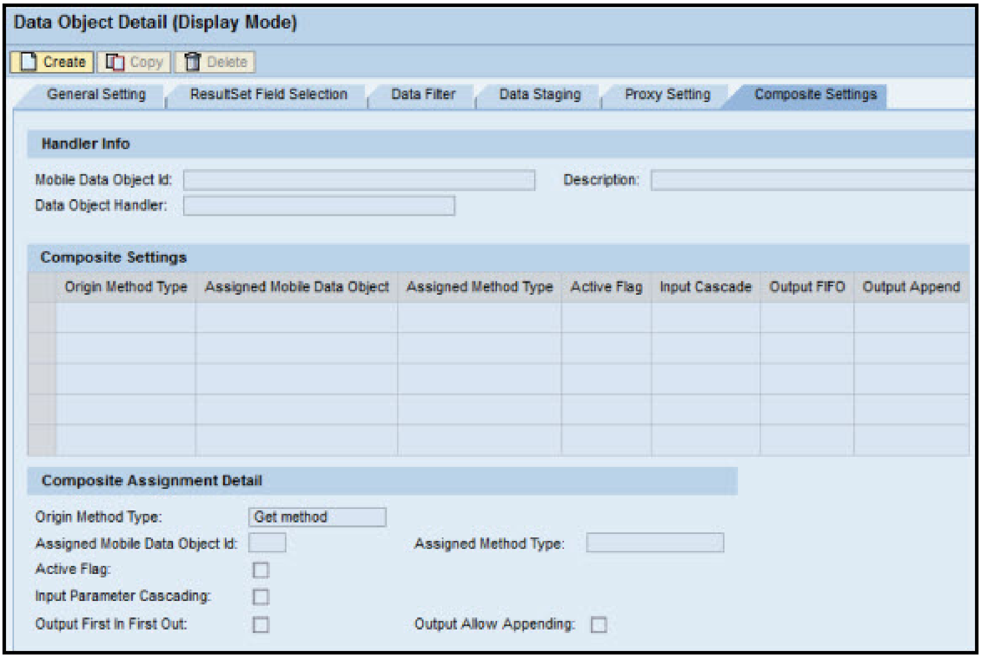
The following four settings are available for each of the methods.

- **System Component:** Identifies the remote system

- **Proxy Type:** BAPI Proxy is the only supported proxy type
- **Proxy Name:** BAPI name of the remote system
- **Proxy Active:** When checked, proxy settings are active for the specific method.

Mobile Data Object - Composite Settings Tab

Use to Composite Settings tab to define the data object type Composite Settings. This allows you to combine and use multiple MDOs.



Origin Method Type	Assigned Mobile Data Object	Assigned Method Type	Active Flag	Input Cascade	Output FIFO	Output Append

Origin Method Type: Get method

Handler Info

- **Mobile Data Object ID:** the ID of the MDO to be used
- **Description:** description of the MDO being used
- **Data Object Handler:** name of the handler for the data object

Composite Settings

- **Origin Method Type:** by default this is GET method
- **Assigned Mobile Data Object:** the MDO ID
- **Assigned Method Type:** the defined method type
- **Active Flag:** indicates if this setting is active or not
- **Input Cascade:** indicates whether cascading has been allowed
- **Output FIFO:** output settings indicate first in, first out

- **Output Append:** indicates if results will be aggregated

Composite Assignment Detail

- **Origin Method Type:** by default, this is GET method
- **Assigned Mobile Data Object ID:** enter the MDO ID
- **Assigned Method Type:** enter the method type
- **Active Flag:** check this to indicate active status
- **Input Parameter Cascading:** when the BAPI is called, this allows a chain of MDOs to execute
- **Output First In First Out:** check this todefine the output as first in, first out
- **Output Allow Appending:** check this to aggregate results

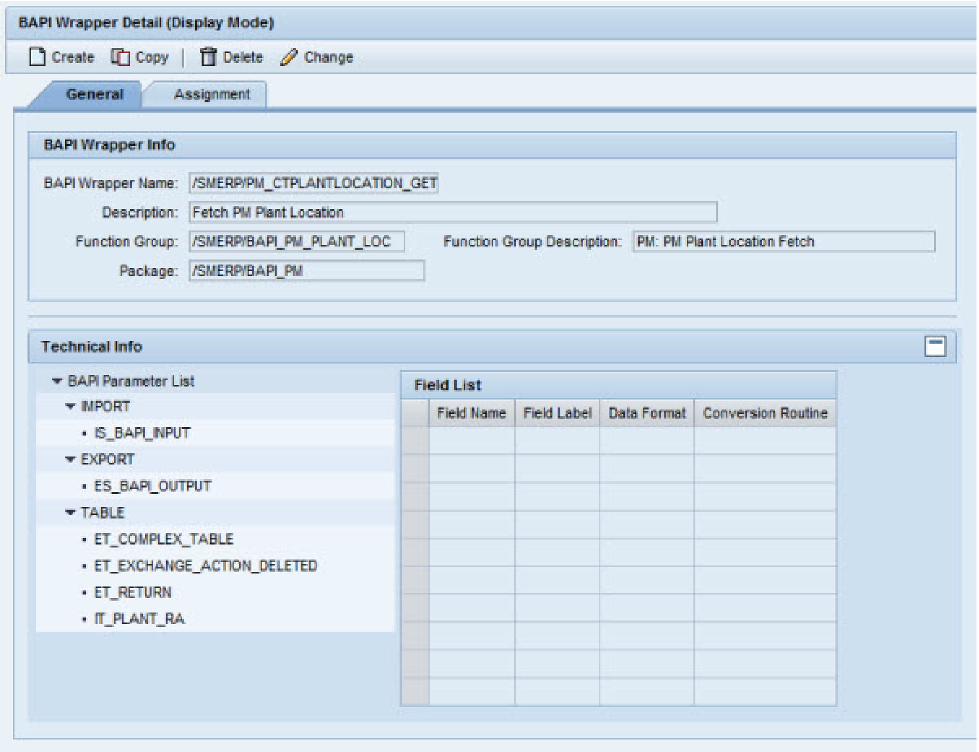
BAPI Wrapper Configuration

A BAPI wrapper is created by the application developer to expose SAP data and business logic to the mobile application. By design, the BAPI wrapper does not contain any business logic. Each BAPI wrapper must be assigned to a specific method type (GET, CREATE, UPDATE, or DELETE) of a mobile data object to perform the required business logic. By decoupling the business logic from the BAPI wrapper, it is possible to switch mobile data objects without affecting the underlying mobile application definition.

BAPI Wrapper - General Settings

Use the General tab to modify the general settings for a chosen BAPI wrapper.

BAPI Wrapper - General Settings



BAPI Wrapper Info

BAPI Wrapper Name: /SMERP/PM_CTPLANTLOCATION_GET

Description: Fetch PM Plant Location

Function Group: /SMERP/BAPI_PM_PLANT_LOC

Function Group Description: PM: PM Plant Location Fetch

Package: /SMERP/BAPI_PM

Technical Info

- BAPI Parameter List
  - IMPORT
    - IS_BAPI_INPUT
  - EXPORT
    - ES_BAPI_OUTPUT
  - TABLE
    - ET_COMPLEX_TABLE
    - ET_EXCHANGE_ACTION_DELETED
    - ET_RETURN
    - IT_PLANT_RA

Field List

	Field Name	Field Label	Data Format	Conversion Routine

BAPI Wrapper Info

- **BAPI Wrapper Name:** Technical name of the Remote Function Call (RFC) function module defined in the system
- **Description:** Description of the BAPI wrapper, limited to 60 characters. This is a required field.
- **Function Group:** Function group to which the BAPI belongs
- **Function Group Description:** Description of the function group, limited to 60 characters.
- **Package:** SAP group to which the function group and the BAPI wrapper both belong

BAPI Wrapper List

The BAPI wrapper list provides an expandable tree of the available BAPI wrapper function groups and the BAPI wrapper name(s) associated with each function group. Use the arrows to the left of the function name to display all BAPI wrapper names under the function group.

Technical Info

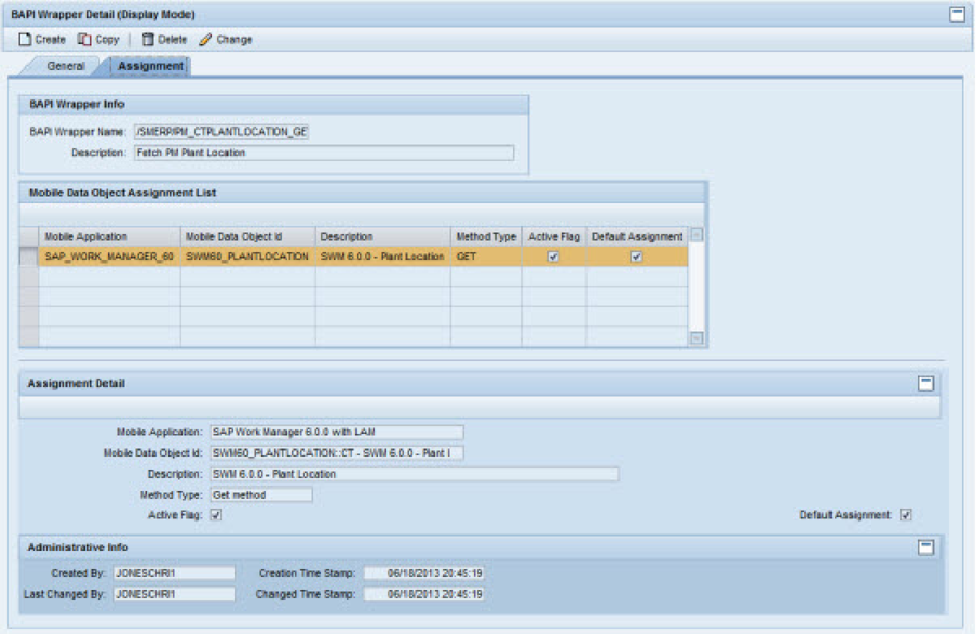
- **Field Name column:** Field name of the structure
- **Field Label column:** Text describing the structure

- **Data Format column:** Standard SAP data type
- **Conversion Routine column:** SAP-defined conversion routine

BAPI Wrapper - Assignment Settings

Use the Assignment tab to hook up BAPI wrapper assignments to mobile data objects. In this tab, you can change a BAPI wrapper's assignment to specific mobile data objects, assign a new mobile data object to the BAPI wrapper, or delete a mobile data object assignment from the BAPI wrapper.

BAPI Wrapper - Assignment Settings



BAPI Wrapper Info

BAPI Wrapper Name: /SMERP/PM_CTPLANTLOCATION_GE

Description: Fetch PM Plant Location

Mobile Data Object Assignment List

Mobile Application	Mobile Data Object Id	Description	Method Type	Active Flag	Default Assignment
SAP_WORK_MANAGER_60	SWM60_PLANTLOCATION	SWM 6.0.0 - Plant Location	GET	☑	☑

Assignment Detail

Mobile Application: SAP Work Manager 6.0.0 with LAM

Mobile Data Object Id: SWM60_PLANTLOCATION::CT - SWM 6.0.0 - Plant l

Description: SWM 6.0.0 - Plant Location

Method Type: Get method

Active Flag: ☑

Default Assignment: ☑

Administrative Info

Created By: JONESCHRI1

Creation Time Stamp: 06/18/2013 20:45:19

Last Changed By: JONESCHRI1

Changed Time Stamp: 06/18/2013 20:45:19

BAPI Wrapper Info

- **BAPI Wrapper Name:** Technical name of the Remote Function Call (RFC) function module defined in the system
- **Description:** Description of the BAPI wrapper, limited to 60 characters. This is a required field.

Mobile Data Object Assignment List

Mobile Data Object Assignment table: Table that displays which mobile data objects are assigned to each BAPI wrapper. All column information is replicated in the Assignment Detail section directly below the table.

To highlight an individual row, click on the grey square to the left of the Mobile Application column in that row.

Assignment Detail

Information in this section will change depending on which row is highlighted in the Mobile Data Assignment List section table.

- **Mobile Application:** The specific mobile application and its release number. This field is non-editable.
- **Mobile Data Object ID:** The name of the mobile data object in a drop-down field.
- **Description:** A brief, easy to understand description of the mobile data object, limited to 60 characters. This field is non-editable.
- **Method Type:** Mobile data object method (GET, CREATE, UPDATE, or DELETE) that is assigned to the BAPI wrapper.
- **Active Flag:** When checked, the mobile data object is in an active state. If unchecked, the mobile data object performs no actions.
- **Default Assignment:** When checked, the specific mobile data object assigned to that BAPI wrapper is primary. If no mobile data object ID is specified in the standard BAPI wrapper input parameter ‘IS_BAPI_INPUT-DO_ID’, then the primary mobile data object is used during runtime.

  It is possible to assign multiple MDOs to the same BAPI in a single mobile application. To override the default MDO assignment at runtime and to specify the desired MDO, the DO_ID field must be defined in IS_BAPI_INPUT.

Administrative Info

- **Created By:** SAP user ID of the person who created the BAPI wrapper assignments
- **Creation Time Stamp:** Date and time of the creation of the BAPI wrapper assignments
- **Last Changed By:** SAP user ID of the person who last changed the BAPI wrapper assignments
- **Changed Time Stamp:** Date and time of the change to the BAPI wrapper assignments

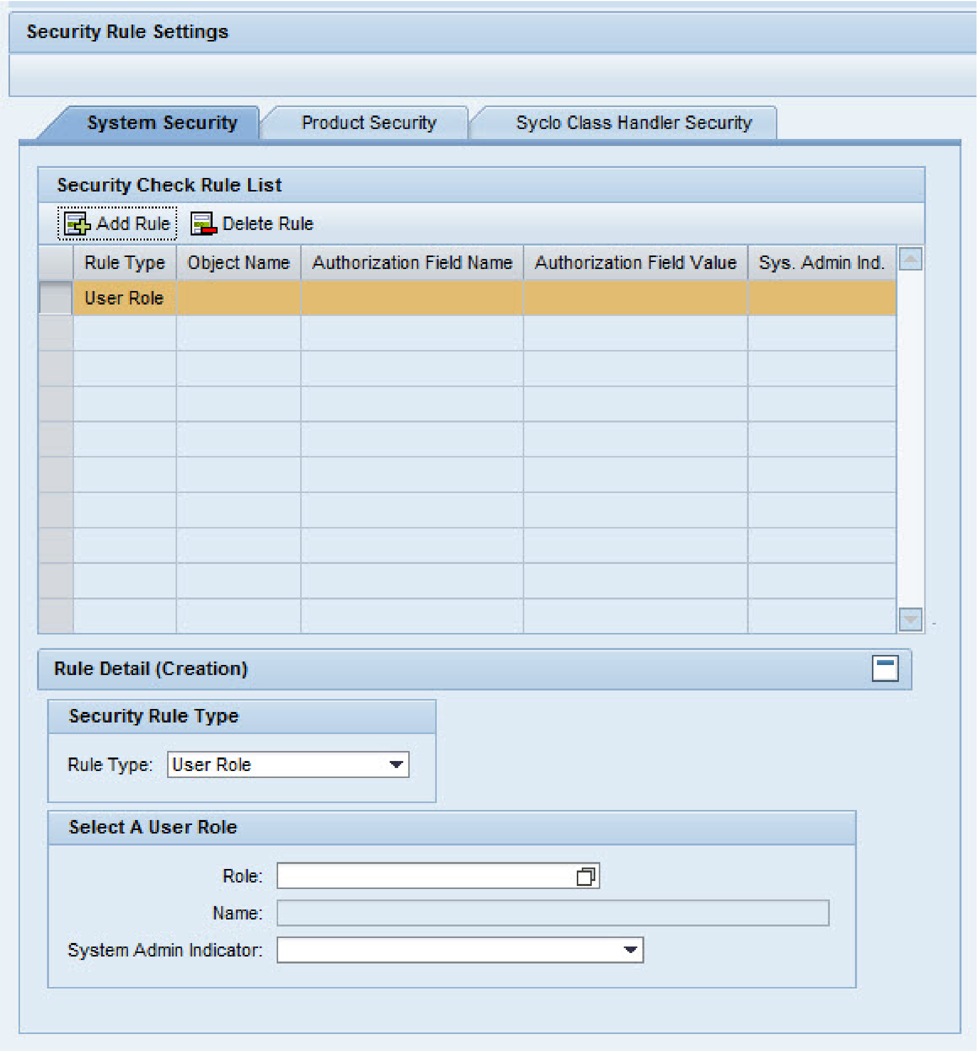
Security Settings

Security settings are used to provide additional rules and roles on top of the standard SAP-provided rules and roles.

Security Settings - System Security

Use this tab to configure system security settings that are mobile application-independent. System security settings apply to all applications running on the framework.

Security Settings - System Security



Rule Type	Object Name	Authorization Field Name	Authorization Field Value	Sys. Admin Ind.
User Role				

Rule Type: User Role

Role:

Name:

System Admin Indicator:

Security Check Rule List

- **Add Rule button:** Press the **Add Rule** button to add a new system security rule. Fill in the fields in the Rule Detail section to automatically fill in the fields in this table.
- **Delete Rule button:** Press the **Delete Rule** button to delete an system security rule. To delete a system security rule, press the rectangle to the left of the Rule Type column in the row you wish to delete and press the **Delete Rule** button.
- **Rule Type:** Rule Type from the Rule Detail section. This is a non-editable field.
- **Object Name:** Taken from the Profile field in the Rule Detail section if rule type Authorization Object is chosen. This is a non-editable field.

- **Authorization Field Name:** Standard SAP authorization object name. This is a non-editable field.
- **Authorization Field Value:** Free-text field. Text entered depends on developer implementation in SAP. This is a non-editable field.
- **System Admin Indicator:** When Rule Type: User Role is selected, taken from the System Admin Indicator field in the Rule Detail section. This is a non-editable field.

Rule Detail - Rule Type: User Role

- **Rule Type: User Role:** In addition to the standard SAP user profile rules, the user role can add restrictions on what a user can or cannot see in menus or other mobile application screens.
- **Role:** User role in SAP. To search for a user role, click on the white box icon to the right of the Role field to display the Role Selection search window.
- **Name:** Brief description of the role. This is a non-editable field.
- **System Admin Indicator:** Drop-down menu with four choices:
  - **System administrator:** User role can view system activity and make changes to system administration setup in the Administration portal.
  - **System administration - View only:** User role can view the system activity in the Administration portal, but cannot make changes to the setup.
  - **System configurator:** User role can view system configuration and make changes to the setup in the Configuration portal.
  - **System configuration - View only:** User role can view the system configuration in the Configuration portal, but cannot make changes to the setup.

Once the System Admin Indicator roles have been configured, the configurations and roles are available in both the Configuration and the Administration & Monitoring portals.

Rule Detail - Rule Type: Authorization Profile

- **Rule Type: Authorization Profile:** A collection of objects, or roles, such as Technician or Supervisor.
- **Profile:** Authorization profile in SAP. To search for an authorization profile, click on the white box icon to the right of the Profile field. The Profile Selection window displays.
- **Text:** Brief description of the authorization profile. This is a non-editable field.

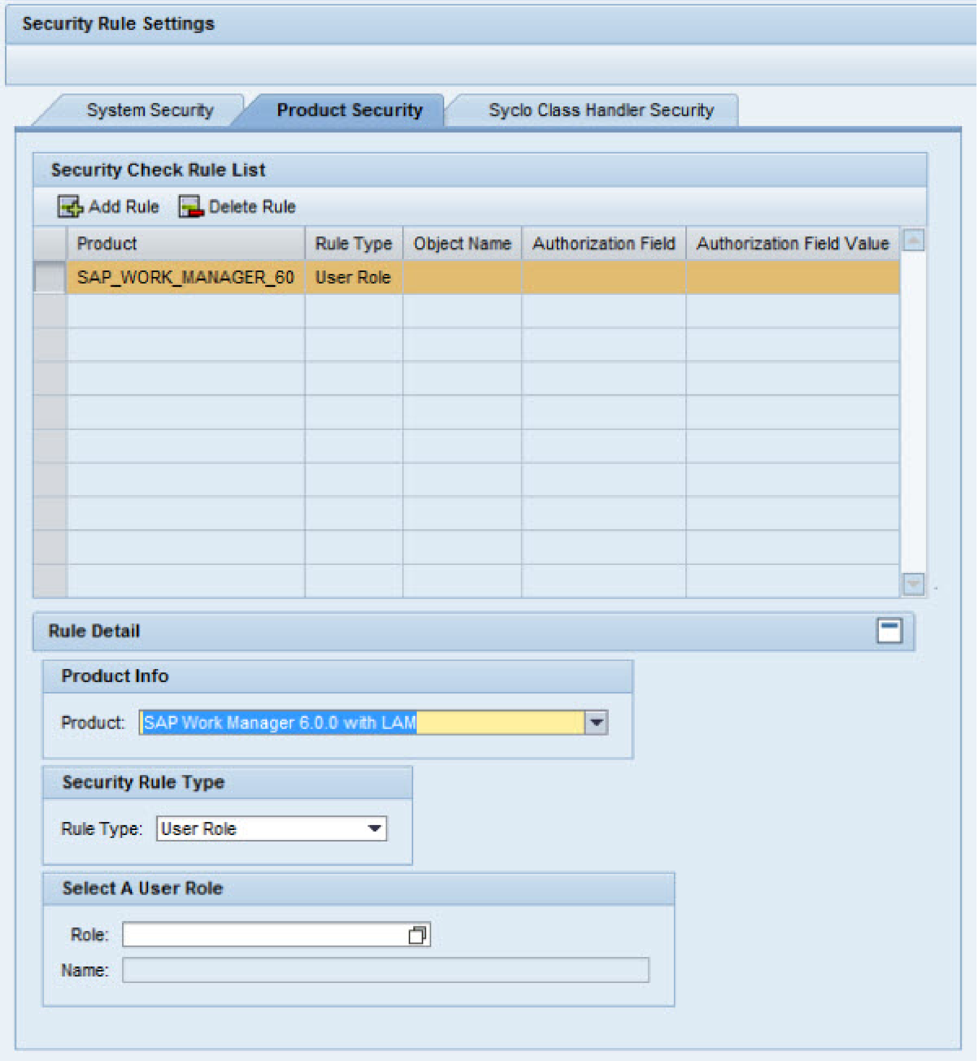
Rule Detail - Rule Type: Authorization Object

- **Rule Type: Authorization Object:** Baseline object used across mobile applications
- **Authorization Object:** Authorization object in SAP.
- **Authorization Field:** Standard SAP authorization object name
- **Authorization Field:** Free-text field. Text entered depends on developer implementation in SAP.

Security Settings - Product Security

Use this tab to configure security settings for a specific mobile application.

Security Settings - Product Security



Product	Rule Type	Object Name	Authorization Field	Authorization Field Value
SAP_WORK_MANAGER_60	User Role			

Product: SAP Work Manager 6.0.0 with LAM

Rule Type: User Role

Role:

Name:

Security Check Rule List

- **Add Rule button:** Press the **Add Rule** button to add a new product security rule. Fill in the fields in the Rule Detail section to automatically fill in the fields in this table.

- **Delete Rule button:** Press the **Delete Rule** button to delete an product security rule. To delete a product security rule, press the rectangle to the left of the Rule Type column in the row you wish to delete and press the Delete Rule button.
- **Product:** Mobile application chosen in the Rule Detail section. This is a non-editable field.
- **Rule Type:** Rule Type from the Rule Detail section. This is a non-editable field.
- **Object Name:** Taken from the Profile field in the Rule Detail section if rule type Authorization Object is chosen. This is a non-editable field.
- **Authorization Field:** Taken from the Authorization Field selection if rule type Authorization Object is chosen. Standard SAP authorization object name. This is a non-editable field.
- **Authorization Field Value:** Taken from the Field Value selection if rule type Authorization Object is chosen.

Rule Detail - Security Rule Type: User Role

- **Product:** Select the mobile application that will contain the product security rule.
- **Rule Type: User Role:** In addition to the standard SAP user profile rules, the user role can add restrictions on what a user can or cannot see in menus or other mobile application screens.
- **Role:** Select a user role contained within SAP.
- **Name:** After the user role is selected, this non-editable field is filled in with the descriptive name of the user role.

Rule Detail - Security Rule Type: Authorization Profile

- **Product:** Select the mobile application that will contain the product security rule.
- **Rule Type: Authorization Profile:** A collection of objects, or roles, such as Technician or Supervisor.
- **Profile:** Select an authorization profile contained within SAP.
- **Text:** After the authorization profile is selected, this non-editable field is filled in with the descriptive name of the authorization profile.

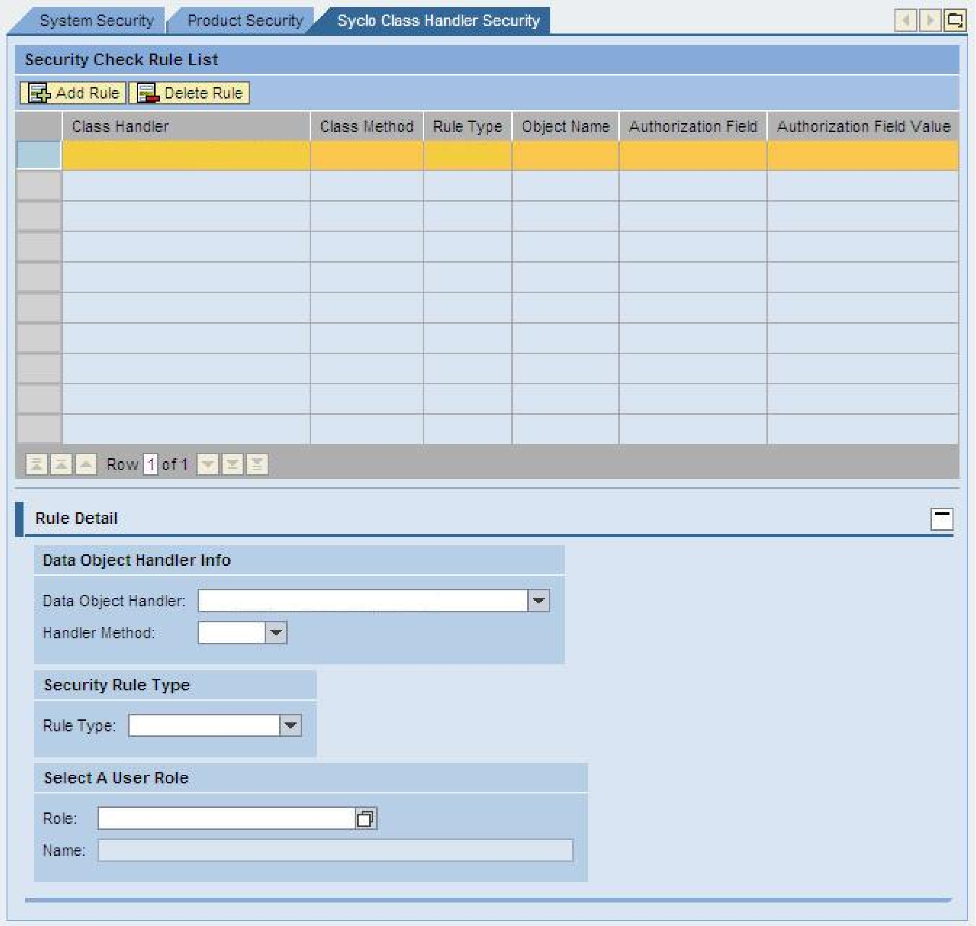
Rule Detail - Security Rule Type: Authorization Object

- **Product:** Select the mobile application that will contain the product security rule.
- **Rule Type: Authorization Object:** Baseline object used across mobile applications.
- **Authorization Object:** Select an authorization object contained within SAP.
- **Authorization Field:** Select a field contained within SAP.
- **Field Value:** Free-text field for additional object configuration. Text entered depends on developer implementation in SAP.

Security Settings - Class Handler Security

Use this tab to configure class handler security settings that cross mobile applications, but are only applicable for the selected data object handler chosen in the Rule Detail pane.

Security Settings - Class Handler Security



Class Handler	Class Method	Rule Type	Object Name	Authorization Field	Authorization Field Value

Data Object Handler:	
Handler Method:	
Rule Type:	
Role:	
Name:	

Security Check Rule List

- **Add Rule button:** Press the **[Add Rule]** button to add a new class handler security rule. Fill in the fields in the Rule Detail section to automatically fill in the fields in this table.
- **Delete Rule button:** Press the **[Delete Rule]** button to delete an class handler security rule. To delete a class handler security rule, press the rectangle to the left of the Rule Type column in the row you wish to delete and press the Delete Rule button.

- **Class Handler:** Taken from the Data Object Handler field in the Rule Detail section. This is a non-editable field.
- **Class Method:** Taken from the Handler Method field in the Rule Detail section. This is a non-editable field.
- **Rule Type:** Rule Type from the Rule Detail section. This is a non-editable field.
- **Object Name:** Taken from the Profile field in the Rule Detail section if rule type Authorization Object is chosen. This is a non-editable field.
- **Authorization Field:** Taken from the Authorization Field selection if rule type Authorization Object is chosen. Standard SAP authorization object name. This is a non-editable field.
- **Authorization Field Value:** Taken from the Field Value selection if rule type Authorization Object is chosen.

Rule Detail - Security Rule Type: User Role

- **Data Object Handler:** Select the desired class handler from the drop-down list.
- **Handler Method:** Select the desired handler method from the drop-down list.
- **Rule Type: User Role:** In addition to the standard SAP user profile rules, the user role can add restrictions on what a user can or cannot see in menus or other mobile application screens.
- **Role:** Select a user role contained within SAP.
- **Name:** After the user role is selected, this non-editable field is filled in with the descriptive name of the user role.

Rule Detail - Security Rule Type: Authorization Profile

- **Data Object Handler:** Select the desired class handler from the drop-down list.
- **Handler Method:** Select the desired handler method from the drop-down list.
- **Rule Type: Authorization Profile:** A collection of objects, or roles, such as Technician or Supervisor.
- **Profile:** Select an authorization profile contained within SAP.
- **Text:** After the authorization profile is selected, this non-editable field is filled in with the descriptive name of the authorization profile.

Rule Detail - Security Rule Type: Authorization Object

- **Data Object Handler:** Select the desired class handler from the drop-down list.
- **Handler Method:** Select the desired handler method from the drop-down list.
- **Rule Type: Authorization Object:** Baseline object used across mobile applications
- **Authorization Object:** Select an authorization object contained within SAP.
- **Authorization Field:** Select a field contained within SAP.
- **Field Value:** Free-text field for additional object configuration. Text entered depends on developer implementation in SAP.

System Administration

All components of the Agentry SAP Framework Administration, such as BAPI wrappers, mobile data objects and exchange objects, support logging. Activity logs generated by the Agentry SAP Framework are integrated into the standard SAP Application Log database.

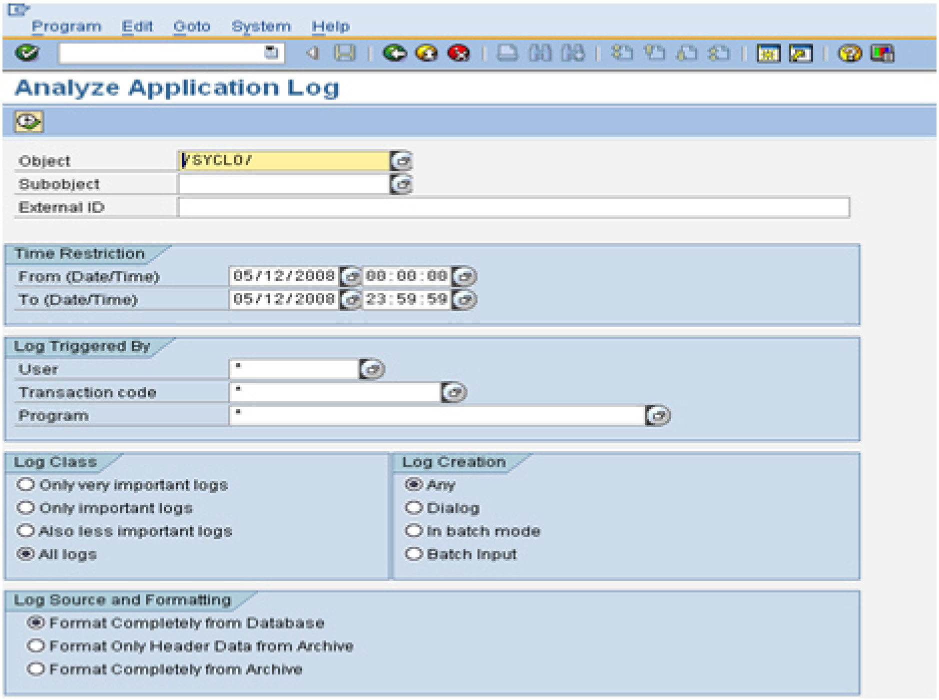
The recommended time frame for keeping the logs is:

- **Customization Table logs -** Do not delete
- **Exchange Table logs -** Keep no more than 6 months (180 days)
- **System logs -** Keep no more than 30 days

Viewing the Mobile Application Log

To view the application logs:

1. Start the Mobile Administration Menu by running the command `/n/syclo/smart` from the command field of SAPGUI session.
2. Select the transaction **/SYCLO/SLG1 - Application Log: Display Logs** to launch the start screen of the log display.



Analyze Application Log

Object: /SYCLO/
Subobject:
External ID:

Time Restriction
From (Date/Time): 05/12/2008 00:00:00
To (Date/Time): 05/12/2008 23:59:59

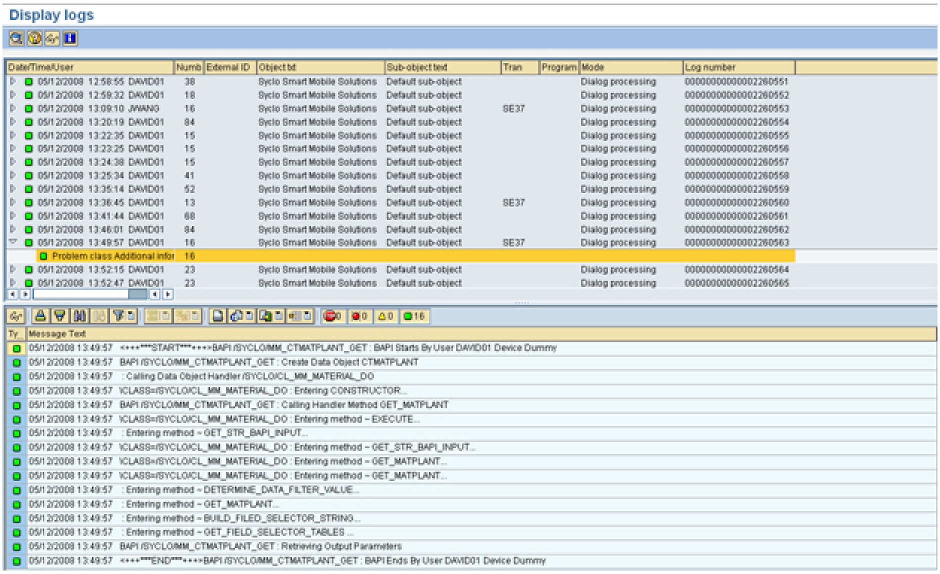
Log Triggered By
User: *
Transaction code: *
Program: *

Log Class
○ Only very important logs
○ Only important logs
○ Also less important logs
● All logs

Log Creation
● Any
○ Dialog
○ In batch mode
○ Batch Input

Log Source and Formatting
● Format Completely from Database
○ Format Only Header Data from Archive
○ Format Completely from Archive

3. Enter the desired the selection criteria. Make sure Object is set to `/SYCLO/`. Execute the transaction to view log details.



Display logs

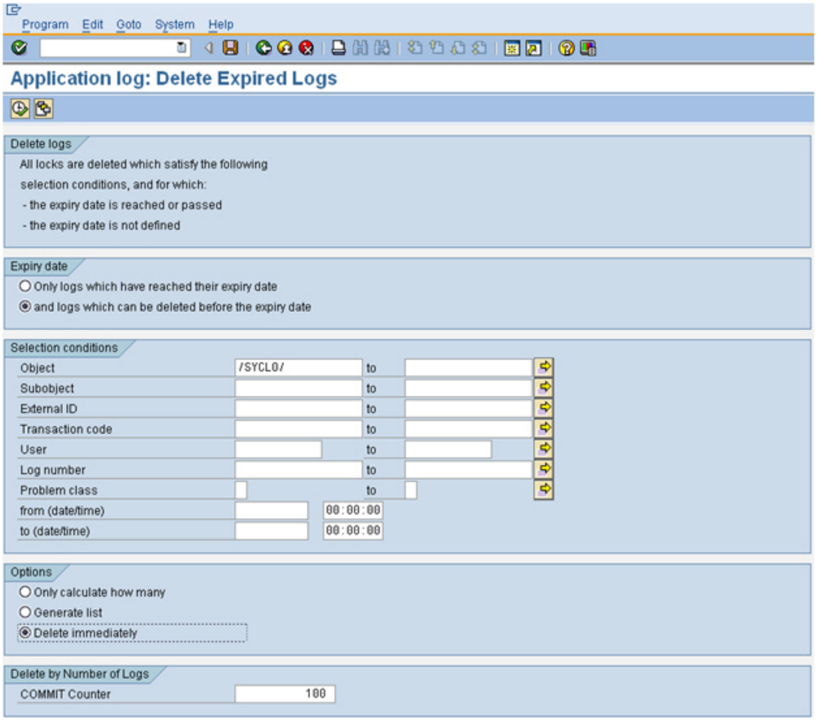
Date/Time/User	Numb	External ID	Object txt	Sub-object text	Tran	Program	Mode	Log number
▷ 🟩 05/12/2008 12:58:55 DAVID01	38		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260551
▷ 🟩 05/12/2008 12:59:32 DAVID01	18		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260552
▷ 🟩 05/12/2008 13:09:10 JWANG	16		Syclo Smart Mobile Solutions	Default sub-object	SE37		Dialog processing	00000000000002260553
▷ 🟩 05/12/2008 13:20:19 DAVID01	84		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260554
▷ 🟩 05/12/2008 13:22:35 DAVID01	15		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260555
▷ 🟩 05/12/2008 13:23:25 DAVID01	15		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260556
▷ 🟩 05/12/2008 13:24:38 DAVID01	15		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260557
▷ 🟩 05/12/2008 13:25:34 DAVID01	41		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260558
▷ 🟩 05/12/2008 13:35:14 DAVID01	52		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260559
▷ 🟩 05/12/2008 13:36:45 DAVID01	13		Syclo Smart Mobile Solutions	Default sub-object	SE37		Dialog processing	00000000000002260560
▷ 🟩 05/12/2008 13:41:44 DAVID01	68		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260561
▷ 🟩 05/12/2008 13:46:01 DAVID01	84		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260562
▽ 🟩 05/12/2008 13:49:57 DAVID01	16		Syclo Smart Mobile Solutions	Default sub-object	SE37		Dialog processing	00000000000002260563
🟩 Problem class Additional infor	16							
▷ 🟩 05/12/2008 13:52:15 DAVID01	23		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260564
▷ 🟩 05/12/2008 13:52:47 DAVID01	23		Syclo Smart Mobile Solutions	Default sub-object			Dialog processing	00000000000002260565

Ty_	Message Text
🟩	05/12/2008 13:49:57 <++++***START***++++>BAPI /SYCLO/MM_CTMATPLANT_GET : BAPI Starts By User DAVID01 Device Dummy
🟩	05/12/2008 13:49:57 BAPI /SYCLO/MM_CTMATPLANT_GET : Create Data Object CTMATPLANT
🟩	05/12/2008 13:49:57 : Calling Data Object Handler /SYCLO/CL_MM_MATERIAL_DO
🟩	05/12/2008 13:49:57 \CLASS=/SYCLO/CL_MM_MATERIAL_DO : Entering CONSTRUCTOR...
🟩	05/12/2008 13:49:57 BAPI /SYCLO/MM_CTMATPLANT_GET : Calling Handler Method GET_MATPLANT
🟩	05/12/2008 13:49:57 \CLASS=/SYCLO/CL_MM_MATERIAL_DO : Entering method ~ EXECUTE...
🟩	05/12/2008 13:49:57 : Entering method ~ GET_STR_BAPI_INPUT...
🟩	05/12/2008 13:49:57 \CLASS=/SYCLO/CL_MM_MATERIAL_DO : Entering method ~ GET_STR_BAPI_INPUT...
🟩	05/12/2008 13:49:57 \CLASS=/SYCLO/CL_MM_MATERIAL_DO : Entering method ~ GET_MATPLANT...
🟩	05/12/2008 13:49:57 \CLASS=/SYCLO/CL_MM_MATERIAL_DO : Entering method ~ GET_MATPLANT...
🟩	05/12/2008 13:49:57 : Entering method ~ DETERMINE_DATA_FILTER_VALUE...
🟩	05/12/2008 13:49:57 : Entering method ~ GET_MATPLANT...
🟩	05/12/2008 13:49:57 : Entering method ~ BUILD_FILED_SELECTOR_STRING...
🟩	05/12/2008 13:49:57 : Entering method ~ GET_FIELD_SELECTOR_TABLES ...
🟩	05/12/2008 13:49:57 BAPI /SYCLO/MM_CTMATPLANT_GET : Retrieving Output Parameters
🟩	05/12/2008 13:49:57 <++++***END***++++>BAPI /SYCLO/MM_CTMATPLANT_GET : BAPI Ends By User DAVID01 Device Dummy

Deleting an SAP Mobile Application Log

You should purge these logs every 30 days. To delete an application log:

1. Select transaction **/SYCLO/SLG2 - Application Log: Delete Logs** to launch the selection screen.



Application log: Delete Expired Logs

Delete logs

All locks are deleted which satisfy the following
selection conditions, and for which:
- the expiry date is reached or passed
- the expiry date is not defined

Expiry date

○ Only logs which have reached their expiry date
● and logs which can be deleted before the expiry date

Selection conditions

Object	/SYCLO/	to	
Subobject		to	
External ID		to	
Transaction code		to	
User		to	
Log number		to	
Problem class		to	
from (date/time)		00:00:00	
to (date/time)		00:00:00	

Options

○ Only calculate how many
○ Generate list
● Delete immediately

Delete by Number of Logs

COMMIT Counter 100

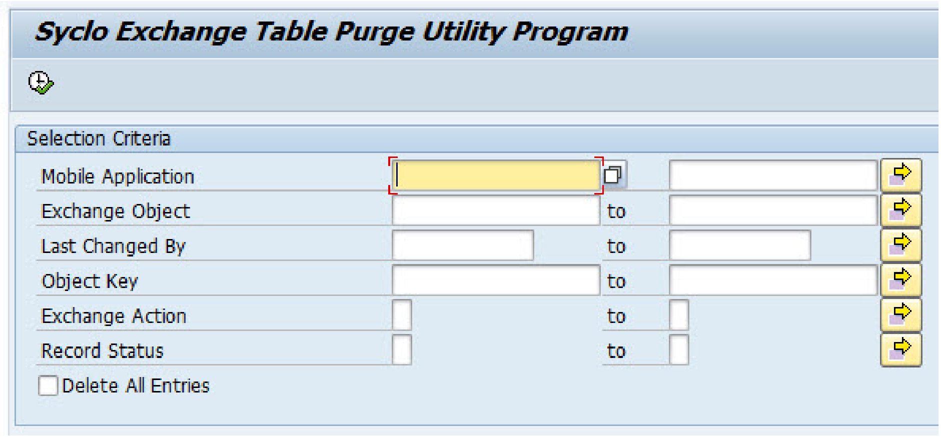
2. Enter selection criteria, and execute the transaction to purge the log database. The system will confirm deletion of logs in a popup window before deleting them.

Purge Utility for Exchange Persistent Layer

The purge utility for the exchange persistent layer is a tool that allows the system administrator to purge the content of the exchange tables. Exchange tables are not intended to be history tables. They should be purged periodically to maintain the exchange performance with the mobile application. It is recommended that you keep these tables no longer than every six months (180 days).

To purge the exchange table online:

1. Select transaction **/SYCLO/EX_PURGE - Exchange Table Purge Utility** to launch the selection screen.



Syclo Exchange Table Purge Utility Program

Selection Criteria

Mobile Application				
Exchange Object		to		
Last Changed By		to		
Object Key		to		
Exchange Action		to		
Record Status		to		

□ Delete All Entries

2. Specify the list of **Mobile Applications** to be included. Leave this blank to include all applications installed.
3. Specify the list of **Exchange Objects** to be included. Leave this blank to include all exchange objects.
4. Enable the option **Delete All Entries** to delete all records in the exchange tables. If the delete all option is not chosen, check the **Value of 'Days To Keep History** in the exchange object settings to determine what data should be purged.
5. Use the program **/SYCLO/CORE_EXCH_PURGE_PROG** to schedule a background job to carry out a periodic purge automatically.

Accessing the Administration and Monitoring Portal

Prerequisites

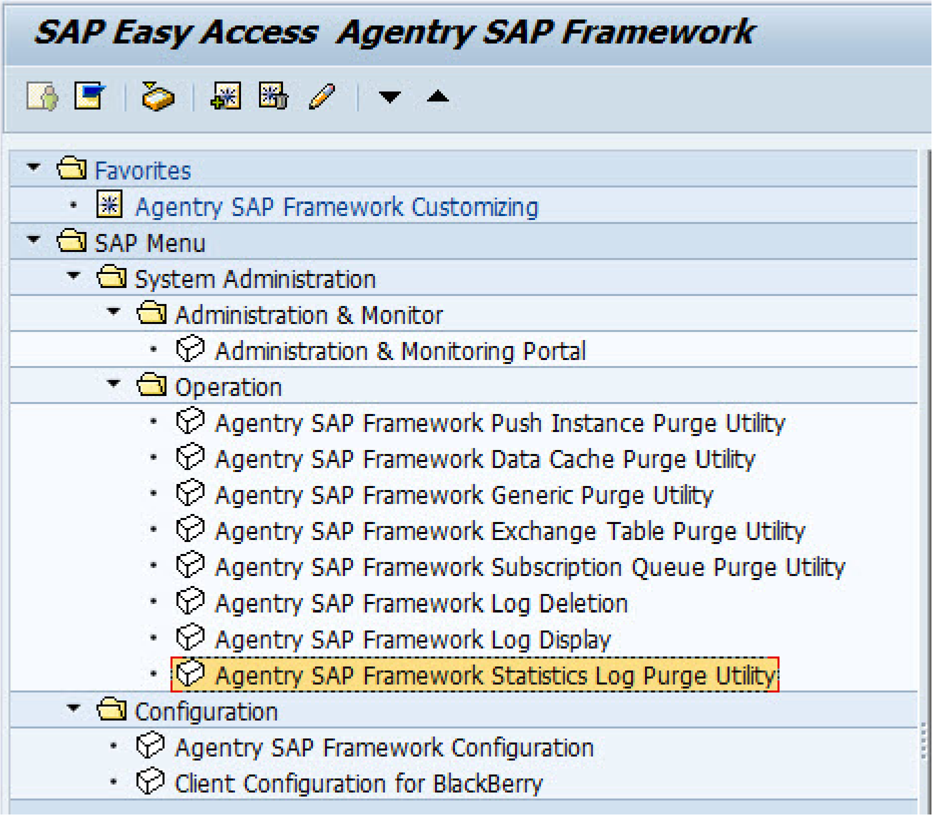
The person performing this procedure must log into the SAP system as a user with the following authorizations:

- *Authorization object* - S_ICF
- *Authorization field* - ICF_FIELD - SERVICE
- *Authorization field* - ICF_VALUE - SYCLOADM
- *Authorization object* - S_TCODE
- Authorization field - TCD - /SYCLO/ADMIN

Task

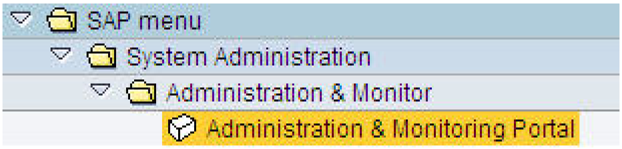
1. Log into SAP under an account with one of the authorizations provided in the prerequisites to this procedure.
2. Type `/n/syclo/smart` into the command field and click the green check mark to the left of the field, or press Enter.

   The SAP Easy Access Agentry SAP Framework window displays.


SAP Easy Access Agentry SAP Framework
Favorites
Agentry SAP Framework Customizing
SAP Menu
System Administration
Administration & Monitor
Administration & Monitoring Portal
Operation
Agentry SAP Framework Push Instance Purge Utility
Agentry SAP Framework Data Cache Purge Utility
Agentry SAP Framework Generic Purge Utility
Agentry SAP Framework Exchange Table Purge Utility
Agentry SAP Framework Subscription Queue Purge Utility
Agentry SAP Framework Log Deletion
Agentry SAP Framework Log Display
Agentry SAP Framework Statistics Log Purge Utility
Configuration
Agentry SAP Framework Configuration
Client Configuration for BlackBerry

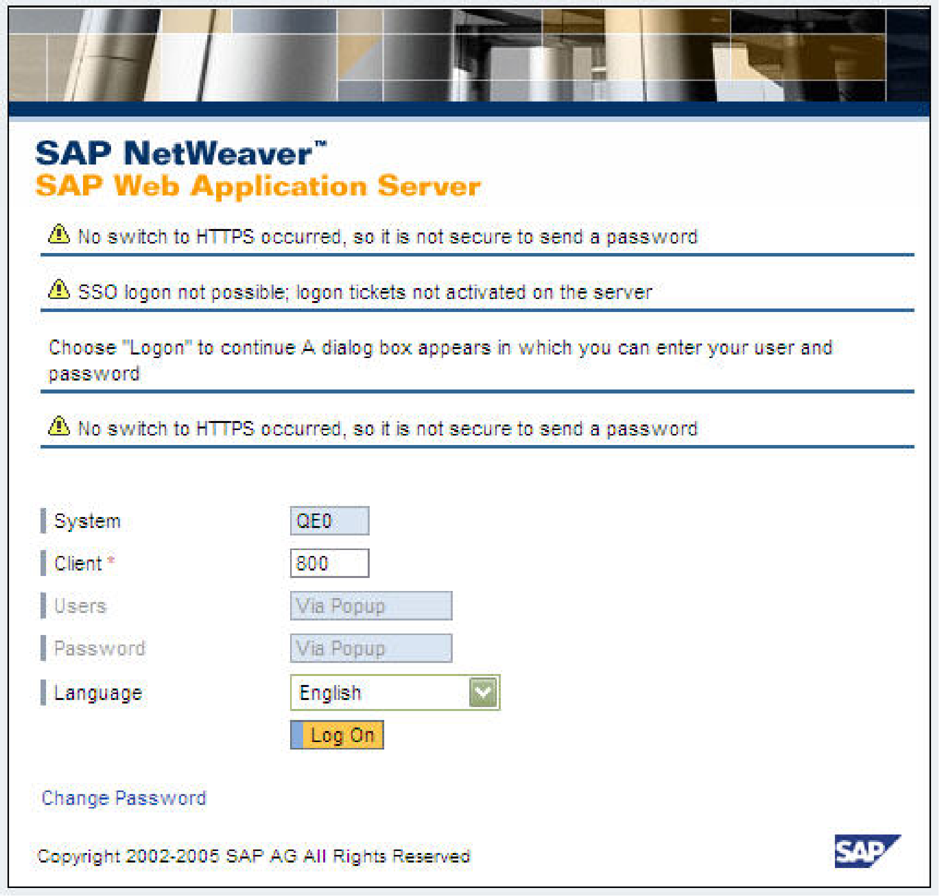
3. Expand the SAP menu tree by clicking the arrows to the left of the menu items. Expand as follows: **SAP menu | System Administration |Administration & Monitor**.

   The available administrative functions display.



4. Double-click on the Administration & Monitoring Portal menu item.

   The SAP NetWeaver Web Application Server logon screen opens in a browser window.



SAP NetWeaver™
SAP Web Application Server

⚠ No switch to HTTPS occurred, so it is not secure to send a password

⚠ SSO logon not possible; logon tickets not activated on the server

Choose "Logon" to continue A dialog box appears in which you can enter your user and password

⚠ No switch to HTTPS occurred, so it is not secure to send a password

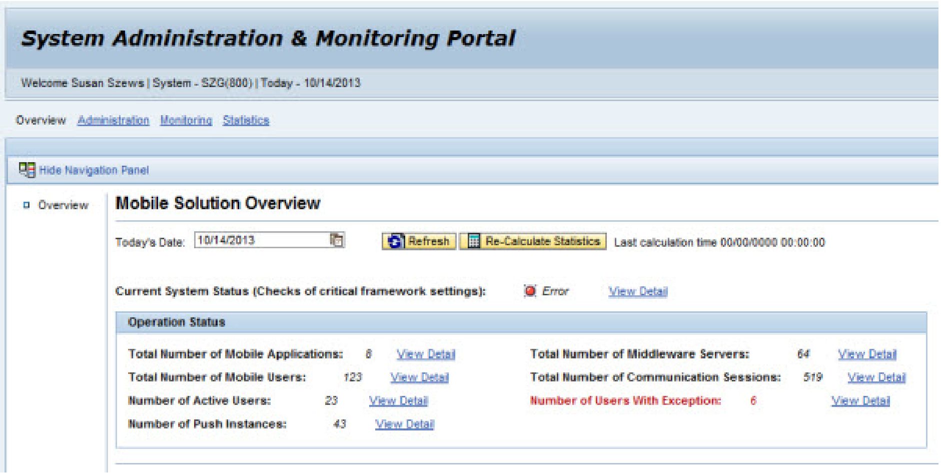
System	QE0
Client *	800
Users	Via Popup
Password	Via Popup
Language	English

Copyright 2002-2005 SAP AG All Rights Reserved

Note: Depending on your configuration, the log on screen may be different than what is shown.

5. Fill in any necessary fields for your specific implementation and click the **Log On** button.

   The System Administration & Monitoring portal opens in the browser window.



Welcome Susan Szews | System - SZG(800) | Today - 10/14/2013

Mobile Solution Overview

Today's Date: 10/14/2013

Last calculation time 00/00/0000 00:00:00

Current System Status (Checks of critical framework settings): Error

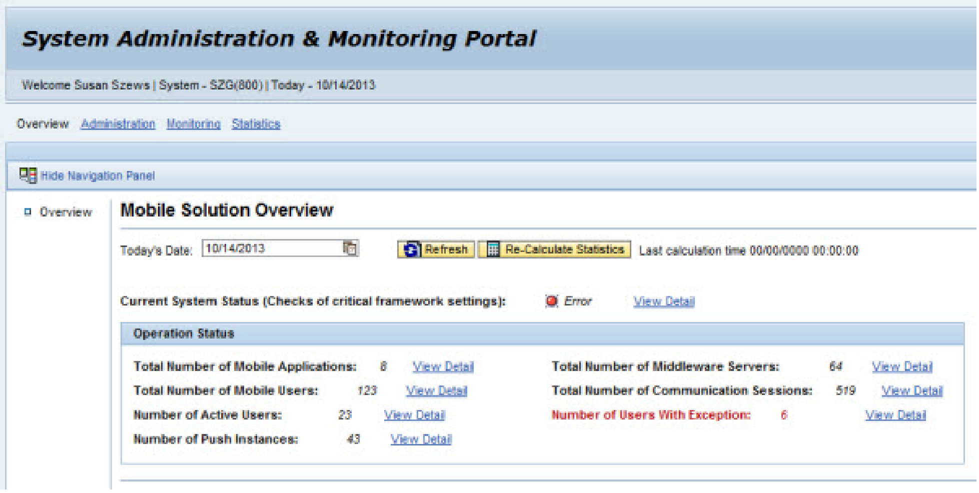
Operation Status

Total Number of Mobile Applications:	8	Total Number of Middleware Servers:	64
Total Number of Mobile Users:	123	Total Number of Communication Sessions:	519
Number of Active Users:	23	Number of Users With Exception:	6
Number of Push Instances:	43		

Administration Portal - Mobile Solution Overview

An administrator can monitor the current system status through the main overview panel.

Mobile Solution Overview Panel


System Administration & Monitoring Portal

Welcome Susan Szews | System - SZG(800) | Today - 10/14/2013

Mobile Solution Overview

Today's Date: 10/14/2013

Last calculation time 00/00/0000 00:00:00

Current System Status (Checks of critical framework settings): Error

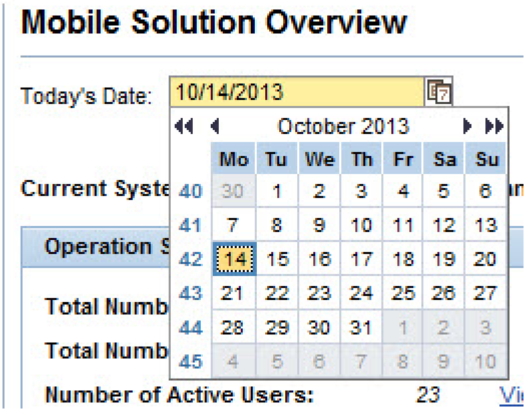
Operation Status

Total Number of Mobile Applications: 8 — Total Number of Middleware Servers: 64
Total Number of Mobile Users: 123 — Total Number of Communication Sessions: 519
Number of Active Users: 23 — Number of Users With Exception: 6
Number of Push Instances: 43

Today’s Date: The date is automatically set to the current date when the Administration portal is first accessed. To view system status for past dates: click the calendar icon to the right of the date field and choose the desired date.

1. Click the calendar icon to the right of the date field and choose the desired date.

   Mobile Solution Overview - Date Selection



Mobile Solution Overview

Today's Date: 10/14/2013

Current Syste

Operation S

Total Numb

Total Numb

Number of Active Users: 23 Vi

October 2013

	Mo	Tu	We	Th	Fr	Sa	Su
40	30	1	2	3	4	5	6
41	7	8	9	10	11	12	13
42	14	15	16	17	18	19	20
43	21	22	23	24	25	26	27
44	28	29	30	31	1	2	3
45	4	5	6	7	8	9	10

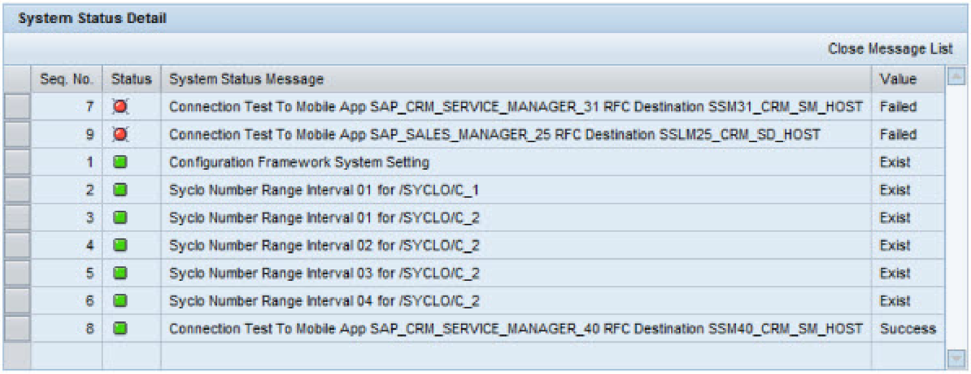
2. Click the **Refresh** button to refresh the Operation Status statistics for the chosen date.
3. To revert back to the current date, select the calendar icon again and choose the current date, highlighted by the blue box. Then click the **Refresh** button to refresh the Operation Status statistics.

Current System Status (Checks of critical framework settings): Brief overview of the current system status through use of an icon and status text. Click on the **View Detail** hyperlink to view the System Status Detail table as shown in the following example.

Current System Status Detail

The System Status Detail table appears when the View Detail hyperlink to the right of the Current Systems Status (Checks of critical framework settings) line is clicked. This table provides an account of the framework and its status during the selected date.

Mobile Solution Overview - System Status Detail



System Status Detail

Close Message List

Seq. No.	Status	System Status Message	Value
7	🔴	Connection Test To Mobile App SAP_CRM_SERVICE_MANAGER_31 RFC Destination SSM31_CRM_SM_HOST	Failed
9	🔴	Connection Test To Mobile App SAP_SALES_MANAGER_25 RFC Destination SSLM25_CRM_SD_HOST	Failed
1	🟩	Configuration Framework System Setting	Exist
2	🟩	Syclo Number Range Interval 01 for /SYCLO/C_1	Exist
3	🟩	Syclo Number Range Interval 01 for /SYCLO/C_2	Exist
4	🟩	Syclo Number Range Interval 02 for /SYCLO/C_2	Exist
5	🟩	Syclo Number Range Interval 03 for /SYCLO/C_2	Exist
6	🟩	Syclo Number Range Interval 04 for /SYCLO/C_2	Exist
8	🟩	Connection Test To Mobile App SAP_CRM_SERVICE_MANAGER_40 RFC Destination SSM40_CRM_SM_HOST	Success

- **Seq. No.:** Row number of the system tests performed
- **Status:** Icon display of the system status

- **System Status Message:** Text message giving a status description. There are three main system tests performed:
  - **Configuration framework system setting -** If this setting does not exist, or returns an error status message, applications will not be able to run on the framework.
  - **Syclo number range interval -** Automatic number counter. If this is not set up during the system installation, the counters will not work. See the installation guide for more details.
  - **Connection test -** This test is dynamic and only appears if the system is multi-back end enabled. For the system to post a Success value for this test, the RFC destinations must be defined.
- **Value:** Text status of the system test

Click the **Close Message List** button to close the table view.

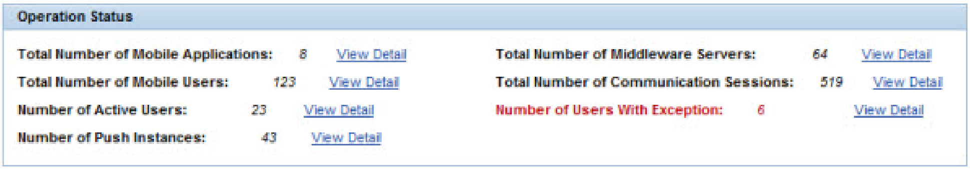
Mobile Solution Overview - Operation Status

The Operation Status panel of the Mobile Solution Overview provides a high-level status of activity on the system. Clicking on the View Detail link to the right of each status displays a more detailed status table below the Operation Status panel.

To access operation status details for a different date, choose a different date using the calendar icon at the top of the panel. Be sure to click **Refresh** in order to refresh the panel statistics to the newly-selected date.

Note: If statistics collection is not enabled in SAP, some operation status details will not be available through the Administration & Monitoring portal.

Mobile Solution Overview - Operation Status



Operation Status

Total Number of Mobile Applications: 8 View Detail

Total Number of Mobile Users: 123 View Detail

Number of Active Users: 23 View Detail

Number of Push Instances: 43 View Detail

Total Number of Middleware Servers: 64 View Detail

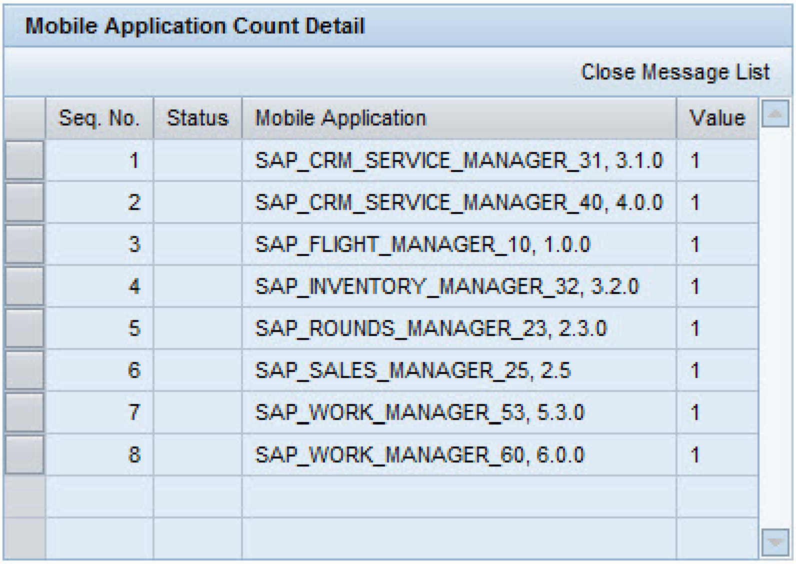
Total Number of Communication Sessions: 519 View Detail

Number of Users With Exception: 6 View Detail

Total Number of Mobile Applications

Lists the number of mobile applications in use on the system.

Mobile Solution Overview - Mobile Application Count Detail



Mobile Application Count Detail

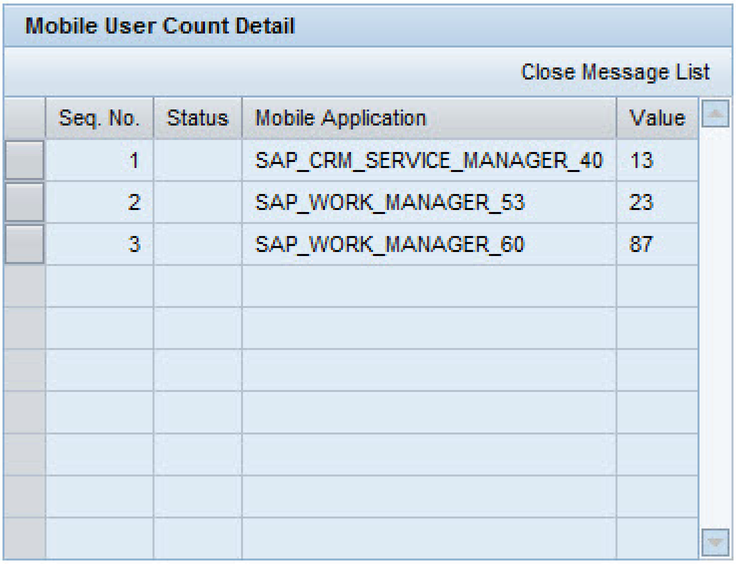
Seq. No.	Status	Mobile Application	Value
1		SAP_CRM_SERVICE_MANAGER_31, 3.1.0	1
2		SAP_CRM_SERVICE_MANAGER_40, 4.0.0	1
3		SAP_FLIGHT_MANAGER_10, 1.0.0	1
4		SAP_INVENTORY_MANAGER_32, 3.2.0	1
5		SAP_ROUNDS_MANAGER_23, 2.3.0	1
6		SAP_SALES_MANAGER_25, 2.5	1
7		SAP_WORK_MANAGER_53, 5.3.0	1
8		SAP_WORK_MANAGER_60, 6.0.0	1

- **Seq. No.:** Row number in the table
- **Status:** Not used in this table
- **Mobile Application:** Name and release of the mobile application
- **Value:** Number of specific mobile applications in use on the system

Total Number of Mobile Users

Lists the total number of mobile users on the system on the date chosen.

Mobile Solution Overview - Mobile User Count Detail



Mobile User Count Detail

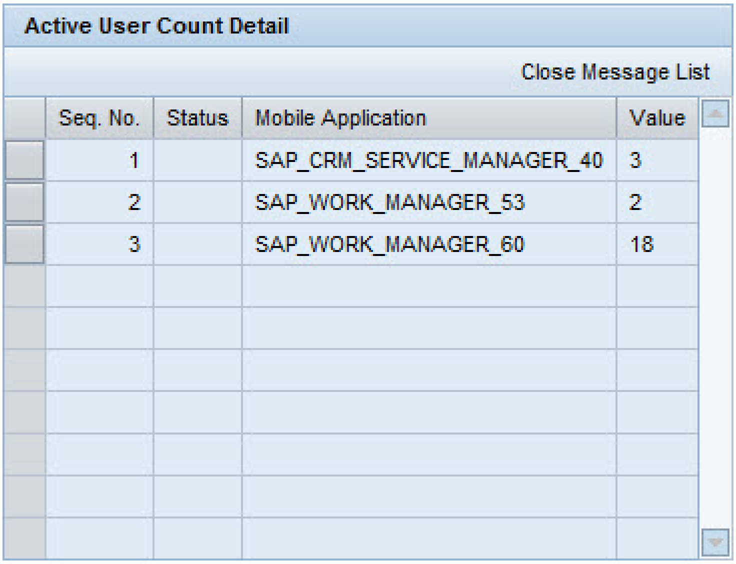
Seq. No.	Status	Mobile Application	Value
1		SAP_CRM_SERVICE_MANAGER_40	13
2		SAP_WORK_MANAGER_53	23
3		SAP_WORK_MANAGER_60	87

- **Seq. No.:** Row number in the table
- **Status:** Not used in this table
- **Mobile Application:** Name of the mobile application
- **Value:** Number of active users on the specific mobile application

Number of Active Users

Lists the total number of active users on the system on the date chosen.

Mobile Solution Overview - Active User Count Detail



Active User Count Detail

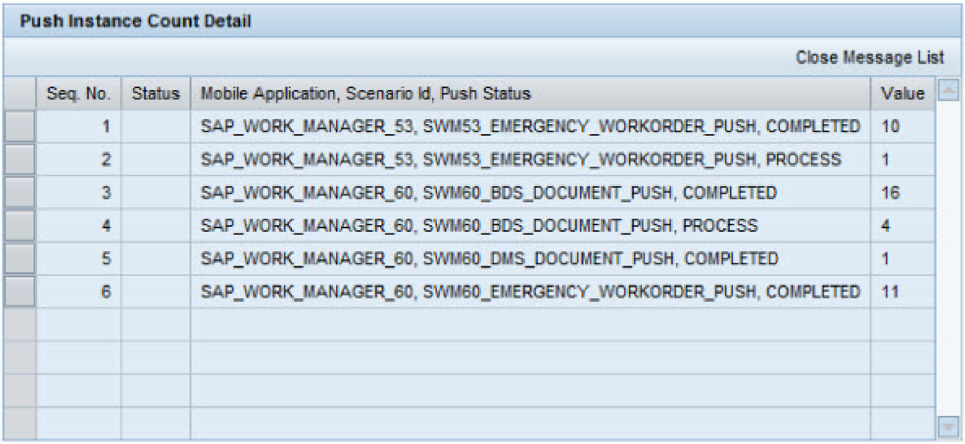
Seq. No.	Status	Mobile Application	Value
1		SAP_CRM_SERVICE_MANAGER_40	3
2		SAP_WORK_MANAGER_53	2
3		SAP_WORK_MANAGER_60	18

- **Seq. No.:** Row number in the table
- **Status:** Not used in this table
- **Mobile Application:** Name of the mobile application
- **Value:** Number of active users on the mobile application

Number of Push Instances

Lists the total number of pushes initiated on the system on the date chosen.

Mobile Solution Overview - Push Instance Count Detail



Push Instance Count Detail

Seq. No.	Status	Mobile Application, Scenario Id, Push Status	Value
1		SAP_WORK_MANAGER_53, SWM53_EMERGENCY_WORKORDER_PUSH, COMPLETED	10
2		SAP_WORK_MANAGER_53, SWM53_EMERGENCY_WORKORDER_PUSH, PROCESS	1
3		SAP_WORK_MANAGER_60, SWM60_BDS_DOCUMENT_PUSH, COMPLETED	16
4		SAP_WORK_MANAGER_60, SWM60_BDS_DOCUMENT_PUSH, PROCESS	4
5		SAP_WORK_MANAGER_60, SWM60_DMS_DOCUMENT_PUSH, COMPLETED	1
6		SAP_WORK_MANAGER_60, SWM60_EMERGENCY_WORKORDER_PUSH, COMPLETED	11

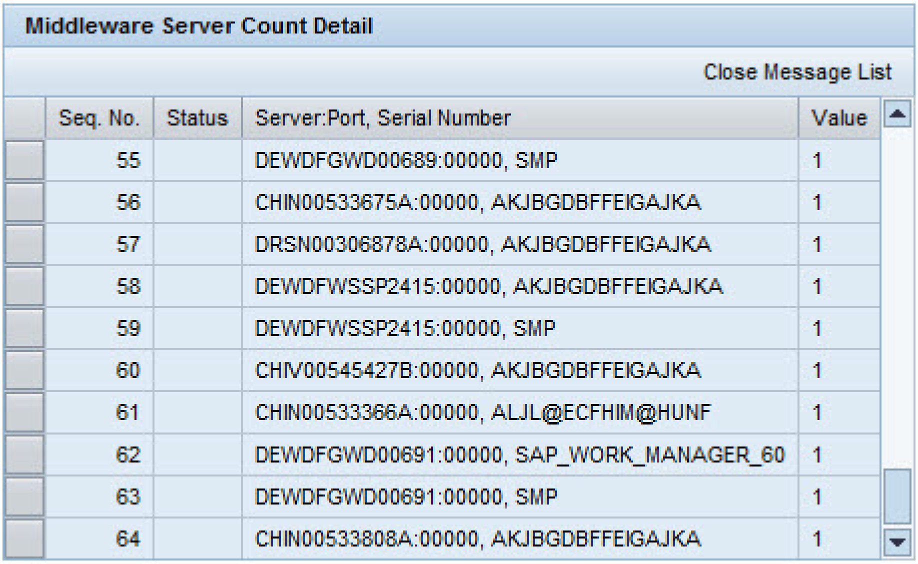
- **Seq. No.:** Row number in the table

- **Status:** Not used in this table
- **Mobile Application, Scenario ID, Push Status:** Name of the mobile application, name of the push instance, and status of the push instance
- **Value:** Number of push instances initiated on each mobile application

Total Number of Middleware Servers

Lists the total number of middleware servers present on the system.

Mobile Solution Overview - Middleware Server Count Detail



Middleware Server Count Detail

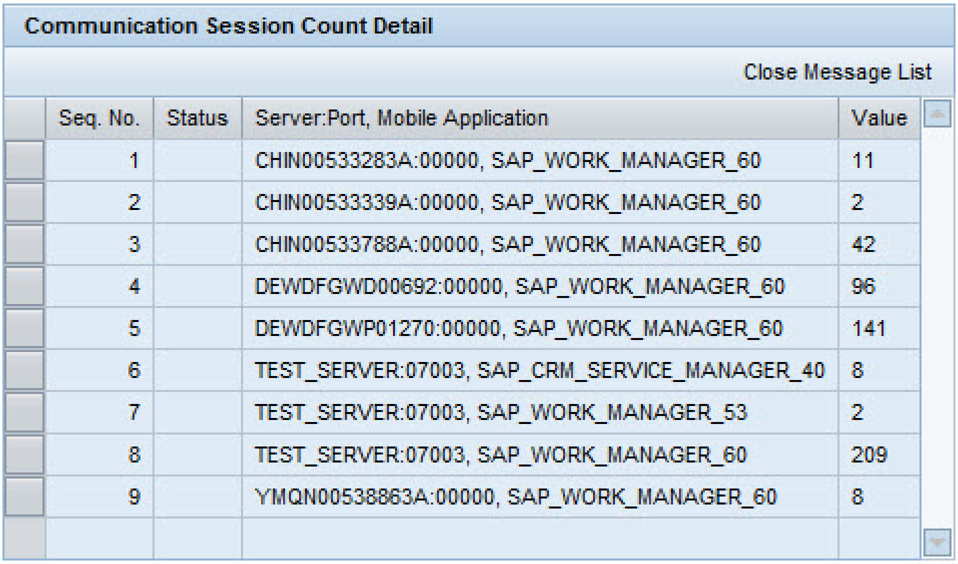
Seq. No.	Status	Server:Port, Serial Number	Value
55		DEWDFGWD00689:00000, SMP	1
56		CHIN00533675A:00000, AKJBGDBFFEIGAJKA	1
57		DRSN00306878A:00000, AKJBGDBFFEIGAJKA	1
58		DEWDFWSSP2415:00000, AKJBGDBFFEIGAJKA	1
59		DEWDFWSSP2415:00000, SMP	1
60		CHIV00545427B:00000, AKJBGDBFFEIGAJKA	1
61		CHIN00533366A:00000, ALJL@ECFHIM@HUNF	1
62		DEWDFGWD00691:00000, SAP_WORK_MANAGER_60	1
63		DEWDFGWD00691:00000, SMP	1
64		CHIN00533808A:00000, AKJBGDBFFEIGAJKA	1

- **Seq. No.:**Row number in the table
- **Status:** Not used in this table
- **Server: Port, Serial Number:** Name of the middleware server, the port it uses to connect to the framework, and the serial number assigned to it in SAP.
- **Value:** Number of middleware servers with that server name, port and serial number. This number will always be 1.

Total Number of Communication Sessions

Lists the total number of communication sessions for the date chosen.

Mobile Solution Overview - Communication Session Count Detail



Communication Session Count Detail

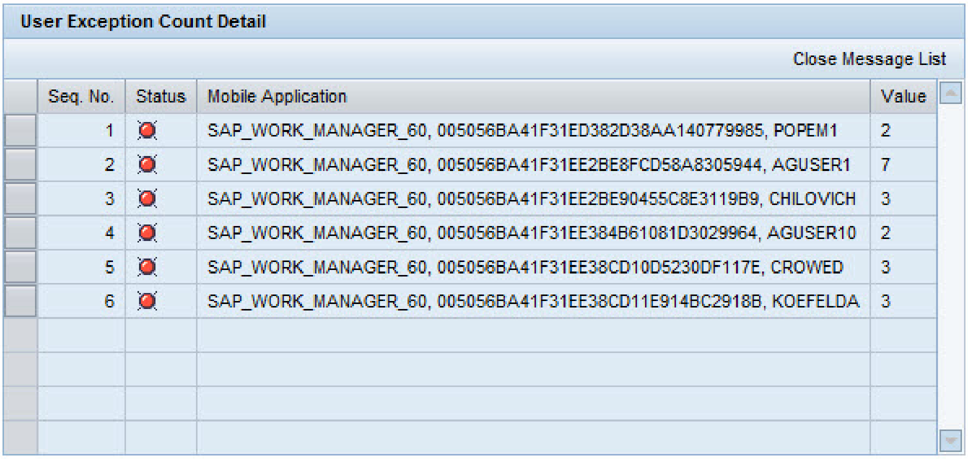
Seq. No.	Status	Server:Port, Mobile Application	Value
1		CHIN00533283A:00000, SAP_WORK_MANAGER_60	11
2		CHIN00533339A:00000, SAP_WORK_MANAGER_60	2
3		CHIN00533788A:00000, SAP_WORK_MANAGER_60	42
4		DEWDFGWD00692:00000, SAP_WORK_MANAGER_60	96
5		DEWDFGWP01270:00000, SAP_WORK_MANAGER_60	141
6		TEST_SERVER:07003, SAP_CRM_SERVICE_MANAGER_40	8
7		TEST_SERVER:07003, SAP_WORK_MANAGER_53	2
8		TEST_SERVER:07003, SAP_WORK_MANAGER_60	209
9		YMQN00538863A:00000, SAP_WORK_MANAGER_60	8

- **Seq. No.:** Row number in the table
- **Status:** Not used in this table
- **Server: Port, Mobile Application:** Name of the server, the port it uses to connect to the framework, and the mobile application used on the server for the communication sessions
- **Value:** Number of communication sessions on the specific server listed

Number of Users with Exception

If a user encounters issues during a transmit, the user ID, the mobile application, and the application ID is logged.

Mobile Solution Overview - User Exception Count Detail



User Exception Count Detail

Seq. No.	Status	Mobile Application	Value
1	●	SAP_WORK_MANAGER_60, 005056BA41F31ED382D38AA140779985, POPEM1	2
2	●	SAP_WORK_MANAGER_60, 005056BA41F31EE2BE8FCD58A8305944, AGUSER1	7
3	●	SAP_WORK_MANAGER_60, 005056BA41F31EE2BE90455C8E3119B9, CHILOVICH	3
4	●	SAP_WORK_MANAGER_60, 005056BA41F31EE384B61081D3029964, AGUSER10	2
5	●	SAP_WORK_MANAGER_60, 005056BA41F31EE38CD10D5230DF117E, CROWED	3
6	●	SAP_WORK_MANAGER_60, 005056BA41F31EE38CD11E914BC2918B, KOEFELDA	3

- **Seq. No.:** Row number in the table
- **Status:** Visual indicator showing the severity of the exception
- **Mobile Application:** Name of the mobile application, GUID of the mobile application, and user name causing the exception
- **Value:** Number of exceptions the listed user has on the listed mobile application

Administration Portal - Administration

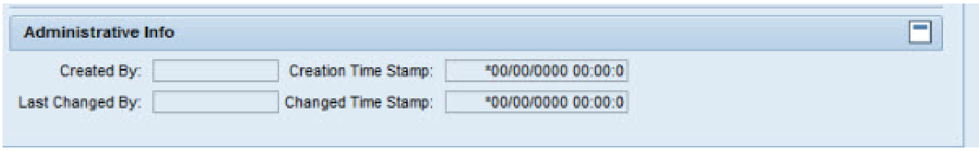
The Administration panel is used to create and manage the following three areas:

- **User Management -** System administrators can create new mobile user IDs, as well as manage all settings associated with specific user IDs.
- **Server Management -** System administrators can create new connections to new servers, as well as manage all settings associated with specific servers.
- **Runtime Logging Level Setting -** System administrators can create or modify new logging parameters associated with specific user GUIDs.
- **Mobile Application Parameters -** Use the Mobile Application Parameter Management panel to override parameter settings in a production environment for a specific user or for the whole application, depending on the selection you make for the parameter scope.

Administrative Info

Each management panel contains a section at the bottom of the page called Administrative Info. Expand the section by clicking on the white box icon to the right of the Administrative Info heading.

Administration Portal - Administrative Info



Administrative Info

Created By:

Creation Time Stamp: *00/00/0000 00:00:0

Last Changed By:

Changed Time Stamp: *00/00/0000 00:00:0

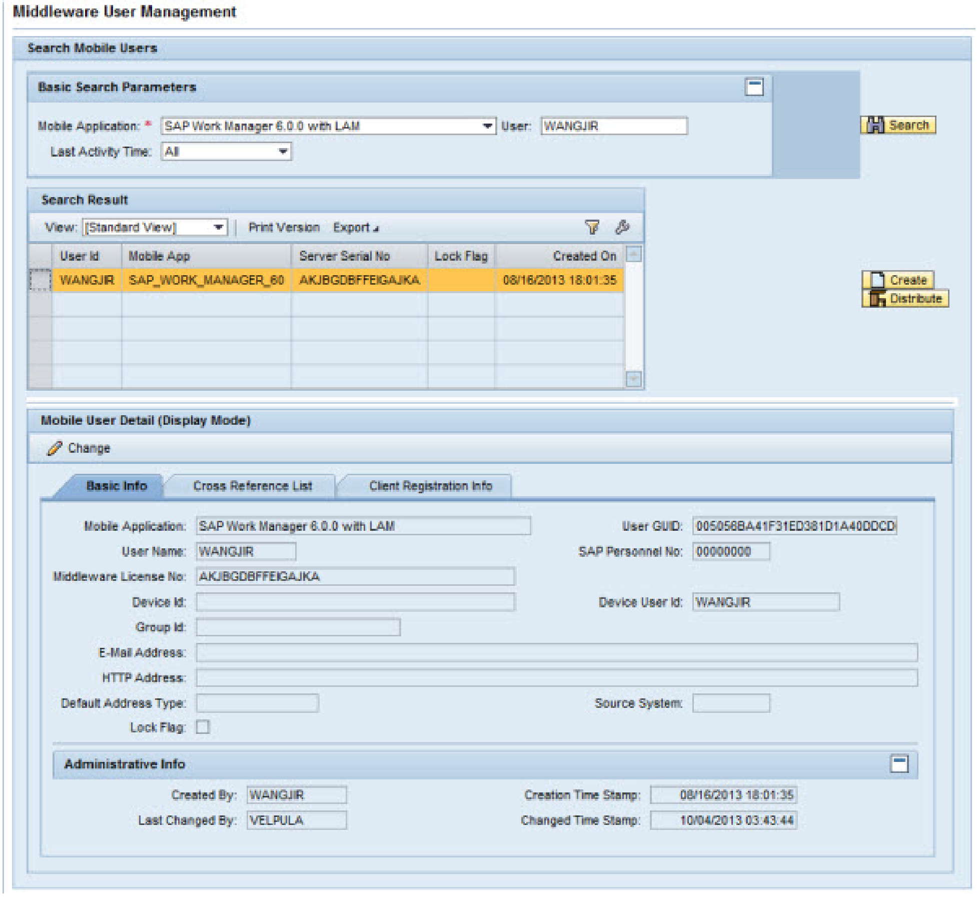
- **Created By:** SAP user ID of the person who created the information
- **Creation Time Stamp:** Time and date the information was created
- **Last Changed By:** SAP user ID of the person who modified information in the panel
- **Changed Time Stamp:** Time and date the information was modified

Administration - User Management

The Middleware User Management panel is used to create and manage mobile client user IDs. These IDs are layered on top of the user's SAP user ID.

A user must have an SAP user ID before being able to access any mobile applications. A user must also have a separate user GUID for each mobile application they access. In this way, the system can track and calculate exchange episodes for each application a user accesses while keeping the exchanges separated by mobile application.

Administration - Middleware User Management



Middleware User Management

Search Mobile Users

Basic Search Parameters

Mobile Application: * SAP Work Manager 6.0.0 with LAM
User: WANGJIR
Last Activity Time: All

Search Result

View: [Standard View]   Print Version   Export

User Id	Mobile App	Server Serial No	Lock Flag	Created On
WANGJIR	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		08/16/2013 18:01:35

Mobile User Detail (Display Mode)

Mobile Application: SAP Work Manager 6.0.0 with LAM
User GUID: 005056BA41F31ED381D1A40DDCD
User Name: WANGJIR
SAP Personnel No: 00000000
Middleware License No: AKJBGDBFFEIGAJKA
Device Id:
Device User Id: WANGJIR
Group Id:
E-Mail Address:
HTTP Address:
Default Address Type:
Source System:
Lock Flag: □

Administrative Info

Created By: WANGJIR
Creation Time Stamp: 08/16/2013 18:01:35
Last Changed By: VELPULA
Changed Time Stamp: 10/04/2013 03:43:44

Basic Search Parameters

Use this field to search for users on a specific mobile application, or refine the search even further by the user ID or last activity time. Once specific user(s) are found, their information can be changed if necessary.

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application. This is a required field to enable the Search function.
- **User:** SAP user ID
- **Last Activity Time:** Use the drop-down menu to select a window of time. The default is set to All, which is equal to selecting no specific time period.
- **Search:** Click the **Search** button once after you have entered all required and additional search parameters. If the search returns results, they are displayed in the Search Result section. If the search parameters are not fulfilled, a message stating “No data found” displays in the Search Result section.

Search Result

When you click the **Search** button in the Basic Search Parameters section, the Search Result table is populated. You can also create a new user in this section using the **Create** button. In this case, it is not necessary to search for an existing user.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet.
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Mobile User Detail - Basic Info

When a user is selected in the Search Results table, the fields are populated in the Basic Info section. Click **Change** to modify editable fields. Click **Create** in the Search Result section to create a new user through the editable fields. Fields with a red asterisk beside them are mandatory.

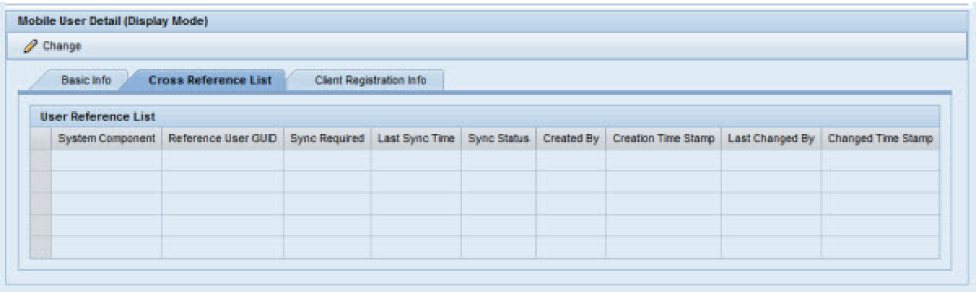
- **Mobile Application:** Name of the mobile application chosen in the Basic Search Parameters section
- **User GUID:** User GUID for the specific mobile application chosen. Each user has a different GUID for each mobile application assigned in SAP.
- **User Name:** User ID chosen from the table in the Search Result section
- **SAP Personnel Number:** User ID number assigned to the user in SAP
- **Middleware License Number:** Server serial number of the middleware server that is running the chosen mobile application
- **Device ID:** If a user is assigned to a mobile device connected to the chosen application and the mobile device has an ID, the field is populated.
- **Device User ID:** User ID assigned to the mobile device
- **Group ID:** Allows you to define additional information about a user, such as a crew number or team name. This is a free-text field.
- **Email Address:** Email address of the user. Email are sent through the Push Scenario Definition panel in the Configuration portal. Email are also used in the User Monitor panel under Monitoring in the Administration portal.

- **HTTP Address:** Web address or site links that users can access, such as a Twitter feed
- **Default Address Type:** Select either Email Address or HTTP Address from the drop-down menu for the default email sending method
- **Source System:** Originating SAP system of the user ID. This is useful in multi-backend systems.
- **Lock Flag:** If checked, user is unable to access the mobile application from the mobile device. This is used if a mobile device is lost or stolen and the application data must be made inaccessible.

Mobile User Detail - Cross Reference List

The Cross Reference list is used in systems that contain multi-back ends. When a user is created or modified, that user's information will need to be replicated across the systems. This detail view provides statistics and information for each user on a multi-back end system.

Figure 14: Mobile User Detail - Cross Reference List



Mobile User Detail (Display Mode)

User Reference List

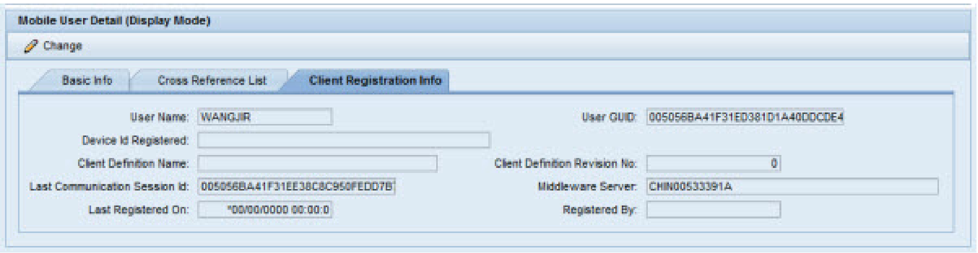
System Component	Reference User GUID	Sync Required	Last Sync Time	Sync Status	Created By	Creation Time Stamp	Last Changed By	Changed Time Stamp

- **System Component:** System where the user ID resides
- **Reference User GUID:** SAP user GUID
- **Sync Required:** If box is checked, synchronization is required
- **Last Sync Time:** Time and date of last synchronization
- **Sync Status:** Descriptive text detailing the synchronization status
- **Created By:** SAP ID of the user who created the new user ID
- **Creation Time Stamp:** Time and date of the
- **Last Changed By:** SAP ID of the user who made the change
- **Changed Time Stamp:** Time and date of last change

Mobile User Detail - Client Registration Info

The Client Registration Info tab is used for RIM products only.

Figure 15: Mobile User Detail - Client Registration Info



User Name: WANGJIR

User GUID: 005056BA41F31ED381D1A40DDCDE4

Device Id Registered:

Client Definition Name:

Client Definition Revision No: 0

Last Communication Session Id: 005056BA41F31EE38C8C950FEDD7B

Middleware Server: CHIN00533391A

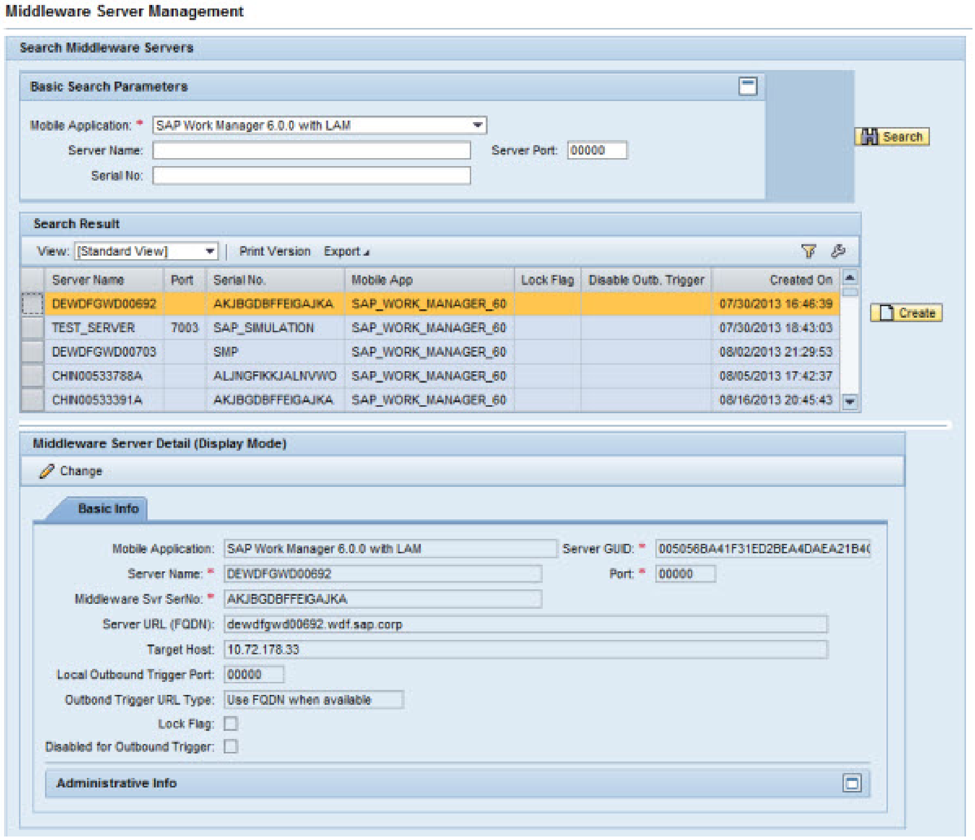
Last Registered On: *00/00/0000 00:00:0

Registered By:

Administration - Server Management

The Middleware Server Management panel is used to create and manage the middleware servers on the system.

Administration - Middleware Server Management



Mobile Application: * SAP Work Manager 6.0.0 with LAM

Server Name:

Server Port: 00000

Serial No:

View: [Standard View]

Server Name	Port	Serial No.	Mobile App	Lock Flag	Disable Outb. Trigger	Created On
DEWDFGWD00692		AKJBGDBFFEIGAJKA	SAP_WORK_MANAGER_60			07/30/2013 16:46:39
TEST_SERVER	7003	SAP_SIMULATION	SAP_WORK_MANAGER_60			07/30/2013 18:43:03
DEWDFGWD00703		SMP	SAP_WORK_MANAGER_60			08/02/2013 21:29:53
CHIN00533788A		ALJNGFIKKJALNVWO	SAP_WORK_MANAGER_60			08/05/2013 17:42:37
CHIN00533391A		AKJBGDBFFEIGAJKA	SAP_WORK_MANAGER_60			08/16/2013 20:45:43

Mobile Application: SAP Work Manager 6.0.0 with LAM

Server GUID: * 005056BA41F31ED2BEA4DAEA21B4(

Server Name: * DEWDFGWD00692

Port: * 00000

Middleware Svr SerNo: * AKJBGDBFFEIGAJKA

Server URL (FQDN): dewdfgwd00692.wdf.sap.corp

Target Host: 10.72.178.33

Local Outbound Trigger Port: 00000

Outbond Trigger URL Type: Use FQDN when available

Lock Flag: □

Disabled for Outbound Trigger: □

Basic Search Parameters

Use this field to search for users on a specific mobile application, or refine the search even further by the user ID or last activity time. Once specific user(s) are found, their information can be changed if necessary.

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application. This is a required field to enable the Search function.
- **Server Name:** Name given to the middleware server either through SAP or when creating a new server configuration in this panel.
- **Server Port:** Port on the server assigned to the selected mobile application
- **Serial Number:** The serial number of the server assigned to the mobile application
- **Search:** Click **Search** once all required and additional search parameters are filled. If the search returns results, they are displayed in the Search Result section. If the search parameters are not fulfilled, a message stating “No data found” displays in the Search Result section.

Search Result

When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. You can also create a new server in this section using **Create**. In this case, it is not necessary to search for an existing server.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet.
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click on the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click on Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Middleware Server Detail - Basic Info

When you select a server in the Search Results table, the fields are populated in the Basic Info section. Click **Change** to modify editable fields. Click **Create** in the Search Result section to create a new server by populating the editable fields. Fields with a red asterisk beside them are mandatory.

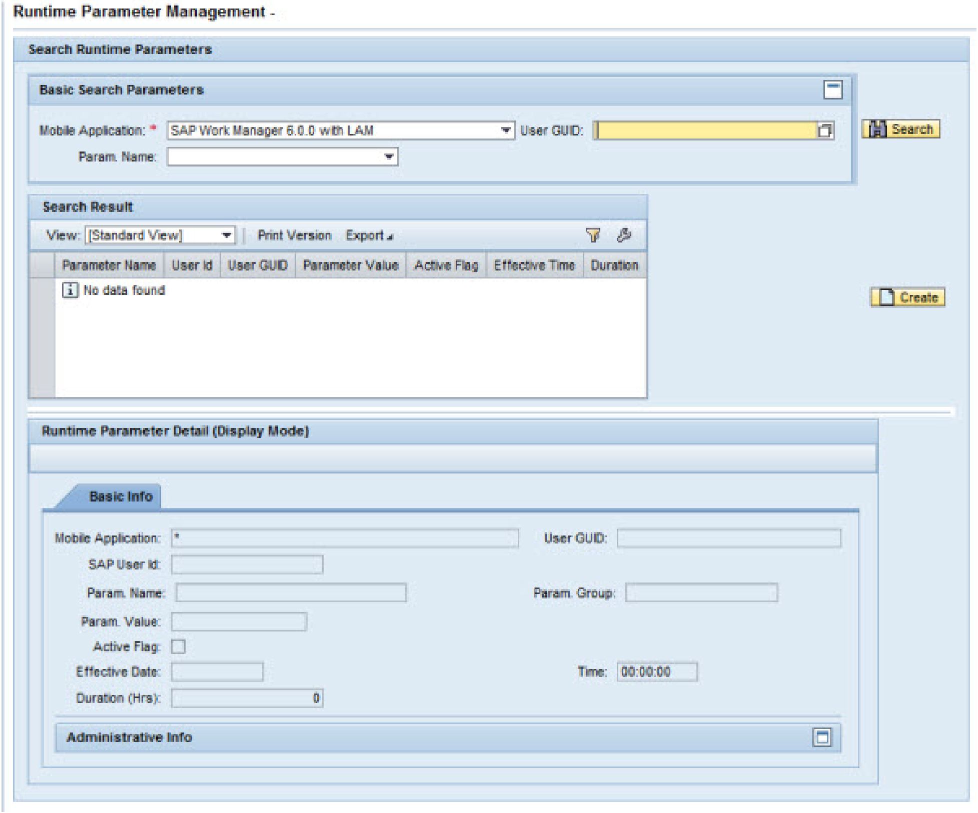
- **Mobile Application:** Name of the mobile application chosen in the Basic Search Parameters section
- **Server GUID:** Server GUID for the specific mobile application chosen. Each server has a different GUID for each mobile application assigned in SAP.
- **Server Name:** Server name chosen from the table in the Search Result section
- **Port:** Port the Agentry application in the SAP Mobile Server has assigned to the specific mobile application chosen
- **Middleware Svr (Server) SerNo (Serial Number):** Customer license number
- **Server URL (FQDN):** Fully qualified domain name that can identify the Agentry application defined in the SAP Mobile Server. In this way, SAP can identify and broadcast to the middleware server without using an IP address. If the IP address of the middleware server changes, use this method as the preferred way to communicate.
- **Target Host:** If a target host is specified, a broadcast of push notification is sent to the IP or URL specified in the Target Host field, instead of to the middleware server’s own IP or URL. Use this if a dedicated middleware or server should receive the broadcast from SAP.
- **Local Outbound Trigger Port:** By default, each middleware server running the same mobile application use the same port number to receive push notification broadcasts from SAP. This is the default in the outbound trigger setting Target Host Port Number. If, for some reason, individual middleware servers listen to push notifications on a different port, specify the port number here. The listed port number in this field will then supercede the port number set in Outbound Trigger Settings.
- **Outbound Trigger URL Type:** Use either the IP address or server URL when broadcasting push notifications.
- **Lock Flag:** If checked, the server will no longer accept users attempting to connect. Set the lock flag to temporarily lock down a server without deleting it from the system.
- **Disabled for Outbound Trigger:** If checked, the Outbound Trigger is unavailable for the selected server.

Administration - Runtime Logging Level Setting

The Runtime Logging Level Setting panel is used to create and manage logging levels for specific mobile applications and user GUIDs. These settings override the logging level setting defined in the Technical Settings panel of the Configuration portal. The logging level setting defined in the Configuration portal applies to all applications running on the framework. However, at times more detail on a specific user or mobile application is desired without making changes to the entire framework.

With the runtime logging level setting, administrators can make dynamic adjustments to the logging level settings. Overrides to the framework settings can be made in two different areas: an individual mobile application or individual users running a specific mobile application. When troubleshooting of the user or the mobile application is complete, the administrator can reset the logging levels back to the default settings specified in the Configuration portal by un-checking the active flag.

Administration - Runtime Parameter Management



Runtime Parameter Management -

Search Runtime Parameters

Basic Search Parameters

Mobile Application: * SAP Work Manager 6.0.0 with LAM

User GUID:

Param. Name:

Search Result

View: [Standard View]

Parameter Name	User Id	User GUID	Parameter Value	Active Flag	Effective Time	Duration
No data found						

Runtime Parameter Detail (Display Mode)

Basic Info

Mobile Application: *

User GUID:

SAP User Id:

Param. Name:

Param. Group:

Param. Value:

Active Flag: □

Effective Date:

Time: 00:00:00

Duration (Hrs): 0

Administrative Info

Basic Search Parameters

Use these fields to search for logging parameters on a specific mobile application, or refine the search even further by the user ID. Once specific logging parameters are found, their information can be changed if necessary.

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application. This is a required field to enable the search function.
- **User GUID:** User GUID associated with the selected mobile application
- **Param Name:** Name of the selected parameter. Parameters are set and configured in the Configuration portal.
- **Search:** Click **Search** once all required and additional search parameters are set. If the search returns results, they are displayed in the Search Result section. If the search parameters are not fulfilled, a message stating “No data found” displays in the Search Result section.

Search Result

When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. You can also create a new runtime parameter in this section using the **Create** button. In this case, it is not necessary to search for an existing runtime parameter.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click on the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click on Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Runtime Parameter Detail - Basic Info

- **Mobile Application:** Name of the mobile application chosen in the Basic Search Parameters section
- **User GUID:** User GUID for the specific mobile application chosen. Each user has a different GUID for each mobile application assigned in SAP.
- **SAP User ID:** User ID assigned in SAP
- **Param. Name:** Name assigned to the logging parameter
- **Parameter Group:** A non-editable field that identifies the system category of the parameter group
- **Param. Value:** A numerical value corresponding to logging levels:
  - 0 - No logging
  - 1 - Abort
  - 2 - Error
  - 3 - Warning
  - 4 - Info
  - 5 - Debug
  - 6 - Trace
- **Active Flag:** Check this box in order to enable the logging levels set through this panel. Uncheck to disable the logging levels set through this panel and to return to the logging levels set through the Technical Settings panel in the Configuration Panel.

- **Effective Date / Time:** Effective date and time of the logging level change
- **Duration (Hrs):** Duration, in hours, of the logging level change

Administration Portal - Settings

All of the screens in the Administration portal contain a Settings link in the Search Result section. Configuring the settings for results display can assist in filtering and sorting user data, especially if there are many user results returned in a search.

To access the Settings section, click on the Settings link at the top right of the Search Result section. This opens the Settings view.

Note: Each of the portal screens requires individual setup of the settings.

- **View:** If multiple views are saved, use the drop-down menu to select the appropriate view to change.

  Use the **Save As** button to save the particular configuration created or modified with a descriptive title. Use the View drop-down menu in the Search Result section to select the desired view. The created view is available in all Settings tabs.

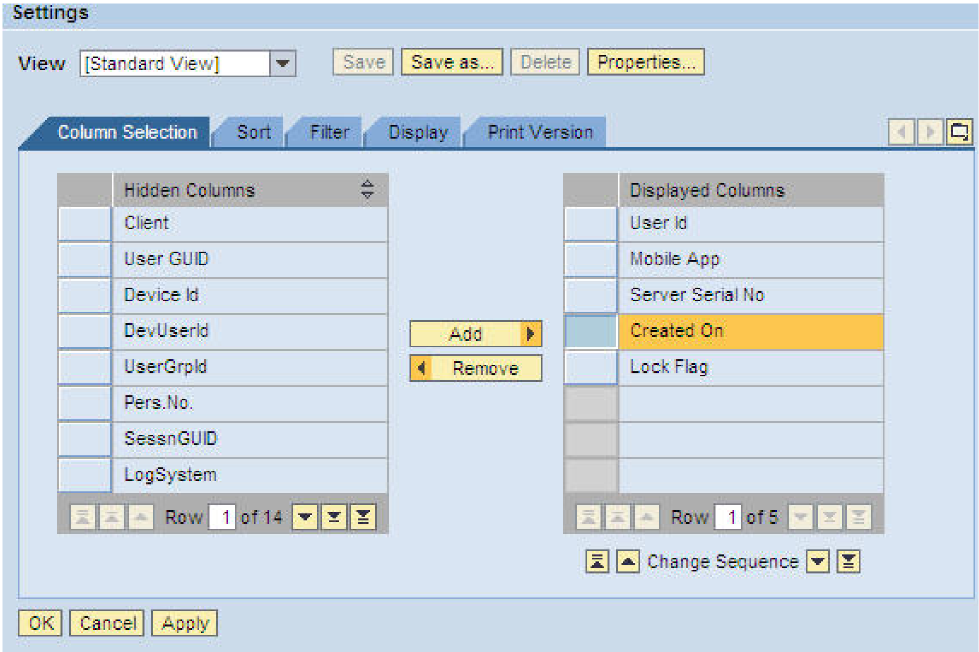
  If a view is no longer needed, select the view in the drop-down menu and click **Delete**.

  To display view properties or to rename the view, click the Properties button.
- **OK:** When done with configuration in the tabs, click **OK** to change the settings in the Search Result table or test the printing configuration. Clicking **OK** does not commit the changes permanently.
- **Cancel:** Click **Cancel** to cancel any changes made during the configuration session.
- **Apply:** Click **Apply** when all configuration is finished and tested. Clicking **Apply** commits the changes so they are available from session to session.

Column Selection

The Column Selection tab configures the columns and their results that are displayed in the search result table.

Management Settings - Column Selection



View: [Standard View]

Hidden Columns
Client
User GUID
Device Id
DevUserId
UserGrpId
Pers.No.
SessnGUID
LogSystem

Row 1 of 14

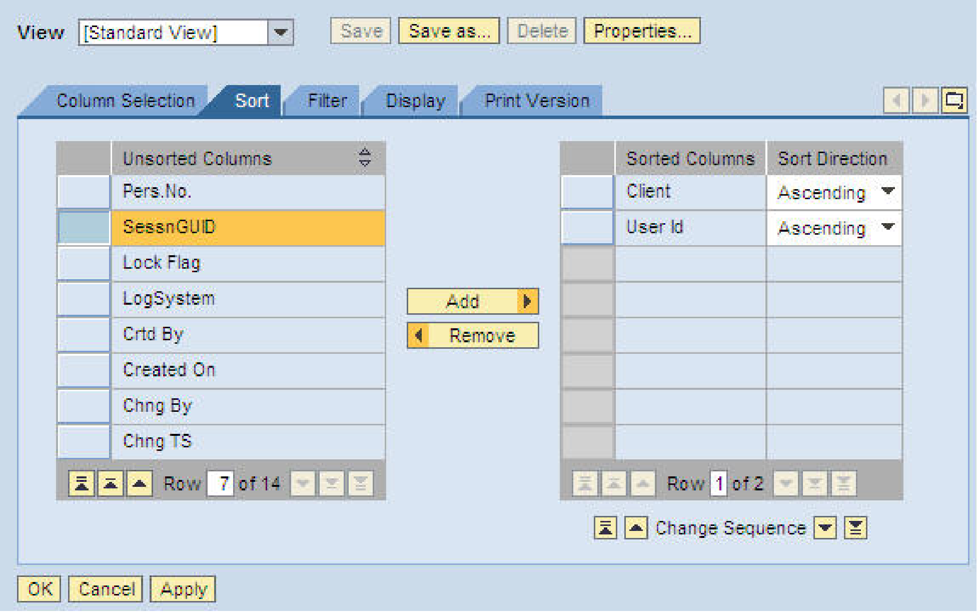
Displayed Columns
User Id
Mobile App
Server Serial No
Created On
Lock Flag

Row 1 of 5

Sort

At times a search performed by using the standard Basic Search Parameters can result in a large amount of information displayed. Use the Sort tab to determine which columns are used to sort the information, depending on the needs of the administrator. The sorting function works in that the first row in the Sorted Columns is the primary source for sorting. Each additional row refines the sorting further. For instance, if a user is active on more than one server, sort by user name and then Server Serial No to display that user’s activity in order of the serial number.

Management Settings - Sort



View [Standard View]

Unsorted Columns
Pers.No.
SessnGUID
Lock Flag
LogSystem
Crtd By
Created On
Chng By
Chng TS

Row 7 of 14

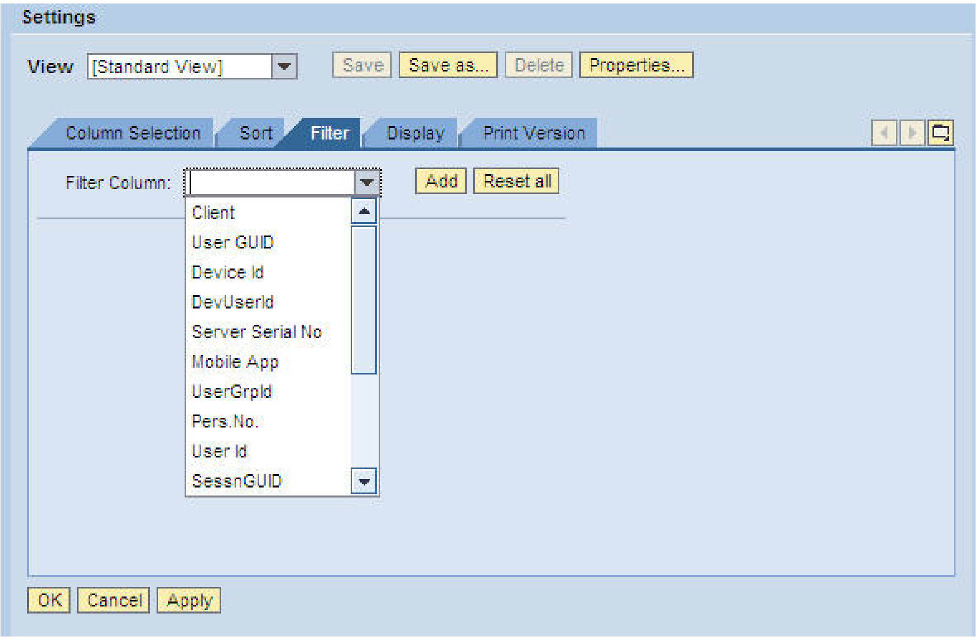
Sorted Columns	Sort Direction
Client	Ascending
User Id	Ascending

Row 1 of 2

Filter

Administrators can create different filters in order obtain a more detailed view from the search results. When the filter is no longer needed on the search results table, click the Delete Filter link.

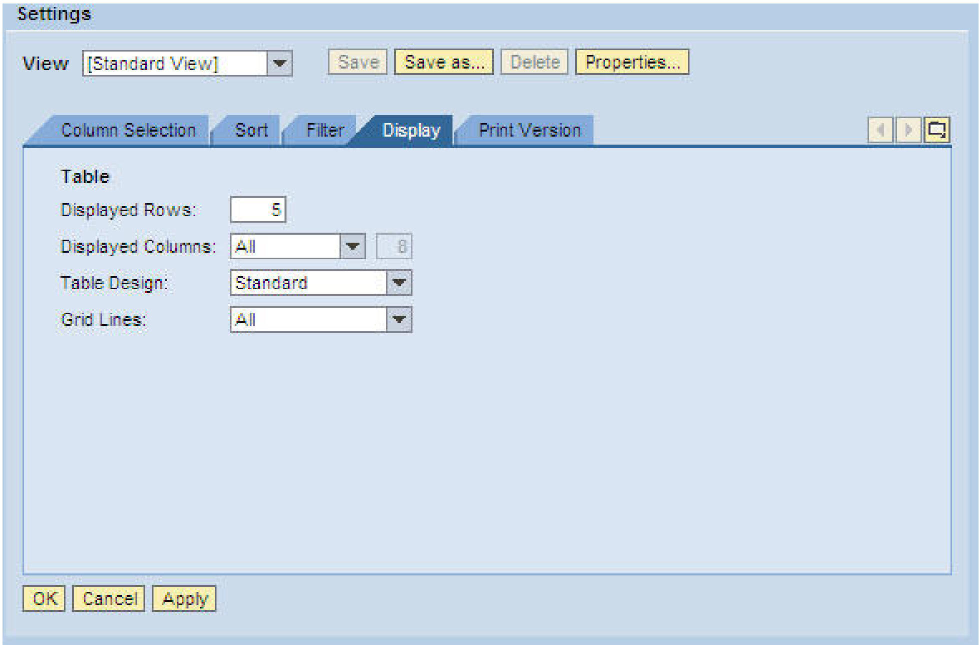
Management Settings - Filter



Display

Use the Display tab to control how the table in the Search Result section displays.

Management Settings - Display



View: [Standard View]

Table

Displayed Rows: 5

Displayed Columns: All 8

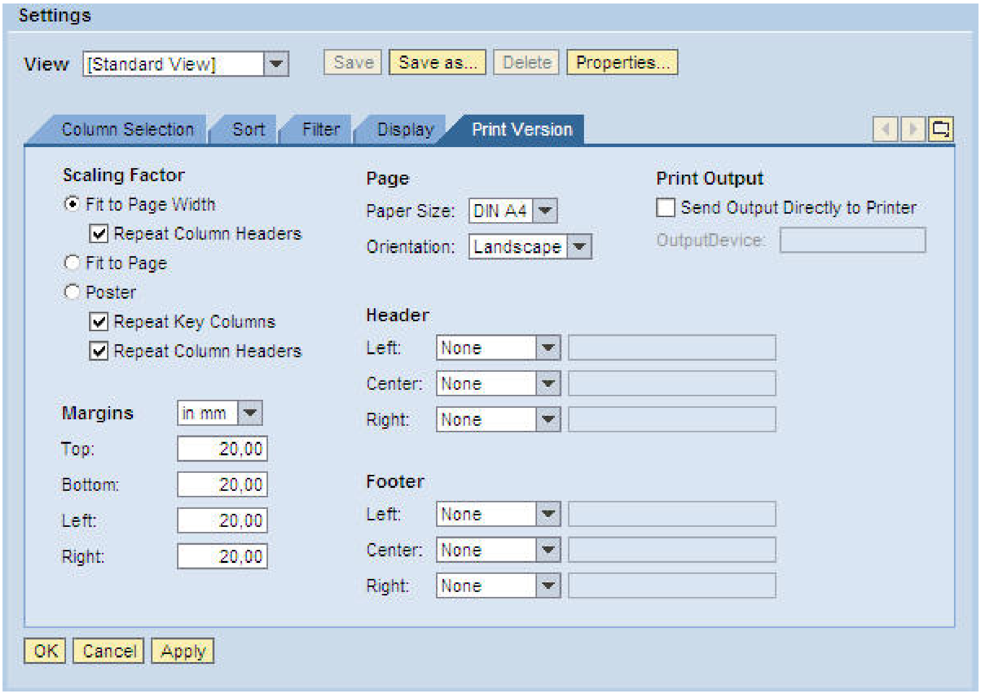
Table Design: Standard

Grid Lines: All

Print Version

The Print Version tab configures specific print settings.

Management Settings - Print Version



View: [Standard View]

Scaling Factor
- ● Fit to Page Width
  - ☑ Repeat Column Headers
- ○ Fit to Page
- ○ Poster
  - ☑ Repeat Key Columns
  - ☑ Repeat Column Headers

Margins: in mm
Top: 20,00
Bottom: 20,00
Left: 20,00
Right: 20,00

Page
Paper Size: DIN A4
Orientation: Landscape

Header
Left: None
Center: None
Right: None

Footer
Left: None
Center: None
Right: None

Print Output
□ Send Output Directly to Printer
OutputDevice:

Administration Portal - Monitoring

The Monitoring panel is used to monitor the following areas:

- User Monitor
- Push Instance Monitor
- Communication Session Monitor
- Object Mobile Status Monitor
- Mobile Transaction History Monitor
- Subscription Queue Monitor

The Monitoring panels are only used for monitoring users and activity. An administrator cannot create new information or change existing information through these panels. To create or modify information, navigate to the appropriate panel in the portal and save the changes.

Administration Portal - Settings

All of the screens in the Administration portal contain a Settings link in the Search Result section. Configuring the settings for results display can assist in filtering and sorting user data, especially if there are many user results returned in a search.

To access the Settings section, click on the Settings link at the top right of the Search Result section. This opens the Settings view.

Note: Each of the portal screens requires individual setup of the settings.

- **View:** If multiple views are saved, use the drop-down menu to select the appropriate view to change.

  Use the **Save As** button to save the particular configuration created or modified with a descriptive title. Use the View drop-down menu in the Search Result section to select the desired view. The created view is available in all Settings tabs.

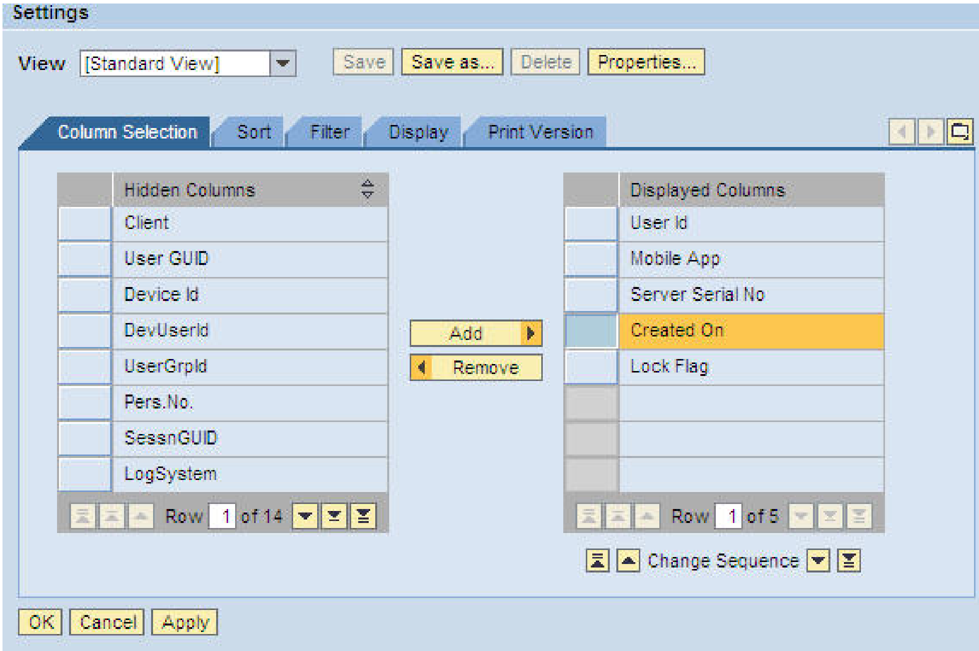
  If a view is no longer needed, select the view in the drop-down menu and click **Delete**.

  To display view properties or to rename the view, click the Properties button.
- **OK:** When done with configuration in the tabs, click **OK** to change the settings in the Search Result table or test the printing configuration. Clicking **OK** does not commit the changes permanently.
- **Cancel:** Click **Cancel** to cancel any changes made during the configuration session.
- **Apply:** Click **Apply** when all configuration is finished and tested. Clicking **Apply** commits the changes so they are available from session to session.

Column Selection

The Column Selection tab configures the columns and their results that are displayed in the search result table.

Management Settings - Column Selection



View [Standard View]

Hidden Columns
Client
User GUID
Device Id
DevUserId
UserGrpId
Pers.No.
SessnGUID
LogSystem

Row 1 of 14

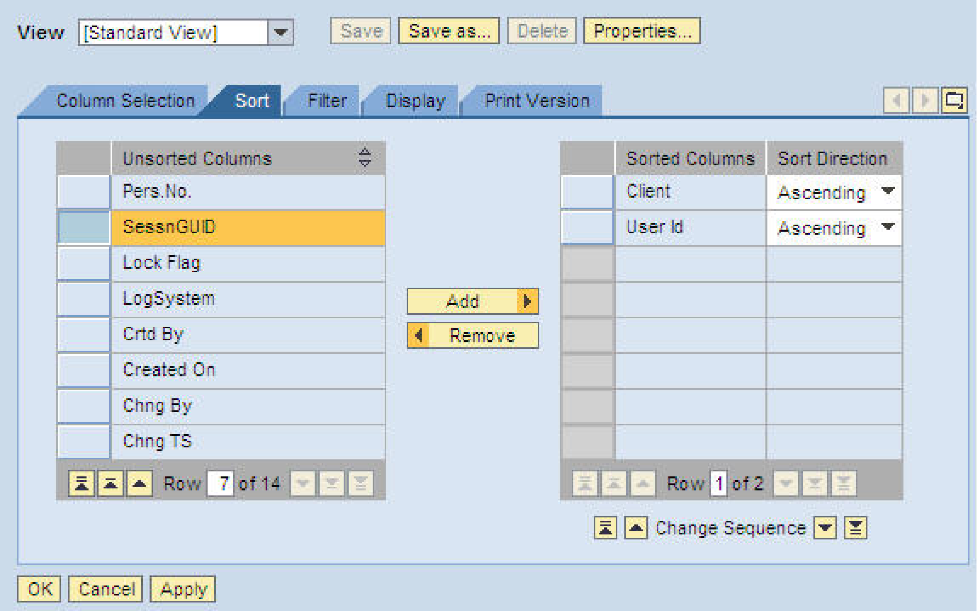
Displayed Columns
User Id
Mobile App
Server Serial No
Created On
Lock Flag

Row 1 of 5

Sort

At times a search performed by using the standard Basic Search Parameters can result in a large amount of information displayed. Use the Sort tab to determine which columns are used to sort the information, depending on the needs of the administrator. The sorting function works in that the first row in the Sorted Columns is the primary source for sorting. Each additional row refines the sorting further. For instance, if a user is active on more than one server, sort by user name and then Server Serial No to display that user’s activity in order of the serial number.

Management Settings - Sort



View: [Standard View]

Unsorted Columns
Pers.No.
SessnGUID
Lock Flag
LogSystem
Crtd By
Created On
Chng By
Chng TS

Row 7 of 14

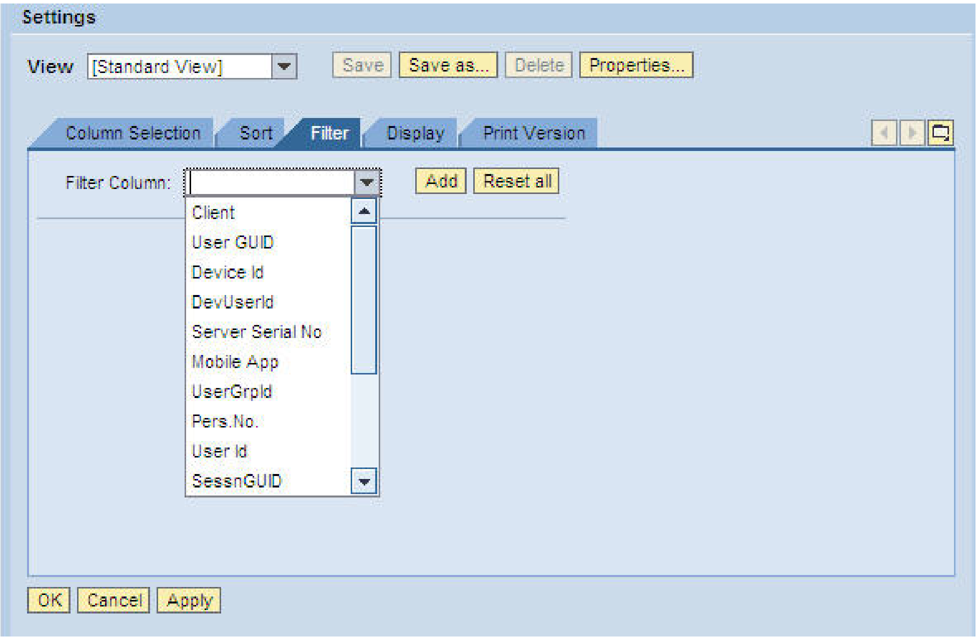
Sorted Columns	Sort Direction
Client	Ascending
User Id	Ascending

Row 1 of 2

Filter

Administrators can create different filters in order obtain a more detailed view from the search results. When the filter is no longer needed on the search results table, click the Delete Filter link.

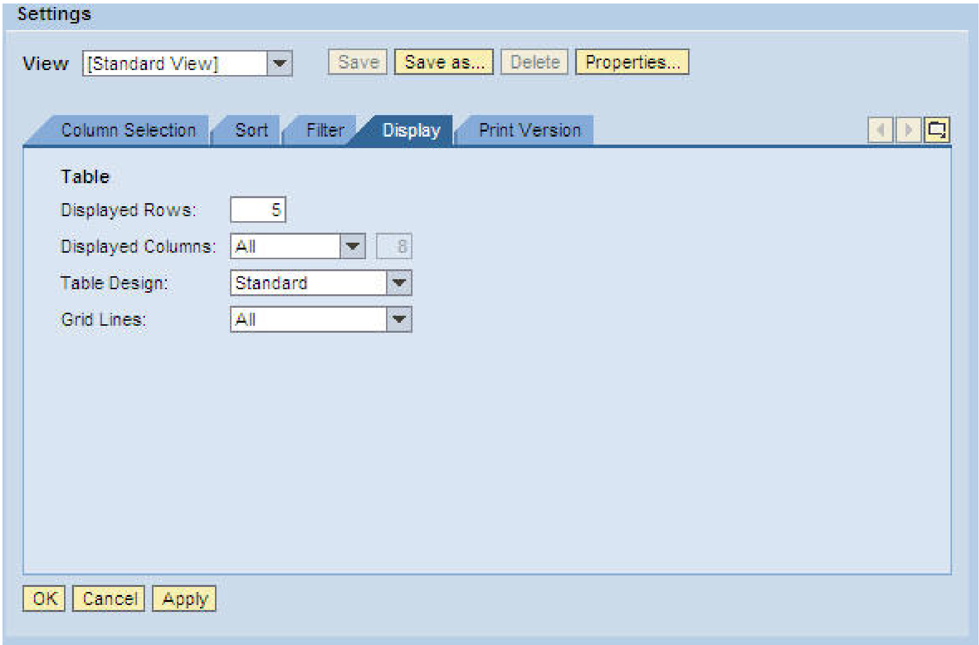
Management Settings - Filter



Display

Use the Display tab to control how the table in the Search Result section displays.

Management Settings - Display



Settings

View [Standard View]

Table

Displayed Rows: 5

Displayed Columns: All 8

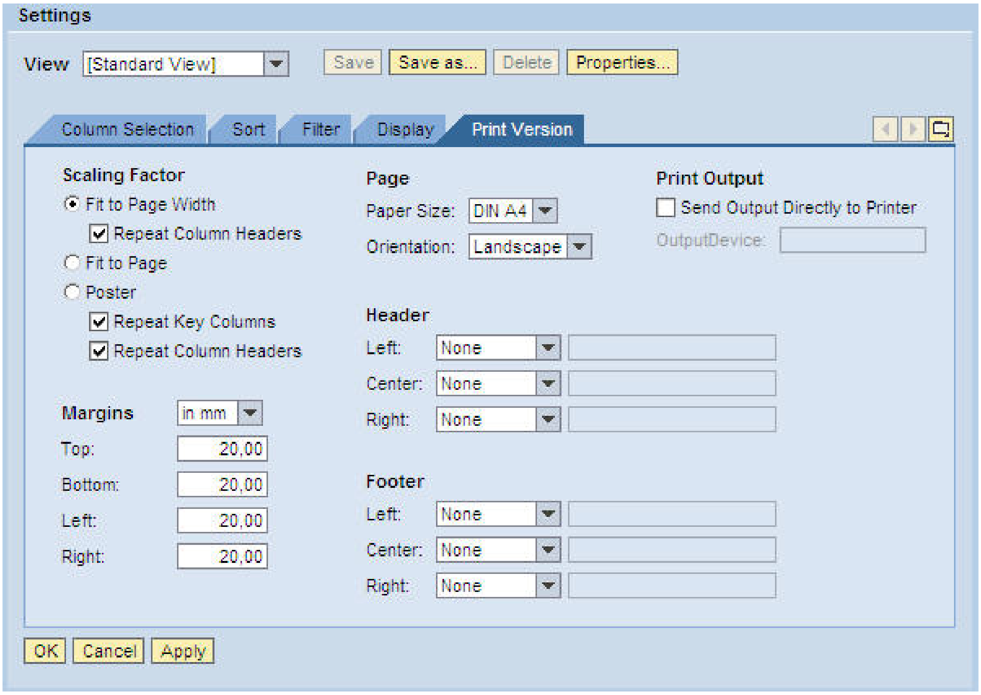
Table Design: Standard

Grid Lines: All

Print Version

The Print Version tab configures specific print settings.

Management Settings - Print Version



Settings

View [Standard View]

Scaling Factor

● Fit to Page Width

☑ Repeat Column Headers

○ Fit to Page

○ Poster

☑ Repeat Key Columns

☑ Repeat Column Headers

Margins in mm

Top: 20,00

Bottom: 20,00

Left: 20,00

Right: 20,00

Page

Paper Size: DIN A4

Orientation: Landscape

Header

Left: None

Center: None

Right: None

Footer

Left: None

Center: None

Right: None

Print Output

□ Send Output Directly to Printer

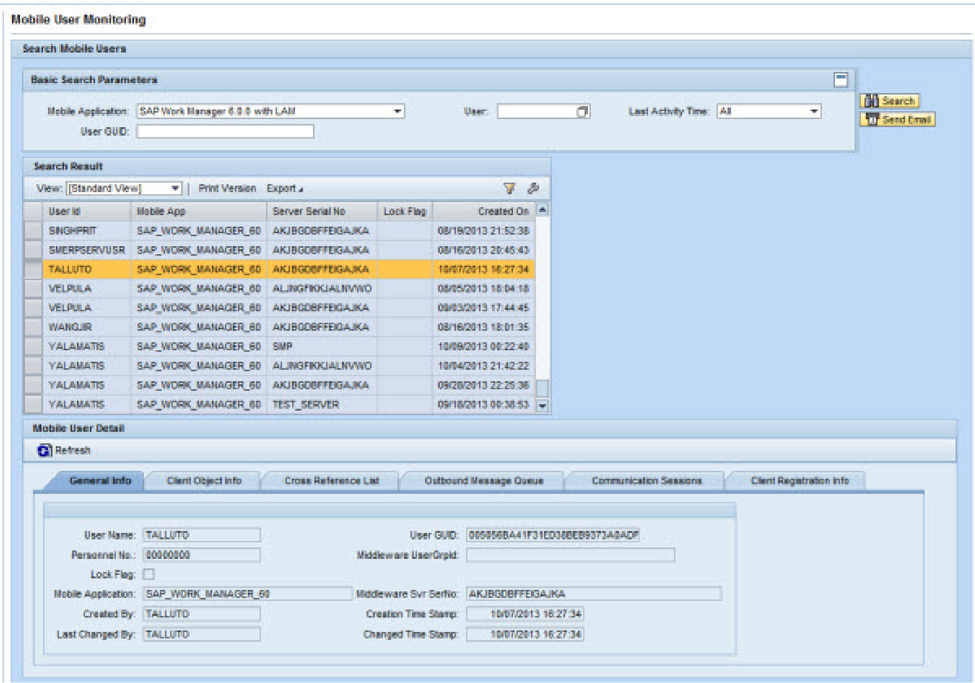
OutputDevice:

Monitoring - User Monitor

The User Monitor panel allows an administrator to locate and view specific users and mobile application use. An administrator can also send specific users email to their mobile devices or HTTP addresses through this panel.

Note: The Client Registration Info tab is for RIM products only and will not be covered in this manual.

Monitoring - Mobile User Monitoring



Mobile Application: SAP Work Manager 6.0.0 with LAM

User:

Last Activity Time: All

User GUID:

View: [Standard View]

User Id	Mobile App	Server Serial No	Lock Flag	Created On
SINGHPRIT	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		08/19/2013 21:52:38
SMERPSERVUSR	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		08/16/2013 20:45:43
TALLUTO	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		10/07/2013 16:27:34
VELPULA	SAP_WORK_MANAGER_60	ALJNGFIKKJALNVWO		08/05/2013 18:04:18
VELPULA	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		09/03/2013 17:44:45
WANGJIR	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		08/16/2013 18:01:35
YALAMATIS	SAP_WORK_MANAGER_60	SMP		10/09/2013 00:22:40
YALAMATIS	SAP_WORK_MANAGER_60	ALJNGFIKKJALNVWO		10/04/2013 21:42:22
YALAMATIS	SAP_WORK_MANAGER_60	AKJBGDBFFEIGAJKA		09/28/2013 22:25:36
YALAMATIS	SAP_WORK_MANAGER_60	TEST_SERVER		09/18/2013 00:38:53

User Name: TALLUTO

Personnel No.: 00000000

Lock Flag: □

Mobile Application: SAP_WORK_MANAGER_60

Created By: TALLUTO

Last Changed By: TALLUTO

User GUID: 005056BA41F31ED38BEB9373A0ADF

Middleware UserGrpId:

Middleware Svr SerNo: AKJBGDBFFEIGAJKA

Creation Time Stamp: 10/07/2013 16:27:34

Changed Time Stamp: 10/07/2013 16:27:34

Basic Search Parameters

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application.
- **User:** SAP user ID. Manually type in the SAP user ID or click the white box icon to the right of the field for optional search methods.
- **Last Activity Time:** Use the drop-down menu to select a window of time. The default is set to All, which is equal to selecting all activity times. All times are available unless the historical logs have been purged in SAP.
- **User GUID:** User GUID assigned to their mobile device.
- **Search:** Click **Search** to initiate the search process. If no results are valid in the search parameters, the message No Data Found displays in the Search Results table. If valid data is returned, it displays in the Search Results table according to the Settings and Filter setup configured by the administrator.
- **Send Email:** Press to start the Send System Emails action.

Search Result

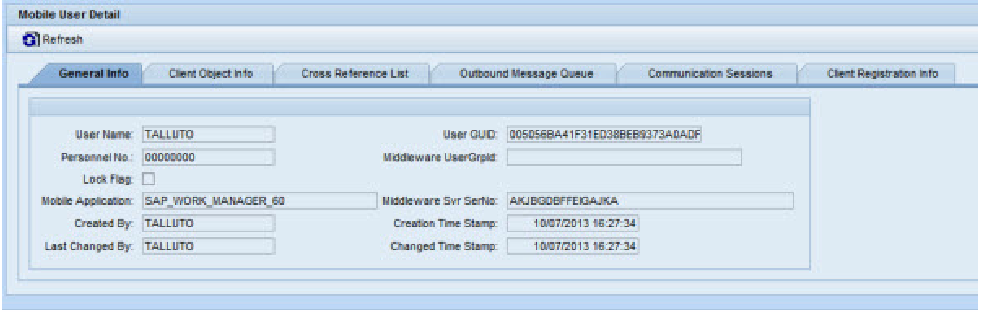
When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. Click on the rectangle to the left of the first column in the desired row to populate the tabs in the Mobile User Detail section below.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Mobile User Detail - General Info

The General Info tab displays detailed information about the specific user highlighted in the Search Results table.

Mobile User Detail - General Info



User Name: TALLUTO
User GUID: 005056BA41F31ED38BEB9373A0ADF
Personnel No.: 00000000
Middleware UserGrpId:
Lock Flag: □
Mobile Application: SAP_WORK_MANAGER_60
Middleware Svr SerNo: AKJBGDBFFEIGAJKA
Created By: TALLUTO
Creation Time Stamp: 10/07/2013 16:27:34
Last Changed By: TALLUTO
Changed Time Stamp: 10/07/2013 16:27:34

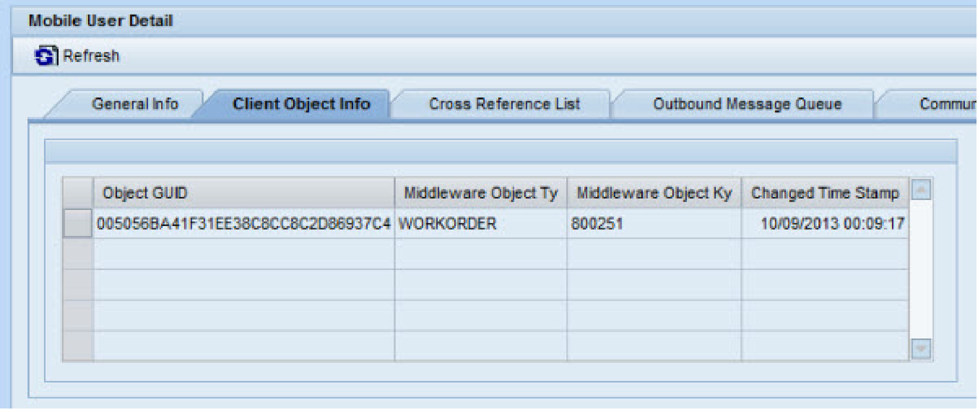
- **User Name:** SAP User ID
- **User GUID:** User GUID for the specific mobile application chosen. Each user has a different GUID for each mobile application assigned in SAP.
- **SAP Personnel Number:** User ID number assigned to the user in SAP
- **Middleware User Group ID:** Group ID assigned to the specific user through the User Management panel in the Administration portal
- **Lock Flag:** If checked in the User Management panel, the user is unable to access the mobile application from the mobile device. This is used if a mobile device is lost or stolen and the application data must be made inaccessible.
- **Mobile Application:** Name of the mobile application chosen in the Basic Search Parameters section

- **Middleware Svr SerNo:** Server serial number of the middleware server that is running the chosen mobile application
- **Created By:** SAP user ID of the person who created the user shown
- **Creation Time Stamp:** Creation time and date of the mobile user
- **Last Changed By:** SAP user ID of the person who modified information of the mobile user shown
- **Changed Time Stamp:** Time and date when the mobile user's information was modified

Mobile User Detail - Client Object Info

The Client Object Info tab displays everything that is contained on the client device associated with a specific mobile user GUID as of the last transmit.

Mobile User Detail - Client Object Info



Object GUID	Middleware Object Ty	Middleware Object Ky	Changed Time Stamp
005056BA41F31EE38C8CC8C2D86937C4	WORKORDER	800251	10/09/2013 00:09:17

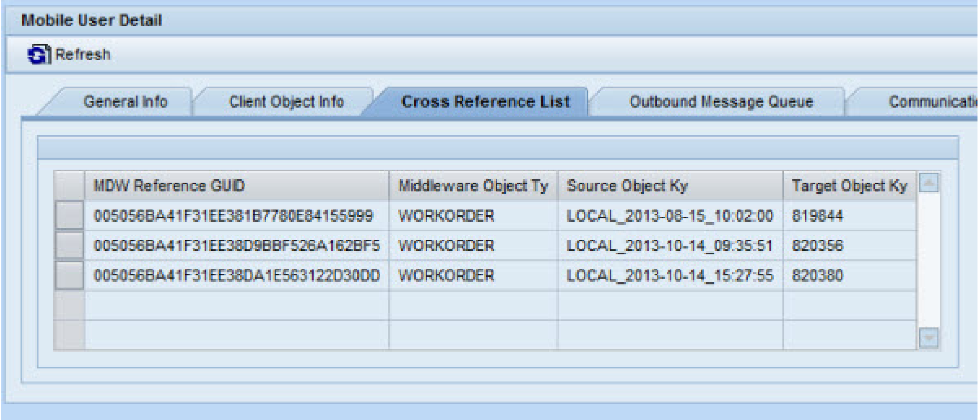
- **Object GUID:** Internal identifier of the record
- **Middleware Object Ty:** Type of object, such as notification or work order
- **Middleware Object Ky:** Object key, or ID of the object
- **Changed Time Stamp:** Time and date the object was entered into the system

Note: For details on how to disable and re-enable items, see the topic "Disabling and Re-enabling Contacts, Transactions, and FactSheets" in the "Common Changes" chapter.

Mobile User Detail - Cross Reference List

When a user creates an object on their mobile device, that object generates a local key. Upon the next transmit, SAP assigns a key to the object and maps the newly-assigned object key to the old local object key that was generated by the client device. The cross reference list displays the object GUIDs for a mobile application on a mobile device and their local object keys mapped to their SAP object keys.

Mobile User Detail - Cross Reference List



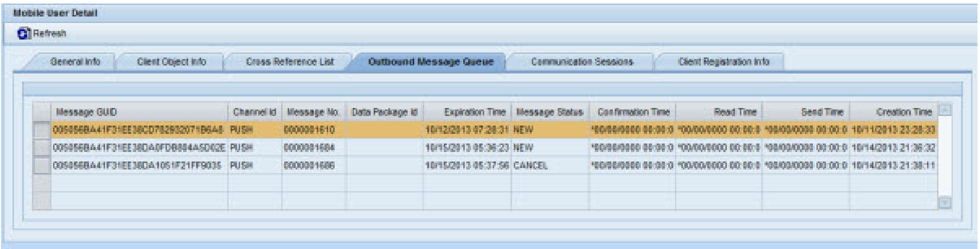
MDW Reference GUID	Middleware Object Ty	Source Object Ky	Target Object Ky
005056BA41F31EE381B7780E84155999	WORKORDER	LOCAL_2013-08-15_10:02:00	819844
005056BA41F31EE38D9BBF526A162BF5	WORKORDER	LOCAL_2013-10-14_09:35:51	820356
005056BA41F31EE38DA1E563122D30DD	WORKORDER	LOCAL_2013-10-14_15:27:55	820380

- **MDW Reference GUID:** Standard internal ID of the record
- **Middleware Object Ty:** Object type, such as work order or notification
- **Source Object Ky:** Object key originating from the client device, which is then mapped to the key assigned to the object by SAP
- **Target Object Ky:** Key that SAP assigns to the object, which is then mapped to the key originating from the client device

Mobile User Detail - Outbound Message Queue

The Outbound Message Queue tab displays all messages in the SAP outbound message queue sent through the client device associated with the user and their mobile application highlighted in the Search Results table.

Mobile User Details - Outbound Message Queue



Message GUID	Channel Id	Message No.	Data Package Id	Expiration Time	Message Status	Confirmation Time	Read Time	Send Time	Creation Time
005056BA41F31EE38CD782932071B6A8	PUSH	0000001610		10/12/2013 07:28:31	NEW	'00/00/0000 00:00:0	'00/00/0000 00:00:0	'00/00/0000 00:00:0	10/11/2013 23:28:33
005056BA41F31EE38DA0FDB884A5D02E	PUSH	0000001684		10/15/2013 05:36:23	NEW	'00/00/0000 00:00:0	'00/00/0000 00:00:0	'00/00/0000 00:00:0	10/14/2013 21:36:32
005056BA41F31EE38DA1051F21FF9035	PUSH	0000001686		10/15/2013 05:37:56	CANCEL	'00/00/0000 00:00:0	'00/00/0000 00:00:0	'00/00/0000 00:00:0	10/14/2013 21:38:11

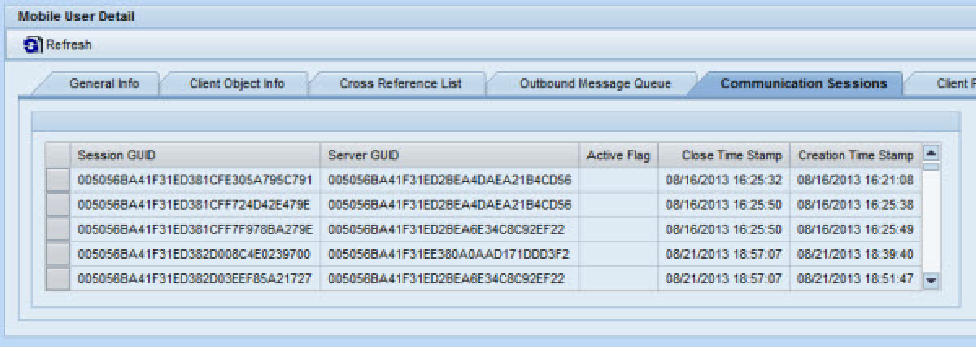
- **Message GUID:** GUID assigned to the message when it was created
- **Channel ID:** Channel for the outbound message that identifies the source of the message
- **Message No.:** Readable identifier of the message as an incremental integer
- **Data Package ID:** The data package identifier, if a data package is associated with the outbound message
- **Expiration Time:** Time set for the message to expire
- **Message Status:** Status of the message. Statuses include New, Read, Send, and Confirm.

- **Confirmation Time:** Time the message was successfully received on the device
- **Read Time:** Time the message was opened, or read, on the device
- **Send Time:** Time message is sent from the mobile device
- **Creation Time Stamp:** Time message was created, or started, on the system

Mobile User Detail - Communication Sessions

The Communication Sessions tab displays all transmissions associated with the user and the mobile application highlighted in the Search Results table.

Mobile User Detail - Communication Sessions



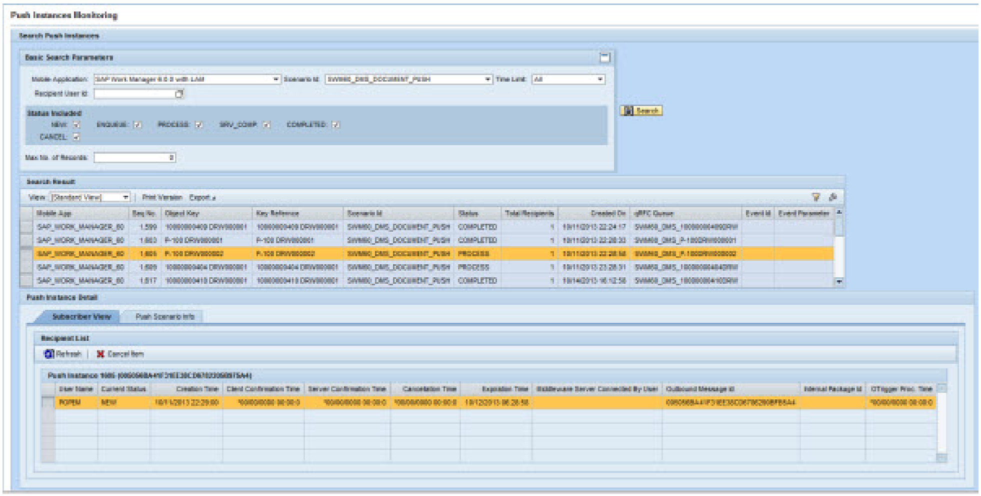
Session GUID	Server GUID	Active Flag	Close Time Stamp	Creation Time Stamp
005056BA41F31ED381CFE305A795C791	005056BA41F31ED2BEA4DAEA21B4CD56		08/16/2013 16:25:32	08/16/2013 16:21:08
005056BA41F31ED381CFF724D42E479E	005056BA41F31ED2BEA4DAEA21B4CD56		08/16/2013 16:25:50	08/16/2013 16:25:38
005056BA41F31ED381CFF7F978BA279E	005056BA41F31ED2BEA6E34C8C92EF22		08/16/2013 16:25:50	08/16/2013 16:25:49
005056BA41F31ED382D008C4E0239700	005056BA41F31EE380A0AAD171DDD3F2		08/21/2013 18:57:07	08/21/2013 18:39:40
005056BA41F31ED382D03EEF85A21727	005056BA41F31ED2BEA6E34C8C92EF22		08/21/2013 18:57:07	08/21/2013 18:51:47

- **Session GUID:** Internal identifier of the session
- **Server GUID:** Identifier of the server on which the user initiated the session
- **Active Flag:** Indicator that a user stayed connected to the server throughout the session. If a client remains connected, the active flag is shown as **1**.
- **Close Time Stamp:** Time and date the session was ended
- **Creation Time Stamp:** Time and date the session was initiated

Monitoring - Push Instance Monitor

The Push Instance Monitor panel allows an administrator to search for and view details of push instances by specific mobile applications.

Monitoring - Push Instance Monitor



Push Instances Monitoring

Search Push Instances

Basic Search Parameters

Mobile Application: SAP Work Manager 6.0.0 with LAM

Scenario Id: SWM60_DMS_DOCUMENT_PUSH

Time Limit: All

Recipient User Id:

Status Included:

NEW: ☑ ENQUEUE: ☑ PROCESS: ☑ SRV_COMP: ☑ COMPLETED: ☑

CANCEL: ☑

Max No. of Records: 0

Search Result

View: [Standard View]

Mobile App	Seq No.	Object Key	Key Reference	Scenario Id	Status	Total Recipients	Created On	qRFC Queue	Event Id	Event Parameter
SAP_WORK_MANAGER_60	1,599	10000000409 DRW000001	10000000409 DRW000001	SWM60_DMS_DOCUMENT_PUSH	COMPLETED	1	10/11/2013 22:24:17	SWM60_DMS_10000000409DRW		
SAP_WORK_MANAGER_60	1,603	P-100 DRW000001	P-100 DRW000001	SWM60_DMS_DOCUMENT_PUSH	COMPLETED	1	10/11/2013 22:28:33	SWM60_DMS_P-100DRW000001		
SAP_WORK_MANAGER_60	1,605	P-100 DRW000002	P-100 DRW000002	SWM60_DMS_DOCUMENT_PUSH	PROCESS	1	10/11/2013 22:28:58	SWM60_DMS_P-100DRW000002		
SAP_WORK_MANAGER_60	1,609	10000000404 DRW000001	10000000404 DRW000001	SWM60_DMS_DOCUMENT_PUSH	PROCESS	1	10/11/2013 23:28:31	SWM60_DMS_10000000404DRW		
SAP_WORK_MANAGER_60	1,617	10000000410 DRW000001	10000000410 DRW000001	SWM60_DMS_DOCUMENT_PUSH	COMPLETED	1	10/14/2013 16:12:58	SWM60_DMS_10000000410DRW		

Push Instance Detail

Recipient List

Push Instance 1605 (0050568A41F31EE38CD67023080F5A4)

User Name	Current Status	Creation Time	Client Confirmation Time	Server Confirmation Time	Cancelation Time	Expiration Time	Middleware Server Connected By User	Outbound Message Id	Internal Package Id	OTrigger Proc. Time
POPEM	NEW	10/11/2013 22:29:00	00/00/0000 00:00:0	00/00/0000 00:00:0	00/00/0000 00:00:0	10/12/2013 06:28:58		0050568A41F31EE38C0678C280BFE5A4		00/00/0000 00:00:0

Basic Search Parameters

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application.
- **Scenario ID:** Type of push, selected from the drop-down menu, to search for in the pushes in the application
- **Time Limit:** Use the drop-down menu to select a window of time. The default is set to All, which is equal to selecting all push instances. All instances are available unless the historical logs have been purged in SAP.
- **Recipient User ID:** User ID receiving the push
- **Status Included:** Status of the push instance on the middleware server. Choosing no status returns all push instances on the mobile application. Multiple statuses can be chosen for the search. The following are the different status levels and their meanings:
  - **NEW -** Set when data in SAP has changed and the system configuration indicates that a push process is needed. No other information is available or set when status is in the NEW stage.
  - **ENQUEUE -** Set when the push process program is running on the data that triggered the push process. During this time, the data is locked so that it cannot be changed during the push process.
  - **PROCESS -** Standard status. The push process agent processes the instance in the push register and determines the proper recipients of the push data. The push data have been prepared for each recipient in the outbound queue in order for Agentry to pick it up.
  - CANCEL - Set for a push instance if there are subsequent newer push instances in the push register for the same work order. In this case, only the last push instance is processed in order to prevent multiple pushes for the same work order sent to the same mobile device.

  - **COMPLETED -** Set when either no recipient is determined or all push recipients for the push have CLNT_CONF status with respect to the recipient's push message.
  - **SRV_COMP -** Set for a push instance when all recipients have SRV_CONF status with respect to the individual push recipient's push message. If there are multiple push messages for the same work order and the recipients are waiting for Agentry to pick up the work orders, only the latest push event is sent to Agentry and the rest are set to SRV_CONF automatically by the push fetch BAPI. This prevents multiple copies of the same work order sent to each client.
- **Max. No. of Records:** Default is set to 2,000. Type in maximum number of records returned.
- **Search:** Click **Search** to initiate the search process. If no results are valid in the search parameters, the message No Data Found displays in the Search Results table. If valid data is returned, it displays in the Search Results table according to the Settings and Filter setup configured by the administrator.

Search Result

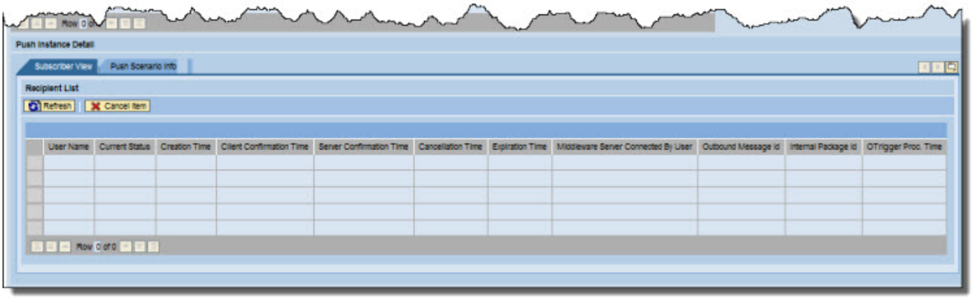
When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. Click on the rectangle to the left of the first column in the desired row to populate the tabs in the Push Instance Detail section below.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Push Instance Monitor - Subscriber View

The Subscriber View tab on the Push Instance Monitor panel provides a recipient list containing all users and detailed information about the push instance associated with their user IDs.

Push Instance Detail - Subscriber View



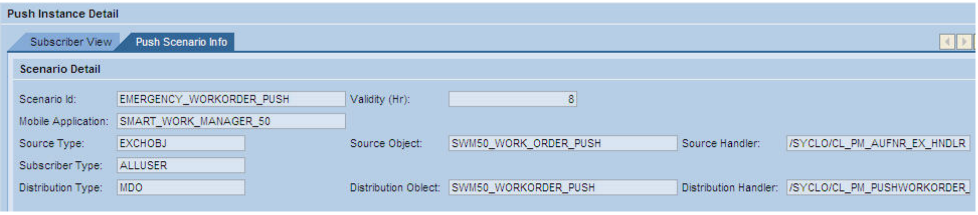
	User Name	Current Status	Creation Time	Client Confirmation Time	Server Confirmation Time	Cancellation Time	Expiration Time	Middleware Server Connected By User	Outbound Message Id	Internal Package Id	OTrigger Proc. Time

- **User Name:** Name of user who received the push instance
- **Current Status:** Status of the push instance at the time the search was performed
- **Creation Time:** Time the push instance was created
- **Client Confirmation Time:** Time and date that the client confirms it has received the push
- **Server Confirmation Time:** Time and date when the middleware server confirmed it received the push in its cache
- **Cancellation Time:** Time and date of a manual or automatic cancellation of a push. For example, if a new push is created, the old pushes can be manually cancelled, as they are now out of date.
- **Expiration Time:** Time and date of a push expiration. If a client does not pick up a push before it expires, it is shown here.
- **Middleware Server Connection By User:** Name of the Agentry server that the user is connected to.
- **Outbound Message ID:** Message ID
- **Internal Package ID:** Data package ID
- **OTrigger Proc. Time:** Time and date when the outbound trigger was processed

Push Instance Monitor - Push Scenario Info

The Push Scenario Info tab on the Push Instance Monitor panel provides technical information about the selected push instance highlighted in the Search Result table. Information displayed in this tab is created and configured using the Configuration portal, Push Scenario Definitions panel.

Push Instance Detail - Push Scenario Info



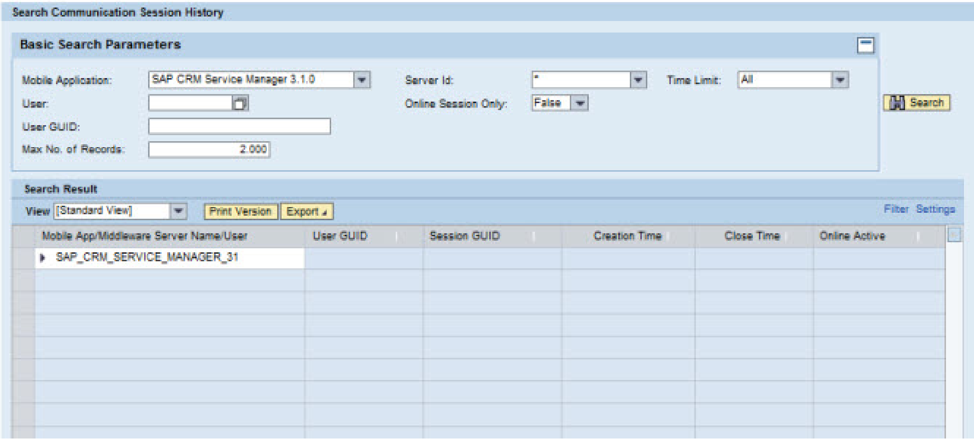
Scenario Id: EMERGENCY_WORKORDER_PUSH
Validity (Hr): 8
Mobile Application: SMART_WORK_MANAGER_50
Source Type: EXCHOBJ
Source Object: SWM50_WORK_ORDER_PUSH
Source Handler: /SYCLO/CL_PM_AUFNR_EX_HNDLR
Subscriber Type: ALLUSER
Distribution Type: MDO
Distribution Oblect: SWM50_WORKORDER_PUSH
Distribution Handler: /SYCLO/CL_PM_PUSHWORKORDER_

- **Scenario ID:** Name of the push instance scenario
- **Validity (Hr):** Hours that the push scenario remains valid after the initial push to clients
- **Mobile Application:** Name of the mobile application where the push instance resides
- **Source Type:** Type of source object associated with the push scenario
- **Source Object:** Name of the mobile data object that contains the push scenario. The source object is created or modified in the Mobile Data Object configuration panel in the Configuration portal.
- **Source Handler:** Class handler associated with the source object for the push scenario
- **Subscriber Type:** Corresponds to the Subscriber Type in the Subscriber Settings section of the Push Scenario Definition panel in the Configuration portal.
- **Distribution Type:** Corresponds to the Distribution Type in the Distribution Settings section of the Push Scenario Definition panel in the Configuration portal. MDO, or mobile data object, is the only setting available.
- **Distribution Object:** Name of the distribution object that is set in the Distribution Settings section of the Push Scenario Definition panel in the Configuration portal.
- **Distribution Handler:** Name of the distribution handler associated with the mobile data object contained in the push instance. The distribution handler is selected or changed in the Mobile Data Object Configuration panel, General Setting tab in the Configuration portal.

Monitoring - Communication Session Monitor

The Communication Session Monitor panel provides an administrator a detailed history of communications on a specific mobile application. The communications monitor can provide a history for everything in the system, unless the history has been purged from SAP.

Monitoring - Communication Session History Monitor



Search Communication Session History

Basic Search Parameters

Mobile Application: SAP CRM Service Manager 3.1.0
User:
User GUID:
Max No. of Records: 2.000
Server Id: *
Online Session Only: False
Time Limit: All

Search Result

View [Standard View]

Mobile App/Middleware Server Name/User	User GUID	Session GUID	Creation Time	Close Time	Online Active
SAP_CRM_SERVICE_MANAGER_31					

Basic Search Parameters

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application.
- **Server ID:** Name of the middleware server
- **Time Limit:** Use the drop-down menu to select a window of time. The default is set to All, which is equal to selecting all communication sessions. All sessions are available unless the historical logs have been purged in SAP.
- **User:** SAP user ID. Manually type in the SAP user ID or click the white box icon to the right of the field for optional search methods.
- **Online Session Only:** Select from True or False:
  - True - Picks up online-only communication sessions
  - False - Picks up both online and offline communication sessions
- **User GUID:** User GUID assigned to their mobile device.
- **Max. No. of Records:** Default is set to 2,000. Type in maximum number of records returned.

Search Result

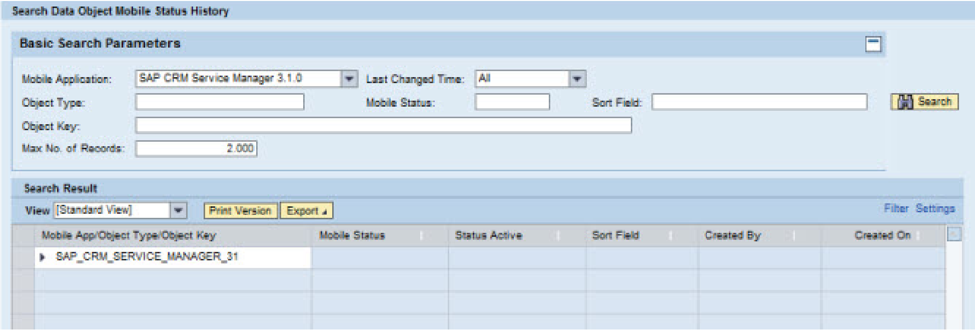
When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. Click on the triangle to the left of the initial search result row to display the tree of results.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Monitoring - Object Mobile Status Monitor

The Data Object Mobile Status Monitor provides the administrator a history of SAP objects affected by a mobile user’s actions. The actions are recorded in SAP and the monitor panel provides a user-friendly way to access the information.

Monitoring - Data Object Mobile Status History Monitor



Mobile Application: SAP CRM Service Manager 3.1.0
Last Changed Time: All
Object Type:
Mobile Status:
Sort Field:
Object Key:
Max No. of Records: 2.000

View [Standard View]

Mobile App/Object Type/Object Key	Mobile Status	Status Active	Sort Field	Created By	Created On
▸ SAP_CRM_SERVICE_MANAGER_31					

Basic Search Parameters

- **Mobile Application:** Name of the mobile application. Use the drop-down menu to choose the correct mobile application.
- **Last Changed Time:** Use the drop-down menu to select a window of time. The default is set to All, which is equal to selecting all mobile object change times. All logs are available unless the they have been purged in SAP.
- **Object Type:** Type of object defined in the mobile application definition.
- **Mobile Status:** Historical view of status changes for the mobile object
- **Sort Field:** Additional information about the object, if any is provided
- **Object Key:** Internal ID of the object
- **Max. No. of Records:** Default is set to 2,000. Type in maximum number of records returned.

Search Result

When you click **Search** in the Basic Search Parameters section, the Search Result table is populated. Click on the triangle to the left of the initial search result row to display the tree of results.

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.

- **Search Result Table:** Table that displays the search results. Columns are dependent on the configuration built through the Settings link above the table. See the section *Administration Portal - Management Settings* for details on specific settings.

Administration Portal - Statistics

The Statistics panel is used to view statistics in the following three areas:

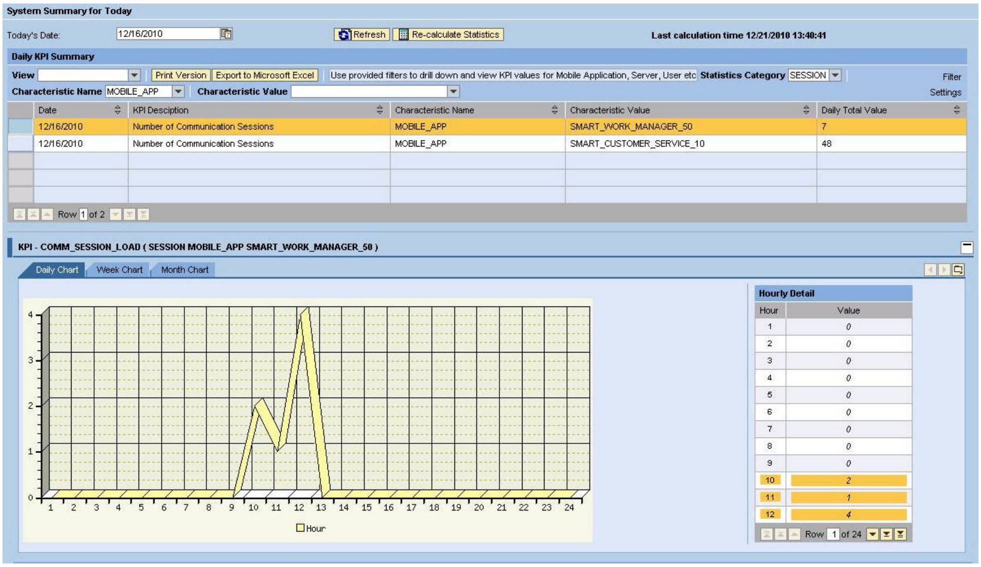
- Communication Session Statistics
- Application BAPI Wrapper Call Statistics
- Push Scenario Statistics

The statistics panels display graphical views of various key performance indicator (KPI) data. The Statistics panels are view-only. An administrator cannot create new information or change existing information through these panels.

There are three tabs for each statistic, each allowing a different graphical view: Daily, Weekly, and Monthly. The following examples depict a representation of each type of graph.

The Daily Chart tab shows a graphical representation of the chosen statistics broken down into hours.

Figure 16: Daily KPI Chart



System Summary for Today

Today's Date: 12/16/2010

Last calculation time 12/21/2010 13:40:41

Daily KPI Summary

View:

Use provided filters to drill down and view KPI values for Mobile Application, Server, User etc

Statistics Category: SESSION

Characteristic Name: MOBILE_APP

Characteristic Value:

Date	KPI Desciption	Characteristic Name	Characteristic Value	Daily Total Value
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_WORK_MANAGER_50	7
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_CUSTOMER_SERVICE_10	48

Row 1 of 2

KPI - COMM_SESSION_LOAD (SESSION MOBILE_APP SMART_WORK_MANAGER_50)

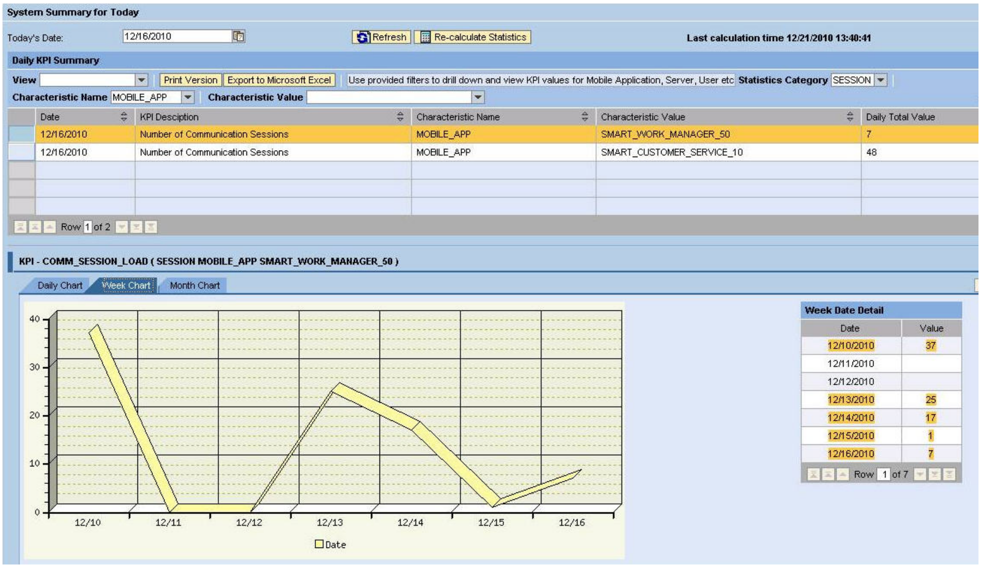
Hourly Detail

Hour	Value
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	2
11	1
12	4

Row 1 of 24

The Weekly Chart tab shows a graphical representation of the chosen statistics broken down into days, with a total of seven days.

Figure 17: Weekly KPI Chart



System Summary for Today

Today's Date: 12/16/2010 | Refresh | Re-calculate Statistics | Last calculation time 12/21/2010 13:40:41

Daily KPI Summary

View | Print Version | Export to Microsoft Excel | Use provided filters to drill down and view KPI values for Mobile Application, Server, User etc **Statistics Category** SESSION

Characteristic Name MOBILE_APP **Characteristic Value**

Date	KPI Desciption	Characteristic Name	Characteristic Value	Daily Total Value
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_WORK_MANAGER_50	7
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_CUSTOMER_SERVICE_10	48

Row 1 of 2

KPI - COMM_SESSION_LOAD (SESSION MOBILE_APP SMART_WORK_MANAGER_50)

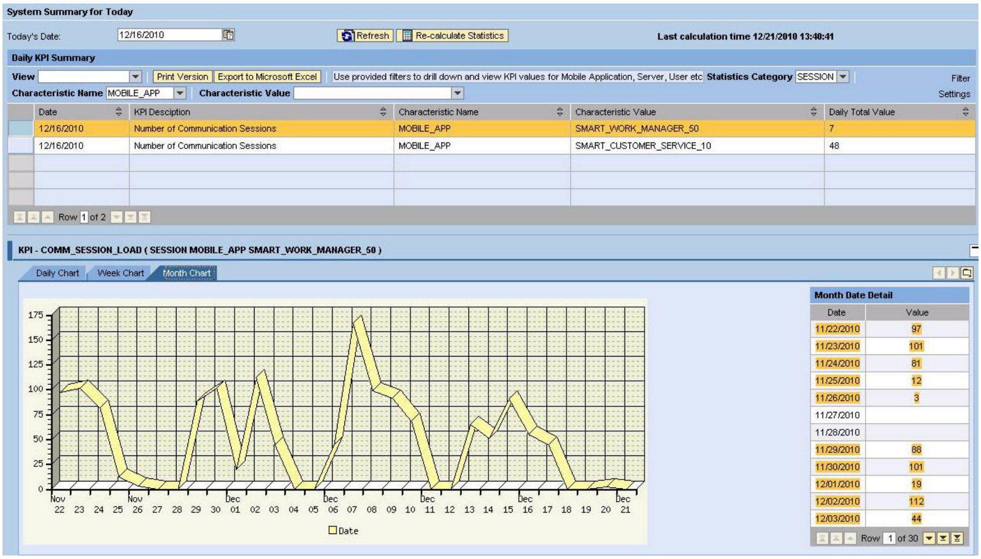
Week Date Detail

Date	Value
12/10/2010	37
12/11/2010	
12/12/2010	
12/13/2010	25
12/14/2010	17
12/15/2010	1
12/16/2010	7

Row 1 of 7

The Monthly Chart tab shows a graphical representation of the chosen statistics broken down into days, with a total of the amount of days in the selected month.

Figure 18: Monthly KPI Chart



System Summary for Today

Today's Date: 12/16/2010 | Refresh | Re-calculate Statistics | Last calculation time 12/21/2010 13:40:41

Daily KPI Summary

View | Print Version | Export to Microsoft Excel | Use provided filters to drill down and view KPI values for Mobile Application, Server, User etc **Statistics Category** SESSION | Filter Settings

Characteristic Name MOBILE_APP **Characteristic Value**

Date	KPI Desciption	Characteristic Name	Characteristic Value	Daily Total Value
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_WORK_MANAGER_50	7
12/16/2010	Number of Communication Sessions	MOBILE_APP	SMART_CUSTOMER_SERVICE_10	48

Row 1 of 2

KPI - COMM_SESSION_LOAD (SESSION MOBILE_APP SMART_WORK_MANAGER_50)

Month Date Detail

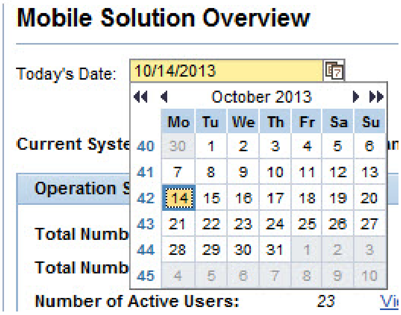
Date	Value
11/22/2010	97
11/23/2010	101
11/24/2010	81
11/25/2010	12
11/26/2010	3
11/27/2010	
11/28/2010	
11/29/2010	88
11/30/2010	101
12/01/2010	19
12/02/2010	112
12/03/2010	44

Row 1 of 30

Retrieving and Recalculating Statistics

Each panel in the Statistics section automatically displays the current date's statistics. If statistics are needed for prior dates, use the following procedure.

1. Click the calendar icon to the right of the date field and choose the desired date.



Mobile Solution Overview

Today's Date: 10/14/2013

October 2013

	Mo	Tu	We	Th	Fr	Sa	Su
40	30	1	2	3	4	5	6
41	7	8	9	10	11	12	13
42	14	15	16	17	18	19	20
43	21	22	23	24	25	26	27
44	28	29	30	31	1	2	3
45	4	5	6	7	8	9	10

Current Syste

Operation S

Total Numb

Total Numb

Number of Active Users: 23 Vi

2. Click the **[Refresh]** button to refresh the statistics for the chosen date.

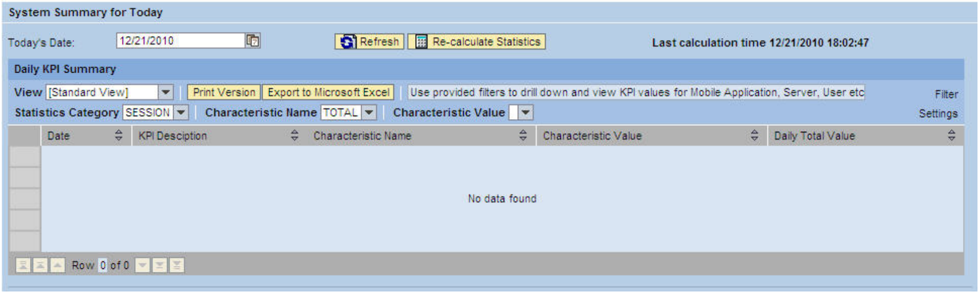
   The new statistics graph displays in the KPI table.

3. If the current date is selected, to check if the statistics have changed, click **[Re-calculate Statistics]** in order to recalculate.

Statistics - Communication Session Statistics

The KPIs available through the Communication Session Statistics panel display information on the system operation status for the chosen date.

Statistics - Communication Session Statistics



System Summary for Today

Today's Date: 12/21/2010

Last calculation time 12/21/2010 18:02:47

Daily KPI Summary

View [Standard View]

Use provided filters to drill down and view KPI values for Mobile Application, Server, User etc

Statistics Category SESSION   Characteristic Name TOTAL   Characteristic Value

Date	KPI Desciption	Characteristic Name	Characteristic Value	Daily Total Value

No data found

Row 0 of 0

Daily KPI Summary

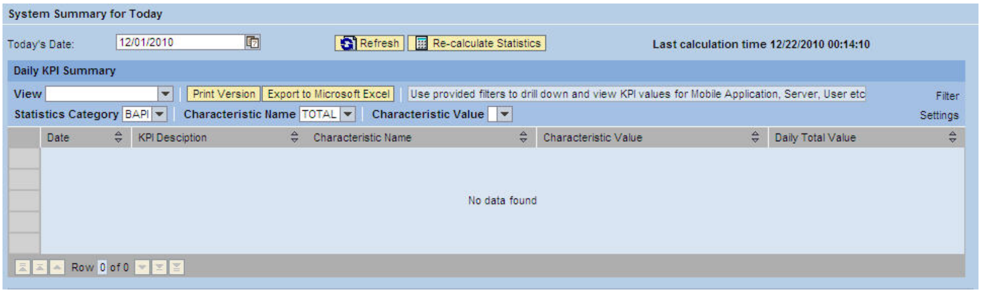
- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.

- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Statistics Category/Characteristic Name/Characteristic Value:** Use to filter statistics further. Select from the choices available through the dropdown menus. If the field is blank, no choices are available.
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Daily KPI Summary table:** Contains rows that display which statistics are available for the selected date. Highlight a row to display that graph. The graphs available are as follows:
    - **KPI - COMM_SESSION_LOAD:** The KPI - COMM_SESSION_LOAD is a graphical representation of the system load for the date chosen. The more sessions there are per hour, the heavier the load on the system, which could cause performance issues. Session load statistics are based on the number of sessions connecting; it does not take into account duration of the sessions.
    - **KPI - CONNECTING_USER_COUNT:** The KPI - CONNECTING_USER_COUNT is a graphical representation of the total users connecting to the system for the date chosen. Data gathered for this graph does not take into account a single user logging into multiple sessions; it only counts unique user IDs connecting to the system.

Statistics - Application BAPI Wrapper Call Statistics

The KPIs available through the Application BAPI Wrapper Call Stats panel display information on the number of BAPI calls for the chosen date. This statistical information can be useful in troubleshooting end user behaviors and use of the mobile application(s).

Statistics - Application BAPI Wrapper Call Statistics



System Summary for Today

Today's Date: 12/01/2010

Last calculation time 12/22/2010 00:14:10

Daily KPI Summary

View

Use provided filters to drill down and view KPI values for Mobile Application, Server, User etc

Statistics Category BAPI | Characteristic Name TOTAL | Characteristic Value

Date	KPI Desciption	Characteristic Name	Characteristic Value	Daily Total Value

No data found

Row 0 of 0

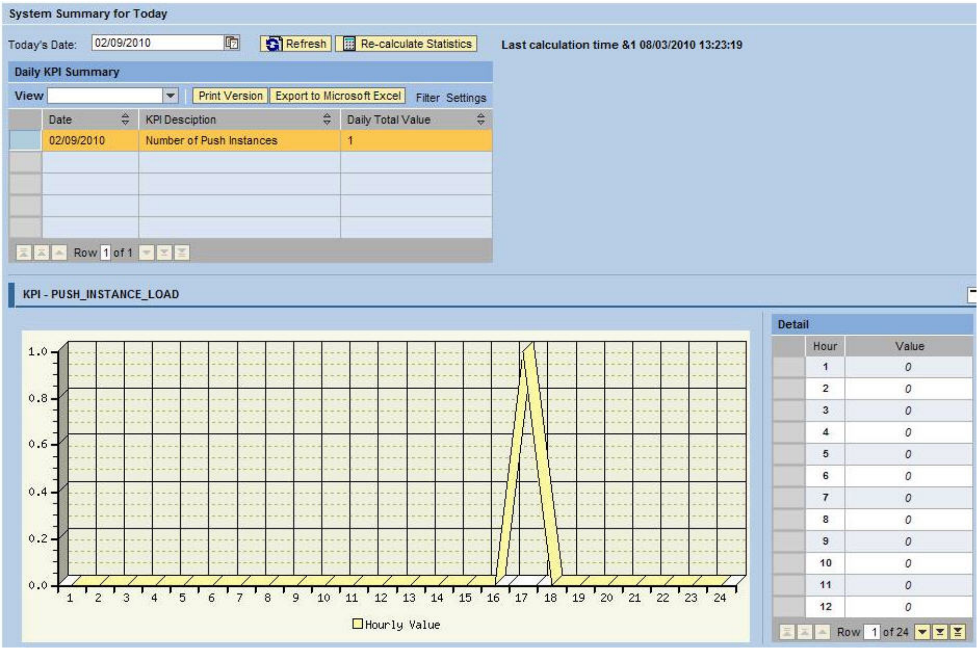
Daily KPI Summary

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Statistics Category/Characteristic Name/Characteristic Value:** Use to filter statistics further. Select from the choices available through the drop-down menus. If field is blank, no choices are available.
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.
- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Daily KPI Summary table:** Contains rows that display which statistics are available for the selected date. Highlight a row to display that graph. The graphs available are as follows:
  - **KPI - CRT_BAPI_CALL_COUNT:** The KPI - CRT_BAPI_CALL_COUNT is a graphical representation of how many BAPI calls were detected. The more calls there are per hour, the heavier the load on the backend system, which could cause performance issues. BAPI call statistics are based on the number of calls; it does not take into account what is called during the transaction process.
  - **KPI - ERROR_BAPI_CALL_COUNT:** The KPI - ERROR_BAPI_CALL_COUNT is a graphical representation of how many BAPI calls returned with an error message. These statistics can provide a good indicator to begin troubleshooting performance issues with the system, if necessary.
  - **KPI - GET_BAPI_CALL_COUNT:** The KPI - GET_BAPI_CALL_COUNT is a graphical representation of how many GET BAPI calls were made.
  - **KPI - DEL_BAPI_CALL_COUNT:** The KPI - DEL_BAPI_CALL_COUNT is a graphical representation of how many DELETE BAPI calls were made.
  - **KPI - UPD_BAPI_CALL_COUNT:** The KPI - UPD_BAPI_CALL_COUNT is a graphical representation of how many UPDATE BAPI calls were made.
  - **KPI - INITSYNC_BAPI_CALL_COUNT:** The KPI - INITSYNC_BAPI_CALL_COUNT is a graphical representation of initial synchronizations (i.e., all data is transmitted to a mobile device, not just data changed since last transmit) on the system. If a large amount of initial synchronization transmissions are present on the system for a specific time period, it could indicate issues with the system.
  - **KPI - ALL_BAPI_CALL_COUNT:** The KPI - ALL_BAPI_CALL_COUNT is a graphical representation of all BAPI calls made during the time period.

Statistics - Push Scenario Statistics

The KPIs available through the Push Scenario Statistics panel display the amount of pushes that occurred for the chosen date on an hourly basis. This statistical information can be useful in determining peak loads and to assess factors that drive push scenarios.

Statistics - Push Scenario Statistics



System Summary for Today

Today's Date: 02/09/2010

Last calculation time &1 08/03/2010 13:23:19

Daily KPI Summary

Date	KPI Desciption	Daily Total Value
02/09/2010	Number of Push Instances	1

Row 1 of 1

KPI - PUSH_INSTANCE_LOAD

Detail

Hour	Value
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0

Row 1 of 24

Daily KPI Summary

- **View:** If the administrator sets up different views using the Settings link, the drop-down menu will display those view names. Select a different view for specific data needs.
- **Print Version:** If enabled, creates a PDF version of the data in the Search Results table.
- **Export to Microsoft Excel:** Exports all data in the Search Results table to an Excel spreadsheet
- **Statistics Category/Characteristic Name/Characteristic Value:** Use to filter statistics further. Select from the choices available through the drop-down menus. If field is blank, no choices are available.
- **Filter/Delete Filter:** If the Filter tab is utilized in the Settings window accessed by the Settings link, click the Filter link to display filter choices in order to further filter the data displayed. If a filter is in use, click Delete Filter to remove the filter and display all data returned by the initial search performed.

- **Settings:** Click this link to display a Settings panel in order to modify how search results are displayed. See the section on changing Administrator settings for more details.
- **Daily KPI Summary table:** Contains rows that display which statistics are available for the selected date. Highlight a row to display that graph. The graphs available are as follows:
  - **KPI - PUSH_INSTANCE_LOAD:** The KPI - PUSH_INSTANCE_LOAD is a graphical representation of total number of pushes on the system.
  - **KPI - PUSH _INST_CANCEL_COUNT:** The KPI - PUSH_INSTANCE_CANCEL_COUNT is a graphical representation of total number of push instances with a CANCEL status.
  - **KPI - PUSH _AVG_TIME_CMPLETE:** The KPI - PUSH_AVG_TIME_CMPLETE is a graphical representation of the average time between when a push instance is created and when it reaches a COMPLETE status.
  - **KPI - PUSH _AVG_TIME_PROCESS:** The KPI - PUSH_AVG_TIME_PROCESS is a graphical representation of the average time between when a push instance is created and when it is processed by the push processor.
  - **KPI - PUSH _AVG_TIME_SVR_CONF:** The KPI - PUSH_AVG_TIME_SVR_CONF is a graphical representation of the average time between when a push instance is created and when it reaches SVR_CONF status.
  - **KPI - PUSH _AVG_TIME_CLNT_CONF:** The KPI - PUSH_AVG_TIME_CLNT_CONF is a graphical representation of the average time between when a push instance is created and when it reaches CLNT_CONF status.
  - **KPI - PUSH _CLNTCNF_TIME_TOPRANK:** The KPI - PUSH_CLNTCNF_TIME_TOPRANK is a graphical representation of the shortest time between when a push instance is created and when it reaches CLNT_CONF status.
  - **KPI - PUSH _CLNTCNF_TIME_LOWRANK:** The KPI - PUSH_CLNTCNF_TIME_LOWRANK is a graphical representation of the longest time between when a push instance is created and when it reaches CLNT_CONF status.

Copying an Object to the Customer Namespace

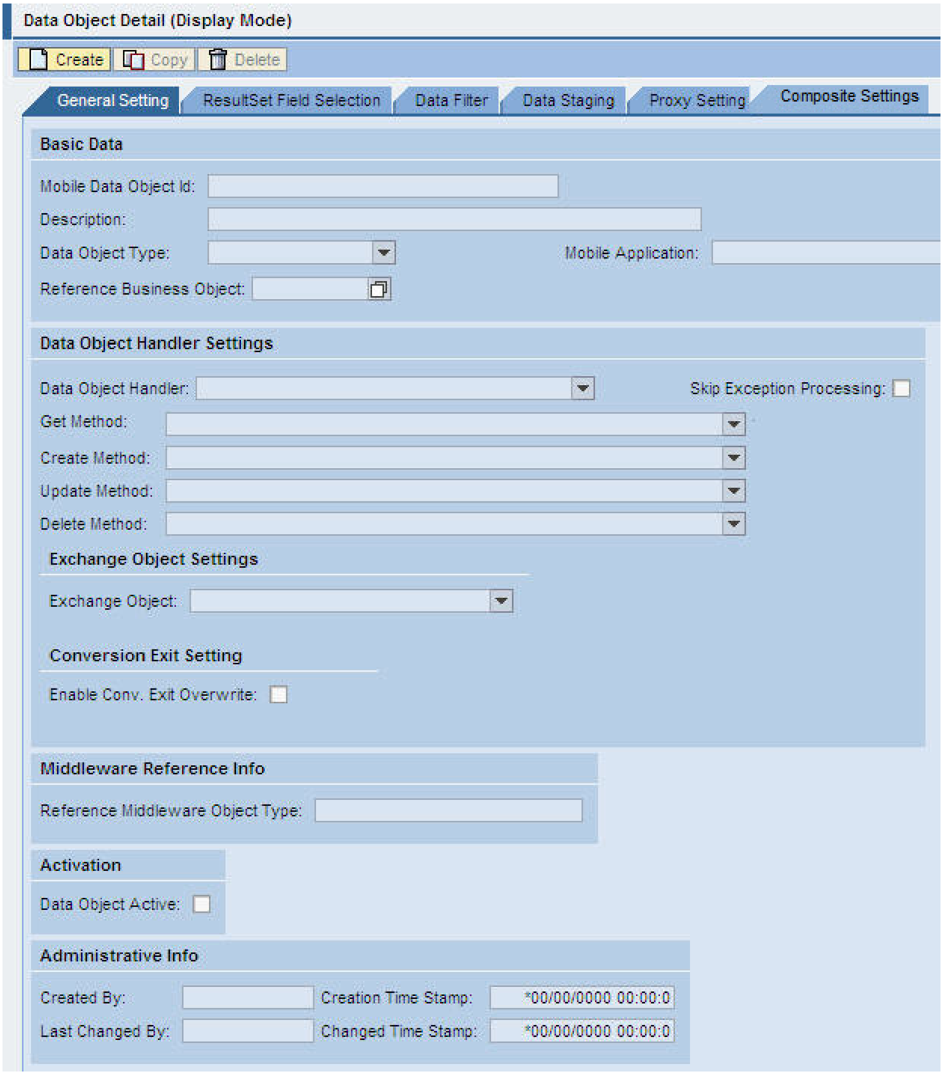
The following procedure provides information on making a copy of a mobile data object (MDO) or exchange object within the Agentry SAP Framework. For configuration changes where an MDO or exchange object is being modified, it is recommended that a copy is first made and placed in the customer namespace. In any of the procedures provided where an MDO or an exchange object should be copied, refer to this procedure for instructions. Copying these elements before making a modification ensures changes made to the application can be easily rolled back without affecting the originals.

Once a copy is made of an MDO and that copy is then modified for a configuration change, it will likely be necessary to adjust the BAPI wrapper assignment to reference the new MDO. Similarly, when an exchange object is copied and then modified, the EFI trigger assignment may need to be changed to the new exchange object. These procedures are covered separately.

1. Log into the Configuration Panel of the Agentry SAP Framework.
2. Click either **Exchange Object Configuration** or **Mobile Data Object Configuration** from the navigation menu.

   The Data Object Detail panel opens.

   Note: Figures shown in this procedure are taken from the Mobile Data Object configuration page. Screens may look different when configuring an exchange object. For either, the ability to copy is provided.



Data Object Detail (Display Mode)

Basic Data

Mobile Data Object Id:

Description:

Data Object Type: Mobile Application:

Reference Business Object:

Data Object Handler Settings

Data Object Handler: Skip Exception Processing: □

Get Method:

Create Method:

Update Method:

Delete Method:

Exchange Object Settings

Exchange Object:

Conversion Exit Setting

Enable Conv. Exit Overwrite: □

Middleware Reference Info

Reference Middleware Object Type:

Activation

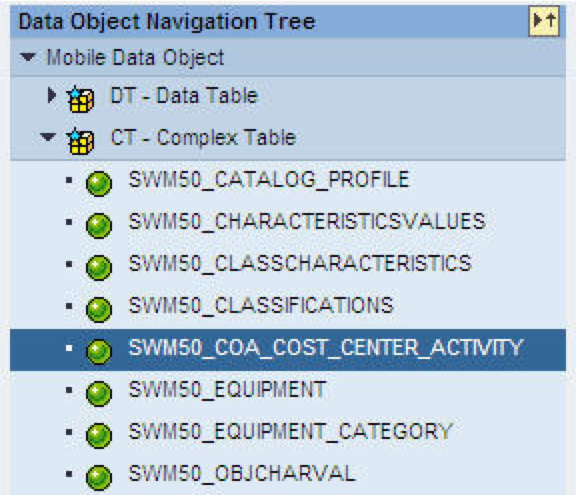
Data Object Active: □

Administrative Info

Created By: Creation Time Stamp: *00/00/0000 00:00:0

Last Changed By: Changed Time Stamp: *00/00/0000 00:00:0

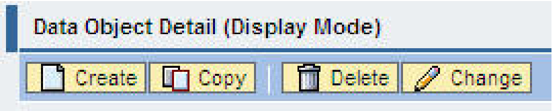
3. In the list of MDOs or exchange objects, select the one to be copied.



Data Object Navigation Tree

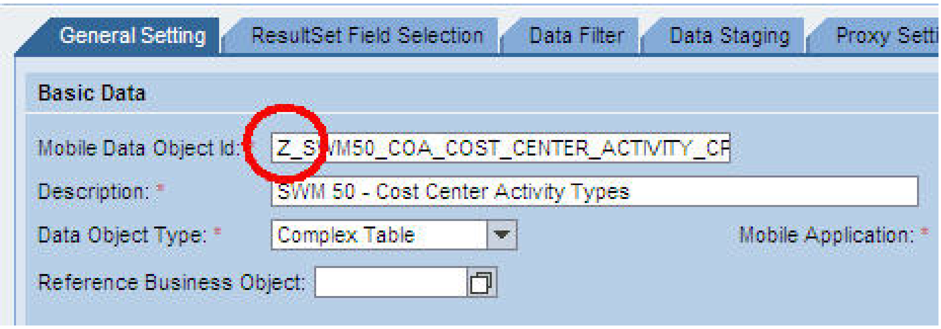
- Mobile Data Object
  - DT - Data Table
  - CT - Complex Table
    - SWM50_CATALOG_PROFILE
    - SWM50_CHARACTERISTICSVALUES
    - SWM50_CLASSCHARACTERISTICS
    - SWM50_CLASSIFICATIONS
    - SWM50_COA_COST_CENTER_ACTIVITY
    - SWM50_EQUIPMENT
    - SWM50_EQUIPMENT_CATEGORY
    - SWM50_OBJCHARVAL

4. Click **Copy**.



Data Object Detail (Display Mode)

5. In the Basic Data, or main panel object ID field, add a ‘Z’ to the beginning of the object name.



Basic Data

Mobile Data Object Id: Z_SWM50_COA_COST_CENTER_ACTIVITY_CF

Description: * SWM 50 - Cost Center Activity Types

Data Object Type: * Complex Table

Mobile Application: *

Reference Business Object:

6. When finished, click **Save** to save the object copy.

   A copy of the original object is created in the SAP customer namespace. The element can now be modified, with the back up element available for rollback purposes, if necessary.

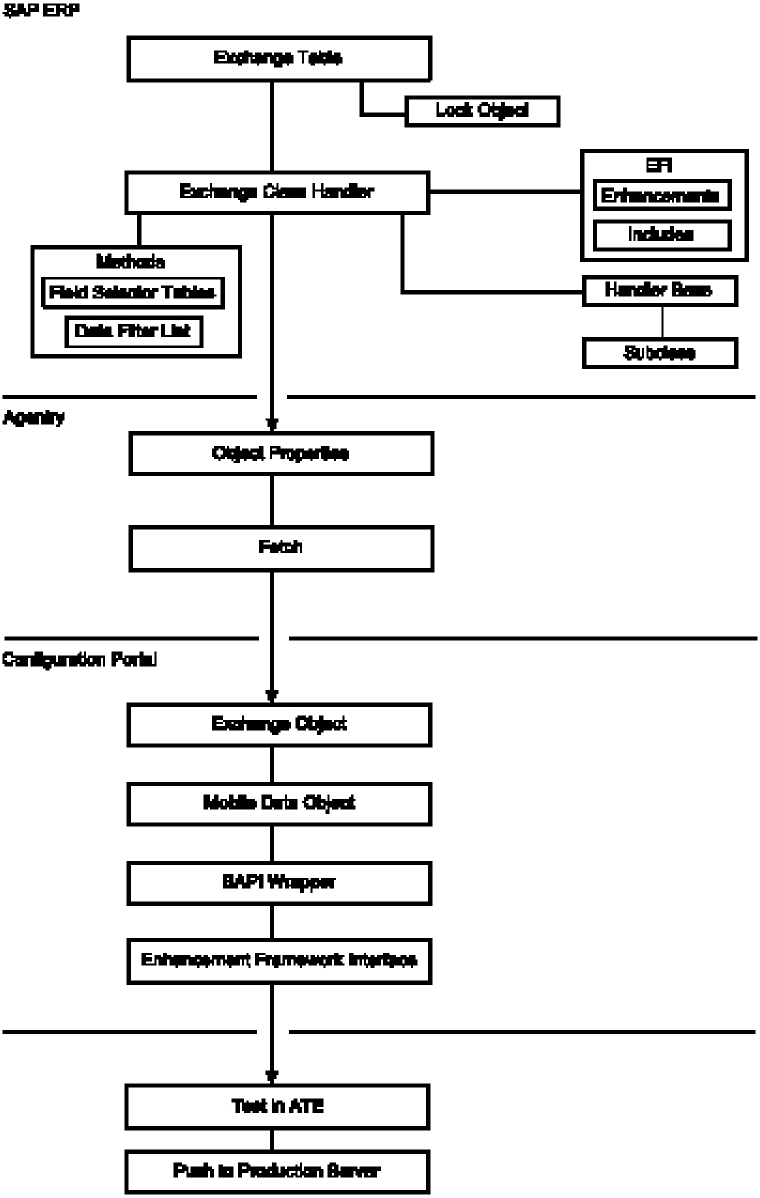
Adding a New Downstream Synchronization Process

The exchange process detects changes from transaction updates and records the change activity into an exchange table **/SYCLO/<fieldname>_EX**. The exchange table information is used by the mobile data object class handler to accomplish the delta exchange process so that the mobile application only sends and receives delta data using a time stamp.

The source code plug-in is implemented using the enhancement framework. To implement this, either the multi-instance BADI or the enhancement spot must include source code to update the exchange table for the current change. In order to implement the enhancement, the location of the enhancement must be identified. The location of the enhancement should be such that the old/new values of the changed records are available. These old/new values will be used by the exchange process to identify changes based on field selections and filter conditions configured for the exchange object.

The source code plug-in is either an include program **/SYCLO/<module>_EFI_<key field name>_EX_INCL** or an update module **/SYCLO/<module>_<exchange name>_SAVE_EX**. If the enhancement is being implemented in an UPDATE module, the plug-in is an INCLUDE program. Otherwise, it is an UPDATE module (the INCLUDE program is called within this module) to ensure the exchange table is updated in an UPDATE TASK.

The INCLUDE program is then assigned to an exchange object in the Agentry SAP Framework Configuration portal to activate the actual change detection process.



Implementing Downstream Synchronization

1. Implement the exchange process in SAP:

If there are similar objects already existing in SAP for any of these steps, it may be easier to select that object, right click, and choose the Copy menu item. After the object is copied, right click and select the Change menu item to make the necessary changes.

a) Add or modify the appropriate exchange table to store the change activity record.

b) Add or modify the appropriate lock object to trigger the lock mechanism for the update process.

c) Add or modify the exchange class handler.

d) Add or modify the methods by redefining the method GET_FIELD_SELECTOR_TABLES to enable the field selector function for the exchange object.

e) List the database tables supported by the exchange handler for change detection field selection and pass it to ET_FLDSEL_TABLE_LIST.

f) If required, redefine the method GET_TABLE_FILTER_LIST to enable the change detection filter function for the class handler. This provides the ability to restrict change detection based on data content.

g) List the filters supported by this exchange class handler along with the table name and the field name and pass it to the output table ET_DATA_FILTERS. Take the data object filter name from the technical properties of the field.

h) Add or modify the EFI enhancements.

i) Add or modify the EFI includes and insert in the appropriate enhancement location.

   The following areas may need to be added or modified when developing the source code plug-in:

   - Determine the current transaction update mechanism.
   - Construct an object key for the exchange table.
   - Build the data table with old and new values to detect changes based on field selections and filter conditions configured for the exchange object.
   - Call the exchange object class handler method UPDATE_EXCHANGE to update the exchange table with the current action and timestamp.

j) Add or modify the handler base by inheriting all the properties from the superclass /SYCLO/CL_CORE_EX_HANDLER_BASE. This superclass contains the base methods to fulfill the business logic related to the exchange process.

k) Add or modify the subclass.

2. **Implement the downstream synchronization in Agentry Editor:**

   a) Create new steplets for fetches and transactions.

   b) Create the associated stephandlers.

   c) Create the new POJO for the exchange object.

3. **Add the new exchange object in the Agentry SAP Framework Configuration portal:**

   a) Add or modify the exchange object.

   b) Add or modify the mobile data object.

   c) Add or modify the BAPI wrapper and assign the object to the BAPI wrapper.

   d) Add or modify the EFI.

Working with BAPI Wrappers

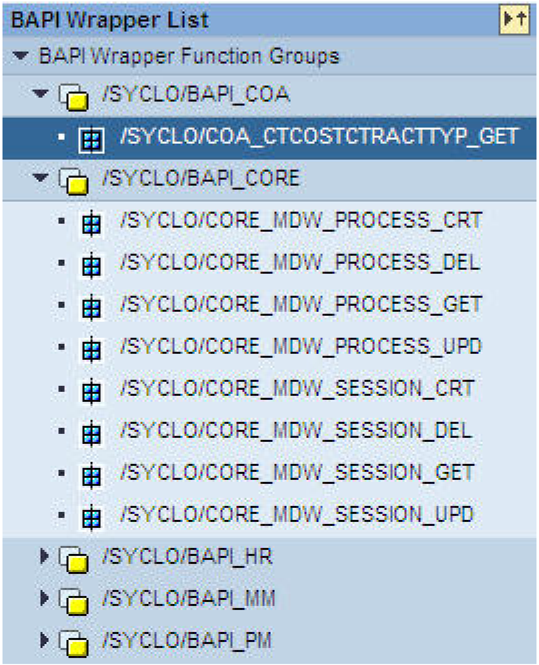
BAPI wrappers are an element type within the Agentry SAP Framework. The BAPI wrappers are the elements through which calls are made by the SAP Agentry Server to the elements configured in the MDOs, (i.e., the `GET`, `CREATE`, `UPDATE`, and `DELETE` methods). A BAPI wrapper is assigned to an MDO in the Agentry SAP Framework. The BAPI wrapper defines the inputs and outputs for the calls made to it and the data it returns to the SAP Agentry Server.

When making modifications or configuration changes to MDOs, and when a copy of the MDO is made in the customer namespace as a a part of the modification, the assignment settings of a BAPI wrapper must be changed to reference the new copy.

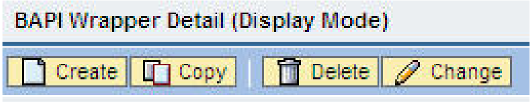
Changing the MDO Assignment of a BAPI Wrapper

Once an MDO is created, it must be assigned to a BAPI wrapper. During runtime, the MDO invoked is determined based on the BAPI wrapper assignment. This procedure describes the steps needed to change a BAPI wrapper’s MDO assignment.

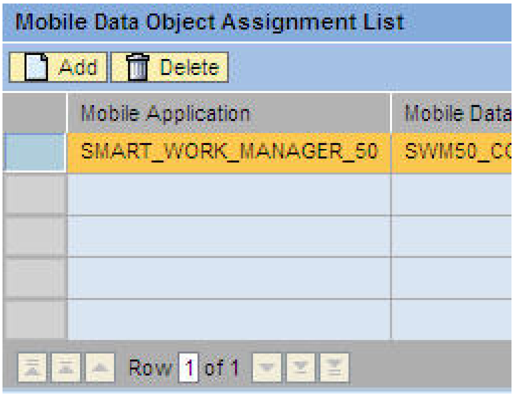
1. Open the Syclo Agentry SAP Framework Configuration portal through SAP.
2. Click the **BAPI Wrapper Configuration** menu item.
3. Navigate to and highlight the BAPI wrapper that will use the new or modified MDO or Z-MDO in the BAPI Wrapper List tree.


BAPI Wrapper List
BAPI Wrapper Function Groups
/SYCLO/BAPI_COA
/SYCLO/COA_CTCOSTCTRACTTYP_GET
/SYCLO/BAPI_CORE
/SYCLO/CORE_MDW_PROCESS_CRT
/SYCLO/CORE_MDW_PROCESS_DEL
/SYCLO/CORE_MDW_PROCESS_GET
/SYCLO/CORE_MDW_PROCESS_UPD
/SYCLO/CORE_MDW_SESSION_CRT
/SYCLO/CORE_MDW_SESSION_DEL
/SYCLO/CORE_MDW_SESSION_GET
/SYCLO/CORE_MDW_SESSION_UPD
/SYCLO/BAPI_HR
/SYCLO/BAPI_MM
/SYCLO/BAPI_PM

4. Click the **Assignment** tab.
5. Click **Change**.

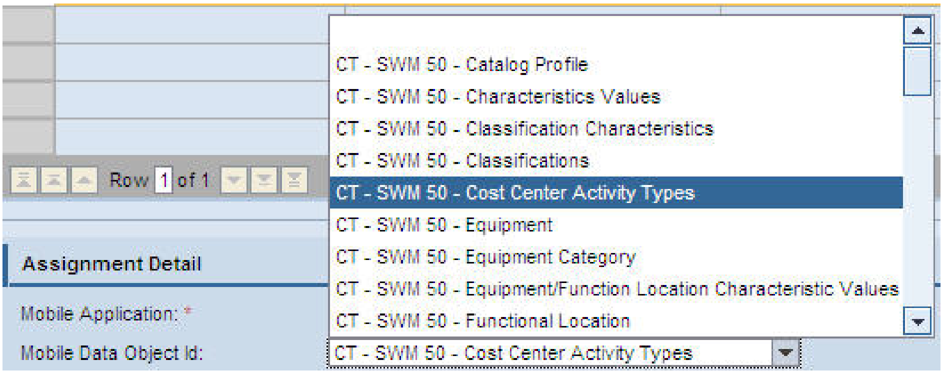


6. Use the **Add** button in the Mobile Data Object Assignment List panel to add the new Z-MDO.



Mobile Application	Mobile Data
SMART_WORK_MANAGER_50	SWM50_CC

7. Click the arrow to the right of the Mobile Data Object ID field and select the desired MDO from the drop-down list.


Row 1 of 1
Assignment Detail
CT - SWM 50 - Catalog Profile
CT - SWM 50 - Characteristics Values
CT - SWM 50 - Classification Characteristics
CT - SWM 50 - Classifications
CT - SWM 50 - Cost Center Activity Types
CT - SWM 50 - Equipment
CT - SWM 50 - Equipment Category
CT - SWM 50 - Equipment/Function Location Characteristic Values
CT - SWM 50 - Functional Location

Mobile Application: *

Mobile Data Object Id: CT - SWM 50 - Cost Center Activity Types

8. Change the Method Type field to select the MDO method to be called by the BAPI wrapper.
9. Highlight the original MDO in the Mobile Data Object Assignment List table and click **Delete**.

   The original is no longer assigned to the BAPI wrapper, leaving the newly-added Z-MDO.
10. Click **Save** to save the changes.

Changing MDO Filter Rules

Many of the common configuration changes made for an implementation involve the modification or addition of one or more filter rules within an MDO. In SAP, each user is assigned a role-based profile with authorization permissions on viewable data and available activities. For example, a user working in one plant should not be able to view data for another plant. When business activities performed by a user are mobilized through the mobile application, the ability to extend the same restrictions to the mobile application is necessary. Data filter rules provide the function to restrict data access for mobile applications.

This procedure describes the steps necessary, in general, to modify a data filter rule for an MDO. The specific settings of a given rule will vary depending on the overall nature of the change being made. Subsequent procedures will reference this process and provide the detailed values and settings for the filter rules involved in the specific change.

1. Open the Agentry SAP Framework Configuration portal through SAP.
2. From the ConfigPanel Home page, click the **Mobile Data Object Configuration** menu item.

   The Mobile Data Object Configuration page displays.
3. Expand the Mobile Data Object tree in the Data Object Navigation Tree section and highlight the MDO created earlier in this procedure.
4. Expand the desired method in the Defined Filters list.
5. Under the method are listed all current filters defined for the method. Select the filter whose rules should be modified.

   The current rule filter settings are displayed in the Rule Editor section. All existing rules for the filter are displayed in the Rule List table.
6. To add a new rule, edit an existing one, or delete a rule from the filter, click **Change** at the top of the page. Many of the fields in the Rule Editor section become editable, and two new buttons are displayed to the right of the Rule List field, **Create** and **Delete**.
7. To delete a rule from the filter, select that rule in the list and click **Delete**. If no further changes are to be made, click **Save** at the top of the page to completely remove the rule from the filter. If additional changes are still needed, the filter can be saved after all changes are complete.
8. To add a new rule to the filter, click **Create** to the right of the Rule List. To edit an existing rule within the filter, select it in the Rule List. This will display the current settings for the rule in the DOF Rule Type, Sign, Option, Low Value, High Value, and Active Flag fields within the Rule Editor section.
9. Set or modify the editable fields according to the needs of the application. For a detailed description of all fields, see *Mobile Data Object - Data Filter*.

10. Be sure the Active Flag field is set to true for each added or edited field before saving changes. Inactive filter rules will have no effect on the synchronization processing. An Active Flag is set by clicking in the box in order to display a checkmark.
11. When all desired changes have been made to the filter rules click **Save** to apply those changes.

Creating a New Filter Rule

1. Click the BAPI Wrapper Settings in the Navigation Panel.
2. Navigate to the BAPI Wrapper that uses the MDO you are modifying, in the BAPI Wrapper List.
3. Click the **Assignment** Tab and click the **Change** Button.
4. Use the Add button to add ZMDO version of the complex table, with the newly created field. Use the delete button to delete the original complex table.
5. Click the **Save** button to save the changes.
6. From the SAP user menu, click **SMART Mobile Suite Configuration**.

   This will launch a web browser, which will log into the SMART Mobile Suite Configuration Site.

7. Click the **Mobile Data Object Settings**.

   This will display the Mobile Data Object Settings Drop down menu:

8. Expand this Menu and select desired Data object for the data filter.
9. Select the Data Filter Tab.
10. Click the Create button near the top of the screen.
11. Use the filer settings to define the new filter.
12. Click the **Save** button to save your changes.
13. Use the ATE to verify the filters perform as expected on the client. Validate the expected material is fetched to the device, by paying attention to the transmit log as it completes.

Adding a New Data Table

1. **Create a new data table in the Configuration portal.**
   a) Create a new subclass in the Z namespace from **/syclo/cl_dt_DO**.
   b) Override **get_data_table** method to retrieve data from SAP.
   c) Override **get_data_filter_list** method to support any filter logic.
   d) Create a new mobile data object in the Z namespace, referred to as a Z-MDO.
2. **Add the new data table definition to Agentry.**

a) Open the SAP project in Agentry.
b) Create the new data table.
c) Make any other applicable changes to the application that is using the new data table, such as displaying the new value or modifying rules to search on the new values.

3. **Configure Java synchronization between the data table and Agentry.**
   a) Open the appropriate Java project for the SAP application.
   b) Modify the new data table class created when the data table definitions was added to the Agentry application project.
   c) Create the associated stephandlers.
   d) Create the new POJO for the data table.
   e) Publish the application to the SAP Agentry Development Server in preparation for testing. Restart the Server as changes were also made to the Java logic in support of the new data table.

Next

Perform a transmit from the Agentry Test Environment and verify that the new data table and all related functionality are producing desired results.

Adding a New Complex Table

Adding a new complex table to the application requires four main tasks:

- Creating or modifying the exchange tables and associated objects in SAP.
- Using the Agentry SAP Framework Configuration portal to create a new MDO or Z-MDO for the complex table.
- Creating new complex table Java class, step handler and POJO for the new complex table in the Java editor in Eclipse.
- Using Agentry Editor to modify the application to make use of and support the new complex table.

1. **Create or modify the exchange process in SAP**
   a) Determine the enhancement spot in SAP if an exchange process is required. An enhancement spot is a location in the SAP code where access is available to the data that is changing either through a transaction or a BAPI.
   b) Create the enhancement implementation. The enhancement implementation should be in an update module.
   c) Create the exchange table.
   d) Create the lock object associated with the exchange table.
   e) Create the EFI implementation. The EFI implementation reads the data being changed in SAP and calls the method **/syclo/cl_cor_exch_serve=>update_exchange** to update the exchange table.

   f) Determine and create the structure of the data that is sent back in the ET_COMPLEX_TABLE parameter of the BAPI wrapper.
   g) Determine and create the structure of the deleted records that are sent back in the ET_EXCHANGE_ACTION_DELETED parameter of the BAPI wrapper.
   h) Create the MDO handler class, which will inherit from the class **/SYCLO/CL_CORE_CT_HANDLER**.

2. **Create a new complex table in the Configuration portal**
   a) Create a new complex table MDO or copy an existing MDO to the Z-namespace.
   b) If needed, create a new BAPI wrapper in the Z-namespace using the BAPI include template **/syclo/core_bapi_template_incl**.
   c) Assign the new Z-MDO to the new BAPI wrapper, or assign the new Z-MDO to an existing BAPI wrapper, depending on your configuration.
3. **Create the complex table in the Agentry Editor**
   a) Create a new complex table in Agentry Editor to make use of and support the data in SAP.
   b) Create the field definitions within the complex table to
   c) Add any necessary indexes to the complex table.
4. **Configure Java synchronization between the complex table and Agentry**
   a) Open the appropriate Java project for the SAP application.
   b) Modify the new complex table class created when the complex table definitions was added to the Agentry application project.
   c) Create the associated step handlers.
   d) Create the new POJO for the complex table.
   e) Publish the application to the SAP Agentry Development Server in preparation for testing. Restart the Server as changes were also made to the Java logic in support of the new complex table.

Next

Perform a transmit from the Agentry Test Environment and verify that the new complex table and all related functionality are producing desired results.

Adding a New Data Object

Adding a new data object to the application requires four main tasks:

- Redefine methods in SAP to allow for the new data object.
- Using the Agentry SAP Framework Configuration portal to create a new MDO or Z-MDO.
- Creating new data object Java class, stephandler and POJO for the new data object in the the Java editor in Eclipse.
- Using the Agentry Editor to modify the application to make use of and support the new data object.

1. **Redefine methods in SAP**
   a) Redefine the **get_field_selector_tables** method for the class handler to enable field selector function for the mobile data object.
   b) In the **get_field_selector_tables** method, list the database tables supported by the class handler GET method for the result set field selection **ET_FLDSEL_TABLE_LIST**.
   c) List out the mandatory key fields from the required tables and pass them to the output table **ET_REQ_FIELD_LIST** to ensure that all key fields are included in the output selection.
   d) Redefine the **get_data_filter_list** method to enable data filter function for the class handler and define the necessary filter rules to control what data can be viewed by the mobile application.
   e) List the filters supported by the **get_data_filter_list** method along with the table and field names and pass them to the output table **ET_DATA_FILTERS**. Take the data object filter (DOF) name from the technical properties of the field.
2. **Create a new data object in the Configuration portal**
   a) Create a new mobile data object in the Z namespace, referred to as a Z-MDO.
   b) Create a new subclass in the Z namespace from **/syclo/cl_DO_Handler_base**.
   c) Override the GET, CREATE, UPDATE, and DELETE methods as needed to retrieve data from SAP.
   d) Create a new BAPI Wrapper in the Z namespace using the BAPI include template: **/syclo/core_bapi_template_incl**. Or, assign the Z-MDO to an existing BAPI wrapper.
   e) Assign the new Z-MDO to the new BAPI wrapper if it was not assigned to an existing BAPI wrapper.
3. **Configure Java synchronization between the data object and Agentry**
   a) Create new steplets for fetchs and transactions.
   b) Create the associated stephandlers.
   c) Create the new POJO for the data object.
4. **Add the new data object definitions to Agentry**
   a) Open the SAP project in Agentry.
   b) Create the new data object to represent the business object from SAP.
   c) Create the transactions and read steps of the data object.
   d) Create any necessary fetches or transactions associated with the data object.
   e) Publish the application to the SAP Agentry Development Server in preparation for testing. Restart the Server as changes were also made to the Java logic in support of the new data object.

Next

1. Perform a transmit from the Agentry Test Environment and verify that the new data object and all related functionality are producing desired results.

2. Publish the application, including the supporting Java synchronization logic, to the SAP Agentry Production Server for deployment.

Adding new Values to be Retrieved for Mobile Application Definitions

Prerequisites

The following items must be addressed prior to performing this procedure:

- The desired values to be added to those being retrieved should be determined and noted, as should the tables within which they reside within SAP.
- The person performing this procedure must have access to the Agentry SAP Framework Configuration Panel and have permissions to change the configuration settings of the elements within it.

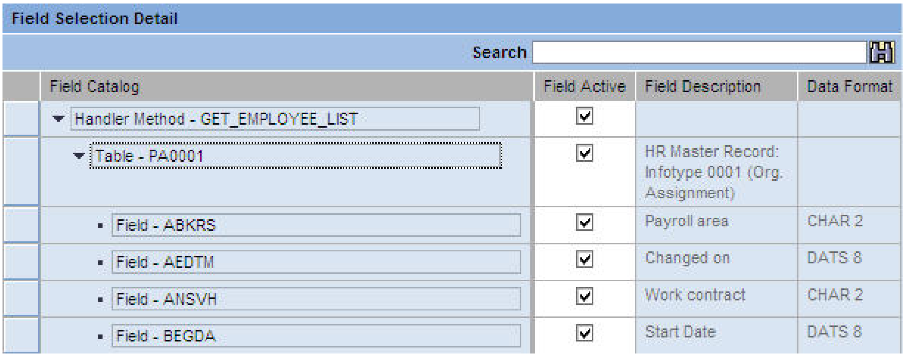
Task

Use this procedure to add new fields to complex tables or objects. The following procedure uses a complex table as an example. However, these steps will accomplish the same goal to add new values to object definitions as well. Where there are differences, they are noted in this procedure.

1. Within the Agentry SAP Framework Configuration Panel ConfigPanel Home page, select the menu link **Mobile Data Object Configuration**. Select the proper MDO from the list on the left of the configuration page, which displays its current settings.
2. Copy this MDO to the Z namespace. All changes to the MDO should be made to this new copy in the Z namespace. The original MDO should not be modified.

   For information on copying an MDO, see the procedure section “Copying an MDO or Exchange Object to the Customer Namespace.”
3. **Select the additional fields in the MDO:**
   a) Navigate to the Mobile Data Object Configuration panel in the Configuration Panel and double-click the appropriate MDO.
   b) Click the **Result Set Field Selection** tab and click the **Change** button.
   c) In the Field Selection Detail pane, expand the table to which active fields will be added.

   The Field Active column displays in white, with check mark boxes enabled.



Field Selection Detail

Search

Field Catalog	Field Active	Field Description	Data Format
▾ Handler Method - GET_EMPLOYEE_LIST	☑		
▾ Table - PA0001	☑	HR Master Record: Infotype 0001 (Org. Assignment)	
▪ Field - ABKRS	☑	Payroll area	CHAR 2
▪ Field - AEDTM	☑	Changed on	DATS 8
▪ Field - ANSVH	☑	Work contract	CHAR 2
▪ Field - BEGDA	☑	Start Date	DATS 8

d) Scroll through the fields to determine which ones to enable or disable. Checking the box enables the field, while un-checking the box disables the field.

e) Click **Save** to save your changes.

4. Navigate to the ConfigPanel Home page and select the **BAPI Configuration** link. In this page, change the BAPI wrapper assignment of the proper BAPI wrapper to use the MDO in the Z namespace just modified in the previous step.

For information and instructions on changing a BAPI wrapper's MDO assignment, see the procedure section "Changing the MDO Assignment of a BAPI Wrapper."

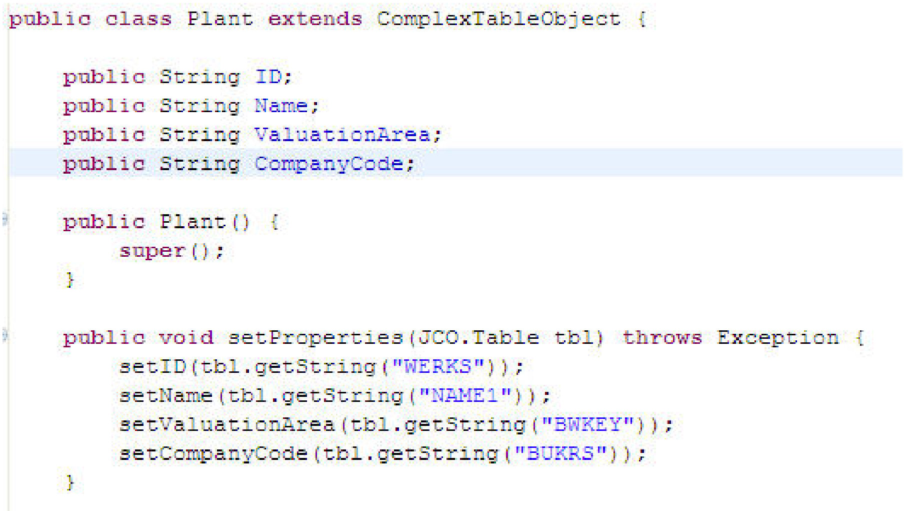
5. Complete Java synchronization through Eclipse:

a) Open the appropriate Java project in Eclipse for the mobile application.

b) Locate the appropriate BAPI wrapper Java file in the project. You can find this by locating the BAPI wrapper name that is associated with the MDO in the Configuration portal. Copy this BAPI wrapper name and perform a file search within Eclipse to locate the correct Java file.

c) Open the POJO declaration within the Java file.

d) Locate the public strings and the properties. Add new data members to the class matching the fields selected for retrieval in the MDO. Also define the new setter methods for those selected fields.



```
public class Plant extends ComplexTableObject {

    public String ID;
    public String Name;
    public String ValuationArea;
    public String CompanyCode;

    public Plant() {
        super();
    }

    public void setProperties(JCO.Table tbl) throws Exception {
        setID(tbl.getString("WERKS"));
        setName(tbl.getString("NAME1"));
        setValuationArea(tbl.getString("BWKEY"));
        setCompanyCode(tbl.getString("BUKRS"));
    }
```

e) Save and compile the Java code by running the `build-mm.xml` script. This should initially be performed on the SAP Agentry Development Server to allow for testing before publishing to the Production server.

6. **Add the new fields to the complex table or object definition in Agentry:**

a) Open the SAP project in Agentry.

b) Navigate to the complex table or object that will contain the newly-added fields.

c) For complex tables, select the **Fields** tab and click the green plus icon to add a new field.

d) Add the new field using the wizard, naming it to match the name given to the values in the Java class. When done, click the **Finish** button.

e) Click on the **Indexes** tab to add new indexes if necessary.

f) For objects, select the **Properties** tab and click the green plus icon to add a new field.

g) Add the new property using the wizard, naming it to match the name given to the values in the Java class. When done, click the **Finish** button.

h) Make any other applicable changes to the application that is using the complex table or object, such as displaying the new value or modifying rules to search on the new values.

i) Publish the application to the SAP Agentry Development Server in preparation for testing. Restart the SAP Agentry Development Server as changes were also made to the Java logic in support of the new fields.

With the completion of this procedure, one or more new values will be retrieved as a part of the data for the object or complex table definition. These new values may be displayed, edited, searched on, or used in any other appropriate manner on the client.

Next

When these modifications are complete, the application should be thoroughly tested. It should be verified that the values added are retrieved as expected. Any functionality related to these values should also be tested. Once testing is successful, the modifications made should be migrated to the production system according to migration processes in place at the implementation site.

Adding New Fields to an Exchange Object

Use this procedure to add new fields exchange objects.

The following are accomplished in this procedure:

- A new field is enabled in the exchange object through the Configuration portal
- Java synchronization is achieved in Eclipse
- The new object is added in the Agentry editor

1. If a new exchange object is needed, copy an existing exchange object to the Z namespace in SAP.

   For more information, see the section Copying an Object to the Z Namespace.

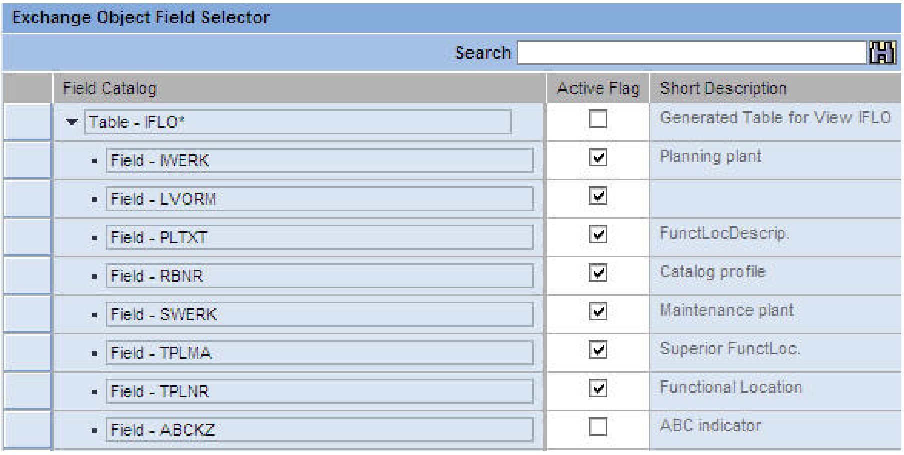
2. **Add new fields to the exchange object:**

   a) Navigate to the Exchange Object Configuration panel in the Configuration portal and double-click the appropriate exchange object.

   b) Click the **Change Detection Field Selection** tab and click the **Change** the button.

   c) In the Exchange Object Field Selector Detail pane, expand the table to which active fields will be added.

   The Field Active column displays in white, with checkmark boxes enabled.



Exchange Object Field Selector

Search

Field Catalog	Active Flag	Short Description
Table - IFLO*	☐	Generated Table for View IFLO
Field - IWERK	☑	Planning plant
Field - LVORM	☑	
Field - PLTXT	☑	FunctLocDescrip.
Field - RBNR	☑	Catalog profile
Field - SWERK	☑	Maintenance plant
Field - TPLMA	☑	Superior FunctLoc.
Field - TPLNR	☑	Functional Location
Field - ABCKZ	☐	ABC indicator

   d) Scroll through the fields to determine which ones to enable or disable. Checking the box enables the field, which unchecking the box disables the field.

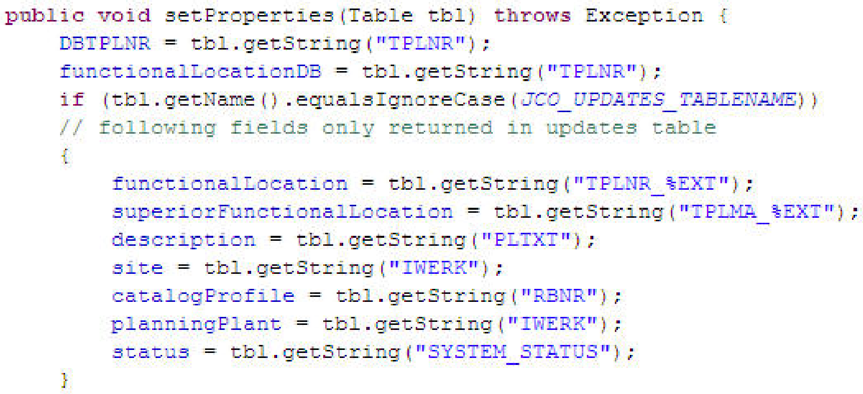
   e) Save your changes by clicking the **Save** button.

3. **Complete Java synchronization through Eclipse:**

a) Open the appropriate project in Java, or create a new project.

b) Locate and open the appropriate Java file in the project.

c) Locate the public strings and the properties. Add the new table fields checked in the Exchange Object Field Selector Detail pane in the Configuration portal.



```
public void setProperties(Table tbl) throws Exception {
    DBTPLNR = tbl.getString("TPLNR");
    functionalLocationDB = tbl.getString("TPLNR");
    if (tbl.getName().equalsIgnoreCase(JCO_UPDATES_TABLENAME))
    // following fields only returned in updates table
    {
        functionalLocation = tbl.getString("TPLNR_%EXT");
        superiorFunctionalLocation = tbl.getString("TPLMA_%EXT");
        description = tbl.getString("PLTXT");
        site = tbl.getString("IWERK");
        catalogProfile = tbl.getString("RBNR");
        planningPlant = tbl.getString("IWERK");
        status = tbl.getString("SYSTEM_STATUS");
    }
```

d) Save and compile the Java code by running the `build-mm.xml` script. This should initially be performed on the Development server to allow for testing before publishing to the Production server.

4. Add the new fields to the exchange object in Agentry:

a) Open the SAP project in Agentry.

b) Navigate to the appropriate object.

c) Add any new required properties, transactions, or read steps associated with the new fields.

d) Click the **Finish** button when done.

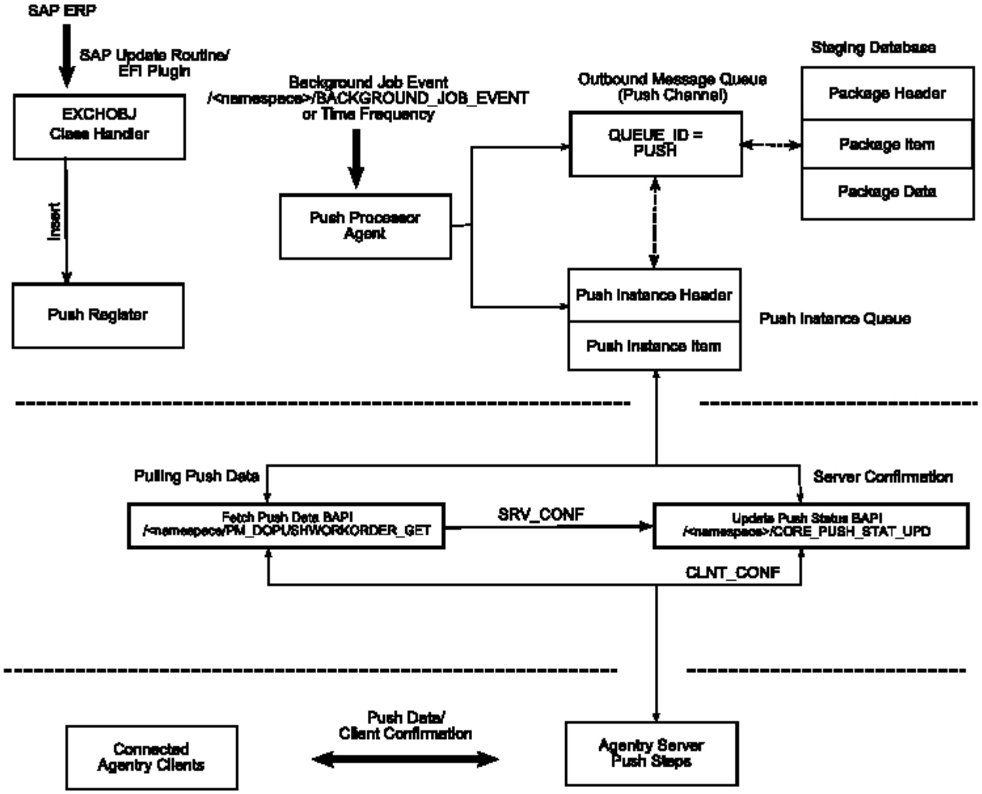
e) Make any other applicable changes to the application that is using the modified object.

f) Publish the application to the SAP Agentry Development Server in preparation for testing. Restart the Server as changes were also made to the Java logic in support of the new fields.

Working with Push Scenarios

A push scenario pushes emergency work orders to the corresponding recipients. Use the following diagram and steps to follow a push instance from generation in SAP to reception on the Client.

Push Process Flow


SAP ERP
SAP Update Routine/ EFI Plugin
EXCHOBJ Class Handler
Insert
Push Register
Background Job Event /<namespace>/BACKGROUND_JOB_EVENT or Time Frequency
Push Processor Agent
Outbound Message Queue (Push Channel)
QUEUE_ID = PUSH
Staging Database
Package Header
Package Item
Package Data
Push Instance Header
Push Instance Item
Push Instance Queue
Pulling Push Data
Server Confirmation
Fetch Push Data BAPI /<namespace/PM_DOPUSHWORKORDER_GET
SRV_CONF
Update Push Status BAPI /<namespace>/CORE_PUSH_STAT_UPD
CLNT_CONF
Connected Agentry Clients
Push Data/ Client Confirmation
Agentry Server Push Steps

1. The push exchange process initiates the push trigger based on the push conditions. Conditions are defined as filter rules in the push exchange object. For instance, `work order priority = 1` is considered an emergency work order in the base product release.
2. The work order that satisfies the push conditions inserts a record into the push register table /SYCLO/PSH01 with an object key as the work order number and a push status of NEW.
3. The event /SYCLO/BACKGROUND_JOB_EVENT is raised after the work order is saved, which triggers the background job for the push processer agent.
4. The push processer job /SYCLO/CORE_PUSH_PROC_PROG is triggered, either by using the event or the time frequency. This trigger is based on specific customer processes.
5. The push processer determines the recipients for the push work order and builds the data for each recipient as a separate instance. The instance is stored in the outbound message queue /SYCLO/PSH02 with queue ID = PUSH, using the staging database.
6. The push instance displays one of the following statuses, viewable in the push monitor in the Administration Panel:
   - NEW
   - PROCESS

- CANCEL
- COMPLETED
- SRV_COMP

7. The Agentry application within the SAP Mobile Server calls the push BAPI /SYCLO/PM_DOPUSHWORKORDER_GET for every predefined time interval and checks the push queue for new items.
8. The Agentry application within the SAP Mobile Server sends the push data to the respective Clients depending on the user credentials that match the push instance.
9. Once the Client receives the push message, it sends the Client confirmation back to the Server and the Server calls the BAPI /SYCLO/CORE_PUSH_STAT_UPD to update the confirmation with status CLNT_CONF back to SAP.

Adding a New Push Scenario

In order to create a new push scenario, the following need to be defined:

- Source attributes
- Distribution settings
- Subscriber list

1. **Generate the MDO class handler for the work order push:**
   a) Create a new class interface by inheriting all the properties from the base class /SYCLO/CL_CORE_DO_HANDLER_BASE.
   b) Activate the push by redefining the method CHECK_PUSH_SUPPORTED to determine whether the push service is supported by this handler.
   c) Redefine the method DETERMINE_PUSH_RECIPIENTS to identify valid users for the emergency work order push.
   d) Based on the work order assignment type, determine valid partner information and obtain the recipients from the middleware user registry table /<namespace>/MDW00.
   e) Redefine the method BUILD_PUSH_DATA to prepare user-dependent data to be pushed to the recipients. Each recipient will have a separate set of data collection images (push instance) and moved to the outbound queue.
   f) Redefine the method GET_DATA_FILTER_LIST to enable the data filter function for the class handler.
   g) List the filters supported by this class handler method along with the table name and filter name and pass it to the output table ET_DATA_FILTERS.
2. **Redefine the GET method to enable the fetch process for the class handler:**
   a) Convert the RFC parameter list into OO parameter format.
   b) Build filter rules from the data object filter service and BAPI input parameters.
   c) Get the outbound message queue data from the push registry cache tables for the MDW user GUID range supplied through the BAPI parameter interface.
   d) Build the output data in OO parameter format.

e) Cache the exceptions, if any, to the output return table.

f) In the Agentry SAP Framework Configuration portal, navigate to the Mobile Data Object panel and assign the GET class handler to the appropriate mobile data object.

3. **Create the BAPI wrapper for the push work order fetch (GET):**

a) Create a new RFC.

b) Assign the input parameter IS_BAPI_INPUT Type: /SYCLO/CORE_BAPI_INPUT_STR, which is the Syclo BAPI wrapper standard input setting. This structure contains the mobile object data such as the mobile user name, device ID, timestamp from the mobile, mobile data object ID, staging information, session and user GUIDs, etc.

c) Assign the export parameter ES_BAPI_OUTPUT Type: /SYCLO/CORE_BAPI_OUTPUT_STR, which is the Syclo BAPI wrapper standard output structure. This structure contains the timestamp from SAP to the mobile device, package information, etc.

d) Assign the appropriate tables parameters. This tab contains all the filter ranges and output data structures.

4. **Implement the exchange process for the work order push:**

a) Create a new configuration entry for the exchange process that is utilizing the push.

b) Enable the push settings and additional filter conditions relevant to the push instance.

c) Configure the exchange push settings to identify whether the push is active or not, as well as additional filter criteria used to maintain push conditions.

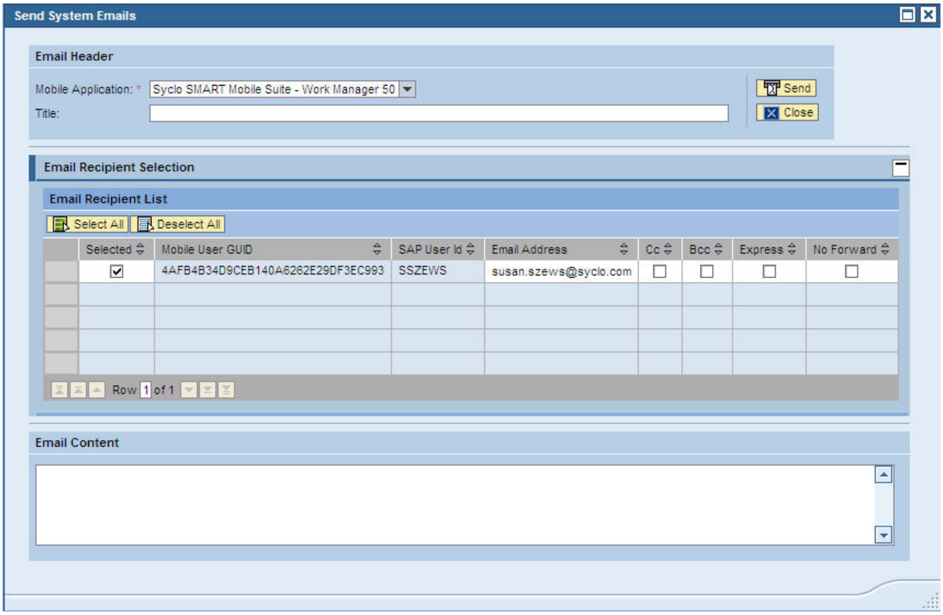
Sending Email Using the Administration & Monitoring Portal

At times, administrators may need to broadcast system messages or other information to a group of users or all users on the system. The Administration portal provides a way to communicate with users through email or text messages rather than through the mobile device, which may not be connected and available.

Note: User email or HTTP addresses and preferences are set in the Administration portal, Administration menu, User Management panel.

1. Access the Administration & Monitoring portal.
2. Click on the **Monitoring** hyperlink at the top of the screen, and then click the **User Monitor** menu option in the navigation panel.
3. Click the **[Send Email]** button.

   The Send System Emails screen displays.



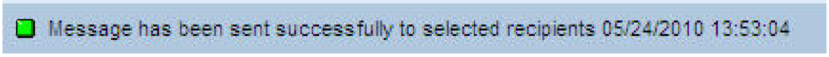
Mobile Application: * Syclo SMART Mobile Suite - Work Manager 50

Title:

Selected	Mobile User GUID	SAP User Id	Email Address	Cc	Bcc	Express	No Forward
☑	4AFB4B34D9CEB140A6262E29DF3EC993	SSZEWS	susan.szews@syclo.com	□	□	□	□

4. Check or uncheck the Selected boxes for all desired mobile user GUIDs.
5. Fill out the title, or header of the email.
6. Type in the body of the email in the Email Content box.
7. Click **[Send]** to send the email to all marked user GUIDs or **[Close]** to cancel out of the screen and discard changes.

   If the email is sent successfully, a message displays.



🟩 Message has been sent successfully to selected recipients 05/24/2010 13:53:04