SAS® Solutions Adapter 1.3 for SAP

User's Guide

THE POWER TO KNOW®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2006. *SAS® Solutions Adapter 1.3 for SAP: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® SAS Solutions Adapter 1.3 for SAP: User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 2006

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/pubs** or call 1-800-727-3228.

Contents

CHAPTER 1

Introduction to SAS Solutions Adapter for SAP

Introduction to SAS Solutions Adapter for SAP

The SAS Solutions Adapter for SAP is a collection of ETL jobs that load data from standard SAP tables into standard SAS tables in the SAS Detail Data Store (DDS). These ETL jobs are built on the SAS Data Surveyor and communicate with any SAP R/3 system.

The SAS Solutions Adapter for SAP performs two primary functions:

- It uses the SAS Data Surveyor for SAP software to extract SAP tables. These extractions include transparent table extracts, Business Application Programming Interface (BAPI) calls, and calls to remote-enabled Advanced Business Application Programming (ABAP) functions that SAS provides.
- It takes the raw data that is extracted from SAP and transforms it into the structure that is required to load the data into tables that populate SAS solutions. This transformation logic encapsulates business rules of both SAP and SAS solutions.

Based on the requirements of your source SAP system and your business needs, you can customize the collection of ETL jobs that make up the SAS Solutions Adapter for SAP. The architecture of SAS Data Integration Studio and SAS Data Integration Server solutions provides the environment for these ETL jobs. As a result, you can use SAS Data Integration Studio to view and edit your ETL jobs as needed.

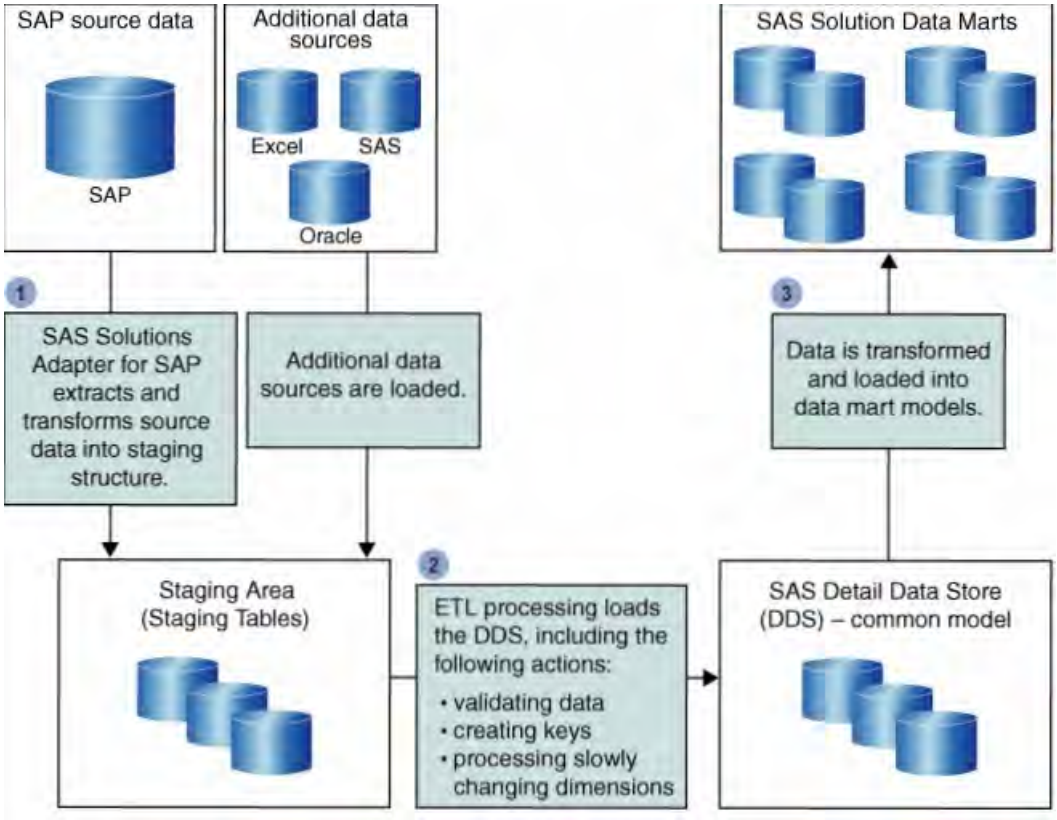
Data Flow Architecture

The SAS Solutions Adapter for SAP facilitates the beginning of a data flow that moves data from an SAP source system into a format that businesses manage and use with SAS solutions.

Source data from various systems must flow through a series of extraction, transformation, and loading steps to convert into a usable format for SAS solution data marts such as SAS Human Capital Management. The figure below details the basic

architecture of this data flow. The SAS Solutions Adapter for SAP acts in the first step of the model.

Figure 1.1 Data Flow Architecture: SAP to SAS Solution Data Marts



1 The SAS Solutions Adapter for SAP extracts and transforms the SAP source data into a staging area. The staging area is the first target after the data is extracted from an operational system. This staging area is a repository for raw data that is extracted from an operational system and prepared for transformation and loading to the SAS Detail Data Store.

2 Once source data is loaded into the staging area, it is ready for ETL processing that loads it into the DDS. The following actions occur during this ETL process:

- validating the data
- creating surrogate keys instead of operational system IDs to identify the data
- introducing date-and-time stamps to reflect data validity if differences in dimensional information arise

The transformed data is stored in the DDS, a "lightly denormalized" relational data model that provides storage flexibility. The DDS is a specific detail data model that represents a standard business function and supports a SAS solution data mart. Some of the data that the DDS captures includes current and historical information such as temporal data (event data that occurs at a particular date and time such as an account inquiry) and nontemporal data (non-event data such as a customer or a financial account).

3 Once data is cleansed, validated, and stored in the DDS, it is ready for transforming and loading to a data mart model.

Benefits

The SAS Solutions Adapter for SAP enables SAS solutions to turn SAP data into business intelligence that is manageable on an enterprise scale. By automating and streamlining parts of the data conversion from SAP to SAS, the adapter decreases the time it takes to retrieve business intelligence from source data. Primarily, it reduces the time needed to set up SAP ETL by 80% or more. This gives you more time to concentrate on generating intelligence from your SAP data.

The SAS Solutions Adapter for SAP also includes SAP knowledge that streamlines the data conversion process. This efficiency reduces your dependence on SAP resources to help you understand how SAP works. It also saves you time and valuable SAP resources so that they're available to work on other projects.

Configuration and Administration

The SAS Solutions Adapter for SAP is configured and managed using the Solution Package Deployment Wizard and SAS Data Integration Studio.

The Solution Package Deployment Wizard, a plug-in for SAS Management Console, enables you to specify the adapter configurations that are required for your unique business environment. You can use this tool to specify parameters during initial installation or import.

After you have configured the SAS Solutions Adapter for SAP, you can use SAS Data Integration Studio to manage and customize the adapter.

Components

The SAS Solutions Adapter for SAP includes various components that you can manage and configure using the Solution Package Deployment Wizard. This wizard enables you to manage and configure the following parameters that you need in order to import a SAS Solutions Adapter for SAP environment for SAS Data Integration Studio:

SAS Data Integration Studio jobs

: extract SAP tables into a partial mirror image of the SAP tables. These jobs mirror only the necessary tables and in many cases transfer only the required data rows and columns. This limited transfer places a smaller load on the SAP operational system. The extractions are typically run as SAP batch processes that do not use complex logic so that the jobs can run in a short time during off-peak SAP system load.

transformation jobs

: create intermediate tables or populate tables in the SAP staging area model.

table metadata

include the standard tables needed by the SAS solutions. In addition, intermediate tables are defined to enable flexibility in customizing the jobs to fit local requirements.

new SAS Data Integration Studio transformations

work in the default jobs or new jobs if necessary. Additional transformations are also available.

SAS code files

install on the SAS application server. These open source files are automatically copied to a location where they can be modified if necessary.

SAS macros

include simple macros that are provided as open source code and complex macros that are provided without source code so that SAS can maintain them easily.

SAS libraries

provide storage for administration and data tables.

Job Organization and Names

The SAS Solutions Adapter for SAP is a collection of ETL jobs that extract, transform, and load data from standard SAP tables into standard SAS tables. These jobs are organized into the following two groups:

- extraction jobs
- transformation jobs

To help identify job functions and streamline the extraction process, extraction job names begin with the letter E and transformation job names begin with T. This naming structure enables all extractions to occur without waiting for follow-on transformation jobs to complete. As a result, extractions can complete in a shorter time period.

Extraction jobs and transformation jobs are also grouped into several SAS Data Integration Studio job groups. Once grouped, each job name in a job group begins with the same initial letter and number. This enables quick navigation in the **`Custom`** tab of SAS Data Integration Studio.

To help identify the job functions of the group, the group names begin with a structure that is similar to the job names. The extraction job groups begin with the letter E followed by a two-digit number. The transformation job groups start with T followed by a four-digit number.

The numbering of transformation jobs within each group generally implies the order of execution unless a specific job does not depend on another. As a result, the specific numbers assigned to jobs within a group allow for new jobs to be inserted numerically between existing jobs if necessary.

Unlike transformation jobs, the extraction jobs generally do not depend on each other. Exceptions to this rule are those jobs that extract new or changed data and must run after a job that did the initial extract.

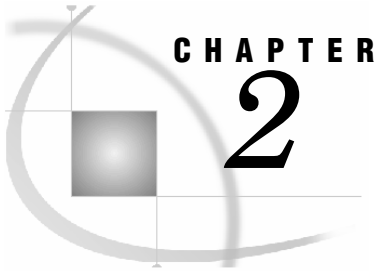
Accessibility

The SAS Solutions Adapter for SAP does not have a stand-alone GUI. It is configured and managed using features of SAS Data Integration Studio and SAS Management Console.

SAS Data Integration Studio and SAS Management Console include accessibility and compatibility features that improve their usability for users with disabilities. These features are related to accessibility standards for electronic information technology that were adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended.

For more information about specific accessibility features of SAS Data Integration Studio and SAS Management Console, refer to their respective documentation available from **`http://support.sas.com`**.

SAS is committed to improving the accessibility and usability of our products. If you have questions or concerns about the accessibility of SAS products, send e-mail to **`accessibility@sas.com`**.



CHAPTER 2

SAP Administration Tasks

Customizing for the Installation Environment

The SAS Solutions Adapter for SAP requires some customization to work in a unique install site. When customizing the SAS Solutions Adapter for SAP for your site, you must configure the SAS/ACCESS interface to R/3 software and customize the SAS Solutions Adapter for SAP for the assigned libraries in your installation environment.

Configuring SAS/ACCESS Interface to R/3 Software

Install Transports and Customize the Installation Environment

The SAS Solutions Adapter for SAP uses the SAS/ACCESS Interface to R/3 software. This requires extensive installation setup before use. For detailed installation instructions and configuration information, refer to the *Installation Instructions for SAS/ACCESS Interface to R/3* document included in your SAS software order.

In addition to the standard SAS/ACCESS Interface to R/3 configuration that is required, the following transport must be copied and installed on the SAP Server to accommodate the SAS Solutions Adapter for SAP:

Table 2.1 Specific Transport Required for the SAS Solutions Adapter for SAP

Transport	Objects	Purpose	To be applied to
SAPKA91302INSAS	Development class /SAS/ADDDS		SAP R/3 Systems
	Function group /SAS/ADDDS1 includes the function modules /SAS/FI_IMPORT_BAL_SHEET_POS and /SAS/FI_IMPORT_BAL_SHEET_TEXT	ADDDS1 for retrieving balance sheet information	
	Function group /SAS/ADDDS2 includes function modules /SAS/HR_READ_SALDO, /SAS/HR_READ_ZES, and /SAS/HR_EVALUATE_WAGES	ADDDS2 for retrieving salary information	
	Referenced structures /SAS/HR_CLUSTER_STRUCT_01 and /SAS/HR_EVALUATE_WAGES		

You can locate this transport in the **`\soladaptsap\sasmisc`** directory that is relative to the location where your SAS Foundation software is installed (for example, **`x:\Program Files\SAS\SAS 9.1\soladaptsap\sasmisc`**).

To customize the installation environment for the SAS Solutions Adapter for SAP, the SAP administrator must also complete the following four tasks:

- Define a role for SAS_HR_CLUSTER_EXTRACTOR.
- Define a destination to enable the extraction of HR cluster tables.
- Define a variant to enable the extraction of HR cluster tables.
- Define user permissions.

The following sections provide general instructions about how to perform these tasks. The specific steps that you must follow to complete each task might vary from these instructions based on your version of SAP software. For more information about how to perform these tasks in your specific environment, refer to the appropriate documentation for your SAP software.

Define a Role for SAS_HR_CLUSTER_EXTRACTOR

The steps below provide general instructions about how to define a role for SAS_HR_CLUSTER_EXTRACTOR.

1. Issue the transaction PFCG in the SAP GUI.
2. Create a role with the title SAS_HR_CLUSTER_EXTRACTOR.
3. From the authorizations section, select to change authorization data.
4. Manually supply the following data:
   - P_ORGIN
   - P_PCLX

5 In the HR: Master Data branch of the resulting tree, provide the following values:
   - Authorization level – R
   - Infotype – 0001
   - All remaining fields - *

6 In the HR: Clusters branch of the resulting tree, provide the following values:
   - Authorization level – R
   - Area Identifier for Cluster - *

7 Save the role.

Define a Destination

The steps below provide general instructions about how to define a destination to enable the extraction of HR cluster tables.

1 Issue transaction SM59 in the SAP GUI and supply the following values:
   - RFC Destination - SAS_HR_CL_EXTRACT
   - Connection Type – T
   - Description – Destination for SAS HR cluster extraction

2 In the section for technical settings, enable the Registered Server Program option and specify the value **`Program ID - SAS_HR_CL_EXTRACT`**.

Define a Variant

The steps below provide general instructions about how to define a variant to enable the extraction of HR cluster tables. You must complete these steps twice in order to create both the GENERAL and LIMITED variants. The LIMITED variant is used for testing purposes.

1 Issue transaction SE38 in the SAP GUI and provide the value Program Name - /SAS/HRCL_EXTRACT.

2 Specify the subobjects as variant.

3 Click **`Display`**.

4 Provide a Variant name of GENERAL or LIMITED. The macro variable **`hr_payroll_abap_variant`** in preprocparms12.sas should be set accordingly.

5 Click **`Create`**.

6 In the SAS HR Cluster Table Support section, change RFC Destination from BACK to SAS_HR_CL_EXTRACT (defined above).

7 In the subsetting section, provide the following data:
   - S_CST – RT
   - S_SRTZA – A
   - All other fields – blank

8 Consider appropriate subsetting conditions based on variant.
   - For the GENERAL variant, subset on the wage types parameter S_WTP to reduce the volume of data by an order of magnitude. For example, derived wage types starting with a forward slash (/) might not be of interest.
   - For the LIMITED variant, consider additional subsetting to enable faster testing.

9 Save the variant.

Define User Permissions

The SAP administrator must define and give appropriate authorizations to the SAP users who can log on and extract data using the SAS Solutions Adapter for SAP. The SAS administrator configures the SAS Solutions Adapter for SAP so that only the desired SAP users can log on.

CHAPTER 3

Customizing the SAS Solutions Adapter for SAP

Preparing to Import the SAS Solutions Adapter for SAP

Before you import the SAS Solutions Adapter for SAP, you must prepare your system environment by completing the following tasks:

- Define user and SAP Login details in SAS Management Console in a foundation repository.
- Define an SAP library and SAP server in SAS Management Console in a custom repository.
- Test and verify the resulting connection in SAS Data Integration Studio.

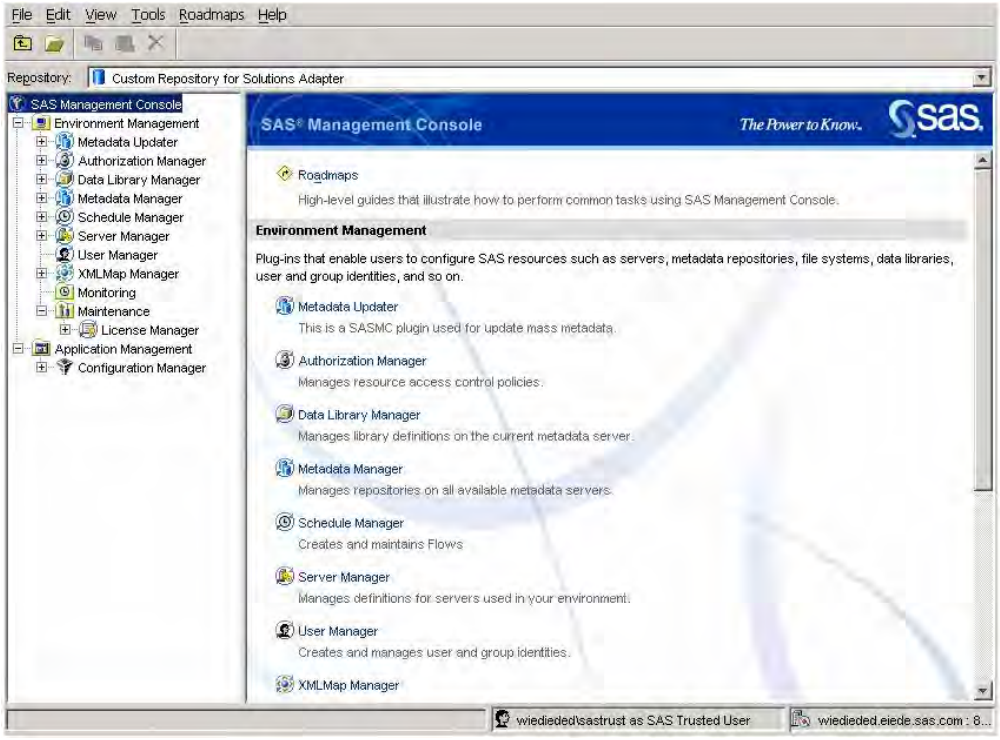
These tasks are required to ensure that the SAS Solutions Adapter for SAP works properly when it is imported. The remaining sections in this chapter provide instructions on how to complete the required tasks for preparing and customizing your system and business environment.

Defining the Environment

Define a User

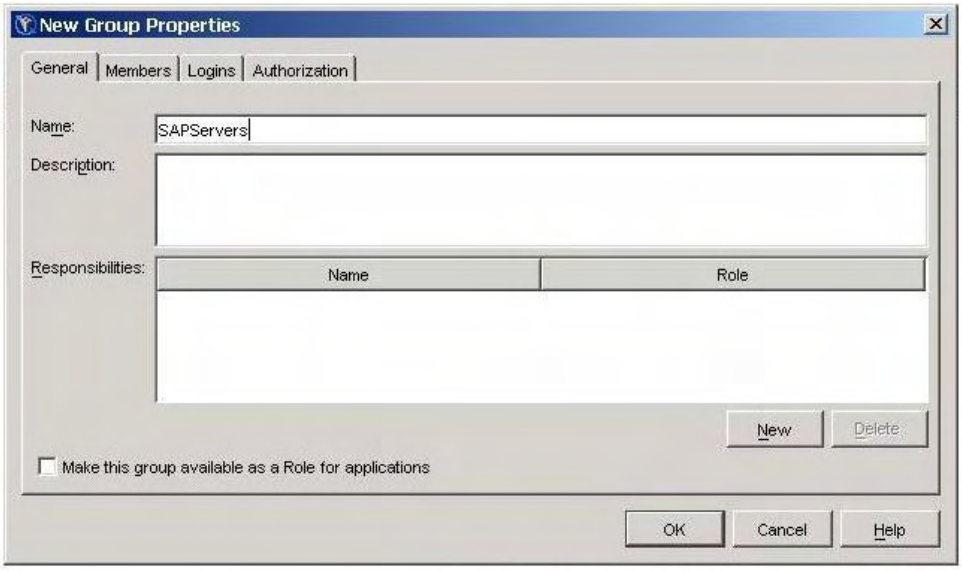
To define a user for the SAS Solutions Adapter for SAP, complete the following steps:

1 Open SAS Management Console.



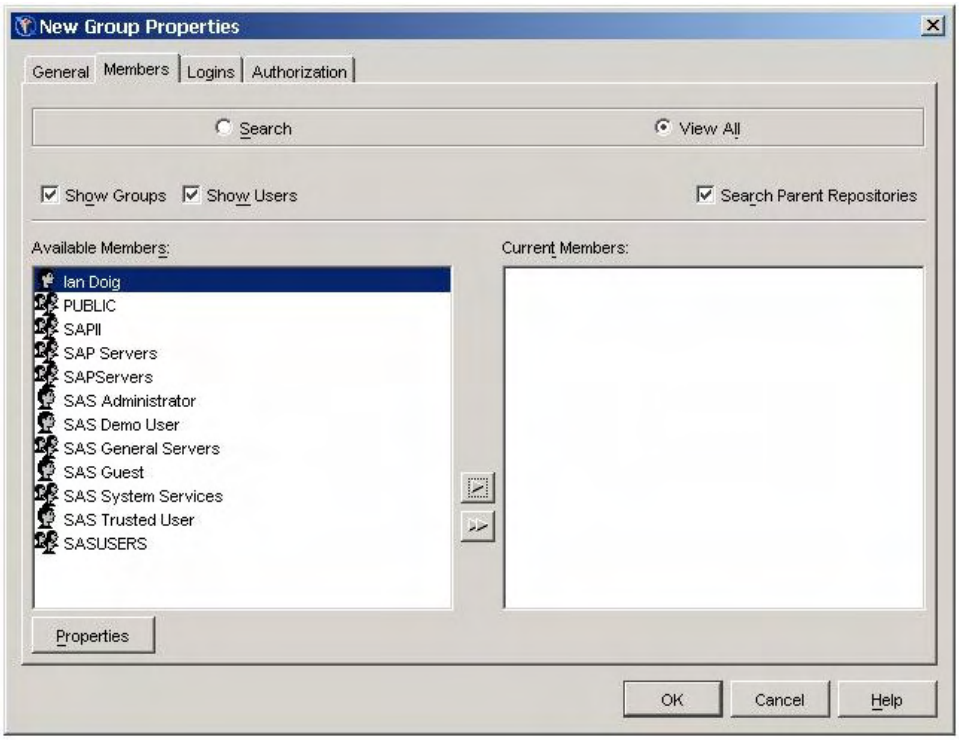
2 Select **`Foundation`** in the **`Repository`** field to connect to the metadata server at the foundation level. You must have administrator access to complete this task.

3 Right-click **`User Manager`** in the hierarchy tree on the left of the SAS Management Console window and select **New>Group**. A New Group Properties dialog box appears.



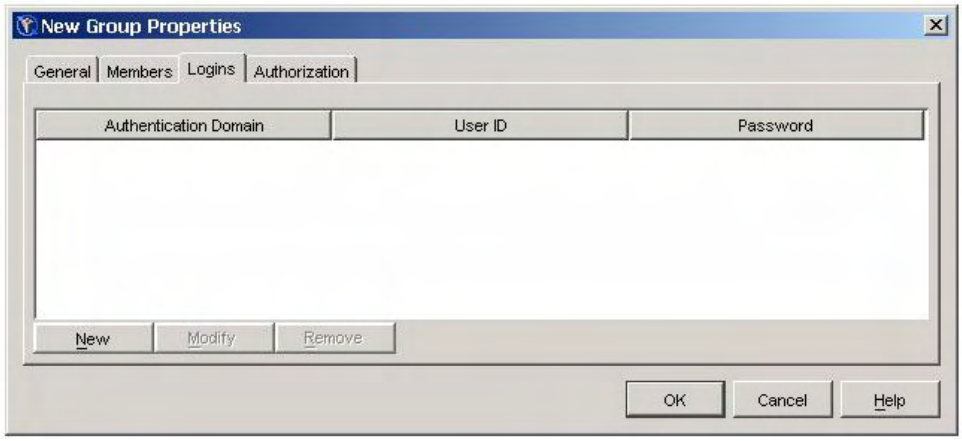
4 Enter **`SAP Servers`** in the **`Name`** field.

5 Click the **`Members`** tab.



6 Select **`SAS Trusted User`** in the **`Available Members`** box and click the arrow button that is pointing to the **`Current Members`** box. The selection moves the **`Current Members`** box.

7 Click the **Logins** tab.



8 Click **New**. The New Login Properties dialog box appears.

9 Complete the following steps to provide the necessary information for the New Login Properties dialog box:

- **a** Enter the user ID for the SAP Server.
- **b** Enter the password for the SAP Server.
- **c** Select **SAPAuth** from the **Authentication Domain** list. If this option is not already available on the list, click **New**, enter **SAPAuth** as the name for the new authentication domain, and enter a short description of your choice. Then, click **OK** to save the new authentication domain.
- **d** Click **OK** to save the new login properties.
- **e** Click **OK** to save the user definitions.

Define an SAP Library and SAP Server

To define an SAP library and SAP server, complete the following steps:

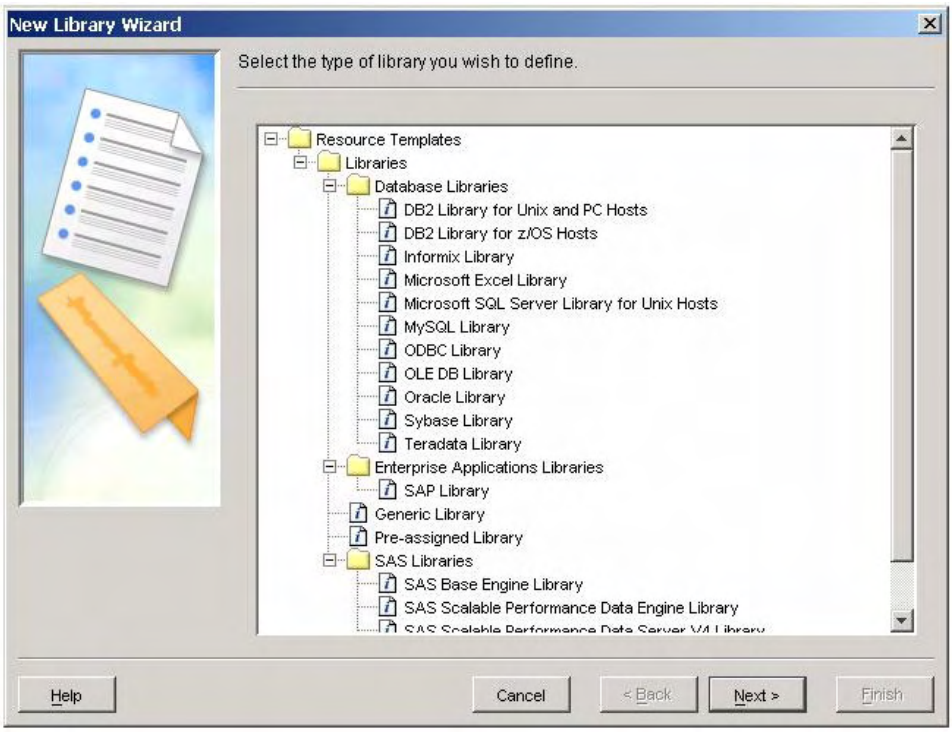
1 Open SAS Management Console.

2 In the **Repository** field, select a repository that is solely dependent on the foundation repository to connect to the metadata server at the foundation level.

Note: Defining an SAP library and an SAP server should occur in a custom repository that is solely dependent on the foundation repository. If you do not have a defined repository that meets this requirement, then you must add one. To do this, use the Metadata Manager in SAS Management Console to access the Active Server node and add a repository.

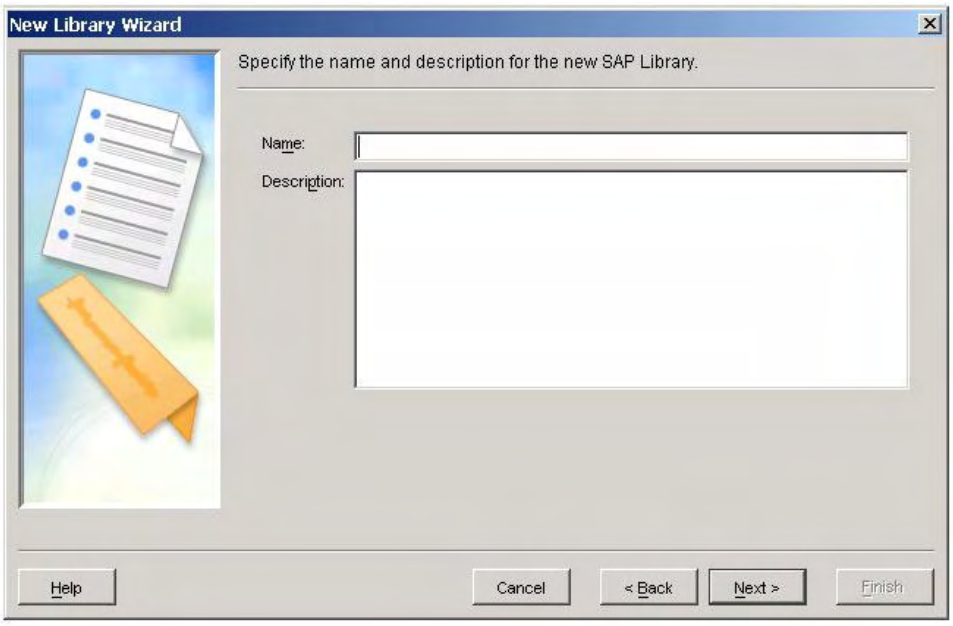
3 Expand the **Data Library Manager** option in the hierarchy tree on the left of the SAS Management Console window.

4 Right-click **SAS Libraries** and select **New Library**. A New Library Wizard dialog box appears.



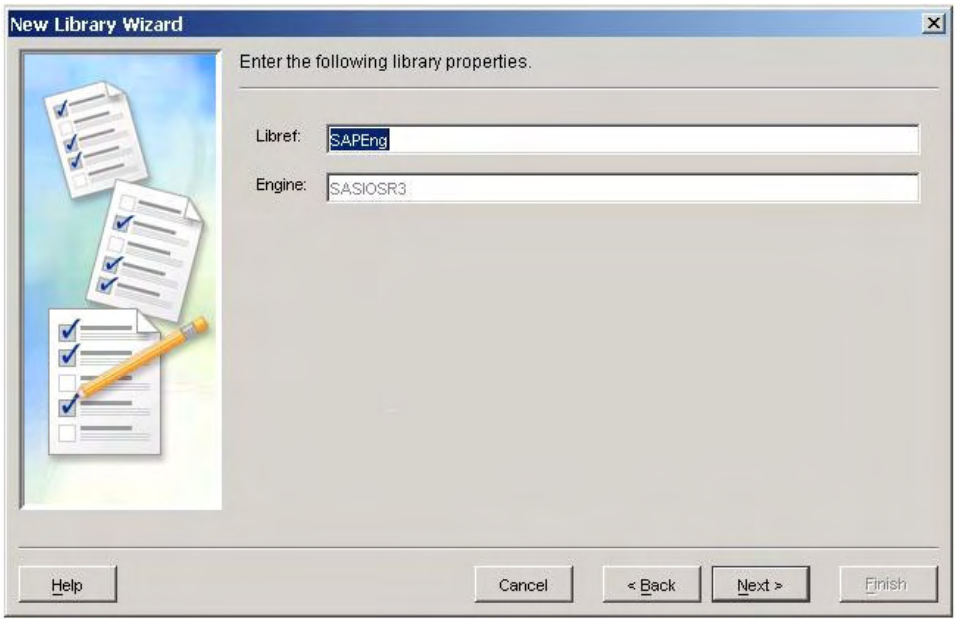
5 Select **SAP Library** from the **Enterprise Applications Libraries** folder.

6 Click **Next** to name the new SAP library.



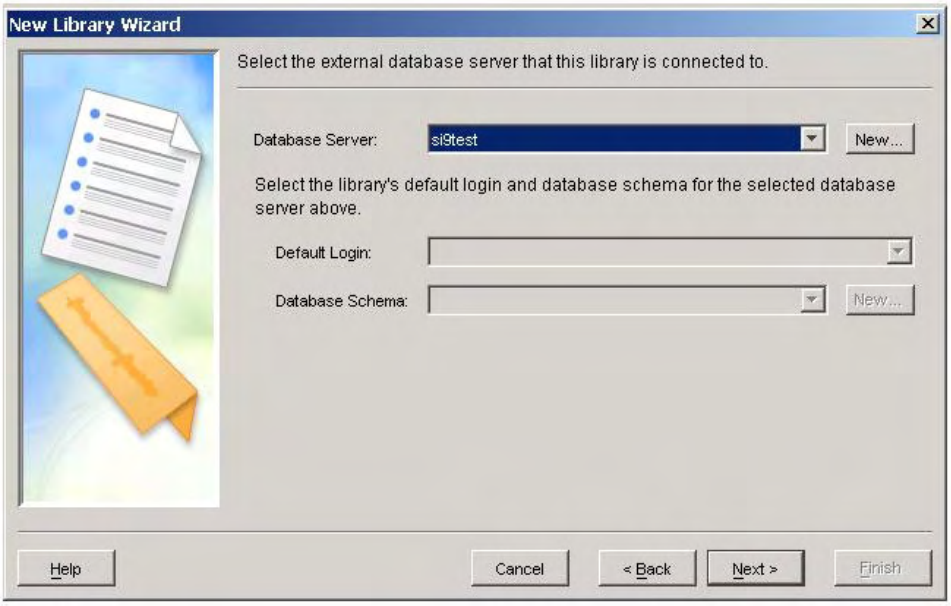
7 Enter a name and description for the SAP library.

8 Click **Next** to specify library properties.



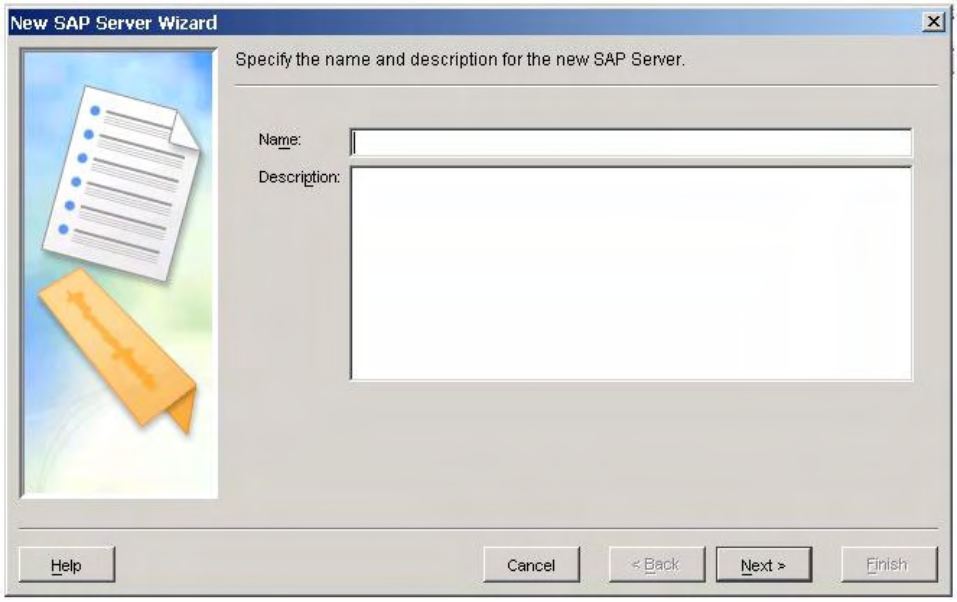
9 Enter **SAPSERVE** in the **Libref** field. SAPSERVE is the default value that the imported metadata uses.

10 Click **Next** to select a database server.



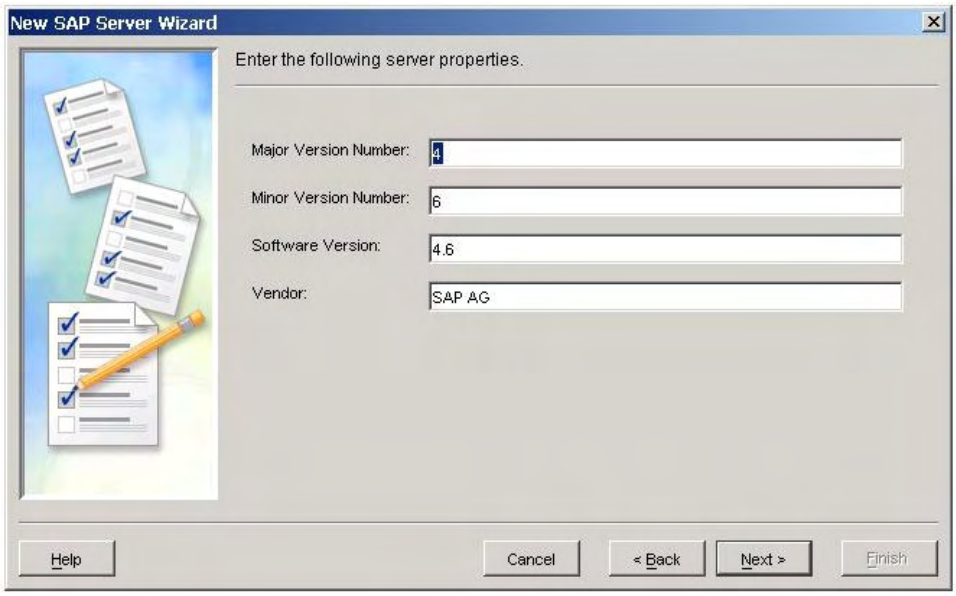
11 In the **`Database Server`** field, select a database server that contains parameters that connect to the SAP server. If an appropriate database server is not already available in the field list, complete the following steps to create one:

a Click **`New`** to open the New SAP Server Wizard.



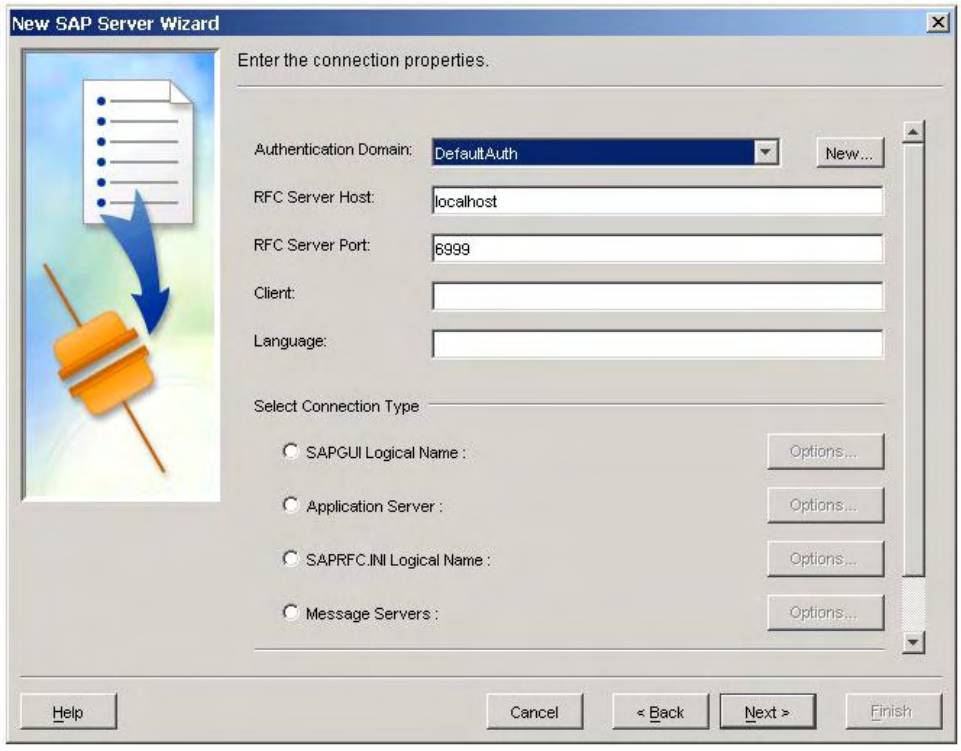
b Enter a name and description for the server.

c Click **`Next`** to enter information about your SAP software.



d Enter the version numbers for your SAP software. These values are optional.

e Click **Next** to enter connection properties.



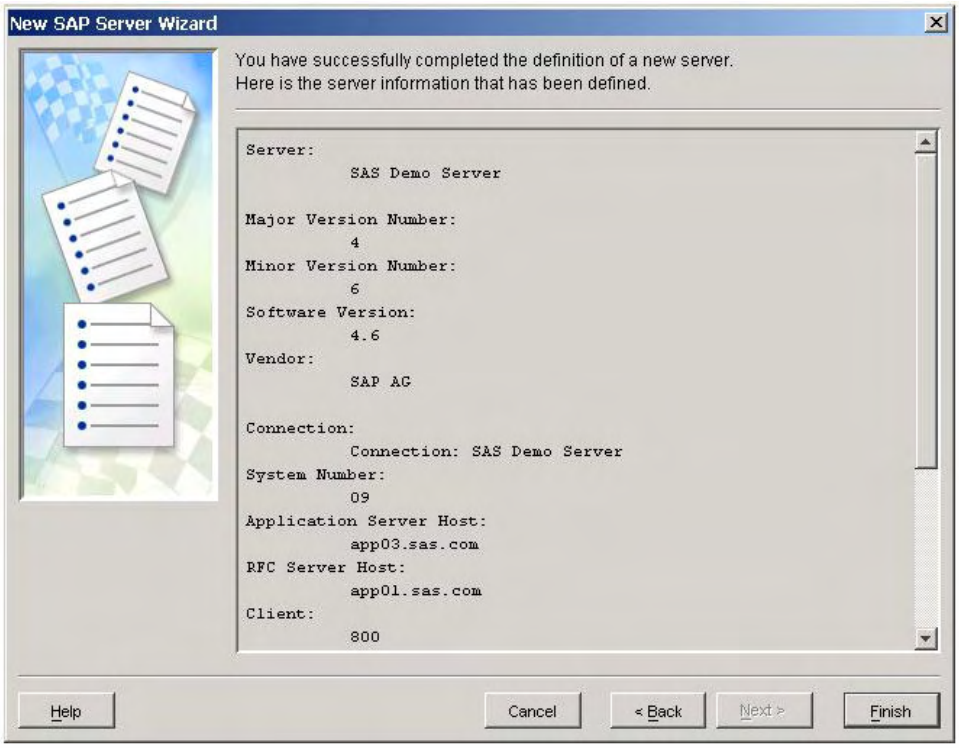
f Select **SAPAuth** in the **Authentication Domain** field.

g Specify the RFC server host, RFC server port, SAP client, and SAP language.

h Select the connection type that corresponds to a normal SAP GUI connection for your system. Then, click the **Options** button that corresponds to your selection and enter the required values.

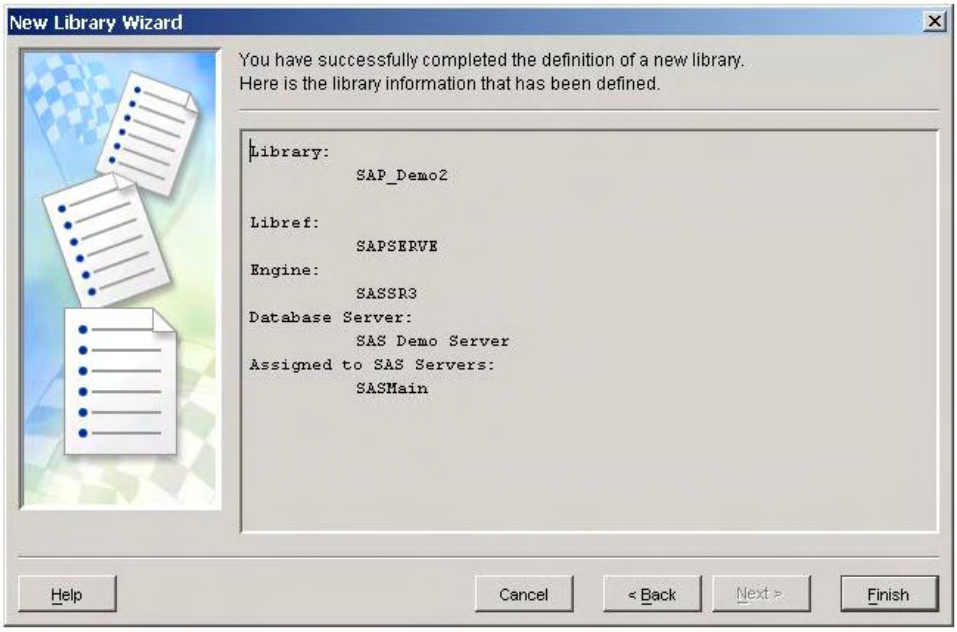
i Scroll down the page to access the **Advanced Options** button. If you have defined batch servers or processes, you might need to specify some of the advanced options that are available from this button.

j Click **Next** to display a summary page of your new specifications.



k Review the set parameters and click **Finish** to save the new SAP server and return to the New Library Wizard. Based on your system specifications up to this point, the wizard might prompt you to select a SAS server where the library is to be assigned. If this page appears, select **SASMain** and click **Next** to continue.

12 Click **Next** to review the library parameters.

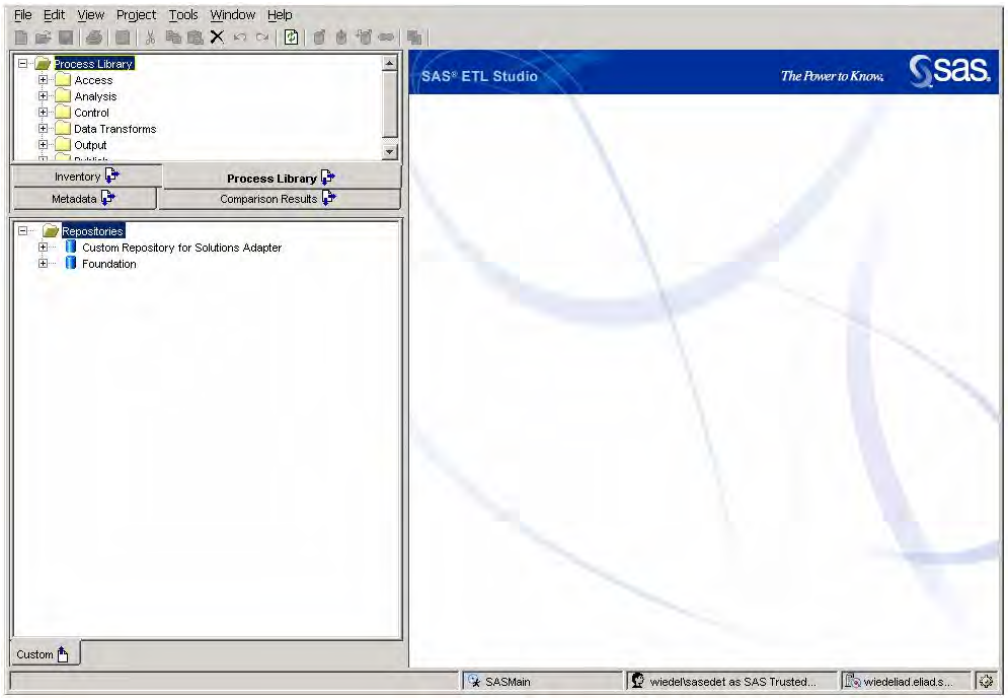


13 Press **`Finish`** to save.

Test the SAP Connection

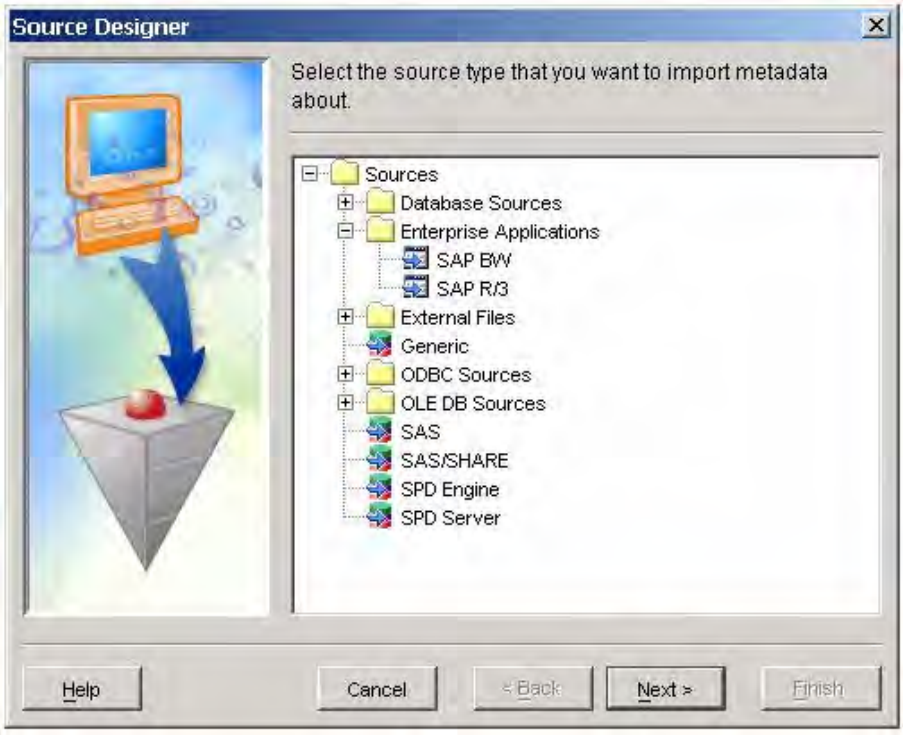
To test the SAP connection, complete the following steps:

1 Open SAS Data Integration Studio.



2 Open the metadata profile that corresponds to the user with access to the custom repository.

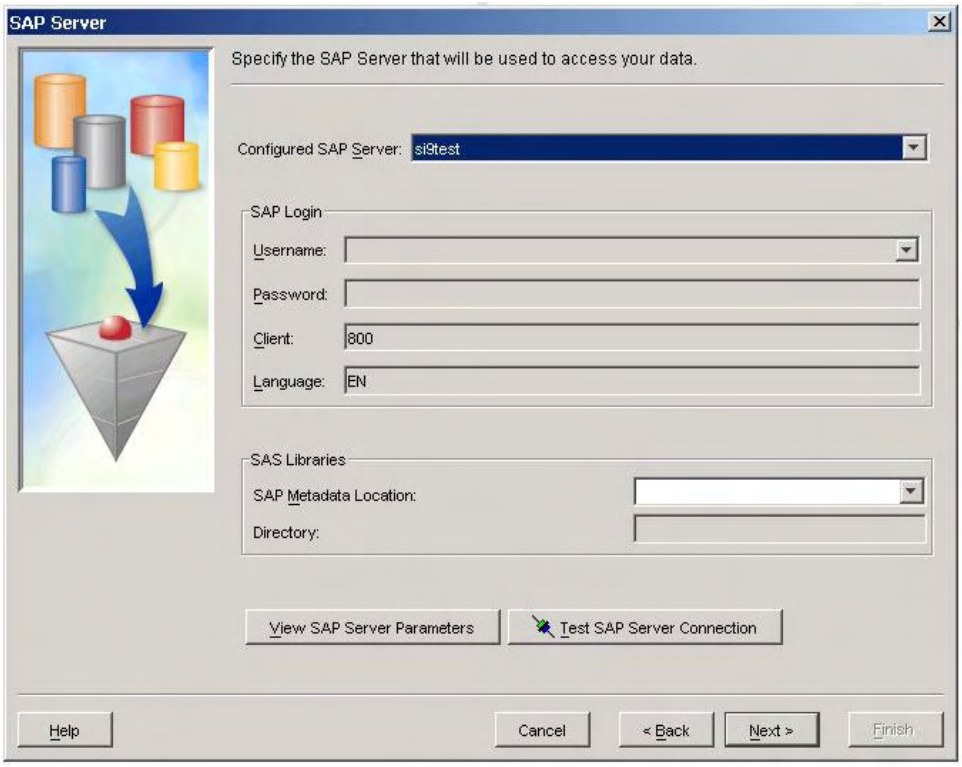
3 Select **Tools>Source Designer**. The Source Designer dialog box appears.



4 Expand the **`Enterprise Applications`** folder and select the **`SAP R/3`** source type.

5 Click **`Next`** to select an SAP server.

Note: Based on your system specifications up to this point, the wizard might prompt you to select a SAS server first. If this page appears, select **`SASMain`** and click **`Next`** to continue.

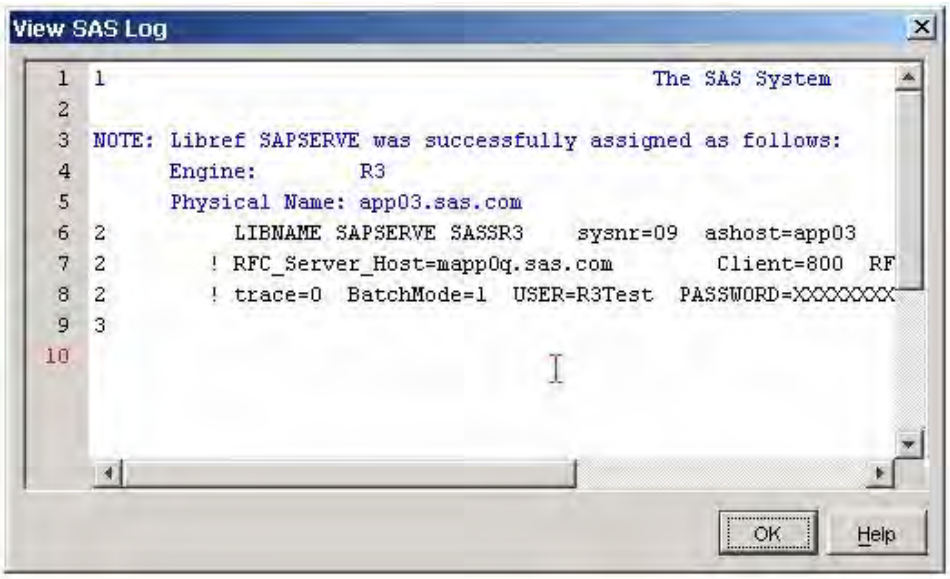


6 In the **`Configured SAP Server`** field, select the name of the SAP server that you defined in SAS Management Console.

7 Click **`Test SAP Server Connection`**. If the connection is successful, then the following message will appear:

Note: If the connection is not successful, the SAS log indicates the problem. For example, RFC server errors might indicate that the RFC server is not running or an invalid User ID might indicate that the password is incorrect. For more information about troubleshooting the connection, refer to the *Installation Instructions for SAS/ACCESS Interface to R/3* document that is included in your SAS software order.

8 Click **Yes** to view the SAS log.



The View SAS Log dialog box displays the LIBNAME statement that is generated using the specified parameters.

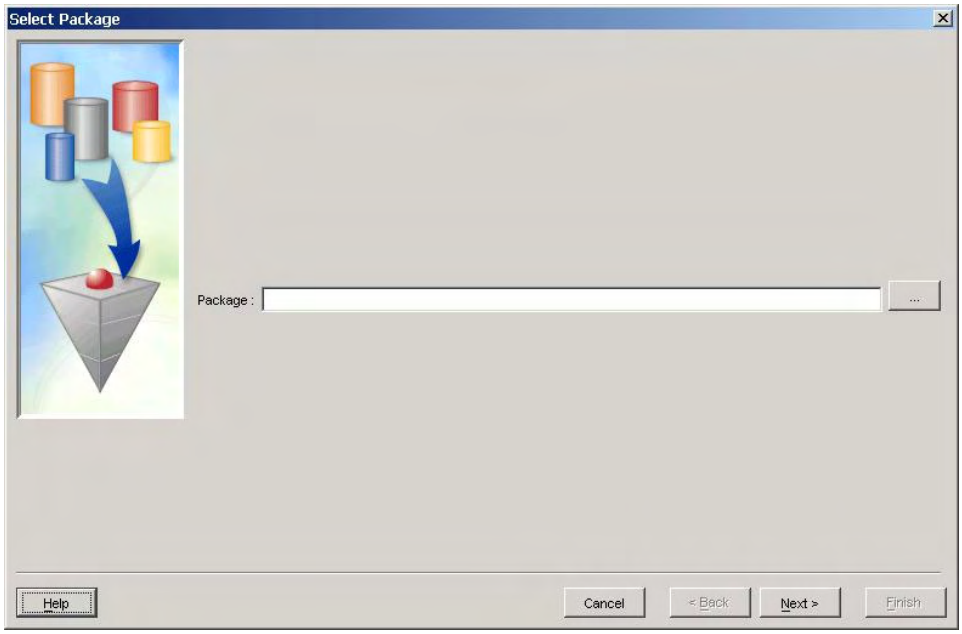
9 Click **OK** to exit the SAS log.

10 Click **Cancel** to exit the Source Designer.

Deploying SAS Solutions Adapter for SAP Jobs and Tables Metadata

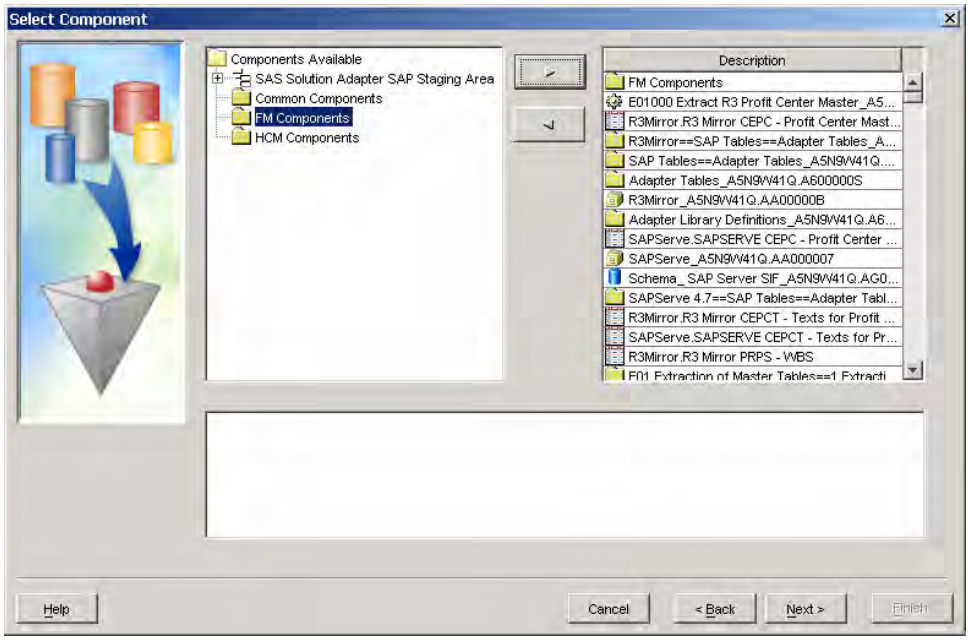
To deploy jobs and tables metadata for the SAS Solutions Adapter for SAP, you can use the Solution Package Deployment Wizard that is available from SAS Management Console. To access this wizard and perform the required configurations, complete the following steps:

1 Open SAS Management Console and select **Tools>Solution Package Deployment Wizard**. The Select Package page of the Solution Package Deployment Wizard appears.



2 Click the button to the right of the **Package** field to browse for the location of the **SolutionsAdapterInstall_13.ipk** package file. Typically, you can find this package file in the location that was designated during initial configuration when the file was extracted from the installation CD. The standard location is X:\Program Files\SAS\SASSolutionsAdapterSAP\1.3\ETLComponentsPackage.

3 Click **Next** to access the Select Component page.



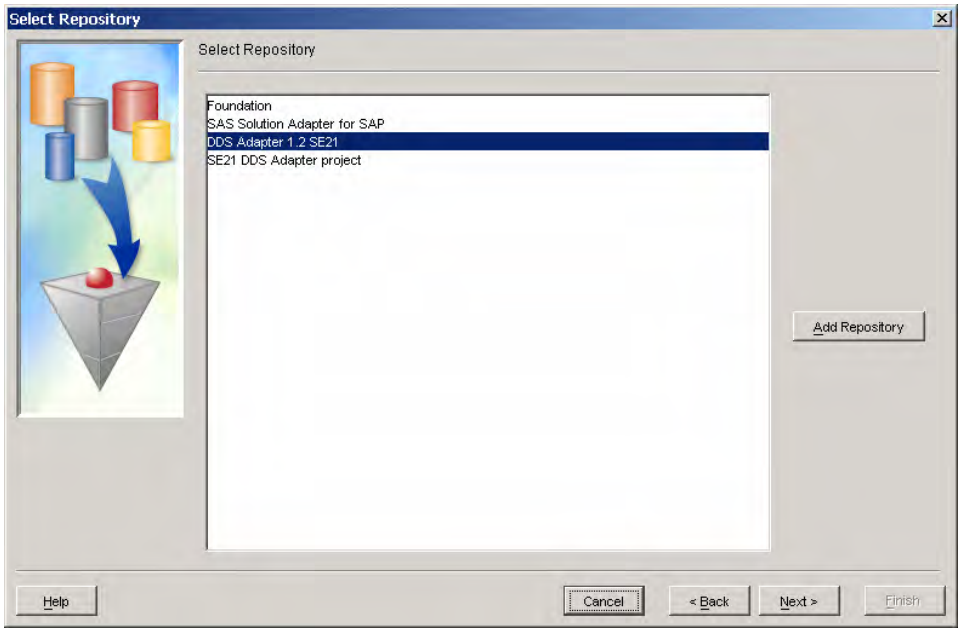
4 Click to select one or more components or groups of components in the Components Available folder.

Groups of components are contained in subfolders, and the Common Components subfolder contains components that are shared by component groups.

When you select a group of components, all related components are automatically selected as well. For example, selecting the FM Components subfolder automatically selects the necessary components in the Common Components subfolder as well as all the dependent data that is related to jobs and tables for those FM components.

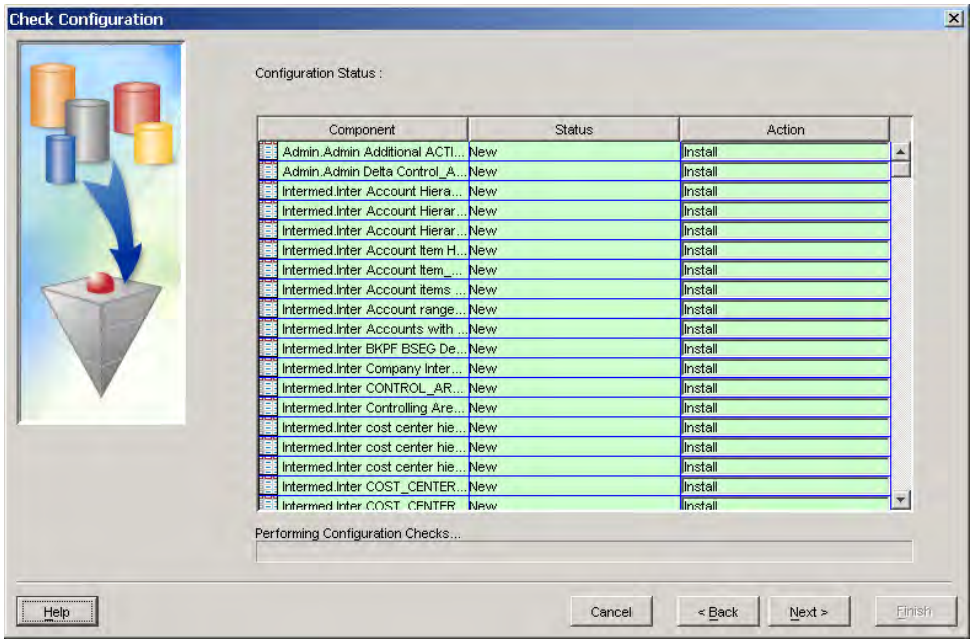
5 Click the arrow button that is pointing to the **Description** box on the right of the Select Component page. The highlighted component(s) appear in the list of selected components in the **Description** box.

6 Click **Next** to continue to the Select Repository page.



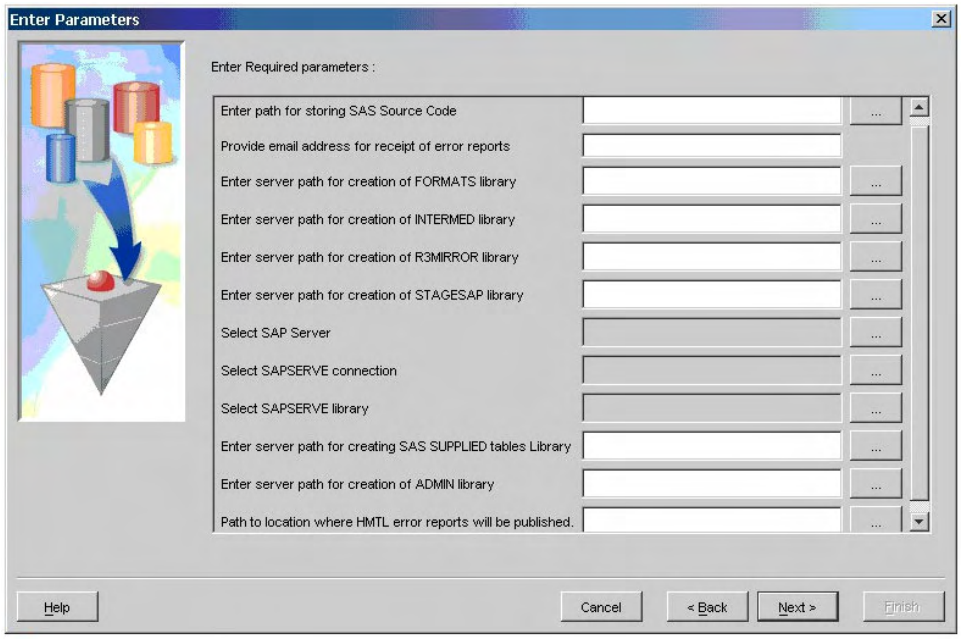
7 Select the custom repository that you defined when you added a server and SAP library in the previous sections. The repository that you select in this step is where you want to import the SAS Solutions Adapter for SAP.

8 Click **Next** to continue to the Check Configuration page.



9 Review the current settings for each component that is listed. A new installation shows a Status of **New** and an Action of **Install** for each component. Click **Help** on this page for more information about alternative options when adding to or modifying an existing SAS Solution Adapter for SAP installation.

10 Click **Next** to continue to the Enter Parameters page.



11 Enter the appropriate values for each parameter that is listed on the Enter Parameters page. These required parameters enable you to define the SAS Solution Adapter for SAP environment.

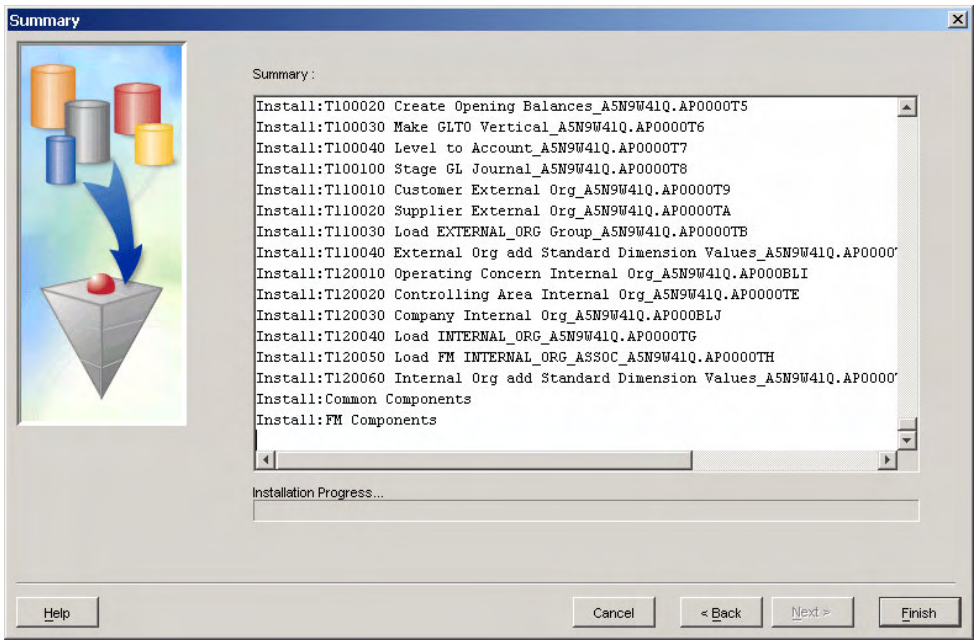
This page includes the following two types of data-entry fields that you can use to enter required values:

- White, editable fields that enable you to enter free-form data.
- Dimmed fields that require you to use the buttons on the right of the fields to locate and select your desired value. For best results, ensure that each dimmed field has a unique value.

For more information about the parameters that you might need to specify on this page, refer to Appendix 1 "Solution Package Deployment Parameters."

Note: Before proceeding to the next page, you might choose to note your parameter values for future reference.

12 Click **Next** to continue to the Summary page.



13 Review the actions that occur as a result of the parameter values that you specified on the previous pages. Click **Finish** to start deploying the components. A progress bar provides feedback and a pop-up window indicates when the process is complete.

14 When the process is complete, close SAS Management Console.

The business environment for data extraction and transformation from SAP is now installed. You can now customize your business environment further in the SAS Data Integration Studio environment as needed.

Customizing SAS Data Integration Studio Jobs and Programs

In order to customize the SAS Solutions Adapter for SAP to fit into your unique business environment, you must review and change several jobs in SAS Data Integration Studio. Several SAS programs within these jobs also require your review and customization.

Although many of the jobs and programs do not require changes, you must review each in order to ensure the results that you desire. This document provides information about each job and program that requires your review and changes. It also describes each job in SAS Data Integration Studio and explains if the job or program requires customization to work successfully.

Customizing the preprocparms12.sas Program

About the preprocparms12.sas Program

The jobs for the SAS Solutions Adapter for SAP use several global macro variables that are initialized in the SAS program preprocparms12.sas. This program serves as the preprocessing step in each job.

The following section describes each part of the preprocparms12.sas program and explains the parts of the program that you must change in order to fit your business environment.

Note: Be sure to make a backup copy of the file before changing the SAS program.

File Location

You can find the preprocparms12.sas program in the install directory. The file location is the one that you specified in the field **`Enter the path for SAS source code`** on the Enter Parameters page of the Solution Package Deployment Wizard in SAS Management Console.

Note: For more information about where and how this location is specified, refer to the following two sections of this document:

- Deploying the SAS Solutions Adapter for SAP Jobs and Table Metadata
- Appendix 1: Solution Package Deployment Parameters

The preprocparms12.sas program file is called by the preprocglobal.sas program. The preprocglobal.sas program does not require editing to fit the local environment.

```
/*************************************************************************/
/* Copyright (c) 2004, 2005 by SAS Institute Inc., Cary, NC 27513, USA   */
/*                                                                       */
/* Name:          preprocglobal.sas                                      */
/*                                                                       */
/*************************************************************************/

%put NOTE: Executing SAS Solution Adapters 1.2 for SAP - Implementation-Version
20051004;

/* Global Preprocessing */

proc display c=sashelp.adptsap.startup.scl batch;
run;

options mstored mrecall;

filename macro '$SAPDDSSOURCE$';
options sasautos = (macro sasautos );

%global spras ktopl versn sakln gjahr _ktopl _versn _start
        dds_source_system_id
        wanted_controlling_areas operating_concern ;


%include "$SAPDDSSOURCE$/preprocparms12.sas";

options fmtsearch = (fmtsearch library.fmt&lang library.fmt&_ktopl.&lang);

/*
 * End of preprocglobal.sas
 */
```

After you locate and open preprocparms12.sas, you can see the banner for the SAS program file. You do not need to change any settings in this section to fit your local environment.

SAP System Parameters

Data Source

After the program file banner, the next section of preprocparms12.sas designates the data source location.

```
%let dds_source_system_id=800; * 3 char ID to tell DDS where info
                               * is coming from.  It could be SAP,
                               * or client number - or SAP sysid - anything
                               * that uniquely represents the current
                               * ETL ;
```

The macro variable **dds_source_system_id** is a three-character identification that informs the SAS Detail Data Store (DDS) where the data is coming from. The value can be anything that uniquely represents the current ETL environment such as SAP, the SAP client number, or the SAP sysid.

This variable populates the source_system_cd column that occurs in multiple DDS tables. If data originates from multiple SAP systems, multiple extracts are necessary with each one having its own preprocparms12.sas file and unique value set for the variable **`let dds_source_system_id`**.

SAP Queries

The SAP queries section of the program enables you to view information about queries that are passed directly to SAP.

```
* Use only when testing to see what is passed to SAP ;
* options debug=dbms_select;
```

The statement **`options debug=dbms_select`** shows detailed information about queries that are passed directly to SAP. Generally, this section is not needed unless you suspect that SAP server-side joins take longer than expected to run. This section provides additional tracking that might help solve problems if necessary.

SAP Language Codes

Use the SAP Language Codes section of the program file to designate the language that SAP uses.

```
%let spras    = 'E';     * single-char SAP language with quotes;
%let lang     = E;       * language without quotes;
* set valid SAS language value to be used in the
* solution data mart ;
%let sas_lang_for_sdm='en';
```

Set the macro variables **`spras`** and **`lang`** to the single-character language code that corresponds to the main language that SAP uses. This is also the language that the program uses to maintain texts. Quoted and unquoted versions exist for easy inclusion into code and for making the code easy to understand and read.

Use the following language code table to determine the valid language value for your local environment.

Table 3.1 Language Codes for preprocparms12.sas

Code	Language	Code	Language
0	Serbian	I	Italian
1	Chinese	J	Japanese
2	Thai	K	Danish
3	Korean	L	Polish
4	Romanian	M	Chinese (traditional)
5	Slovenian	N	Dutch
6	Croatian	O	Norwegian
7	Malaysian	P	Portuguese
8	Ukrainian	Q	Slovakian
9	Estonian	R	Russian
A	Arabic	S	Spanish
B	Hebrew	T	Turkish
C	Czech	U	Finnish
D	German	V	Swedish
E	English	W	Bulgarian
F	French	X	Lithuanian
G	Greek	Y	Latvian
H	Hungarian	Z	Customer Reserve

The macro variable **sas_lang_for_sdm** is the SAS version of the language. This code comprises two characters in lowercase and must be in the SAS supplied table SASHELP.LANGUAGE.

SAP Client

The value of the **let client** variable is the SAP client (column MANDT) that is normally part of the SAP R/3 logon.

```
%let client   = 800 ;    * = MANDT  ;
```

Common Extraction Parameters

E-mail Address for Recipient of Error Reports

The E-mail Contact for Errors Report section of preprocparms12.sas indicates the e-mail address that receives error notifications for any extractions or transformation flows that use a publish-to-e-mail transformation. The initial value of this variable is obtained from the parameters that are supplied in the import steps.

```
*---Email contact for error reports---*;
%let emailerror=xxx@yyy.zzz.domain;
```

Cost Center / Profit Center hierarchies

By default, the SAS Solutions Adapter for SAP extracts the standard hierarchies for cost center and profit center. You can select an alternative hierarchy by specifying alternative hierarchy roots in the macro variables **cost_center_hier_root** and **profit_center_hier_root**.

To accept the default standard hierarchies, leave the macro variables blank.

```
* Set values if you want to override the default hierarchy;
%global cost_center_hier_root profit_center_hier_root;
%let cost_center_hier_root=;
%let profit_center_hier_root=;
```

Company

If you choose to extract only data from SAP for selected companies, you can specify the **wanted_company_codes** macro variable to contain a list of the desired company codes. The list of valid values are in the SAP table TKA02. The companies that you specify should all belong to controlling areas that are selected by the macro variable **wanted_controlling_areas**.

```
* Select wanted company codes from TKA02 -
  List must be entered as quoted company codes
  separated by commas ;
%let wanted_company_codes='1000' , '2000' , '6000';
```

Euro Currency Conversion

The SAS Solutions Adapter for SAP can convert employee compensation amounts that are paid to employees in pre-euro currencies so that all the amounts are in euros. The macro variables **convert_currency_to_euro** and **pre_euro_currency** control this conversion. This conversion is performed because the salary results table does not explicitly contain currency information.

The macro **convert_currency_to_euro** needs to have a value YES or NO. The macro **pre_euro_currency** must be set to one of the standard international currency codes. These codes are three-characters long. The macro **pre_euro_currency** is used only when the macro **convert_currency_to_euro** is set to YES.

Note: The SAS Solutions Adapter for SAP does not use the euro currency conversion macro variables in areas other than the calculation of employee compensation. However, you might choose to use these macro codes elsewhere.

```
/*
 * Currency amount prior to 2001 may be converted from pre-euro currencies
 * to euro using the standard fixed exchange rates.
 * To switch on this conversion the macro variable convert_currency_to_euro
 * has to be set to YES, pre_euro_currency has to be set to the pre-euro currency
 * as 3-character ISO code like DEM, FFR, BEF,...
 * Note that only a single currency is assumed.
 */
%let convert_currency_to_euro = NO;
%let pre_euro_currency = DEM;
```

SAS Financial Management Parameters

Chart of Accounts

The SAS Solutions Adapter for SAP handles a single chart of accounts at a time. You must specify this chart of accounts in the macro variables **ktopl** and **_ktopl**.

A chart of accounts in SAP is a list of all general ledger accounts that one or more company codes uses. For each general ledger account, the chart of accounts contains the account number, account name, and the information that controls how an account functions as well as how it is created in a company code.

```
%let ktopl    = 'INT';   * chart of accounts ;
%let _ktopl   = INT;     * chart of accounts without quotes;
```

If you need more than one chart of accounts for your environment, you must set up additional instances of the SAS Solutions Adapter for SAP and specify the relevant values. In such a case, be sure to select unique values for the data source macro variable **dds_source_system_id**, as shown in the Data Source section of preprocparms12.sas.

To identify a correct value for the chart of accounts, refer to the contents of the SAP R/3 table T004T. This table contains the names of the charts of accounts based on language dependency. You can apply a WHERE clause to select the appropriate language. The chosen value of ktopl must also appear in the ktopl column in table T001.

Once you have determined the correct variable for the chart of accounts, enter the appropriate value. You must place single quotation marks around the **ktopl** value. Do not place quotation marks around the **_ktopl** value.

Financial Statement

The SAS Solutions Adapter for SAP handles one financial statement at a time. You must specify this financial statement in the macro variables **versn** and **_versnl** as shown below.

```
* Versn is a value taken from T011;
%let versn    = 'ERL';   * Financial statement version Version ;
%let _versn   = ERL;     * Financial statement version without quotes;
```

To identify a correct value for the financial statement, refer to the contents of the R/3 table T011. You can refer to T011T for text descriptions if necessary. You can also apply a WHERE clause to T011 to select the **ktopl**.

Using these references, you can choose and enter one of the matching **versn** values. You must place single quotation marks around the **versn** value. Do not place quotation marks around the **_versn** value.

Digits in the Account Number

The value for **sakln** determines the number of digits in the account number. If you do not know this value, refer to R/3 Table T004. The SAKLN column in this table corresponds to the chart of accounts (KTOPL) chosen above.

You might also consider that valid account numbers often contain leading zeros. You can choose to drop these leading zeros from the account digit value.

```
%let sakln    = 6;       * number of digits in the account number;
```

Operating Concern

The preprocparms12.sas program file enables you to specify the operating concern in the macro variable **operating_concern**, and a corresponding country code in the macro variable **operating_concern_country_cd**.

```
* List of valid operating concerns is in
* TKEB,  language-dependent descriptions are in TKEBT ;
%let operating_concern=IDEA;
* specify country where the operating concern HQ is located;
%let operating_concern_country_cd=DEU;
```

To identify a correct value for the operating concern and country code, refer to the R/3 table TKEB. Descriptions based on language dependencies are in TKEBT.

External Organizations

The macro variable **ext_org_assoc_type_cd** specifies the default value for external organizations that are not found in the customer and vendor hierarchies from SAP. This value populates the external_org_assoc_type_cd column in the external_org_assoc_type table of the staging area.

```
* Specify ext_org_assoc_type_cd to use as the default value for
  external organizations that are not found in the customer and
  vendor (supplier)hierarchies from SAP;
%let ext_org_assoc_type_cd=A;
```

The default SAP value for **ext_org_assoc_type_cd** is **A**. To see other valid values, refer to the R/3 tables THITT and TLHITT.

Controlling Areas

You can use controlling areas to form the second level of the internal organizational hierarchy. You can also use them to select cost center and profit center standard hierarchies. In addition, you must specify the corresponding country for each controlling area.

```
%let wanted_controlling_areas=1000 6000;
%let controlling_area_countries=DEU MEX;
```

To set the variables for controlling area, you first need to specify a list of controlling areas that are required in the macro variable **wanted_controlling_areas**. This list should contain values from the KOKRS column in the TKA01 table. When obtaining the list, make sure that the controlling areas are all in the same operating concern. For example, select only values for **KOKRS** where **ERKRS="&operating_concern"**.

Once you have selected the controlling areas that are required, you need to indicate the corresponding countries for each. Each country is identified with a three-digit ISO code in **sashelp.sas_country**.

Use the macro variable **controlling_area_countries** to specify the corresponding countries to each of the desired controlling areas. The first country code must correspond to the first controlling area, the second country must correspond with the second controlling area, and so forth.

R/3 Table GLT0 Data Subsets

Use the macro variables **rldnr**, **rrcty**, and **rvers** to subset the data that is extracted from the R/3 table GLT0. The GLT0 table contains figures that are summed by transaction.

```
* Define macro variables used as "constants" in the
* where clause of R/3 table GLT0 .    ;
%let rldnr=00;    * Ledger ;        * see T881 or T881T for value ;
%let rrcty=0;     * Record Type ;   * 0=Actual,  1=Planned;
%let rvers=001;   * Version ;       * see T894 or T894T
                                    * (apply WHERE clause where rldnr="&rldnr");
```

Although the default settings in this section are generally acceptable, be sure to check them against valid values.

- For the valid ledger values, refer to table T881 and table T881T.
- For record type, **0** is the typical value for actual data.
- For version values that are valid, refer to table T894. The version descriptions are in T894T.

Period Type for Exchange Rates

Use the macro variable **period_type_cd_for_exrates** to specify the period type that is to be associated with exchange rates. A list of valid values is in the SAS supplied table sashelp.sas_period.

```
%let period_type_cd_for_exrates=MO; * This value must be chosen from the table
                                    * sashelp.sas_period_type. This is used
                                    * in the jobs that build table
                                    * STAGE_CURRENCY_EXCHANGE_RATE;
```

Financial Statement

The financial statement that is extracted from SAP is used to build the accounts dimension hierarchy. This hierarchy has no common root because it consists of disjointed subtrees. If necessary, you can add a common root by specifying a description for it in the macro variable **Financial_statement_root_txt**.

Note: It is recommended that you leave this macro variable blank.

```
* The following macro variable is used to specify text for the
* top combination node of finanancial statement hierarchy (root)-
* Otherwise the hierarchy contains disjoint subtrees.;
%let Financial_statement_root_txt = %nrquote(Financial Statement);
```

Fiscal Year Variant

SAP enables companies to have multiple fiscal year variants, although most companies only use one. If your organization uses multiple variants, you must include the variant name in period IDs and descriptions to avoid ambiguity. For best results, be sure to use clear IDs and descriptions that don't include the variant information.

The macro variable **use_fiscal_yr_variant_in_time** can be used to control behavior based on the following two values:

- The value N causes the variant not to be used.
- The value Y forces the variant information into the IDs and descriptions.

```
* The use_fiscal_yr_variant_in_time macro variable determines whether the
*  fiscal variant name is use in the time dimension ID and descriptions.
* In most cases, only one fiscal variant is used, and the default is
* therefore N ;
%let use_fiscal_yr_variant_in_time=N;
```

Extraction Starting Date

You can use the macro variables **_start** and **_startfinyear** to set the starting date for the extraction of financial transactions. These values help reduce the amount of data that is extracted in the initial load by not extracting transactions with dates before the specified date.

```
* If the financial year starts in January then use value 1,
  February then use 2, etc. ;
%let finyear_startmonth=1;

* The following macro variables determine from which date financial
* transactions
* should be extracted from SAP.  This is IMPORTANT when
* initializing the data
* during the initial load phase.  In subsequent extracts,
* only new/changed data
* is extracted;
%let _start = 01JAN2000;   * Start date - First extraction to
                           * take transactions after this date;
%let _startfinyear = 2000;

%let _opening_balance_period=200001; * Opening balance period.
                                     * This must be in the
                                     * format YYYYMM - It should usually
                                     * be the period of _start macro
                                     * variable above.
                                     * If it is set to blank, then No
                                     * opening balance
                                     * is extracted.  ;
```

Set the extraction variables based on your specific business environment.

- The **finyear_startmonth** value is the number of the month in which the financial year starts. January is represented by 1, February by 2, and so forth.
- The **_start** value is the first day of your organization's financial year. This value must be in the SAS DATE9 format.
- The **_startfinyear** value is your organization's financial year. For example, if your organization's financial year for 2005 begins on 01 December 2004, then you must set the **_startfinyear** value to **2005**. Choose a value that reflects the year after which detailed transaction data is to be extracted from SAP.

Determine Current Fiscal Year

The current fiscal year and time period is derived from the current date.

```
*--- Determine the current Fiscal year ---- and period *;
data _null_;
today=today(); * FOR TESTING PURPOSES can be set to a fixed date e.g.'01JAN2000'd;
fiscal_year=year(today);
call symput('GJAHR',put(fiscal_year,4.));
month=month(today);
if month=1 then do;
   prevmonth=12;
   prevmonth_fiscal_year=fiscal_year-1;
end;
else do;
   prevmonth=month-1;
   prevmonth_fiscal_year=fiscal_year;
end;
call symput('current_time_period',put(fiscal_year,4.)!!put(month,z2.));
call symput('previous_time_period',
              put(prevmonth_fiscal_year,4.)!!put(prevmonth,z2.));
run;

* Set the number of years into the future the time dimension should
  cover ;
%let time_dim_future_yrs=10;
```

Security Buffer for Re-extraction

The macro variable **overlap** provides a security buffer that enables you to re-extract data that has changed during the overlap since the last extraction of financial transactions.

The **overlap** value, in number of days, should not be set to less than two because that is the minimum time needed to correctly handle overnight processing and time zone differences. The current default value is four days.

```
%let overlap=4;  * Set number of days to re-extract financial documents;
                 * This may be needed to catch late processing, long
                 * running processes, etc ;
                 * The value should not be made lower than 2!;
```

SAS Human Capital Management Parameters

HR Start Date and Buffer for Re-extraction

This section of preprocparms12.sas enables you to limit HR extractions so that they only include records after a certain date. You can also specify the overlap period for extractions. The default overlap period for extractions is 400 days, slightly more than one year. This default is based on the assumption that historical HR data is correct after it has been on record for more than one year.

```
%let hr_start = 01JAN1900;  * Select an initial date for HR delta extractions.
                            * The date is used to limit HR infotype extractions
                            * with change / begin dates after &hr_start ;

%let hr_overlap=400; * Set number of days to re-extract HR infotype documents;
                     * This should be set to be in range such
                     * that records that were last changed more than
                     * the specified number of days ago will not be re-extracted.
                     * It is necessary to re-extract, since the begda/endda are
                     * keys, but may change between extracts, thereby losing the
                     * link to the initially extracted record.
                     * A value of 400 days is used so that it is slightly longer
                     * than a year;

* Start date for initial extraction of base salaries;
%let hr_base_salary_start = 01Jan1994;

%let absence_unit = DAY; * Unit for employee absences. Must be DAY or HOUR;
%* ISOLANDS: relevant countries in two-char quoted ISO codes as comma-separated list;
%let isolands = %str('DE', 'US', 'GB', 'IT', 'FR', 'ES', 'CA', 'JP', 'PT');
%let isolands = %str('DE');
```

Employee General Information

You can use this section of the preprocparms12.sas program file to include, exclude, or flag certain attributes of the employee data by supplying values to the macro variables from the corresponding tables. This section of the program file begins with the line shown in the box below.

```
*---Employee General Information Macro Variables---*;
```

Ethnicity Codes

To create ethnicity codes, the SAS Solutions Adapter for SAP uses the concatenation of the columns MOLGA and RACKY in the SAP lookup table T505S. The macro variable **minority_ethnic_types** must be initialized to identify the ethnic groups that are to be considered minorities.

```
* Ethnic Minorities - list groups considered to be ethnic minorities;
%let minority_ethnic_types='1001','1002';
```

Union Codes

The SAS Solutions Adapter for SAP uses the column EMFSL in Infotype 57 (SAP table PA0057) to populate the employee's labor union affiliation. You can use a WHERE clause INFTY='0057' to find a list of the valid values in the SAP tables T521B or T521C.

Because infotype can also contain codes that are not related to unions, you must identify the codes that identify real unions. To do this, create a comma-separated list of quoted strings that represent the unions. To set the macro variable, look at the SAP table T521B, apply the WHERE clause INFTY='00057', select the rows with codes that represent unions, and enter the codes into the macro variable.

```
*Union - select only certain codes;
%let labour_union_codes='DIG CHEM','MMETAL';
```

Mobile and Daytime Telephone Numbers

The SAS Solutions Adapter for SAP populates employees' telephone numbers from the columns NUM01 to NUM06 in the HR infotype 0006 (SAP table PA0006). The SAS Solutions Adapter for SAP needs to know the type of communication that represents the daytime phone number by understanding the values in the associated columns COM01 to COM06.

You can find the list of valid values for COM0n columns in the SAP table T536B. To select the daytime number of the NUM0n columns, you must set the macro variable **daytime_phone_com_type_codes** to identify the appropriate daytime code(s), and set the macro variable **mobile_phone_com_type_codes** to identify the appropriate mobile phone number code(s).

```
*Mobile phone label - list of values in SAP table T536B;
%let mobile_phone_com_type_codes='CELL';


*Daytime Telephone label - list of values in SAP table T536B;
%let daytime_phone_com_type_codes='WORK','TEL2';
```

Disciplinary Action Codes

You need to specify the appropriate value for the macro variable **disciplinary_action_codes** so that it lists all action codes that are to be considered disciplinary actions. The SAP tables T548T and T549T provide lists of valid action codes with descriptions.

```
*Disciplinary Actions Codes - SAP tables T548t & T549T describe PA0041 & PA0000,PA302
respectively;
*codes to be listed in Macro variable should be in single quotes '' and the list
comma separated;
%let disciplinary_action_codes=;
```

Applicant Information Parameters

You can use this section of the preprocparms12.sas program file to specify code and status numbers that match those used by the SAP System.

To set the applicant_interviewed_flg column, the SAS Solutions Adapter for SAP needs to know the appropriate value of APSTA in the Applicant HR Infotype PB4000. The table T751B contains a list of possible values with text descriptions. You need to set the macro variable **applicant_interviewed_status** to an appropriate value based on this list.

In order to populate the rejection_reason_cd column, the SAS Solutions Adapter for SAP must know which applicant action code means a rejection. Ensure that the macro

variable **applicant_rejection_code** is set to one of the action codes in the SAP table T751E.

```
*---Applicant Information Macro Variables---*
* Applicant Interview Status;
%let applicant_interviewed_status='1';

*Applicant Rejected Flag;
%let applicant_rejection_code='03';
```

HR Job Position Parameters

This section of the preprocparms12.sas program file enables you to specify the benefit_percent variable for benefits that are expressed as a percentage. This value is processed in the user-written code of job T270080. The selected value must come from the ctype or ctype001-006 fields in the HRP1015 table.

```
*---HR Job Position Macro Variables---*;
%let benefit_percent='US-BENEFITS %';
```

V8 Style Data Extraction

In this section of the preprocparms12.sas program file you can uncomment the necessary code to supply profile, remote, and libname parameters that correspond to your SAS V8 configuration.

```
* Options required if SAS/CONNECT when v8 style extract required;
%let use_v8_extract=NO;

* Uncomment the following section when "use_v8_extract=YES";
* Supply parameters and paths for target environment        ;
/*
%let r3profile=SIX;
options remote=server01;
libname r3lib 'D:\ETLDATA\SolutionFlows\Data\SAPAdapter\R3Lib\R3Lib47';
*/
```

Additional Customizations

This section of the preprocparms12.sas program file enables you to add customizations that relate to other data areas. You can also add these customizations to a new SAS program that can be called by a **%include** statement.

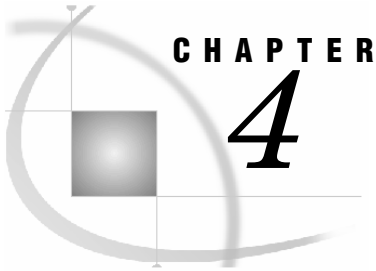
```
*--- Futher customization or overriding parameters can be made in an alternative SAS
program*;
*--- Use %include to reference the appropriate program e.g.
*;
*--- %include 'x:\overridingparms.sas'
*;
```

Customizing the preprocparms13.sas Program

Non-Leaf Text

The preprocparms13.sas program enables you to specify the parameters that identify leaf nodes and non-leaf nodes in the cost center and profit center hierarchies for your site. This file is called by the preprocglobal.sas program that is specified in the pre-processing section of every SAS Solutions Adapter for SAP extract and transformation.

You can use this preprocparms13.sas program to prepend a string to non-leaf node IDs in the cost center and profit center hierarchies. This enables you to easily identify non-leaf nodes in the hierarchies.



CHAPTER 4

Transformations Provided by the SAS Solutions Adapter for SAP

Transformations Provided by the SAS Solutions Adapter for SAP

The SAS Solutions Adapter for SAP provides a set of extraction jobs and transformation jobs that integrate data from standard SAP tables into standard SAS tables for SAS Financial Management and SAS Human Capital Management. Some of these jobs use transformations that are not part of SAS Data Integration Studio, but are supplied as part of the SAS Solutions Adapter for SAP.

Transformations that are provided and used by the SAS Solutions Adapter for SAP can be categorized into the following two types:

- transformations that are specific to SAP-related applications
- generic transformations that are not exclusive to SAP-related applications

This chapter describes these transformations that the SAS Solutions Adapter for SAP provides. The information can help you better understand the existing SAS Solutions Adapter for SAP jobs that use these transformations. You can also use this information to determine whether some of these transformations are useful for other jobs that you might create.

Add Standard Dimension Rows and Add Standard Dimension ASSOC Rows

Overview

The Add Standard Dimension Rows transformation and Add Standard Dimension ASSOC Rows transformation add standard row values to Dimension and Dimension ASSOC tables. Generally, these two transformations work in conjunction with each other.

These transformations are necessary because the DDS does not allow fact tables to have blank values for dimension foreign key columns. As a result, the transformations convert blank values to standard or special values.

Some dimension tables might need additional standard rows. For example, a financial transaction in SAP that does not involve a cost center has a blank value in the cost center column. This blank is then changed to UNASSIGNED.

The following list describes the standard value rows and their corresponding meanings or instances:

ALL	The root node in the hierarchy of certain dimensions. The hierarchy is stored in the _ASSOC table.
UNASSIGNED	The original value in the transaction was blank.
OPENBAL	The transaction was obtained from the opening balance summary information for which there is no detail information for all dimensions.
EXT	This is a required row for the internal organization dimension.

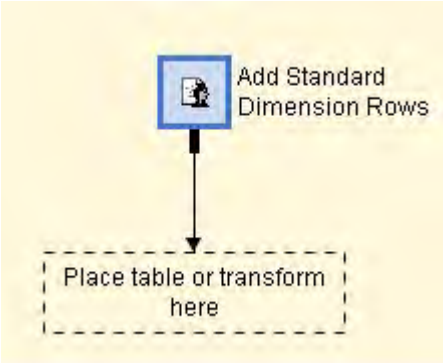
Using the Add Standard Dimension Rows Transformation

The information below describes how to use the Add Standard Dimension Rows transformation in a way that might overlap with basic SAS Data Integration Studio usage.

Note: Other sections of this document concentrate only on what you must do to a specific transformation and do not provide as much detail as this section. These other sections describe specific transformations that are provided by the SAS Solutions Adapter for SAP, but they do not explain the specific steps that you must complete to perform the corresponding tasks. If a new SAS Data Integration Studio uses one of these transformations in a test job, then you should be able to generalize the individual steps described in this section and use them for other transformations.

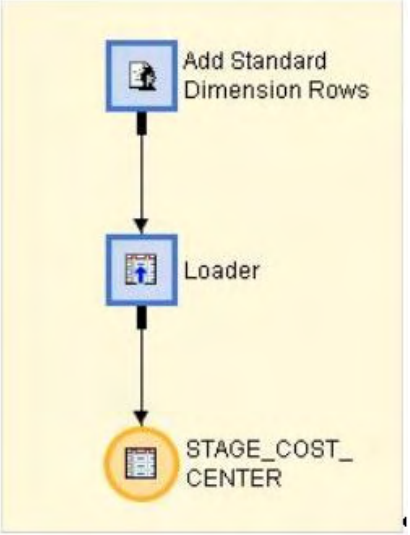
Complete the following steps to use the Add Standard Dimension Rows transformation:

1 Add the transformation to a job.



2 Add the table.

To add the table, drag and drop a dimension table onto the drop zone. The dimension table must have a key column ending in **_ID**.

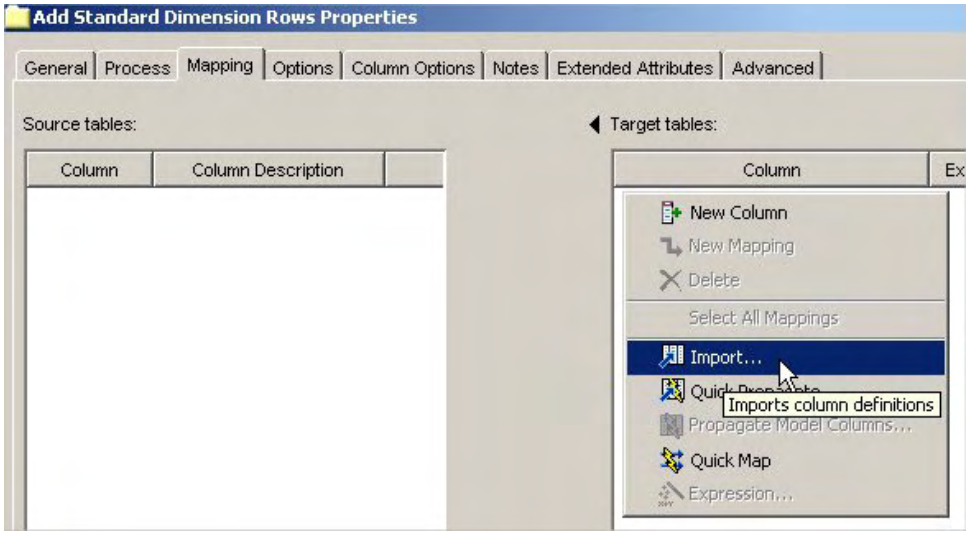


3 Review the **Mapping** tab.

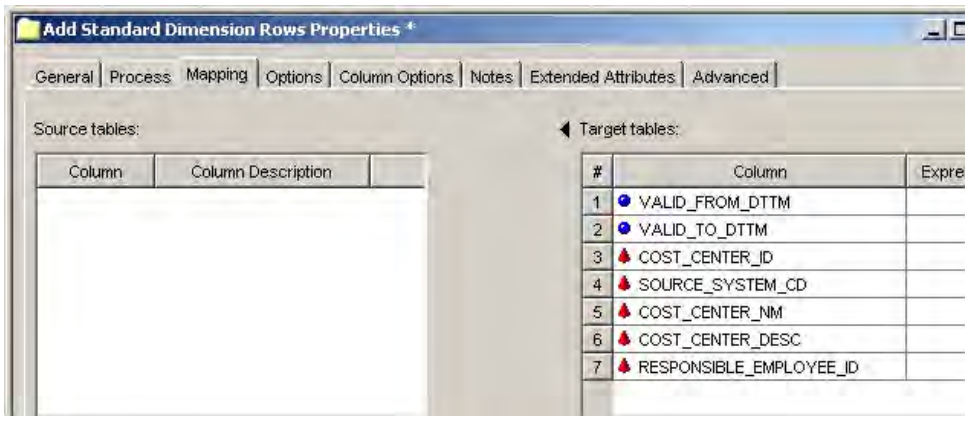
Use the **Mapping** tab to ensure that all columns in the output table are in the target table transformation. Complete the following steps to check these tables:

a Right-click the **Column** box of the Target tables.

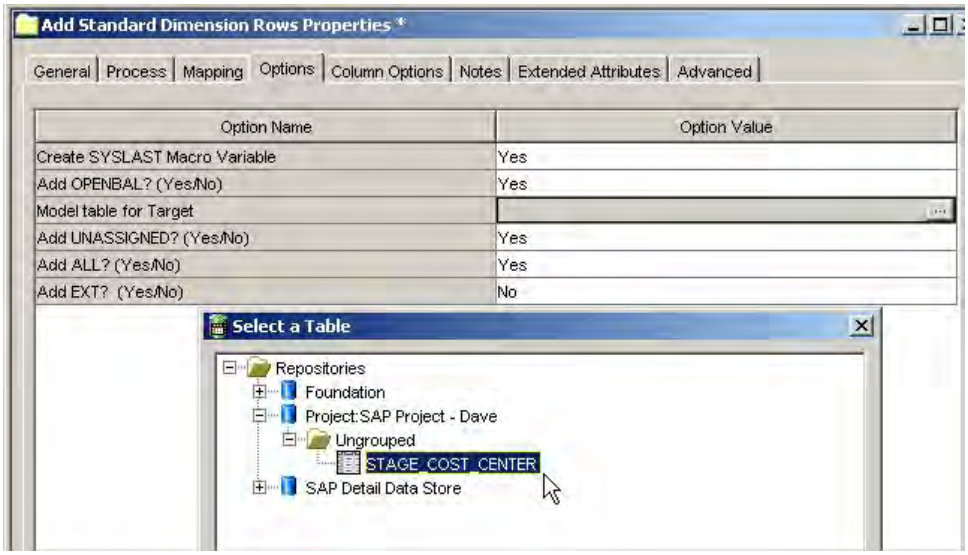
b Select **Import**.



c Select all the columns from the table that you just dragged and dropped onto the output table drop zone. The result should look similar to the figure below:



4 Click the **`Options`** tab.



5 Enter **Yes** or **No** as the option value for the following option names:

- **Add OPENBAL**
- **Add UNASSIGNED**
- **Add ALL**
- **Add EXT**

A **Yes** value adds the corresponding row to the output target table.

6 Use the button to the right of the **Model table for Target** field to select the table that you dragged and dropped onto the drop zone.

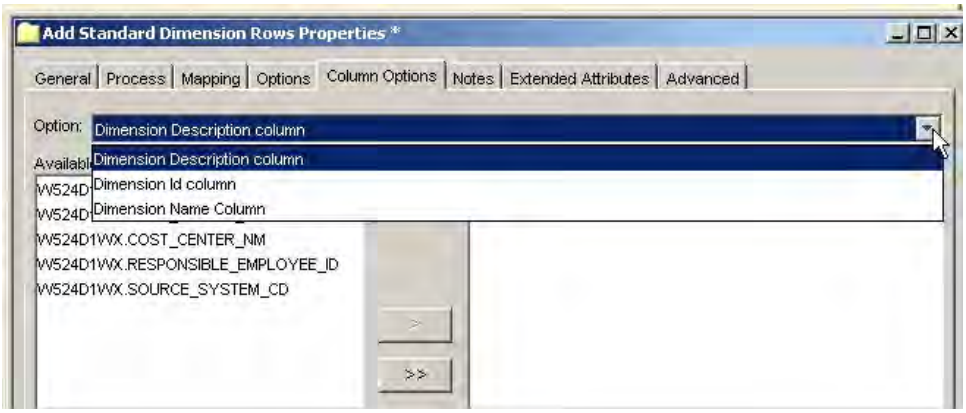
Note: The model table must physically exist before you run the job. Only the structure, not content, is used as a model for all the column definitions.

7 Click the **Column Options** tab.

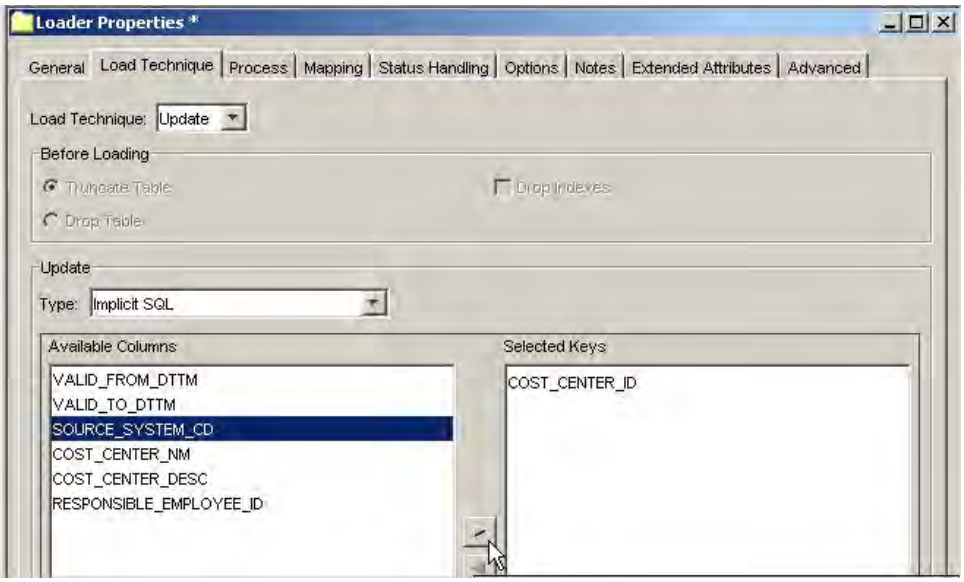
8 Use the **Option** drop-down arrow to select column options and map the columns to their predefined roles as described below.

Dimension Description column	ends in **_DESC** and contains the description of the dimension.
Dimension id column	ends in **_ID** and is the primary key of the dimension table.
Dimension Name Column	ends in **_NM** and contains the name of the description.

Generated code checks for invalid selections, but not immediately upon entry.



9 Click the **Load Technique** tab to review the load step. The generated code creates a table that contains the requested rows in the correct structure. You must still add these rows to the dimension table.



10 Enable the job so that it can run multiple times. To complete this task, you must modify the default settings for the load step as described below:

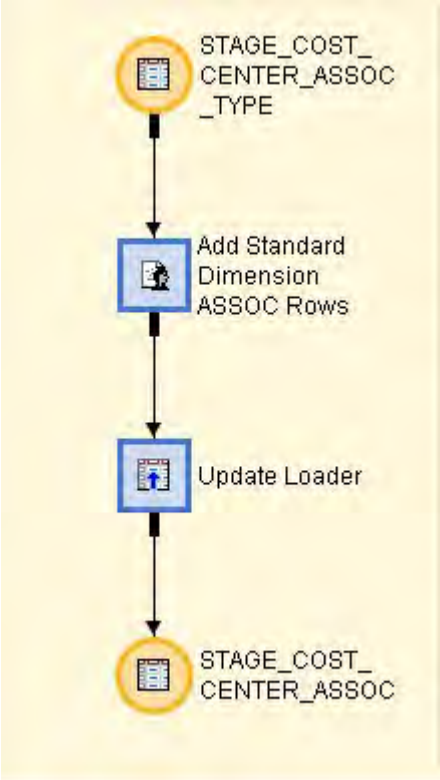
- **a** Set the **Load Technique** to **Update**.
- **b** Select an update **Type**.
- **c** Select the ID column option from the **Available Columns** box, and then click the right arrow to move the option to the **Selected Keys** box. The ID column is the only option that you must move to the **Selected Keys** box.

Using the Add Standard Dimension ASSOC Rows

Using the Add Standard Dimension ASSOC Rows transformation is very similar to using the Add Standard Dimension Rows transformation. Complete the following steps to use the Add Standard Dimension ASSOC Rows transformation:

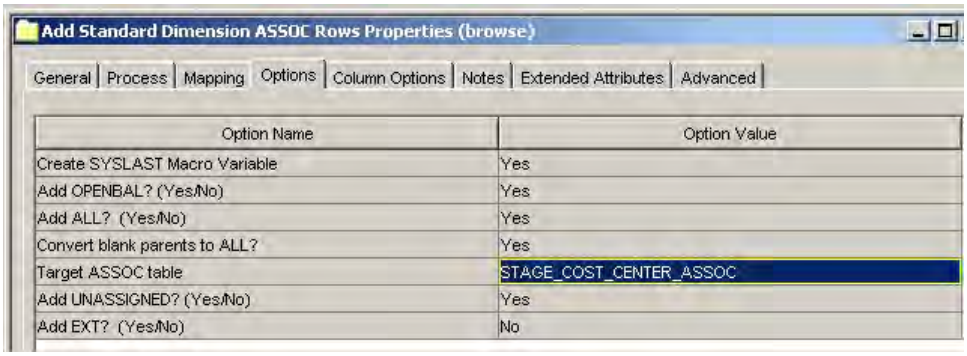
1 Drag and drop the Add Standard Dimension ASSOC Rows transformation onto a job.

2 Drag and drop a dimension _ASSOC_TYPE table onto the job as the input table. Also drag and drop a dimension _ASSOC table onto the job as the output table.



3 Import the columns from the _ASSOC table to the target of the mappings.

4 Click the **`Options`** tab.



5 Enter **Yes** or **No** as the option value for the following option names:

- **Add OPENBAL**
- **Add UNASSIGNED**
- **Add ALL**
- **Add EXT**

A **Yes** value adds the corresponding row to the output target table.

6 Specify **Yes** or **No** for the **Convert blank parents to ALL** option. A **Yes** value for this option changes blank parent IDs to **ALL**. The SAS Detail Data Store requires that a parent is not blank. This requirement includes the root. In that case, the root becomes its own parent.

7 Use the button to the right of the **Model table for Target** field to select the table that you dragged and dropped onto the drop zone.

Note: The model table must physically exist before you run the job. Only the structure, not content, is used as a model for all the column definitions.

8 Click the **Column Options** tab. The available column options on this tab require that you map the columns to their predefined roles. The **Dimension id column** ends in **_ID** and is the primary key of the dimension table.

Note: The name of the parent column is derived from the name of the ID column.

9 Click the **Load Technique** tab to review the load step and ensure that the update **Type** is specified as **Update**.

Add Parent to Hierarchy Table Based on Level and Position

Overview

The Add Parent to Hierarchy Table Based on Level and Position transformation adds a parent column to a table that already has a specified hierarchy in the form of a level and position. This is a common SAP hierarchy storage form.

The level and position uses the root as the first record. To establish the parent of a row, the preceding row with a lower level is used. Therefore, preceding rows with an identical level are siblings.

The macro **%adpt_level_to_parent_child** reflects the level and position hierarchy by including the following parameters:

- in_dsn=&_input0
- out_dsn=&_output0
- level=&level
- parent=&parent
- child=&child
- dstype=&dstype
- keep=&keep

- rootlevel=&rootlevel
- generated_sequence_var=&generated_sequence_var

The interface enables you to define each of these parameters.

Using the Add Parent to Hierarchy Table Based on Level and Position Transformation

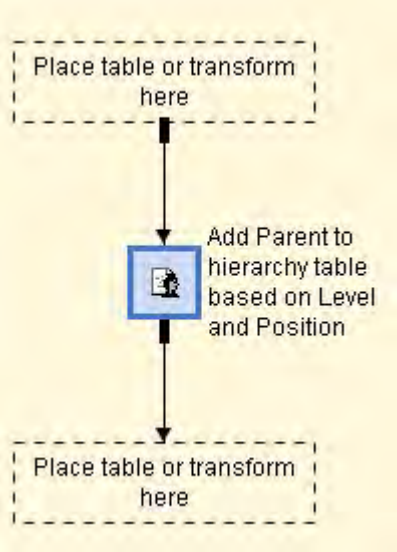
Complete the following steps to use the Add to Parent hierarchy table based on Level and Position transformation:

1 Drag and drop the Add Parent to Hierarchy Table Based on Level and Position transformation onto a job. This transformation requires a single input table and it outputs a single table in return.

2 Drag and drop input and output tables onto the job.

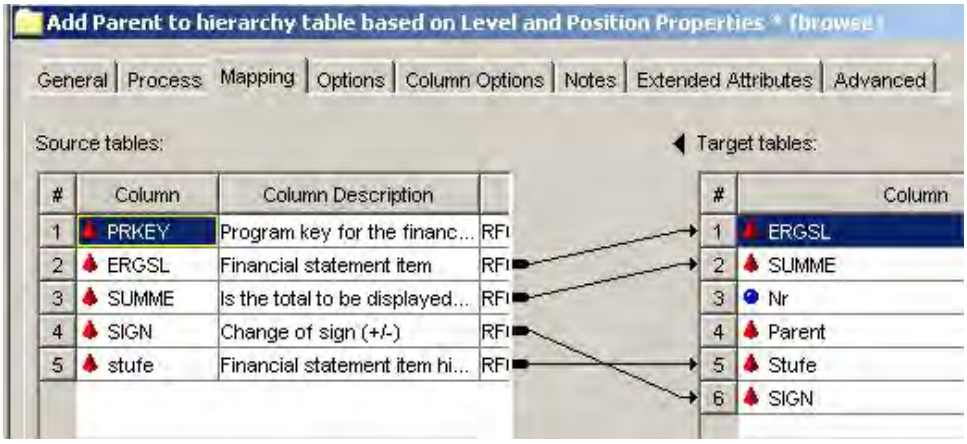
The input table must contain a column that is the identifier or key that contains unique values. The input table must also contain a column that stores the level. This level column can contain contents that are either numeric or character-based. However, character-based contents must contain numeric strings. In addition, the names of the columns must be associated with the roles in the column options.

The output table must contain at least the identifier or key column and a parent_id column. The parent_id must be of the same type and length as the identifier. In some cases, the parent of the root (top) node of the hierarchy can be blank if necessary.



3 Click the **Mapping** tab.

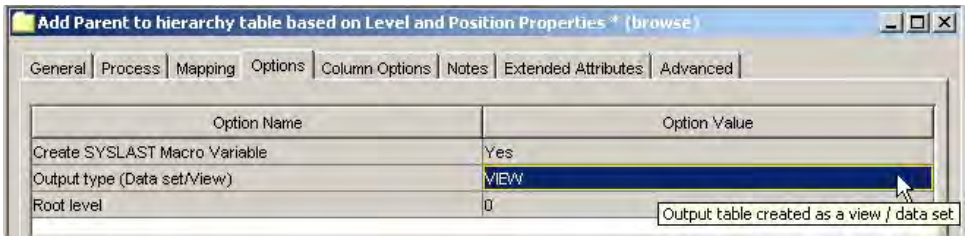
Use the **Mapping** tab to ensure that all columns in the output table are in the target table transformation. The generated code copies all columns that are in both the input table and output table. Although the mappings in this transformation are not required, you should resolve any inconsistencies so that impact analysis has more information if needed.



In the example above, four of the original five columns are propagated into the output structure. In addition, two new columns are added (numeric column Nr and character column Parent).

Do not enter target table columns that do not exist in the input table and are not assigned a parent role or sequence number. For more information about these options, you can review the **`Column Options`** tab.

4 Click the **`Options`** tab.

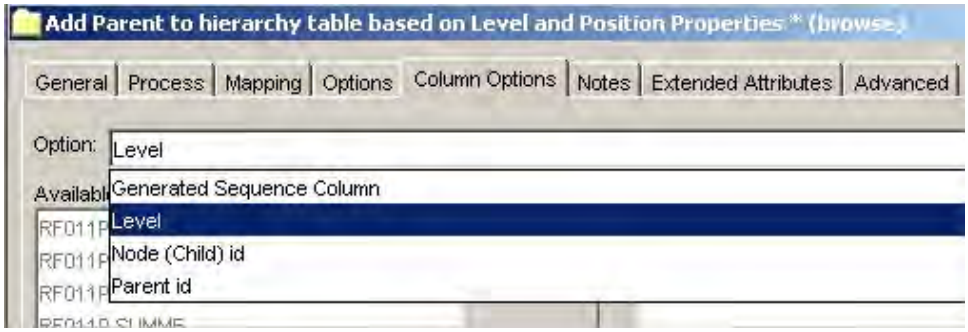


The **Options** tab contains the following two options you must consider setting:

`Output Type` Select **`VIEW`** if the output table is loaded through a standard load step. Select **`DATA`** if the output table is loaded through a load step with code generation switched off (a null loader).

`Root level` Leave this value at the default. If the input table contains multiple disjoint hierarchies (many roots), then the root level forces the parent to be blank.

5 Click the **Column Options** tab.



Use the **Option** drop-down arrow to select column options and map the columns to their predefined roles as described below:

Generated Sequence Column	contains a value that is equal to the row number of the input table. This is a generated column that is numeric. You must add/import this column into the target table in the mappings column before you can select it.
Level	contains the level information. This column is in the input table.
Node (Child) id	identifies each node in the hierarchy. This column is in the input and output tables. It should also be the primary key of the input table.
Parent id	identifies the parent ID (node ID of the parent) in the hierarchy. This column is in the output table. It must be of the same type and length as the node ID. You must add/import this column into the target table in the mappings column before you can select it.

Format Generator

Generating FORMAT CODE

The Format Generator transformation creates code that generates a format from one input table. This transformation uses the generic transformation generator property. The transformation can also run multiple times on a single input table to generate multiple SAS formats.

You can access information about this transformation's options in the **Options** and **Columns** tabs for the transformation.

The code that the Format Generator transformation creates must handle several table properties. These properties include specifications such as format, library catalogs, creating and managing CNTLIN tables, and managing duplicate values.

The following list details some format issues and properties that the generated code must address:

- The format of the generated code can have a description that is also in the description of the format library catalog entry.
- The format must be stored in a library catalog that is available in the current metadata repository. In a project repository, you must fetch/check out the library.
- The format does not have to be saved in a catalog that already exists. When the transformation creates a format, it can also create a catalog in which to save it. A string that includes a macro variable such as **`FORMATS&LOCALE_LANGUAGE`** can specify the catalog.

The generated code must conform to the following properties and requirements for defining library catalogs:

- The default library should be **`LIBRARY.FORMATS`** if the library **`LIBRARY`** is available.
- If metadata does not define the library **`LIBRARY`**, then the default cannot be used. Only libraries available in the current repository can be checked. In this case, a message notifies the user to define and use the LIBNAME **`LIBRARY`**.

The Format Generator transformation creates a temporary format CNTLIN table from the underlying table. (For a description of the CNTLIN table, refer to the online Help for PROC FORMAT.) This process uses metadata to generate each column of the CNTLIN table. When creating this table, the transformation must also address the following issues for the CNTLIN columns:

- The format must have a name (maximum of 30 characters) that is unique within the format catalog. Because the Format Generator transformation does not verify uniqueness, new formats with the same name overwrite preexisting names.
- The START column must be an expression of the columns in the underlying table.
- The END column is optional and can be an expression of the columns in the underlying table.
- The LABEL column must be an expression of the columns in the underlying table.
- A row can be added to map missing values onto a specified value. The special value can be any expression. For example, START might be set to a missing value such as a blank value for characters or a **`.`** value for numeric requirements. The LABEL can be set to **`Missing`**, **`Unknown`**, or an expression such as **`PUT('MISSING', $format_words.)`**. In this case, you need to enter the expression in order to specify the label for **`Missing`**. Note that if you are using an expression that another format uses, the format must already be defined.
- A row can be added to handle the **`OTHER`** concept and the CNTLIN column HLO must be **`O`**. The start label is essentially irrelevant and can be missing. The LABEL can be set to **`OTHER`**, **`Unknown`**, or an expression such as **`PUT('OTHER', $format_words.)`**. In this case, you need to enter the expression to specify the label for **`Missing`**.
- The rows in the **`CNTLIN : WHERE`** clause must be controlled so that the WHERE clause subsets the underlying table into the CNTLIN table on input. It is often necessary to base a format on only some rows of the underlying table such as

language="&LOCALE_LANGUAGE". The WHERE clause is an expression on the columns in the underlying table.

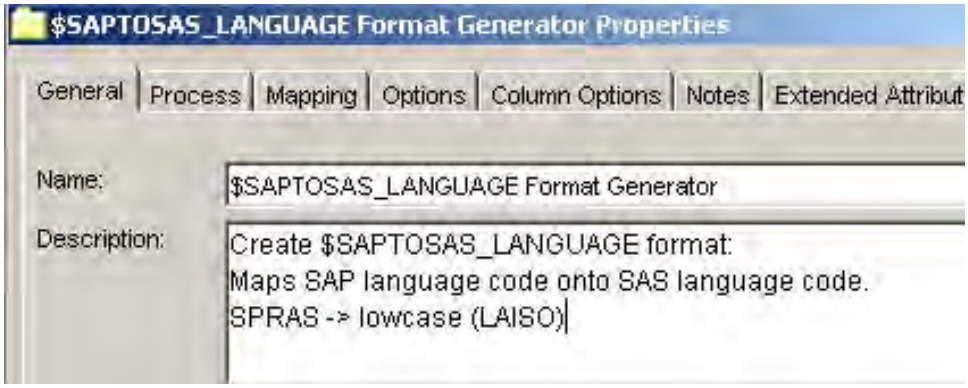
The Format Generator transformation must also consider the DUPLICATE Removal Strategy. The generated CNTLIN table can contain duplicate **START** values or overlapping ranges. Note that the **START** value is inside the range of a START - END combination of another row. The following variables eliminate duplicates in START value:

- **NODUP** generates a **PROC SORT NODUP**; by **START** value. Because this strategy might often throw away the wrong rows, you might also consider using more controlled elimination variables such as **FIRST** and **LAST**.
- **FIRST** and **LAST** variables assume that there are additional columns in the underlying table that enable determination of which row is best to keep. In this case, these additional columns are kept in the temporary CNTLIN table and the subsequent **SORT** variable has these additional columns in the **BY** statement. A second DATA step then keeps only the required rows. The **FIRST** variable keeps the first row in each **BY** group. The **LAST** variable keeps the last row in each **BY** group.

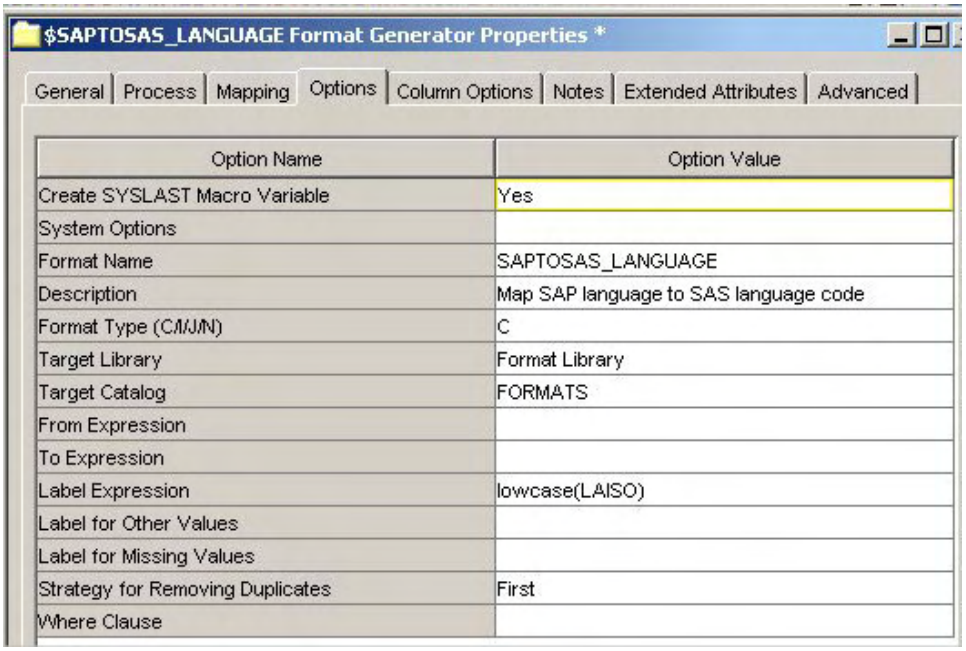
Using the Format Generator Transformation

Follow these steps to use the Format Generator transformation:

1. Drag and drop the Format Generator transformation onto a job that loads the table on which the format is to be based. If possible, use the same job that created or loaded the table because any changes or reloading of the table automatically regenerates the format(s) based on the table.
2. Drag and drop the table onto the job as the input table.
3. Click the **General** tab and change the name of the transformation to **$formatname Format Generator** or a similar name.



4. Click the **Options** tab. This tab provides a list of parameters that you can specify for the Format Generator transformation.



The table below describes the option names and values that are available on the **`Options`** tab. Use this table to verify and edit your Format Generator transformation properties as needed.

Table 4.1 Format Generator Transformation Options

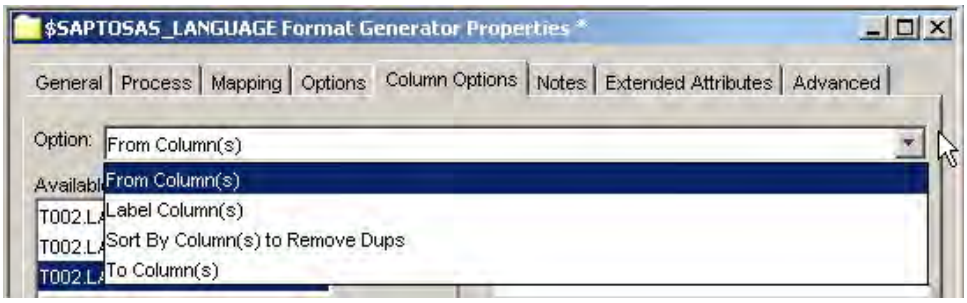
Label	Macro Variable	Description	Type	Req.	Default	Constraint/ valid values
System Options	**options**	Options on a SAS OPTIONS statement.	String	N		
Format Name	**formatName**	Name of the format.	String	Y		
Description	**formatDescription**	Description of the format.	String	N		
Format Type (C/I/J/N)	**formatType**	The type of format. Possible values are **C** for character format, **I** for numeric informat, **J** for character informat, and **N** for numeric format (excluding pictures).	String	Y	C	C N J I
Target Library	**targetLibrary**	Target library for the generated format.	Metadata Library	Y		
Target Catalog	**targetCatalog**	Name of the target catalog for the format.	String	Y	FORMATS	Min string length: 1. Max string length: 32
From Expression	**fromExpression**	Expression to be used for the range's starting value. Use this field to enter an expression or use **From Column(s)** option in the **Column Options** tab to select columns.	String	N		
To Expression	**toExpression**	Expression to be used for the range's ending value. Use this field to enter an expression or use **To Column(s)** option in the **Column Options** tab to select columns.	String	N		
Label Expression	**labelExpression**	Expression to build the informatted or formatted value. Use this field to enter an expression or use **Label Column(s)** option in **Column Options** tab to select columns.	String	N		
Label for Other Values	**otherLabel**	String or expression to be used as the label for the range **OTHER**. This range includes all values that are not in one of the specified ranges.	String	N		
Label for Missing Values	**missingLabel**	String or expression to be used as the label for missing values.	String	N		

Label	Macro Variable	Description	Type	Req.	Default	Constraint/ valid values
Strategy for Removing Duplicates	**dupStrategy**	Specify **NODUP** to remove duplicates (unordered), **FIRST** to keep the first duplicate, and **LAST** to keep last duplicate. Used in conjunction with **Sort By Column(s)** option in **Column Options** tab.	String	N	First	First Last Nodup
WHERE Clause	**whereClause**	WHERE clause to be applied to the input data set.	String	N		

Note: Picture formats are not supported.

5 Click the **Column Options** tab. You can specify the starting and ending values of a range and formatted or informatted values in two different ways.

- If the values are a concatenation of columns of the input table, you can use the **From Column(s)**, **To Column(s)**, and **Label Column(s)** options that are available from the **Column Options** tab.
- If the values are a more complicated expression of columns, you can use the **From Expression**, **To Expression**, and **Label Expression** options that are available from the **Column Options** tab. If specified, these expressions take precedence over the selected columns.



The table below describes the options and values available from the **Column Options** tab. Use this table to verify and edit your Format Generator transformation properties as needed.

Table 4.2 Format Generator Transformation Column Options

Label	Macro Variable	Description (Tool Tip)	Constraint/ Enumeration
From Column(s)	**fromColumns**	Selects column(s) for the format's starting value. Multiple columns are concatenated. Specify either **From Column(s)** or **From Expression**.	No limit on number of selectable columns
To Column(s)	**toColumns**	Selects column(s) for the format's ending value. Multiple columns are concatenated. Specify either **To Column(s)** or **To Expression**.	No limit on number of selectable columns
Label Column(s)	**labelColumns**	Selects column(s) for the informatted or formatted value. Multiple columns are stripped and concatenated with a blank between the columns. Specify either **Label Column(s)** or **Label Expression**.	No limit on number of selectable columns
Sort By Cols to Remove Dups	**sortByColumns**	Sorts the control data set by the selected columns and removes duplicate keys according to Strategy for Removing Duplicates option.	

User Written Code for Multiple Input/Output Tables

Overview

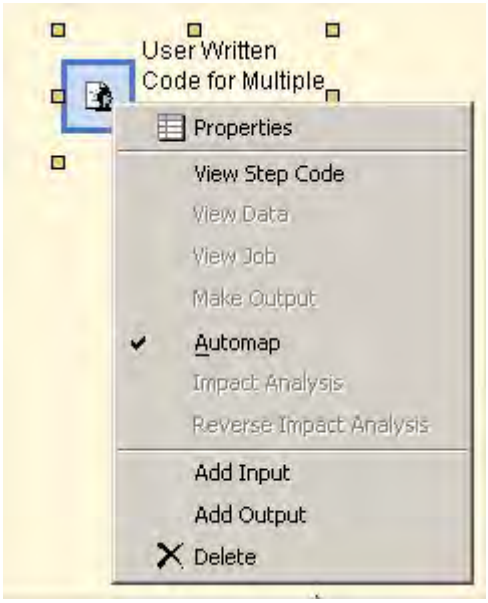
The User Written Code for Multiple Input/Output Tables transformation is a generic, user-written code transformation. In the following two circumstances, use this transformation instead of the standard User-written Code transformation that SAS Data Integration Studio provides:

- when there is more than one input or output table
- when the user-written code needs additional library statements to be available or generated

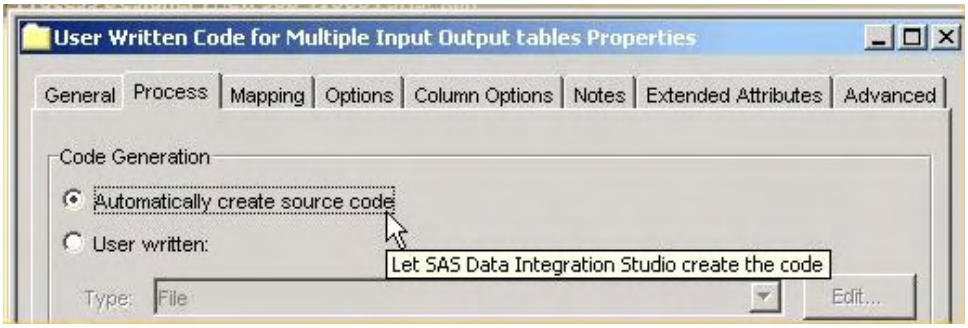
Using the User Written Code for Multiple Input/Output Tables Transformation

Complete the following steps to use the User Written Code for Multiple Input/Output Tables transformation:

1. Drag and drop the User Written Code for Multiple Input/Output Tables transformation onto a job.
2. Right-click on the transformation and select the following options:
   - **`Add Input`** to add drop zones for the input table
   - **`Add Output`** for each required output



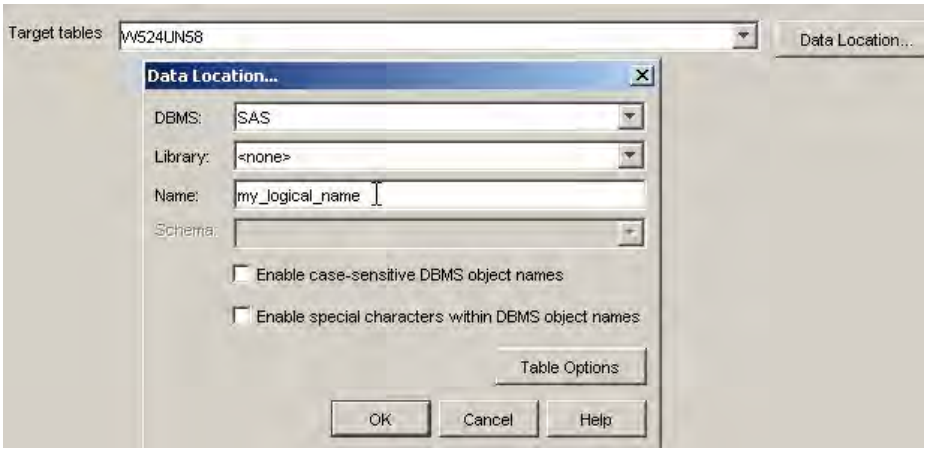
3. Drag and drop tables or transformations onto the new drop zone. You are now ready to review the properties.
4. Click the **`General`** tab and replace the default transformation name with a name that is more specific.
5. Click the **`Process`** tab and complete the following steps:
   a. Do not modify the default settings for **`Code Generation`**.
   b. Ensure that the **`Automatically create source code`** option is selected.



c Review the registered **`Target tables`**. You can specify appropriate parameters for the target tables based on the following three scenarios:

- If the source code refers to tables that are using a macro variable such as **`names &_output`**, or **`&_output0`**, then the default target table names can remain unchanged. In this case, you must ensure that the load steps of the output tables do, in fact, load the real output tables from these temporary tables.

- If the source code creates tables with more logical names, register these tables as target tables. In the example below, **`my_logical_name`** is replacing the default name **`W524UN58`**.



- If your source code directly loads the output tables that you dragged and dropped onto the transformation, then specify the physical name of the table and select the library accordingly. When you finish reviewing the properties, change the load step to not automatically generate code.

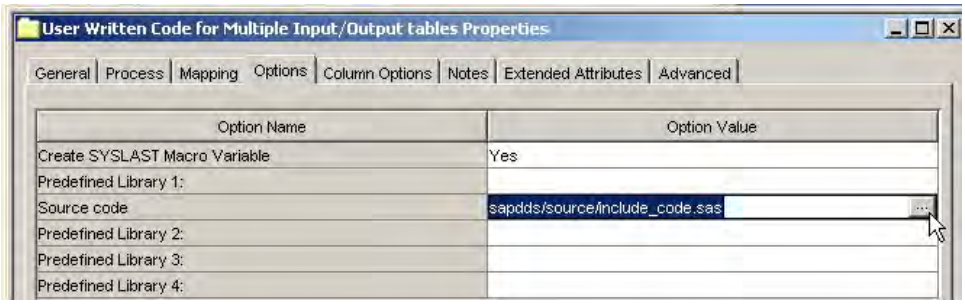
6 Click the **`Mapping`** tab.

In order for the subsequent load step to automatically create the source code for the load, you must import all the columns of the output tables into the corresponding target tables that are listed in the transformation **`Mapping`** tab.

Although the mappings in this transformation are not required to generate code, you should use the mapping assignments to document the mappings made in the

source code. This can help obtain a more accurate reflection from the impact analysis tool.

7 Click the **Options** tab.



8 Replace the default source code **sapdds/source/include_code.sas** with a valid path and the name of the source code that you want to include. The path must be relative to the server environment, as is the case with all other user-written source code locations.

Four additional parameters (**Predefined Library 1** to **Predefined Library 4**) can force library assignments to occur before your own source code executes. The automatic code generation typically assigns libraries for all input tables and all libraries that are entered in the Data Location dialog box for the target tables on the **Process** tab. However, some libraries such as SAP server libraries need to be assigned by specifying them in one of these four parameters.

Changed Data Extraction Using Date or Time-stamp and Overlap

The Changed Data Extraction Using Date or Time-stamp and Overlap transformation creates a SAS Data Integration Studio transformation that extracts changed data. This transformation uses key, datestamp, and timestamp information to identify new or changed data from source tables. It then extracts the data and stores the result in a target table that is conceptually a mirror image of the source table. The target table can also have fewer columns than the source table.

This transformation is often required when extraction of the full data from the source system would be very time-consuming. For example, extracting 20 million rows of data across an entire network might take many hours, whereas extracting only the changes and then merging them into the previous extraction might take only minutes.

The logic for this transformation can be used only with source tables that contain a timestamp, datestamp, or other key column value that increases over time. A new extract can occur for all records with a key or timestamp that is greater than the largest value that was extracted previously. The resulting data can then be appended to the previous data.

The timestamp or key does not need to be increasing exclusively in order to be flagged. To allow for source data that is not strictly increasing, you can use an overlap to re-extract data that is in the overlap range. For example, you might have financial transactions that change in an operational system before the books are closed for a period, yet the operational system does not create a real datestamp or timestamp for the

change. In this case, there is a phase when two or more financial periods are open and the data can change. To manage this scenario, an extraction needs to re-extract data from all open periods. The new records can then replace the previous records with the same keys. If records can be deleted or have keys modified, then it is not always possible to match the changes to the previous corresponding records. This requires that all records previously extracted from an open period need to be removed from the main target table.

After newly extracted data has been successfully appended or merged into the target table, the record must be updated. In addition, a record of previous extracts must be administered and kept to enable the creation of the next extraction condition. If the extraction, append, or merge steps fail, then the administration data should remain unchanged so that the next extraction run gets the data that would have been extracted if the previously attempted extraction had not failed.

Date Join

The Date Join transformation creates a SAS Data Integration Studio transformation that works with tables that have date range columns. The purpose of this transformation is to join two input tables that have a common key into a single output table so that the date ranges interweave.

Many fields such as HR contain historical information. This information can be true for complete periods and also extend into the future. For example, SAP HR infotypes contain effective starting dates and effective ending dates for past and current employees. The Date Join transformation uses a macro that can be used with any effective-dated date, including this type of HR data that is extracted from SAP R/3.

To join tables that are effective-dated, consider that the dates are essentially keys and that a logical match can occur when periods overlap, even if only partially. To join the tables for a fixed date, you can select the date from each contributing table to simplify the join. The algorithm for joining tables with effective dates is not trivial if you consider that the date ranges might or might not partially overlap. Note that a join algorithm that loops for all fixed dates can cause the volume of data to increase exponentially, based on the validity of records that can span from one to ten years.

The Date Join transformation and the underlying **`%DATEJOIN`** macro offer an efficient alternative. You can join tables accurately by introducing a time dimension. This essentially introduces a list of dates, each of which then selects a record from the contributing tables. This can be effective if you want to capture a snapshot of results for something such as employee headcounts at the beginning of each month. However, there is a drawback to this technique in that the number of selected records could increase dramatically. For example, to prepare a table for ad-hoc date queries, you might need to duplicate a table row 365 times per year. As a result, unless you only need to represent a small number of fixed dates, use the table join and **`%DATEJOIN`** macro for exploitation time.

The example below shows how two input tables with a single key value are joined into a single output table. Notice the new start dates and end dates in the tables. Also note how the values in Var1 from Table1 and Var2 from Table2 are populated respectively.

Figure 4.1 Input and Output Tables Using Date Join Transformation

Input Table 1

Key	Begin Date	End Date	Var1
1	01JAN06	15FEB06	1
1	16FEB06	20APR06	2
1	01JUN06	31JUL06	3
1	01AUG06	31AUG06	4

Input Table 2

Key	Begin Date	End Date	Var2
1	01FEB06	31MAR06	A
1	01APR06	31JUL06	B
1	01AUG06	30SEP06	C

Output Table

Key	Begin Date	End Date	Var1	Var2
1	01JAN06	31JAN06	1	
1	01FEB06	15FEB06	1	A
1	16FEB06	31MAR06	2	A
1	01APR06	30APR06	2	B
1	01MAY06	31APR06		B
1	01JUN06	31JUL06	3	B
1	01AUG06	31AUG06	4	C
1	01SEP06	30SEP06		

Period Consolidation

The Period Consolidation transformation creates a SAS Data Integration Studio transformation that works with tables that have date range columns. This transformation identifies single key values where there are no changes in the corresponding, non-key columns that are specified. The similar and sequential data is then consolidated into a single date range. The transformation is written using the transformation generator.

The Period Consolidation transformation uses the **`%consolidate_period`** macro. This macro consolidates, into a single record, consecutive date ranges for a key that is passed as a parameter.

The example below shows a data table before and after the Period Consolidation transformation. The key consists of Key1 and Key2. Begin Date and End Date specify the date range. The important columns for consolidation are non_key1 and non_key3. The non_key2 column is dropped from the output table because its value might not be correct for the full, consolidated date ranges.

Figure 4.2 Input and Output Tables Using Period Consolidation Transformation

Input Table

	Key1	Key2	Begin Date	End Date	non_key1	non_key2 (not important for usage)	non_key3
1	123	A	01JAN2006	31JAN2006	21	ABC	XYZ
2	123	A	01FEB2006	15FEB2006	21	DEF	XYZ
3	123	A	16FEB2006	15JUL2006	21	DEF	XYZZY
4	123	A	16JUL2006	31OCT2006	21	BCD	XYZZY
5	123	A	01NOV2006	31DEC2006	21	CEF	XYZZY
6	125	A	01JAN2006	31AUG2006	21	ABC	CCC
7	125	A	01SEP2006	31DEC2006	22	ABC	CCC
8	126	A	01JAN2006	15FEB2006	22	FED	DDD
9	126	A	16FEB2006	31MAR2006	22	DEF	DDD
10	126	A	01AUG2006	31DEC2006	22	DEF	DDD

Output Table

	Key1	Key2	Begin Date	End Date	non_key1	non_key3	contributing rows from input table
1	123	A	01JAN2006	15FEB2006	21	XYZ	1,2
2	123	A	16FEB2006	31DEC2006	21	XYZZY	3-5
3	125	A	01JAN2006	31DEC2006	21	CCC	6,7
4	126	A	01JAN2006	31MAR2006	22	DDD	8,9
5	126	A	01AUG2006	31DEC2006	22	DDD	10 (This row is not concatenated to 9 because there is a gap between the end date of 9 and the begin date of 10.)

Repeat Record for Date Range Overlap

This transformation reads an input table and writes to an output table with the same structure. In the output table, the records are repeated so that the values in the date column exist for all dates in a time range.

The time range starts at the date that is specified in the date column as corrected by the overlap and interval values. For missing values, the time range starts at the date that is specified by the Initial Start parameter. The time range ends at the latest of two dates: the current date as adjusted with the interval value or the date specified in the Date Column as adjusted with the overlap and interval values.

Here are the parameters that you can specify for this transformation:

Option Name	describes the transformation option.
Date Column	specifies date values that are used to determine start dates for a time range
Overlap	specifies the number of intervals to go back (negative value) or forward (positive value) when determining start and end dates for a time range.
Interval	specifies the calendar interval (month, day, or year) to go back when determining start and end dates for a time range. This parameter also specifies the calendar interval for adding records within a time range. For example, if the interval parameter is monthly, one record is added for each month between the start and end date.
Alignment	specifies if new date values are aligned at the beginning, midpoint, or end of the set interval. The default value is BEGINNING.
Initial Start	specifies the start date of a time range for missing values.
Create Output as View	specifies the output as a view or a table. Specify YES to create the output as a view.

The following example shows a data table before and after the Repeat Record for Date Range Overlap transformation. In this example, the overlap is -2, the interval is MONTH, the Initial Start date is 01JAN2006, and the transformation was run on April 19, 2006.

Figure 4.3 Input and Output Tables Using Repeat Record for Date Range Overlap Transformation

Input Table

	DATECOL
A	
B	01JAN2006
C	01MAR2006
D	01JUL2006

Output Table

	DATECOL
A	01JAN2006
A	01FEB2006
A	01MAR2006
A	01APR2006
B	01NOV2005
B	01DEC2005
B	01JAN2006
B	01FEB2006
B	01MAR2006
B	01APR2006
C	01JAN2006
C	01FEB2006
C	01MAR2006
C	01APR2006
D	01MAY2006
D	01JUN2006
D	01JUL2006

HR Clean Personnel Numbers

The HR Clean Personnel Numbers transformation provides a mechanism whereby HR personnel numbers (PERNR) in SAP are made consistent for a single person. For example, if a person has multiple personnel numbers, this transformation consolidates the data into a single number so that a person is represented with a single, unique personnel number. Additional PERNR numbers are considered superfluous, and only one number is preferred per employee.

This transformation uses a specific format that is specified as an option of the transformation. This format combines personnel numbers with an expression such as **`put(put(pernr,z10.),$R3_to_HRID.)`**.

Some personnel numbers might map to a value that is defined through the DELETEFLG option. This data is moved to a separate table for employee numbers that are deleted. For best results, create the necessary format with the transformation HR Create PERNR Cleaning Format. You can refer to the T200080 Create SAS Format for Personnel Number Clean job as sample.

The HR Clean Personnel Numbers transformation calls the macro **`%adpt_clean_personnel_numbers`**. In addition, a user-exit with additional site-specific code might get executed.

The HR Clean Personnel Numbers transformation uses one input table to which it applies the personnel number cleansing. The following table describes the parameter prompts that are possible in the input table:

Table 4.4 HR Clean Personnel Numbers Transformation Input Table Options

Parameter Prompt	Macro Variable	Description
User Exit Location	**USER_EXIT_CODE**	Fully qualified path and name of user exit code on the SAS server. The code is included in the DATA step that maps the personnel numbers after the mapping. It can use any of the following options: ❑ **What** – description ❑ **&p_outdeleted** – name of output data set for records that are deleted ❑ **cleaned** – intermediate view for the cleaned table ❑ **&p_input** – input table ❑ **new_pernr** – intermediate DATA step variable that contains mapped personnel number ❑ **pernr** – DATA step variable that contains input personnel number
Begin Date	**BEGDA**	Start of date range.
End Date	**ENDDA**	End of date range.
Comparison Column List	**COMPVAR_LIST**	List of comparison columns. The date range is consolidated so each unique combination of comparison columns has a single period.
Should Gaps be Filled	**FILLGAPS**	Specify YES if gaps in the date range should be padded with new records and missing values in the comparison columns.
Non-date-Range Key(s)	**SORTKEY**	List of columns on which the input table is sorted or needs to be sorted.
Additional Columns to Keep	**KEEPVARS**	Additional columns to be kept with the latest value in each consolidated period. Differences in values do not trigger a new period
Output Table Type (VIEW | DATA)	**VIEW**	Option that enables you to specify VIEW if the output table created by the transformation should be a view. This is usually needed when used with a load step or follow-on transformation. Specify DATA to create a table.
Format Name	**FMTNAME**	Name of SAS format to use. The format must be in the format search path that is set up on the workspace server. The default is **R3_to_HRID**.
Delete Flag	**DELETEFLG**	Value of formatted personnel number to be deleted.

The HR Clean Personnel Numbers transformation fills two output tables:

- One table with the drop-prompt **Place table or transform here**. This table contains the cleaned personnel numbers.

- One table with the drop-prompt **`Place table or transform here for deleted records`**. This table contains personnel number records that were not propagated from the input to the main output table because of the DELETEFLG mapping.

HR Payroll Extraction

Overview

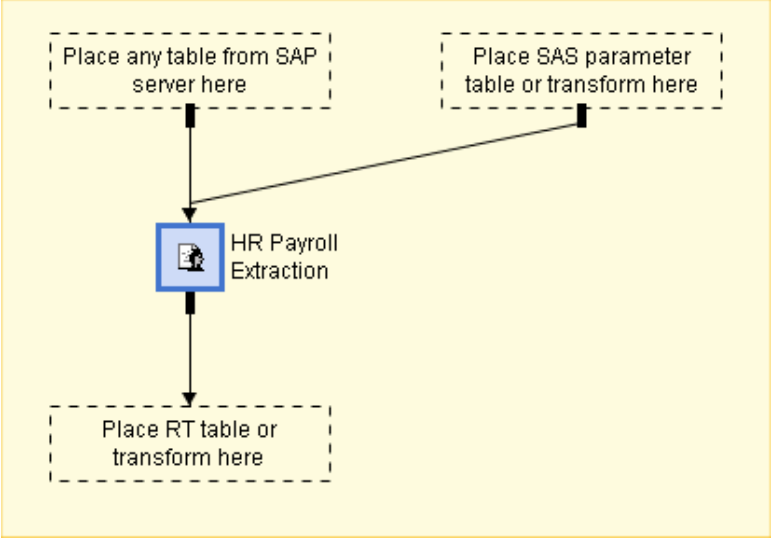
The HR Payroll Extraction transformation extracts HR payroll cluster data from SAP R/3. This transformation has an explicit input that contains the payroll area-date combinations for which the payroll data needs to be extracted. There is also an implicit input from SAP R/3. The transformation does not perform any delta processing because other steps perform the task as needed.

This transformation calls the macro **`%adpt_payroll_xtract`**. For a usage example, refer to the jobs E08040A HR Extract Historical Compensation and E08040B HR Extract Historical Compensation.

Using the HR Payroll Extraction Transformation

Complete the following steps to use the HR Payroll Extraction transformation in a job:

1 Drag and drop the HR Payroll Extraction transformation onto a job. This transformation requires two input tables, and it outputs a single table in return as shown below.



2 Drag and drop two input tables onto the drop zone. The two input tables that you use have specific purposes.

- One of the input tables that you drag and drop can be any table that is associated with the SAP library on the server from which the extraction is to be made. This table is required so that the transformation generates the correct LIBNAME statement as it is needed by the CALLRFC in the source code.
- The other input table should be the parameter table that tells the transformation which HR payroll results to extract. This input parameter table provides a structure that contains the following columns:

Table 4.5 Structure of the HR Payroll Extraction Transformation Input Table

Column	Type	Description
country_cluster	$2	This is a required column. If blank, the corresponding row is ignored. To calculate salaries, it is important to know the country group for wages and salaries (MOLGA). The SAS Solutions Adapter for SAP associates MOLGA with a person by looking up the MOLGA code for the company code (BUKRS) that is stored in PA0001. SAP has a separate cluster table for different countries. The cluster name takes the form Rx where x represents the country. For example, U=USA, D=Germany, J=Japan, C=Switzerland, I=Italy, etc. Refer to R/3 table T500L for the country character.
payroll_area	$2	A separate row should be created for each payroll area that is used in each country
in_period	$6	The in_period is in the form YYYYPP, where YYYY represents the financial year, and PP represents the pay period. The pay period can be 01 to 12 for employees paid monthly and 01 to 53 for employees paid weekly.

Note: All other columns are ignored by the transformation.

3 Drag and drop an output table onto the drop zone. The output table RT must contain at least the subset of columns that are in the SAP RT table shown in the figure below.

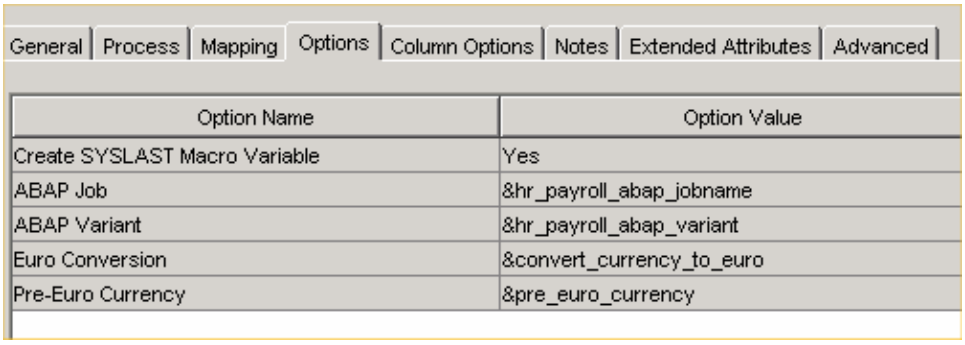
Table 4.6 Structure of the HR Current Payroll Extract Log Input Table

Column	Type	Notes
ABKRS	$2	Payroll area
FPPER	$6	For-period for payroll
FPBEG	Date	Start date of payroll period (For-period)
FPEND	Date	End of payroll period (for-period)
IABKRS	$2	Payroll area
INPER	$6	In-period for payroll
SRTZA	$1	Indicator: status of record
RUNDT	Date	Date of payroll run
PAYDT	Date	Pay date for payroll result
PERNR	$8	Personnel number
LGART	$4	Wage type
BETRG	Numeric 8	HR payroll: amount
AMT_CURR	$5	Currency key

The output table must be used with a null loader because the transformation fills the table directly. This means that the data location must refer directly to the output table. As a result, the table must be a permanent table, not a work table.

4 Click the **Mapping** tab of the transformation and ensure that all the columns in the output table are represented as output columns.

5 Click the **Options** tab.



Use this tab to specify the following parameters:

ABAP Job the name of the ABAP job that is started on the SAP system. This option must not be blank. You can enter the name of the ABAP job directly or use a global macro variable such as that in the example above.

ABAP Variant the ABAP variant for reports. This option must correspond to a variant that the SAP administrator created when customizing the SAP environment for SAS Solutions Adapter for SAP.

Euro Conversion	converts currency amounts from pre-euro currencies to euros using standard, fixed exchange-rates. Must evaluate to YES or NO (default).
Pre-Euro Currency	the currency prior to the euro, such as marks (DEM) or franc (BEF). The conversion assumes single, pre-euro currency. This data is needed because the results table does not always have a currency stored. In this case, an amount of 100 in the pre-euro era would not represent EUR 100, but would represent 100 in the specified currency. If specified and if euro conversion evaluates to YES, the transformation converts the amounts to euros.

HR Current Pay Extraction

The HR Current Pay Extraction transformation is created in the Transformation Generator of SAS Data Integration Studio. This process involves extracting current base salary from SAP HR using the SAP function **/SAS/HR_EVALUATE_WAGES** provided in the SAP transport.

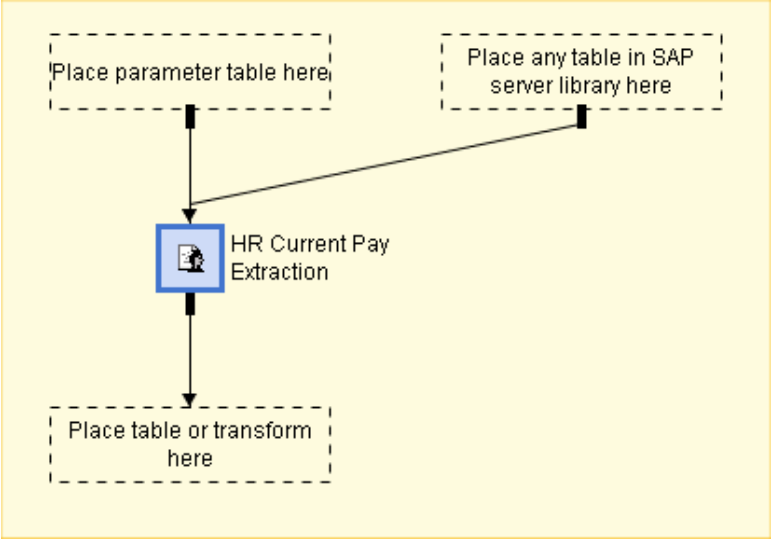
This transformation calls the macro **%adpt_callrfc_evaluate_wages**. Refer to the E08030 Extract Current Pay jobs for usage examples.

This transformation uses the following input tables:

- INPUT_SAP_TABLE
- _INPUT_PARAMETER_TABLE
- _INCLUDE_LIB1 to _INCLUDE_LIB4

Note: The DefaultInfoType option parameter also uses an INPUT_ table. DefaultInfoType specifies the SAP InfoType that is used for extraction unless specified differently in the input parameter table.

The HR Current Pay Extraction transformation can have six input tables. Two of these tables are required. Immediately after you drag and drop the transformation onto a job, the drop zones of the two required tables appear.



This transformation requires the following two table types:

- An input table that includes the drop prompt **`Place any table in SAP server library here`**, which provides important information that the transformation uses. This prompt specifies the SAP server library that is used to call the SAP function module **`/SAS/HR_EVALUATE_WAGES`**. The transformation does not use the actual table itself.
- A table with the drop prompt **`Place parameter table here`** is used to specify input parameters to the ABAP function. This function is called once for each row in the parameter table. The parameters for each individual call are taken from the values of the corresponding columns in the table.

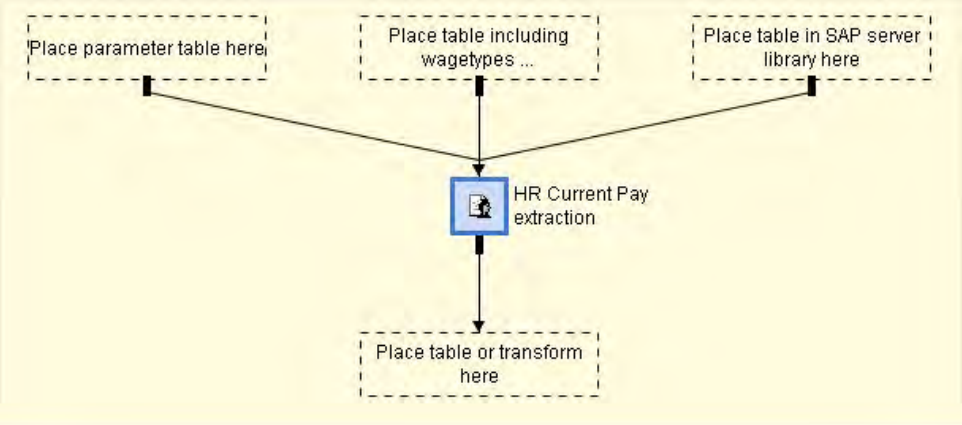
  The columns in the parameter table are required, although the content of columns might be blank. For best results, create and maintain this table with the I0040 Initialize Current Pay Extraction Parameter job.

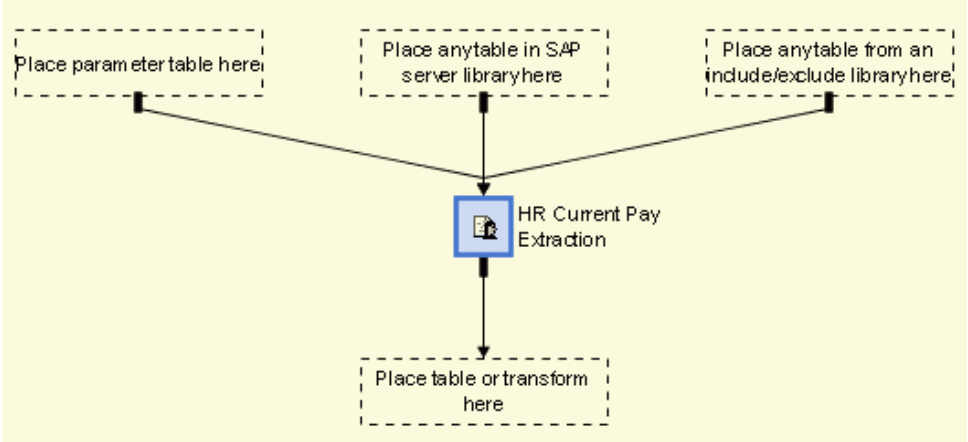
  The following table describes the possible columns in the parameter table:

Table 4.7 Columns in the Parameter Table for HR Current Pay Extraction Transformation

Column	Data Type	Description
PAY_INFOTYPE	$4	Used to select the infotype from which pay info is to be extracted. This is an optional column. The default is 0008.
FROM_PERNR	$8	Specifies the lower limit of personnel numbers. This column can be used together with TO_PERNR to subset the results to a particular group of employees. You can do this to overcome problems if the ABAP function that is called cannot execute in a single run for all employees. This is an optional column. The default is no minimum.
TO_PERNR	$8	Specifies the upper limit of personnel numbers. This column can be used together with FROM_PERNR to subset on employees. This is an option column. The default is no minimum.
INCL_EXCL_TABLE	$41	Specifies the two-level name of a SAS table that contains wage types (in column LGART) to include or exclude. Each SAS library for any of these tables has to be used as an input table of the transformation so that the transformation generates the correct LIBNAME statement. The optional input tables with the drop prompt **Place any table from an include/exclude library here** have to be used for this purpose. This column is optional.
INCL_EXCL_FLG	$1	Specifies whether to include or exclude the wage types that are specified in the INCL_EXCL_TABLE field. Flag with an E to exclude wage types or an I to include wage types. This is an optional column. The default is I.

You can add up to four additional input tables that specify libraries from which wage types are to be included and excluded from the selection.





The table with the drop prompt **`Place any table from an include/exclude library here`** defines the libraries in which the tables are listed in the INCL_EXCL_TABLE column of the parameter table. The table that is dragged and dropped here is used by the library, not the transformation. The wage type table that is used in conjunction with the include/exclude wage types option has a single column. This LGART column has a type of $4 and specifies the wage type.

Note: A different wage type include/exclude table can be used for each row in the parameter table, so any number of include/exclude tables can be added as input to the transformation. However, the wage type include/exclude table is tied to the extraction via the parameter table. Dragging and dropping the include/exclude table onto the job ensures the assignment of the required SAS library. This action also helps represent the logic visually.

The HR Current Pay Extraction transformation returns a results table with the following structure:

Table 4.8 Results Table from the HR Current Pay Extraction Transformation

Name	Data Type	Description
PERNR	$8	Unique identifier for the EMPLOYEE occurrence
INFTY	$4	Infotype relating to wage type
LGART	$4	Wage type
SEQNR	$3	Number of infotype records with the same key
OPKEN	$1	Operation indicator for wage type
INDBW	$1	Indicator for indirect evaluation
BETRG	NUMBER	Salary amount
WAERS	$5	Currency of the associated amount (BETRG)
VALUE_DATE	DATE	Date salary amount is valid

Another output table can be added that specifies a logging table filled with extracted log information. In conjunction with the HR Current Pay Update transformation, this table prevents re-extraction of data that was previously extracted in previous runs of the job. The structure of the logging table is the same as the parameter table, with an additional RUNDATE and EVALDATE columns.

Table 4.9 Logging Table from the HR Current Pay Extraction Transformation

Column	Type	Description
RUNDATE	DATE	Date extract was done
PAY_INFOTYPE	$4	Infotype pay information that is to be extracted
EVALDATE	DATE	Date for the pay information to be valid.
FROM_PERNR	$8	Minimum PERNR in selection
TO_PERNR	$8	Maximum PERNR in selection
INCL_EXCL_TABLE	$41	Name of the SAS table that was used to include and exclude wage types
INCL_EXCL_FLG	$1	Flag E to exclude wage types or I to include wage types. The default is I.

Refer to Appendix 2 "Tables Extracted from SAP," for more information about the associated ABAP function module SAS/HR_EVALUATE_WAGES.

HR Current Pay Update

Overview

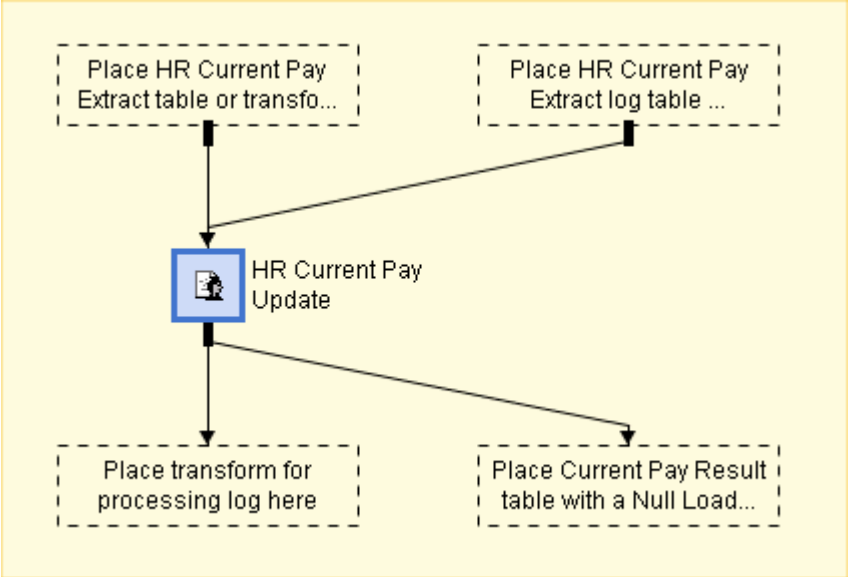
The HR Current Pay Update transformation is used in conjunction with the HR Current Pay Extraction transformation. It is used to post-process the output tables, to update a table that contains all the current pay records from the latest and previous extracts, and to ensure that only later records are extracted in the subsequent extract.

This transformation calls the macro **`%adpt_update_pay_results`**. For a usage example, refer to the T230000 Update Current Pay job.

Using the HR Current Pay Update Transformation

Complete the following steps to use the HR Current Pay Update transformation in a job:

1. Drag and drop the HR Current Pay Update transformation onto a job.
2. Drag and drop input and output tables onto the drop zone.



The input tables for this transformation must be the output tables of an HR Current Payroll Extraction transformation.

The figure below shows the structure of the HR Current Pay Extract input table:

Table 4.10 Structure of the HR Current Pay Extract Input Table

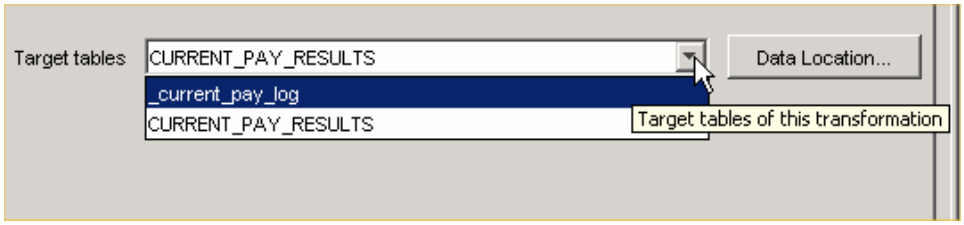
Column	Type	Description
PERNR	$8	Unique identifier for the EMPLOYEE occurrence.
INFTY	$4	Infotype relating to wage type
LGART	$4	Wage type
SEQNR	$3	Number of infotyope records with the same key
OPKEN	$1	Operation indicator for wage type
INDBW	$1	Indicator for indirect evaluation
BETRG	NUMBER	Salary amount
WAERS	$5	Currency of the associated amount (BETRG)
VALUE_DATE	DATE	Date that salary amount is valid

The figure below shows the structure of the HR Current Pay Extract Log input table:

Table 4.11 Structure of the HR Current Pay Extract Log Input Table

Column	Type	Description
RUNDATE	DATE	Date that the extract was done
PAY_INFOTYPE	$4	Which infotype pay information is to be extracted
EVALDATE	DATE	Date for which the pay information is valid.
FROM_PERNR	$8	Minimum PERNR in selection
TO_PERNR	$8	Maximum PERNR in selection
INCL_EXCL_TABLE	$41	Name of the SAS table that was used to include/exclude wage types
INCL_EXCL_FLG	$1	Flag E or I. E=Exclude wage types, I=Include wage types. The default value is I.

3 Click the **Process** tab and provide specifications for the following two target tables for output data:

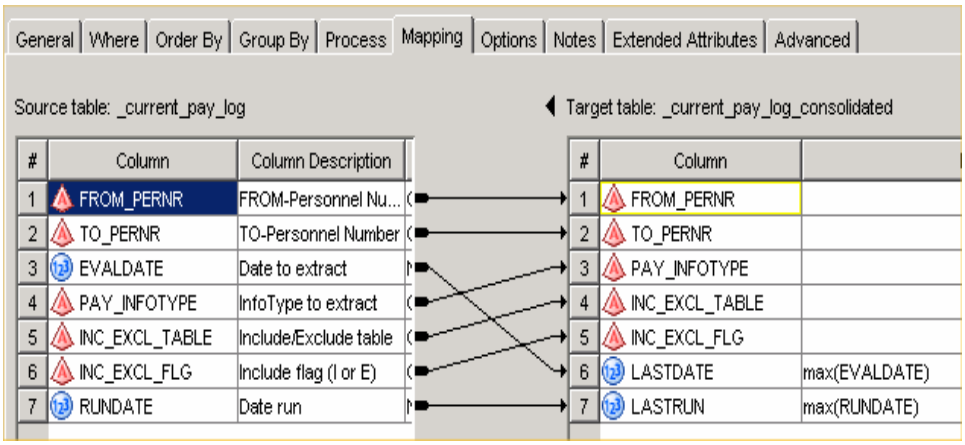


- The main output target table is CURRENT_PAY_RESULTS. This table must contain the complete data for all extraction runs. The update process adds new records and replaces records (by VALUE_DATE) with the new extracted data. Because the HR Current Pay Update transformation handles loading, a null load step must be used.

  The structure of the CURRENT_PAY_RESULTS target table must be the same as the structure of the input HR Current Pay Extract table described above.

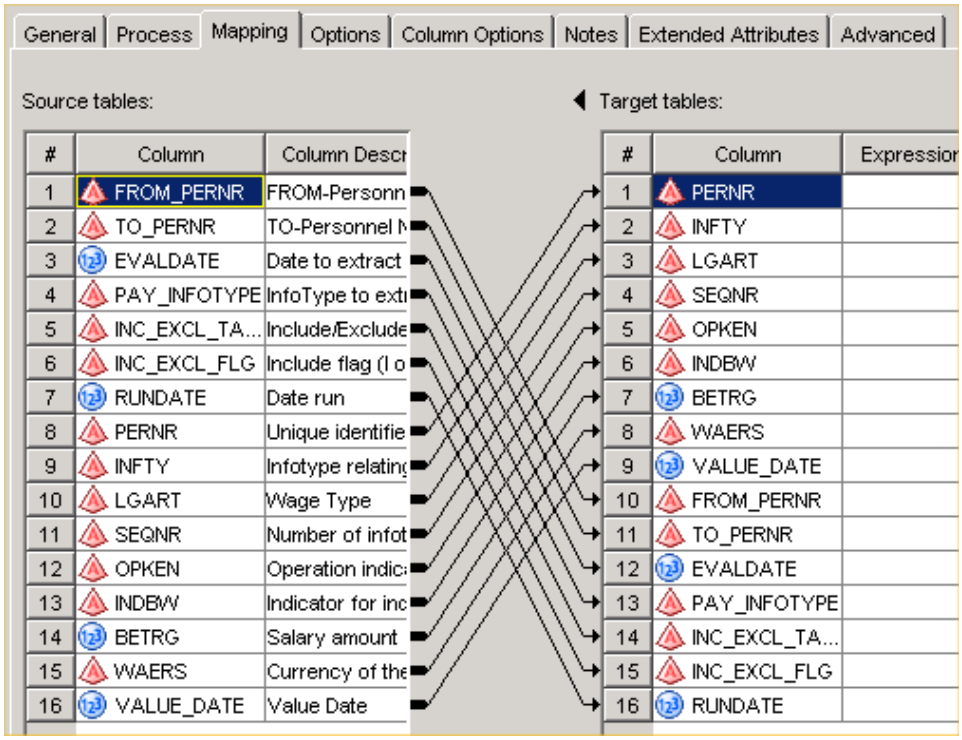
- The secondary output target table is _current_pay_log. This table is populated with information in the same structure as the HR Current Pay Extract Log input table. As a result, the output table or transformation must use the columns in this structure.

  Additional processing for the _current_pay_log output is necessary. To perform this processing, use a transformation that calculates the maximum EVALDATE and RUNDATE. For an illustration of this setup, refer to T230000 Update Current Pay, the job that uses this transaction in the SAS Solutions Adapter for SAP. This job uses the following mapping:

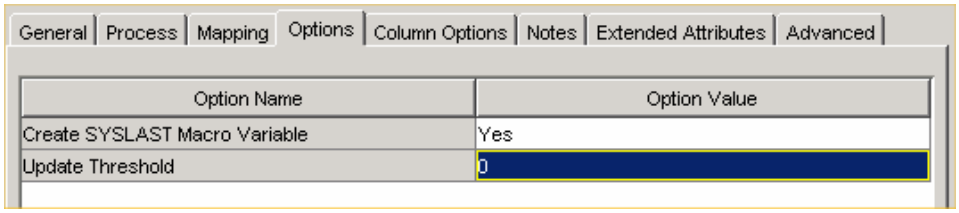


4 Click the **`Mapping`** tab.

5 Set the transformation to use 1:1 mappings for all columns as shown in the example below.



6 Click the **`Options`** tab.



7 Specify the **Update Threshold** option for the HR Current Pay Update transformation. **Update Threshold** sets the upper percentage of deleted records in the table, before the output table is purged. For example, with each update run, old data is deleted from the main output table and new data is added. This process can increase the number of unused records (records marked for deletion) that are in the output table. A value of 25 would indicate that the main output table is updated till 25% of its records are marked for deletion. Then, it is recreated to free this unused space.

The default value for the update threshold is 0. This value indicates that no records that are marked for deletion are kept in the main output table. This specification is comparable to the standard Drop Table Load Technique of the Loader transformation.

Additional Generic Transformations

About Generic Transformations

A group of transformations called data transforms are based on the transformation generator. This group includes transformations that address data processing for the SAS Solutions Adapter for SAP but not solely SAP data processing. As these generic transformations are not restricted to SAP functionality, they can be used in jobs other than those provided by the SAS Solutions Adapter for SAP.

Keep First or Last Record in Group

This transformation keeps the first or last record in a group. It differs from sort nodupkey in that it can control which record is retained.

Data Step Merge

This transformation includes a DATA step that merges a master table (left table) with one or more additional tables. The purpose of this process is to single parse through all tables, which is not possible using SQL left join.

Convert Number Order to Integer Value

Some columns depict numerical order by using floating-point numbers. In these cases, it is sometimes necessary to convert these values to unique integer values for acceptance in a subsequent data model. The Convert Number Order to Integer Value transformation addresses this issue and accepts a single table as INPUT.

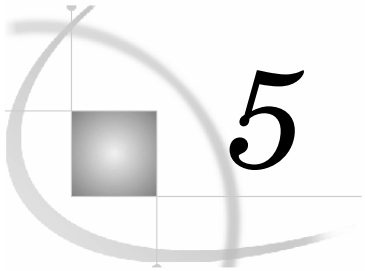
In the **Column Options** tab of the transformation, a single column must be selected. This column has its existing order changed to an integer value. The transformation sorts the input table by the column that was selected. An update step substitutes the existing column value with an integer that corresponds to the DATA step iteration. The value of each subsequent row in the table is then incremented by 1.

This transformation does not have an output table because all work is carried out in place on the input table.

Map SAP Address Lines to SAS Address Lines

SAS data models use the field names Address_Line_1 through Address_Line_4 to store address data that appears after the addressee or company name, but before the city and postcode. SAP has many specialized fields that are available to store address data. The Map SAP Address Lines to SAS Address Lines transformation enables you to select between one and four relevant address columns from SAP data and map them to the SAS columns Address_Line_1 to Adress_Line_4.

In the **`Column Options`** tab of the transformation, select the relevant SAP address columns that correspond to the required SAS address columns. For example, you can define SAP columns CNAME, STRAS, and STR2 to pass to SAS as Address_Line_1, Address_Line_2, and Address_Line_3, respectively. If a particular data record has missing STRAS data, then the STR2 value populates the Address_Line_2 column and the Address_Line_3 column is blank.



5 Initialization Jobs

Introduction to Initialization Jobs

Initialization jobs create Admin tables that extraction jobs and transformation jobs use. Initialization jobs only need to be run once.

These jobs run to initialize your environment and provide data for SAS Human Capital Management.

I0010 Initialize Admin Base Wage Type Groups

This initialization job selects a wage type group that is used to determine the wage types for base salary. To ensure that all the correct wage type groups are selected, review the selection in the WHERE clause of the extract step.

I0020 Load Additional HR Action Types

The Load Additional HR Action Types initialization job creates additional HR action types for actions that are automatically detected. These actions are used when SAP does not explicitly record an action but information columns have changed for an employee who is affected by actions. The additional actions indicate what change was detected in the information columns.

Additional actions include the following action codes:

`action_type_cd='APPRAISAL'`
: is the action code for performance appraisal.

`action_type_cd='0007'`
: indicates a change in working time.

`action_type_cd='0008'`
: indicates a change in pay.

Note: Descriptions provided in the **i0020_init_additional_action_types.sas** source code are in English. You can substitute or use other language descriptions as needed.

I0030 Load Admin Work Hours in Payfrequency

This initialization job is site-dependent and must be reviewed at each local implementation. The job associates the number of hours per pay period, and a value needs to be created for each value in the SAP table T549. As these values might change over time, a date range can be associated for each value. The example source code for this job is **i0030_load_payfreq_hours.sas**.

I0040 Initialize Current Pay Extraction Parameter

This job controls what is extracted by the E08030 Extract Current Pay job, a job that returns the wages for each employee at a specified date. The Initialize Current Pay Extraction Parameter job that is provided at implementation is an example and should be reviewed for your local site.

Here are some ways to control the extraction by the E08030 Extract Current Pay job:

- Select a remuneration infotype. The following infotypes are supported:
  - 0008 = Base salary (default value)
  - 0014 = Recurring payments/deductions
  - 0015 = Additional payments
  - 0052 = Wage maintenance
- Specify a range of personnel numbers to process with min and max values.
- Specify a selection list of wage types that should be included or excluded in the result.

The source code **i0040_create_current_pay_xparms.sas** creates a table that specifies the set parameters.

Because the extraction is performed via CALLRFC, which runs interactively, each extraction must be a reasonable size so that it is able to run without timing out in SAP. An effective way to limit the size of each extraction is to add multiple rows in the parameters table because each row launches a separate extraction. If running multiple extractions is necessary, it is possible to have each row in the parameter table specify a subset of the employees. These subsets are specified by a range of employee numbers (PERNRs) in the columns FROM_PERNR and TO_PERNR.

In addition, it is also possible to control the wage types (LGART) that are extracted. This is done by creating a wage type parameter table that contains a single column called LGART. The name of this wage type parameter table must then be specified in the main parameter table. An INCL_EXCL_FLAG flag is then set to **I** (include) or **E** (exclude). Although this parameter is not used in the example adapter code, it is recommended that you use the mechanism to reduce the amount of extracted data.

I0050 Set Blank Business Area Name

This initialization job assigns a text description to business areas that do not have a description in the transaction data from SAP. The SAS Solutions Adapter for SAP requires business area descriptions when creating default internal organization dimensions. However, descriptions for business areas can be missing in the transaction data from SAP because SAP allows blank descriptions in business area text tables.

The initialization job I0050 Set Blank Business Area Name provides a default description for business areas when needed. When a business area description is blank in the transaction data from SAP, the example source code **`i0050_set_blank_busarea.sas`** assigns a default description of "Business Area not specified."

Note: This initialization job is site-dependent and must be reviewed for each implementation. The code must be adapted to add additional language descriptions.

CHAPTER 6

Extraction Jobs

Extraction Jobs

The SAS Solutions Adapter for SAP uses several extraction jobs to extract data from SAP R/3. Here are the four essential types of extractions that the adapter uses:

Simple complete extractions

extract and load all rows of a table from the SAP server into a SAS mirror table. Rerunning these extractions re-extracts the entire table and automatically retrieves all new information.

Simple partial extractions

extract and load a subset of SAP tables (not all rows and not all columns) into a partial mirror table. Rerunning these extractions re-extracts the same logical parts of the table and automatically gets all new information that is relevant.

Change data capture extractions

extract and load data that was added or changed in SAP since the previous extraction. These extractions are subsequent loads that are used after an initial load is completed by a simple complete extract or a partial extract. The change

data capture extractions are important for large transaction tables. Their complexity is not needed for small amounts of data.

Extractions done via SAP remote-callable functions

serve as the easiest way to extract some SAP data in some cases.

Job Group: E00 Extraction of T Tables

This job group includes common jobs that extract data for both SAS Financial Management and SAS Human Capital Management.

E00010 Extract R3 General and FI Tnnn Tables

This job extracts general and financial SAP R/3 tables that are small and have names prefixed by T. For each table, a mirrored SAS table is created in the library R3mirror. The individual extractions are independent of each other.

This job extracts the following tables:

- T001 – Company Codes
- T002 – Language Keys
- T002T – Language Key Texts
- T003 – Document Types
- T003T – Document Type Texts
- T004 – Directory of Charts of Accounts
- T004T – Chart of Account Names
- T005T – Country Names
- T011 – Financial Statement Versions
- T011T – Financial Statement Version Names
- T880 – Global Company Data (for KONS Ledger)

This job uses the following input:

- T001 from SAP server
- T002 from SAP server
- T002T from SAP server
- T003 from SAP server
- T003T from SAP server
- T004 from SAP server
- T004T from SAP server
- T005T from SAP server
- T011 from SAP server
- T011T from SAP server
- T880 from SAP server

This job creates the following output:

- R3mirror.T001
- R3mirror.T002

- R3mirror.T002T
- R3mirror.T003
- R3mirror.T003T
- R3mirror.T004
- R3mirror.T004T
- R3mirror.T005T
- R3mirror.T011
- R3mirror.T011T
- R3mirror.T880
- **$LOCALCR.**
- **$COM2CAC.**
- **$BUKRS.**
- **$COMPANY_TO_FY_VARIANT.**
- **$TRAD_PTNR_TO_BUKRS.**
- **$SAPTOSAS_LANGUAGE.**

E00011 Formats from R3 General and FI Tnnn Tables

This job creates formats from the tables that are extracted in the job E00010 Extract R3 General and FI Tnnn Tables. Run this job immediately after the E00010 job to ensure that the formats are always as current as the tables on which they are based.

This job uses the following input:

- R3mirror.T001
- R3mirror.T002
- R3mirror.T011

This job creates the following output:

- **$LOCALCR.**
- **$COM2CAC.**
- **$BUKRS.**
- **$COMPANY_TO_FY_VARIANT.**
- **$TRAD_PTNR_TO_BUKRS.**
- **$SAPTOSAS_LANGUAGE.**
-

E00020 Extract TXXX General and FI Tables

This job extracts general and financial SAP R/3 tables that have names prefixed by T. These are generally text description tables. For each table, a mirrored SAS table is created in the library R3mirror. The individual extractions are independent of each other.

This job extracts the following tables:

- TBSLT – Posting Key Names
- TGSBT – Business Area Names
- TKA01 – Controlling Areas

- TKA02 – Controlling Area Assignment
- T077X – Account Group Names (Table T077D)
- TKA00 – Control Parameters for Controlling Areas

This job uses the following input:

- TBSLT from SAP server
- TGSBT from SAP server
- TKA01 from SAP server
- TKA02 from SAP server
- T077X from SAP server
- TKA00 from SAP server

This job creates the following output:

- R3mirror.TBSLT
- R3mirror.TGSBT
- R3mirror.TKA01
- R3mirror.TKA02
- R3mirror.T077X
- R3mirror.TKA00
- **`$CC2CA`** converts Company Code (BUKRS) to Controlling Area (KOKRS).

E00030 Extract Financial Period T Tables

This job extracts financial period SAP R/3 tables that have names prefixed by T009. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- T009 – Fiscal Year Variants
- T009T – Fiscal Year Variant Names
- T009Y – Shortened Fiscal Years in Asset Accounting
- T009B – Fiscal Year Variant Periods
- T009C – Period Names

This job uses the following input:

- T009 from SAP server
- T009T from SAP server
- T009Y from SAP server
- T009B from SAP server
- T009Cfrom SAP server
- &lang (set in the preprocessing step)

This job creates the following output:

- R3mirror.T009
- R3mirror.T009T
- R3mirror.T009Y
- R3mirror.T009B
- R3mirror.T009C

E00031 Formats from Financial Period T Tables

This job creates formats from the tables that are extracted in the job E00030 Extract Financial Period T Tables. Run this job immediately after the E00030 job to ensure that the formats are always as current as the tables on which they are based.

This job uses the following input:

- R3mirror.T009C
- **`&lang`** (set in the preprocessing step)

This job creates the following output:

- **`$PERIOD_DESCRIPTION.`** This output converts concatenated Fiscal Year Variant (PERIV) and Posting Period (POPER) to Period Name Long Text (LTEXT).
- $**`PERIOD_NAME.`** This output converts concatenated Fiscal Year Variant (PERIV) and Posting Period (POPER) to Period Name Short Text (KTEXT).

E00050 Extract Currency T Tables

This job extracts currency SAP R/3 tables that have names prefixed by TCUR. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- TCURR – Exchange Rates
- TCURT – Currency Code Names
- TCURW – Usage of Exchange Rate Types

This job uses the following input:

- TCURR from SAP server
- TCURT from SAP server
- TCURW from SAP server

This job creates the following output:

- R3mirror.TCURR
- R3mirror.TCURT
- R3mirror.TCURW

E00060 Extraction of Account T Tables

This job extracts general and financial SAP R/3 tables that have names prefixed by T030 and are related to accounts. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- T030R – Rules for Determination of Standard Account
- T030 – Standard Accounts Table
- T030C – Global Standard Account Table

- T030D – Acct Determ.for Open Item Exch.Rate Differences
- T030H – Acct Determ.for Open Item Exch.Rate Differences
- T030W – Transaction Key Names

This job uses the following input:

- T030 from SAP server
- T030C from SAP server
- T030D from SAP server
- T030R from SAP server
- T030H from SAP server
- T030W from SAP server

This job creates the following output:

- R3mirror.T030 R/3 Mirror T030 – Standard Accounts Table
- R3mirror.T030C R/3 Mirror T030C – Global Standard Account Table
- R3mirror.T030D R/3 Mirror T030D – Acct Determ.for Open Item Exch.Rate Diffs
- R3mirror.T030H R/3 Mirror T030H – Acct Determ.for Open Item Exch.Rate Diffs
- R3mirror.T030R R3mirror.T030R Mirror T030R – Rules for Determining of Standard Accnts
- R3mirror.T030W R/3 Mirror T030W – Transaction Key Names

E00070 Extract Material Hierarchy Tables

This job extracts material hierarchy SAP R/3 tables with names that are prefixed by T179. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- T179 – Materials: Product Hierarchies
- T179T – Materials: Product Hierarchies: Texts

This job uses the following input:

- T179 from SAP server
- T179T from SAP server

This job creates the following output:

- R3mirror.T179
- R3mirror.T179T

E00090 Extract Operating Concern T Tables

This job extracts Operating Concern SAP R/3 tables. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- TKEB – Management for Operating Concerns (Client-Specific)
- TKEBT – Description of Operating Concern

This job uses the following input:

- TKEB from SAP server
- TKEBT from SAP server

This job creates the following output:

- R3mirror.TKEB
- R3mirror.TKEBT

E00100 Extraction of Additional Tables for Formats

This job extracts miscellaneous SAP R/3 tables that are required for formats and have not been extracted elsewhere. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the table DD07T – DD: Texts for Domain Fixed Values (Language-Dependent).

This job uses the following input:

- DD07T from SAP server
- **`&lang`** (set in the preprocessing step)

This job creates the following output:

- R3mirror.DD07T
- **`$SHKZG.`** This output uses table R3mirror.DD07T to create format **`$SHKZG`**. It also converts Values for Domains: Single Value/Lower Limit (DOMVALUE_L) to short text for fixed values (DDTEXT). This stores format in the Library.fmt&lang catalog which is language-dependent.

E00110 Extraction of Payroll T Tables

This job extracts SAP R/3 tables that are related to payroll. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- T549A Payroll Areas
- T549Q Payroll Periods
- T549R Period Parameters

This job uses the following input:

- T549A from SAP server
- T549Q from SAP server
- T549R from SAP server

This job creates the following output:

- R3mirror.T549A
- R3mirror.T549Q
- R3mirror.T549R
- **`$PAYROLL_AREA_TO_PERIOD.`** This output uses table R3mirror.T549A to create format **`$PAYROLL_AREA_TO_PERIOD`**.

E00120 Extract Time Frequency Measurement Tables

This job extracts miscellaneous SAP R/3 tables that are required for formats and have not been extracted elsewhere. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the table T538T Units of Time/Measurement Texts.

This job uses T538T from SAP server as input.

This job creates R3mirror.T538T as output.

Job Group: E01 Extraction of Master Tables

This job group includes common jobs that extract data for both SAS Financial Management and SAS Human Capital Management.

E01000 Extract R3 Profit Center Master

This job extracts profit center master tables from SAP R/3. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- CEPC – Profit Center Master Data Table
- CEPCT – Texts for Profit Center Master Data

This job uses the following input:

- CEPC from SAP server
- CEPCT from SAP server

This job creates the following output:

- R3mirror.CEPC
- R3mirror.CEPCT

E01010 Extract R3 Cost Center Master

This job extracts cost center master tables from SAP R/3. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- CSKS – Cost Center Master Data
- CSKT – Cost Center Texts

This job uses the following input:

- CSKS from SAP server
- CSKT from SAP server

This job creates the following output:

- R3mirror.CSKS
- R3mirror.CSKT

E01020 Extract R3 Material Master + Hierarchy

This job extracts material master and hierarchy tables from SAP R/3. For each table, a mirrored SAS table is created in the library R3mirror. Note that this job partially overlaps functionality with E0070. If the product hierarchy is needed, E01020 is the preferable job.

This job extracts the following tables:

- MARA – General Material Data
- MAKT – Material Descriptions
- T179 – Materials: Product Hierarchies
- T179T – Materials: Product Hierarchies: Texts

This job uses the following input:

- MARA from SAP server
- MAKT from SAP server
- T179 from SAP server
- T179T from SAP server

This job creates the following output:

- R3mirror.MARA
- R3mirror.MAKT
- R3mirror.T179
- R3mirror.T179T

E01100 Extract R3 Customer Master

This job extracts the customer SAP R/3 master table. A mirrored SAS table is created in the library R3mirror.

This job extracts the table KNA1 – General Data in Customer Master.

This job uses KNA1 from SAP server as input.

This job creates R3mirror.KNA1 as output.

E01200 Extract R3 Vendor Master

This job extracts the vendor SAP R/3 master table. A mirrored SAS table is created in the library R3mirror.

This job extracts the table LFA1 – Vendor Master (General Section).

This job uses LFA1 from the SAP server as input.

This job creates R3mirror.LFA1 as output.

E01400 Extract R3 Account Master

This job extracts SAP R/3 Account Master tables. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- SKA1 – G/L Account Master (Chart of Accounts)
- SKAT – G/L Chart of Accounts Description
- SKB1 – Account Master (Company Code)

This job uses the following input:

- SKA1 from SAP server
- SKAT from SAP server
- SKB1 from SAP server

This job creates the following output:

- R3mirror.SKA1
- R3mirror.SKAT
- R3mirror.SKB1
- **`$ACCBALS.`** This output uses the R3mirror.SKA1 table to create format **`$ACCBALS`**. It also converts Values for G/L Account Number (SAKNR) to Indicator: Account is a Balance Sheet Account? (XBILK). This stores format in the Library.fmt&_ktopl&lang catalog that is for chart of accounts and is language-dependent.

Job Group: E02 Extraction of Transaction Tables

This job group includes jobs that extract data for SAS Financial Management.

E02000 Initial Extraction Financial Documents BKPF and BSEG

Overview

This job does a complete extraction of the SAP R/3 financial document tables. The extraction job E00010 Extract R3 General and FI Tnnn Tables must execute before this job. The extraction job E02005 must run after this job to create a consolidated table of financial documents data.

Refer to the related extract E02210 Extraction of Financial Documents BKPF and BSEG for more information.

This job extracts the following tables:

- BKPF – Accounting Document Header
- BSEG – Accounting Document Segment

This job uses the following input:

- BKPF from SAP server
- BSEG from SAP server
- **`$LOCALCR.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$COM2CAC.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$TRAD_PTNR_TO_BUKRS.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$BUKRS.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`&_startfinyear`** (set in preprocessing step)
- **`&overlap`** (set in preprocessing step)

This job creates the following output:

- R3mirror.BKPF
- R3mirror.BSEG
- Intermed.Financial_Documents

Additional Notes

This job does a complete extract of the financial document tables. It also executes the SAS macro **`%adpt_update_bkpf_delta`** to update the delta control date record for the BKPF table.

Although this job should run once during the first complete run of the job suite for SAS Data Integration Studio, you might need to completely refresh data at a later stage if you suspect that there is incomplete data.

You must also run the E02210 Extraction of Financial Documents BKPF and BSEG job in all subsequent cases. This is necessary because the initial load does a complete extraction of the data from the BKPF and BSEG tables depending on the date setting in macro variable **_startfinyear**. However, subsequent loads save extraction time by only extracting changes made to those tables.

The initial extraction works in the following two stages:

- The first stage extracts data from SAP to SAS data sets. This stage fully extracts BSEG but only partially extracts BKPF. From BKPF, this stage extraction is restricted to those rows that have a Fiscal Year (GJAHR) not less than the value specified in the macro variable **`_startfinyear`**.
- The second stage joins the two resulting SAS data sets. As the BKPF extract is restricted by date, a SQL left join performs, thus joining information from BSEG for the records in BKPF.

E02005 Merge Financial Documents BKPF and BSEG

This job joins the results of job E02000 to provide a consolidated table of the SAP R/3 financial document tables. If customization is required for this job, you can rerun the job. Because E2000 creates a complete extract of the input tables, running this join process again does not create any load on the SAP server.

The job E02000 Initial Extraction Financial Documents BKPF and BSEG must execute before this job.

This job joins the following R3mirror tables:

- BKPF – Accounting Document Header
- BSEG – Accounting Document Segment

This job uses the following input:

- BKPF from R3mirror
- BSEG from R3mirror

This job creates Intermed.Financial_Documents as output.

E02010 Initial Extraction Controlling Objects COBK COEP

This job extracts the Controlling Objects master table. This data is not used but is available if needed.

This job extracts the following tables:

- COBK– Object Document Header
- COEP – Object Line Items

This job uses the following input:

- COBK from SAP R/3 server
- COEP from SAP R/3 server

This job creates the following as output:

- R3mirror.COBK
- R3mirror.COEP

E02210 Extraction of Financial Documents BKPF and BSEG

Overview

This job extracts delta changes made to SAP R/3 financial document tables. Refer to the related extract E02000 Initial Extraction Financial Documents BKPF and BSEG for more information.

The following jobs must execute before this job:

- E00010 Extract R3 General and FI Tnnn Tables
- E02000 Initial Extraction Financial Documents BKPF and BSEG

This job extracts the following tables:

- BKPF – Accounting Document Header
- BSEG – Accounting Document Segment

This job uses the following input:

- BKPF from SAP server
- BSEG from SAP server
- **`$LOCALCR.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$COM2CAC.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$TRAD_PTNR_TO_BUKRS`**. (created in E00010 Extract R3 General and FI Tnnn Tables)

- **`$BUKRS.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`&_startfinyear`** (set in preprocessing step)
- **`&overlap`** (set in preprocessing step)

This job creates the following output:

- R3mirror.BKPF
- R3mirror.BSEG

Additional Notes

This job includes **`e02210_Check_Delta_Control_Date.sas`** and executes the SAS macro **`%adpt_deltactl2`** to obtain the latest delta control date record for table BKPF. This job also includes **`e02210_update_bkpf_delta_control_date.sas`**, which calls the macro **`%adpt_update_bkpf_delta`** to update the delta control date record for table BKPF.

If you require a complete refresh of financial document data, you can run the E02000 Initial Extraction Financial Documents BKPF and BSEG job. This job extracts only delta changes that were made since the last extraction.

Extracting changes to the BKPF and BSEG tables efficiently is one of the most technically demanding tasks for the SAS Solutions Adapter for SAP. Because a full extraction takes many hours and uses a significant amount of SAP R/3 and network resources, it is not desirable to do a full extraction frequently.

There is no perfect solution to this problem because SAP does use a perfect process to create a timestamp for the tables. However, you can use the following columns in BKPF to recognize changes:

- CPUDT (accounting document entry date)
- AEDAT (date of last document change by transaction)
- UPDDT (date of the last document update)

Likewise, BSEG has no date-and-time stamps that are useful for extraction. As a result, the adapter must identify the new and changed records in the heading table BKPF and then extract the matching items from BSEG. A server-side join extracts the changed records in BKPF and BSEG.

This process restricts the extract by checking the three dates above against an extract date that is stored in a SAS data set named deltacontrol in the Admin Library. This extract date in deltacontrol helps identify changed records. The adapter reads and uses this date to subset the extraction. It then updates the new latest extract date in a post-processing step. The macro **`%adpt_deltactl`** updates the deltacontrol table.

You can also use the macro variable **`overlap`** to specify a value of overlap days. This action subtracts the specified number of days from the last extract date and thus increase the amount of data extracted.

Job Group: E03 Extraction of Summary Tables

This job group includes jobs that extract data for SAS Financial Management.

E03000 Extraction of R3 GLT0 Account Transaction Summary

This job extracts SAP R/3 Account Transaction Summary tables. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the table GLT0 – G/L account master record transaction figure.

This job uses GLT0 from SAP server as input.

This job creates R3mirror.GLT0 as output.

Job Group: E04 Extraction of Hierarchies and Structures

This job group includes jobs that extract, transform, and load data for SAS Financial Management.

E04010 Extract R3 Balance Sheet and Financial Statement

This job extracts financial statement data from SAP. The easiest way to get to this data is by calling remote-enabled functions of SAP. However, the SAP functions that produce these reports are not remote-enabled. As a result, the adapter provides additional functions that are remote-enabled and can call the relevant SAP functions.

This job uses the following input:

- T011 from SAP server
- Macro variables such as the following:
  - **`_versn`** Financial statement version from T011 without quotes
  - **`spras`** Language code

This job creates the following output:

- R3mirror.RF011P
- R3mirror.RF011Q
- R3mirror.RF011Z

This job calls E04010_Callrfc_import_balance_sheet.sas and submits a **`PROC CALLRFC`** to execute the following SAP functions:

- **`/SAS/FI_IMPORT_BAL_SHEET_POS`**
- **`/SAS/FI_IMPORT_BAL_SHEET_TEXT`**

E04050 Extract R3 Work Breakdown Structure

This job extracts data from the table PRPS - Work Breakdown Structure data. This data is not used but is available if needed.

This job uses PRPS from the SAP server as input.

This job creates R3mirror.PRPS as output.

E04060 Supplier Hierarchy

This job extracts supplier hierarchy SAP R/3 tables. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- LFMH – Vendor Hierarchy
- TLHITT – Vendor Hierarchy Category (description)

This job uses the following input:

- LFMH from SAP server
- TLHITT from SAP server

This job creates the following output:

- R3mirror.LFMH
- R3mirror.TLHITT

E04070 Customer Hierarchy

This job extracts customer hierarchy SAP R/3 tables. For each table, a mirrored SAS table is created in the library R3mirror.

This job extracts the following tables:

- KNVH – Customer Hierarchies
- THITT – Texts for Customer Hierarchy Types

This job uses the following input:

- KNVH from SAP server
- THITT from SAP server

This job creates the following output:

- R3mirror.KNVH
- R3mirror.THITT

Job Group: E05 Extraction of Miscellaneous Other Tables

This job group includes jobs that extract data for SAS Financial Management.

E05000 Extraction of Number Ranges NRIV

This job extracts the SAP R/3 table NRIV – Number Range Intervals to create a mirrored SAS table in the library R3mirror.

This job uses NRIV from SAP server as input and creates R3mirror.NRIV for output.

Job Group: E06 Extraction of HR PA Infotype Tables

This job group includes jobs that extract data for SAS Human Capital Management.

E06010 Extraction of General Employee Infotypes

This job extracts SAP R/3 tables that are prefixed with PA. The tables relate to general employee information. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the following tables:

- PA0001 Org. Assignment
- PA0002 Personnel Data
- PA0006 Addresses
- PA0016 Contract Elements
- PA0022 Education
- PA0057 Membership Fees
- PA0077 Additional Personnel Data

This job uses the following input:

- PA0001 from SAP server
- PA0002 from SAP server
- PA0006 from SAP server
- PA0016 from SAP server
- PA0022 from SAP server
- PA0057 from SAP server
- PA0077 from Additional Personnel Data

This job creates the following output:

- R3mirror.PA0001
- R3mirror.PA0002
- R3mirror.PA0006
- R3mirror.PA0016
- R3mirror.PA0022
- R3mirror.PA0057
- R3mirror.PA0077

E06020 Extraction of Employee Actions PA0000

This job extracts the SAP R/3 table that relates to employee actions. The Admin Delta control table is used in order to prevent complete extractions if the jobs are rerun.

This job extracts the PA0000 Actions table.

This job uses PA0000 from SAP server for input and creates R3mirror.PA0000 as output.

E06021 Extraction of Additional Empl. Actions PA0302

This job extracts the SAP R/3 table that relates to additional employee actions. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the PA0302 Additional Actions table.

This job uses the PA0302 from SAP server for input and creates R3mirror.PA0302 as output.

E06022 Extraction of Employee Event Dates PA0041

This job extracts the SAP R/3 table that relates to employee event dates. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the PA0041 Date Specifications table.

This job uses PA0041 from SAP server for input and creates R3mirror.PA0041 as output.

E06030 Extraction of Employee Work Schedule Infotype 7

This job extracts the SAP R/3 table that relates to employee work schedule. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the PA0007 Planned Working Time table.

This job uses PA0007 from SAP server for input and creates R3mirror.PA0007 as output.

E06040 Extraction of Employee Compensation Infotype 8

This job extracts the SAP R/3 table that relates to employee compensation. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the PA0008 Basic Pay table.

This job uses PA0008 from SAP server for input and creates R3mirror.PA0008 as output.

E06050 Extraction of HR Time Record Infotype 2001

This job extracts the SAP R/3 table that relates to employee absences. The Admin Delta control table prevents complete extractions if the jobs are rerun.

This job extracts the PA2001 Absences table.

This job uses PA2001 from SAP server for input and creates R3mirror.PA2001 as output.

E06060 Extraction of Additional HR Infotypes

This job extracts the SAP R/3 tables that relate to maternity and compensation data.

This job extracts the following tables:

- PA0080 Maternity Protection
- PA0029 Workers Compensation

This job uses the following input:

- PA0080 from SAP server
- PA0029 from SAP server

This job creates the following output:

- R3mirror.PA0080
- R3mirror.PA0029

E06070 Extraction of German HR Infotypes

This job extracts the SAP R/3 tables that relate to employee absences.

This job extracts the following tables:

- PA0521 Altersteilzeit
- PA0597 Teilzeitarbeit waehrend Elternzeit

This job uses the following input:

- PA0521 from SAP server
- PA0597 from SAP server

This job creates the following output:

- R3mirror.PA0521
- R3mirror.PA0597

E06080 Extraction of Appraisal HR Infotype 25

This job extracts the SAP R/3 table PA0025 that relates to appraisal data.

This job uses PA0025 from the SAP R/3 server for input.

The job creates R3mirror.PA0025 as output.

Job Group: E07 Extraction of HR T Lookup Tables

This job group includes jobs that extract data for SAS Human Capital Management.

E07010 Extraction of HR Action and Date Lookups

This job extracts SAP R/3 tables that relate to HR action and date lookups.

This job extracts the following tables:

- T548T Data Types
- T529T Personnel Action Texts

This job uses the following input:

- T548T from SAP server
- T529T from SAP server

This job creates the following output:

- R3mirror.T548T
- R3mirror.T529T

E07020 Extract HR Education, Ethnicity, Disability Lookups

This job extracts SAP R/3 tables that relate to HR education, ethnicity, and disability lookups.

This job extracts the following tables:

- T523T Challenge Types
- T519T Final Certificates
- T505S Ethnic Origin Texts

This job uses the following input:

- T523T from SAP server
- T519T from SAP server
- T505S from SAP server

This job creates the following output:

- R3mirror.T523T
- R3mirror.T519T
- R3mirrorT505S

E07030 Extract of HR Country Lookup

This job extracts SAP R/3 tables that relate to HR Country lookups.

This job extracts the following tables:

- T500P Personnel Area
- T500L Personnel Country Grouping

This job uses the following input:

- T500P from SAP server
- T500L from SAP server

This job creates the following output:

- R3mirror.T500P
- R3mirror.T500L

E07031 Formats from HR Country Lookup

This job creates formats from the tables that are extracted in the job E07030 Extract of HR Country Lookup. Run this job immediately after the E07030 job to ensure that the formats are always as current as the tables on which they are based.

This job uses the following input:

- R3mirror.T500P
- R3mirror.T500L

This job creates the following output:

- **`$COMPANY_TO_MOLGA`** from T500P maps BUKRS to MOLGA.
- **`$MOLCLUS`** from T500L maps Country Grouping (MOLGA) to the area identifier for cluster in tables PCLx (RELID).
- **`$COUNTRY_TO_MOLGA`** from T500L maps INTCA to MOLGA.

E07040 Extract of HR Contract Lookup

This job extracts SAP R/3 tables that relate to HR Contract date lookups.

This job extracts the following tables:

- T542T Employment Contracts
- T547S Contract Type Texts

This job uses the following input:

- T542T from SAP server
- T547S from SAP server

This job creates the following output:

- R3mirror.T542T
- R3mirror.T547S

E07050 Extract of HR Union Lookup

This job extracts SAP R/3 tables that relate to HR Union lookups.

This job extracts the following tables:

- T521B Payee Keys
- T521C Check Table for Payees

This job uses the following input:

- T521B from SAP server
- T521C from SAP server

This job creates the following output:

- R3mirror.T521B
- R3mirror.T521C

E07060 Extract of HR Communication Type Lookup

This job extracts the SAP R/3 table that relates to HR Communication type.

This job extracts the T536B Payee Keys table.

This job uses T536B from SAP server for input and creates R3mirror.T536B as output.

E07080 Extract Marital Data - T502T

This job extracts SAP R/3 tables that relate to marital data.

This job extracts the T502T Marital Status table.

This job uses T502T from SAP server as the input and creates R3mirror.T502T as output.

E07090 Extract State Region County Data - T005U

This job extracts SAP R/3 tables that relate to state, region, and county.

This job extracts the T005U State/Region/County table.

This job uses T005U from SAP server as input and creates R3mirror.T005U as output.

E07100 Extract Military Experience Data - T505N

This job extracts SAP R/3 tables that relate to military experience data.

This job extracts the T005U State/Region/County table.

This job uses the following input:

- T005U from SAP server
- PA0081 from SAP server
- T591S from SAP server

This job creates the following output:

- R3mirror.T005U
- R3mirror.PA0081
- R3mirror.T591S

E07110 Extract Absence and Attendance Text - T554T

This job extracts SAP R/3 tables that relate to absence and attendance data.

This job extracts the T554T Absence & Attendance Texts table.

This job uses T554T from SAP server as input and creates R3mirror.T554T as output.

E07120 Extract Personnel Area, Subarea - T001P

This job extracts SAP R/3 tables that relate to personnel area data.

This job extracts the following tables:

- T001P Personnel Area/Subarea
- T503 Employee Group Subgroup

This job uses the following input:

- T001P from SAP server
- T503 from SAP server

This job creates the following output:

- R3Mirror.T001P
- R3Mirror.T503
- SAS Format $personnel_subarea_absence
- SAS Format $personnel_subarea_mobur
- SAS Format $subarea_group_for_time
- SAS Format $personnel_subarea_molga
- SAS Format $employee_group_appraisal_subgrp
- SAS Format $employee_group_subgroup

E07121 Formats from Personnel Area, Subarea

This job creates formats from the tables that are extracted in the previous job E07120 Extract Personnel Area, Subarea – T001P. Run this job immediately after the E07120 job to ensure that the formats are always as current as the tables on which they are based.

This job uses the following input:

- R3Mirror.T001P
- R3Mirror.T503

This job creates the following output:

- SAS Format $personnel_subarea_absence
- SAS Format $personnel_subarea_mobur
- SAS Format $subarea_group_for_time
- SAS Format $personnel_subarea_molga
- SAS Format $employee_group_appraisal_subgrp
- SAS Format $employee_group_subgroup

E07130 Extract Jobs Data - T5U13 & T513S

This job extracts SAP R/3 tables that relate to jobs data.

This job extracts the following tables:

- T513S Job Titles
- T5U13 Jobs

This job uses the following input:

- T513S from SAP server
- T5U13 from SAP server

This job creates the following output:

- R3Mirror.T513S
- R3Mirror.T5U13

E07150 Extract AAP Occupational Categories - T5UAA

This job extracts SAP R/3 tables that relate to occupational categories.

This job extracts the T5UAA AAP Occupational Categories table.

This job uses the T5UAA from SAP server for input and creates R3Mirror.T5UAA as output.

E07160 EEO Occupational Categories - T5UEE

This job extracts SAP R/3 tables that relate to occupational categories.

This job extracts the T5UEE EEO Occupational Categories table.

This job uses T5UEE from SAP server for input and creates R3Mirror.T5UEE as output.

E07170 Extract Recruitment Text Tables

This job extracts SAP R/3 tables that relate to extract recruitment data.

This job extracts the T751F - Text for applicant event type table.

This job uses T751F from SAP server as input and creates R3Mirror.T751F as output.

E07180 Extract Wage Type Texts - T512T

This job extracts SAP R/3 tables that relate to extract wage type data.

This job extracts the T512T - Wage Type Texts table.

This job uses the T512T from SAP server as input and creates R3Mirror.T512T as output.

E07185 Extract Base Wage Type Grouping

This job extracts SAP R/3 tables that relate to base wage type data.

This job extracts the T52D7 - Assign Wage Types to Wage Type Groups table.

This job uses the following input:

- T52D7 from SAP server
- Admin Base Wage Type Groups

This job creates R3Mirror.T52D7 as output.

E07190 Extract HR Relationship T Tables

This job extracts SAP R/3 tables that relate to HR relationship data.

This job extracts the following tables:

- T777V Relationship Texts
- T777T Infotypes
- T777U Subtype Texts

This job uses the following input:

- T777V from SAP server
- T777T from SAP server
- T777U from SAP server

This job creates the following output:

- R3Mirror.T777V
- R3Mirror.T777T
- R3Mirror.T777U

E07200 Extract Job Position Tables T528x, etc

This job extracts SAP R/3 tables that relate to data about extract job position.

This job extracts the following tables:

- T528B Positions
- T528T Positions Texts
- T528C Wage Type Catalog
- T510N Pay Scale for Annual Salaries
- T510F Assign Pay Scale to Currency
- DD03T Texts for Fields
- DD04T Data Element Texts
- T549N Period Modifiers
- T5U29 Position Attributes for US

This job uses the following input:

- T528B from SAP server
- T528T from SAP server
- T528C from SAP server
- T510N from SAP server
- T510F from SAP server
- DD03T from SAP server
- DD04T from SAP server
- T549N from SAP server
- T5U29 from SAP server

This job creates the following output:

- R3Mirror. T528B
- R3Mirror. T528T
- R3Mirror. T528C
- R3Mirror. T510N
- R3Mirror. T510F
- R3Mirror. DD03T
- R3Mirror. DD04T
- R3Mirror. T549N
- R3Mirror. T5U29

E07210 Extraction of Action Reasons T530F and T530T

This job extracts SAP R/3 tables that relate to action reasons data.

This job extracts the following tables:

- T530F - Reasons for Changes
- T530T - Reason for Action Texts

This job uses the following input:

- T530F from SAP server
- T530T from SAP server

This job creates the following output:

- R3Mirror.T530F
- R3Mirror.T530T

E07220 Extract Employee Grp-SubGrp Info

This job extracts SAP R/3 tables that relate to employee grouping data.

This job extracts the following tables:

- T503T – Employee Subgroup Names
- T503Z – Country Assignment for Emp Grp-Subgrp

This job uses the following input:

- T503T from SAP server
- T503Z from SAP server

This job creates the following output:

- R3Mirror.T503T
- R3Mirror.T503Z

E07230A Employee Country Association

This job extracts SAP R/3 tables that relate to employee country associations.

This job extracts the following tables:

- R3Mirror.PA0001
- R3Mirror T500P

This job uses the following input:

- R3Mirror.PA0001
- R3Mirror T500P

This job creates Intermed.EMPLOYEE_COUNTRY_ASSOCIATION as output.

E07230B Employee Country Association

This job extracts SAP R/3 tables that relate to employee country associations.

This job extracts the R3Mirror.PA0001 table.

This job uses R3Mirror.PA0001 as input and creates Intermed.EMPLOYEE_COUNTRY_ASSOCIATION as output.

E07240 Extract Appraisal lookups

This job extracts SAP R/3 tables that relate to appraisal lookup data.

This job extracts the following tables:

- T777Q Proficiency Tests
- T513H Appraisal Criteria

This job uses the following input:

- T777Q from SAP server
- T513H from SAP server

This job creates the following output:

- R3Mirror.T777Q
- R3Mirror.T513H

Job Group: E08 Extraction of HR Tables Using ABAP Calls

This group of jobs extracts SAP R/3 tables that are extracted using ABAP calls.

This job group includes jobs that extract data for SAS Human Capital Management.

E08010A Extraction of Employee Time Info - SALDO

This job extracts SAP R/3 tables that relate to employee time information.

Based on how working hours are processed within your SAP R/3 system, you must choose to extract SALDO data (accumulated monthly working hours) or ZES data (daily working hours). Jobs E08010A Extraction of Employee Time Info – SALDO and E08010B Extraction of Employee Time Info – ZES are alternatives to each other based on your extraction choice.

Performance efficiency for extracting SALDO data is superior and recommended; however, ZES data extraction is available and might be preferable or needed in some cases. You can use the extract E08010A or E08010B as appropriate.

The jobs in this group rely on functionality that is delivered with the transport files that are detailed in Chapter 7, "Transformation Jobs."

This job extracts SAP R/3 cluster tables that relate to employee time information from the SALDO cluster table. The Admin Delta control table prevents complete extractions if the jobs are rerun. The SALDO table contains accumulated balances of the hours worked in a month.

This job extracts the T554T Absence & Attendance Texts table.

This job uses the following input:

- T554T from SAP server
- R3mirror.SALDO Employee Time Balance Monthly

This job creates Inter.SALDO Employee time balances monthly as output.

E08010B Extraction of Employee Time Info - ZES

This job extracts SAP R/3 tables that relate to employee time information from the daily working hours. This job is similar to E08010A Extraction of Employee Time Info – SALDO except that it extracts the detailed information in the table ZES instead of the summarized data in SALDO.

This job extracts the T554T Absence & Attendance Texts table.

This job uses the following input:

- T554T from SAP server
- R3mirror.ZES Employee Time Balances

This job creates Inter.Employee time balances as output.

E08020 Build Current Pay Extract Parameter

This job builds parameters that enable the extraction of current pay information. The Admin Current Pay Extraction table controls the creation of an intermediate table. This intermediate table contains the basic selection in the Admin table. It also includes the selection date that is designed so that a new extract does not re-extract data that was previously extracted.

This job creates the table Intermed.Current_pay_extraction_parms. The job E08030 Extract Current Pay uses this table.

E08030 Extract Current Pay

The Extract Current Pay job extracts Current Pay information. The Admin Current Pay Extraction table controls the extraction in coordination with the HR Current Pay Extraction transformation.

The Extract Current Pay job is run together with other jobs. The following list describes each of these jobs that work together:

- The I0040 Initialize Current Pay Extraction Parameter job performs a one-off initialization.
- The E08020 Build Current Pay Extract parameter job uses the Admin Current Pay Extraction History table to create the extraction parameters. This process accommodates for values that have already been extracted.
- The E08030 Extract Current Pay job writes the results. It also writes to the Inter Current Pay Extract log to record what it has done. Note that if this job is rerun, the entire extraction is repeated until the follow-up job (T230000 Update Current Pay) is run.
- The T230000 Update Current Pay job updates the results table and uses the update log to relate with the Admin Current Pay Extraction History. This action prevents re-extraction of data that is already extracted.

The Extract Current Pay job extracts the PA0001 HR Master Record table.

This job uses the following input:

- PA0001 from SAP server
- Admin.Admin Current Pay Extraction Parameter
- Intermed.Current_pay_extraction_parms

This job creates the following output:

- Intermed.Current_Pay_Results
- Intermed.Current_Pay_Extract_Log

E08040A HR Extract Historical Compensation

This job extracts data about payments that are made to employees since the previous extraction and stores the results in the table Intermed.RT. Running this job is the preferred way of extracting historical compensation information. However, it requires that the SAP system populate the HRPY_RGDIR table, an action that is not performed in all SAP implementations.

Here are the steps that the job performs:

1 Prepare a parameter table Intermed.RT_CLUSTER_PARMS.

   - Information about which payments were made is taken from the table HRPY_RGDIR.
   - Information about what was previously extracted is taken from the table Admin.deltacontrol.
   - The table Intermed.employee_country_association is required to determine when an employee was associated with a particular country. This factor is important because the historical salary data in SAP is stored in country-dependent cluster tables.

   The combined information from these tables is used to determine what needs to be extracted. This information is then used to create the parameter table. The source code that performs this action is in a macro in **`e08040_get_cluster_combinations.sas`**.

2 Use the HR Payroll Extraction transformation to perform the extraction.

   Note: The efficiency of this step can be influenced dramatically by the SAP ABAP program variant. See Chapter 2, "SAP Administration Tasks," for more information about defining a variant.

This job uses the following input:

- Intermed.Employee_Country_Association
- Admin.Deltacontrol
- HRPY_RGDIR from SAP Server
- macro variable **`&hr_overlap`**
- macro variable **`&_start`**

This job creates the following output:

- Intermed.RT
- Admin.Deltacontrol

E08040B HR Extract Historical Compensation

This job extracts data about payments that are made to employees since the previous extraction and stores the results in the table Intermed.RT. The job E08040A HR Extract Historical Compensation is the preferred way of extracting historical compensation information. However, it requires that the SAP system populate the HRPY_RGDIR table, an action that is not performed in all SAP implementations.

The job E08040B HR Extract Historical Compensation differs from E08040A HR Extract Historical Compensation in that E08040B determines the extraction parameters by deriving the unique combinations of parameters that are possible. This can mean that extractions are made for which there is no data.

Here are the steps that the job performs:

1 Prepare a parameter table Intermed.RT_CLUSTER_PARMS by finding the unique combinations that are possible to determine from the input tables. The source code that performs this action is in a macro in **`e08040_get_cluster_combinations.sas`**.

2 Use the HR Payroll Extraction transformation to perform the extraction.

Note: The efficiency of this step can be influenced dramatically by the SAP ABAP program variant. Refer to Chapter 2, "SAP Administration Tasks," for more information about defining a variant.

This job uses the following input:

- Intermed.Employee_Country_Association
- Admin.Deltacontrol
- R3mirror.T549Q
- R3mirror.PA0001
- PA0001 from SAP server
- HRPY_RGDIR from SAP server
- macro variable **`&hr_overlap`**
- macro variable **`&_start`**

This job creates the following output:

- Intermed.RT
- Admin.Deltacontrol

Job Group: E09 Extraction of HR Applicant Tables

This job group includes jobs that extract data for SAS Human Capital Management.

E09010 Extraction of Applicant Tnnn Tables

This job extracts SAP R/3 tables that relate to HR applicant data.

This job extracts the following tables:

- T750B Recruitment Advertisement
- T750C Recruitment Instrument
- T750H Text for Recruitment Medium
- T750i Text for an unsolicited application group
- T751B Text for Applicant Status
- T750C Applicant status reason

This job uses the following input:

- T750B from SAP server
- T750C from SAP server
- T750H from SAP server
- T750i from SAP server
- T751B from SAP server
- T750C from SAP server

This job creates the following output:

- R3mirror.T750B
- R3mirror.T750C
- R3mirror.T750H
- R3mirror.T750i
- R3mirror.T751B
- R3mirror.T750C
- **`$recruitment_media.`**
- **`$ad_to_recruitment_ins.`**

E09020 Extraction of General Applicant Infotypes

This job extracts SAP R/3 tables that relate to general HR applicant data.

This job extracts the following tables:

- PB0001 HR Master Record Org. Assignment
- PB0002 HR Master Record Personal Data
- PB0006 HR Master Record Addresses
- PB0007 HR Master Record Planned Working Time
- PB0008 Applicant Master Recruitment Basic Pay
- PB0016 Applicant Master Recruitment Contract Elements
- PB0022 HR Master Record Education
- PB0077 HR Master Record
- PB4000 Applicant Events
- PB4001 Applications
- PB4002 Vacancy Assignment

This job uses the following input:

- PB0001 from SAP server
- PB0002 from SAP server
- PB0006 from SAP server
- PB0007from SAP server
- PB0008 from SAP server
- PB0016 from SAP server
- PB0022 from SAP server
- PB0077 from SAP server
- PB4000 from SAP server
- PB4001 from SAP server
- PB4002 from SAP server

This job creates the following output:

- R3mirror.PB0001
- R3mirror.PB0002
- R3mirror.PB0006

- R3mirror.PB0007
- R3mirror.PB0008
- R3mirror.PB0016
- R3mirror.PB0022
- R3mirror.PB0077
- R3mirror.PB4000
- R3mirror.PB4001
- R3mirror.PB4002

Job Group: E10 Extraction of HR HRP Infotype Tables

This job group includes jobs that extract data for SAS Human Capital Management.

E10010 Extract HR Hierarchy - Infotype 1000, 1001

This job extracts SAP R/3 tables that relate to HR hierarchies.

This job extracts the following tables:

- HRP1000
- HRP1001
- HRP1015

This job uses the following input:

- HRP1000 from SAP server
- HRP1001 from SAP server
- HRP1015 from SAP server

This job creates the following output:

- R3mirror.HRP1000
- R3mirror.HRP1001
- R3mirror.HRP1015

E10020 Extract HR Object Address, Email

This job extracts SAP R/3 tables that relate to HR address and e-mail.

This job extracts the following tables:

- HRP1028
- HRP1032

This job uses the following input:

- HRP1028 from SAP server
- HRP1032 from SAP server

This job creates the following output:

- R3mirror.HRP1028
- R3mirror.HRP1032
- R3mirror.HRP1015

E10030 Extract HRP1045 HRT1045 Emp Appraisal

This job extracts SAP R/3 tables that relate to employee appraisal.

This job extracts the following tables:

- HRP1045
- HRT1045

This job uses the following input:

- HRP1045 from SAP server
- HRT1045 from SAP server

This job creates the following output:

- R3mirror.HRP1045
- R3mirror.HRT1045

CHAPTER 7

Transformation Jobs

Transformation Jobs

The SAS Solutions Adapter for SAP uses several jobs that transform data from the SAP R/3 extracted data into the staging area structure. This chapter provides an overview of these jobs.

Job Group: T0000 Transformations for Common Tables

This job group includes common jobs that transform data for both SAS Financial Management and SAS Human Capital Management.

This job group also contains jobs that logically fit into more than one of the other job groups.

T000010 Load INTERNAL_ORG_ASSOC_TYPE

This job loads the STAGE_INTERNAL_ORG_ASSOC_TYPE table with constant values. The figure below shows the values that are required for the internal organization dimensions.

#	Internal Organization Association Type Code	Language Code	Internal Organization Association Type Description
1	OCC	en	Op Concern, Controlling Area, Company, Cost Center
2	HRG	en	HR Work Groups

You can use the following statement to create and specify the language that you need:

```
LANGUAGE_CD = PUT ("&lang", $SAPTOSAS_LANGUAGE.);
```

The actual descriptions are in English. For descriptions in other languages, you can modify the source code in **`t000010_load_staging_internal_org_assoc_type.sas`** accordingly.

This job uses the following input:

- Macro variable **`&lang`**
- Format **`$SAPTOSAS_LANGUAGE`** (created by E00010 Extract R3 General and FI Tnnn Tables)

This job creates the StageSAP.STAGE_INTERNAL_ORG_ASSOC_TYPE table as output.

T000020 Populate Organization Types

This job loads the STAGE_ORG_TYPE table with constant values. The figure below shows StageSAP.ORG_TYPE table values that are required for both the internal and external organization dimensions.

#	Organization Type Code	Language Code	Organization Type Description
1	ALL	en	All types
2	BA	en	Business Area in Company
3	CAR	en	Controlling Area
4	COM	en	Company
5	CUS	en	Customers
6	HR	en	Human Resources Group
7	NA	en	Not applicable
8	OPC	en	Operating Concern
9	SUP	en	Suppliers

You can use the following statement to create and specify the language that you need:

```
LANGUAGE_CD = PUT ("&lang", $SAPTOSAS_LANGUAGE.);
```

The actual descriptions are in English. For descriptions in other languages, you can modify the source code in **`t000020_populate_org_type.sas`** accordingly.

This job uses the following input:

- Macro variable **`&lang`**
- Format **`$SAPTOSAS_LANGUAGE`** (created by E00010 Extract R3 General and FI Tnnn Tables)

This job creates the StageSAP.ORG_TYPE tables as output.

T000030 - Set Exchange Rate Set

This job uses a user-written code to populate the corresponding table. This table holds a note for the sources of exchange rates. The supplied code gives the value in SAP with the language code en (English) and a description in that language.

You can alter the SAS program to support another language or alternative exchange rates. This is helpful, for example, when exchange rates are obtained or maintained in an external system.

This job uses **t000030_set_exch_rate_set.sas** as input.

This job creates StageSAP.CURRENCY_EXCH_RATE_SET as output.

Job Group: T0001 Account Dimension

Overview

The GL Account hierarchy is based on SAP financial statement reports. The financial statement reports include Profit and Loss (P&L) and balance sheet reports. In SAP, the financial statement is dependent on both language and the chart of accounts.

There is no satisfactory SAP hierarchy that is standard to display account structure. As a result, extraction of the financial statement is done by using the RFC version of the SAP function modules FI_IMPORT_BALANCE_SHEET_POS and FI_IMPORT_BALANCE_SHEET_TEXT .

There is also no generic way to accurately populate some of the classification columns such as account_type_cd for all rows of the STAGE GL_ACCOUNT table. In order to prepare all account dimension data for import into the DDS, nonstandard and customer-specific code must incorporate the additional attributes.

The SAS Solutions Adapter for SAP supplies example code that you must adapt to these installation business rules. This example code helps you get started, but you must modify it to work for your environment. If not modified, the example code produces an error message that informs you accordingly.

Note: The value of ACCOUNT_TYPE_CD (in STAGE_GL_ACCOUNT) must be one of the predefined values for account type in the SAS_GL_ACCOUNT_TYPE table.

This job group includes jobs that transform data for SAS Financial Management.

Inputs and Outputs (Tables, Formats, Macro Variables)

The following list describes general inputs to this job group:

- Descriptions must be in the language that is specified by **&SPRAS** or **&LANG**, as indicated in the global parameters. SPRAS contains quotation marks. LANG does not include quotation marks.
- The value of the source_system_id column is set to the value of **&dds_source_system_id**.
- **&VERSN = Financial Statement Version** and alternatively **&_VERSN quoted Financial Statement Version**.
- The following SAP data sources are also inputs to this job group:
  - SKA1 – G/L Account Master (Chart of Accounts). This source is used directly as a table and is used to create format **$ACCBALS**.
  - SKAT – G/L Chart of Accounts Description.
  - T011 – Financial Statement Versions.
  - T011T – Financial Statement Version Names.

- T030C – Global Standard Account Table that contains retained earnings specification.
- T002 – Language Keys that uses format **$SAPTOSAS_LANGUAGE**.

This information is extracted from SAP R/3 using ABAP functions and CALLRFC for a selected balance sheet report. The following code calls ABAP:

```
PROC CALLRFC libref=sapserve;
call function Z_SAS_FI_IMPORT_BAL_SHEET_POS
EXPORTING version=&_versn
OUTTABLES X011P=R3mirror.RF011P
I011Z=R3mirror.RF011Z;
call function Z_SAS_FI_IMPORT_BAL_SHEET_TEXT
EXPORTING sprache=&spras version=&_versn
OUTTABLES X011Q=R3mirror.RF011Q;
run;
```

The job group also uses the following tables as input:

- R/3 Mirror RF011P Items in Financial Statement
- R/3 Mirror RF011Q Financial Statement text
- R/3 Mirror RF011Z Balance sheet line assignment call function

This job group uses the various input values, tables, and source descriptions to create the following output tables:

StageSAP.GL_ACCOUNT_ASSOC_TYPE

The T000140 GL Account Association Tables job creates this output table. This Association Type table lists codes that identify the types of hierarchies that are represented in the association table.

StageSAP.GL_ACCOUNT_ASSOC

The T000140 GL Account Association Tables job creates this output table. This Association table lists one or more hierarchical parent/child relationships for the GL_Account member IDs. The gl_account_assoc_type_cd distinguishes each unique hierarchy type.

StageSAP.GL_ACCOUNT

The T000150 Create Stage GL Account with Site Specifics job creates this output table. This table lists General Ledger Account members. The table includes both the child and parent GL account members that are used in traditional hierarchy or dimension. Examples include cash, accounts receivable, current assets, and assets.

Administration Factors

Financial Statement Nodes

The SAP financial statement consists of a series of nodes. There are some predefined root nodes such as Assets and Liabilities. Therefore, the child nodes of the Asset root are also all assets. At the lowest level, the node holds a list of accounts or intervals of accounts. At the top-level, there is a node called “Unassigned” accounts. In this solution, all accounts that are not assigned to other parts of the financial statement are automatically assigned here.

Account Hierarchies and Numbering

The base underlying column in the R/3 table is SKA1.SAKNR when it is mapping to create ACCOUNT_ID. This is only unique in SAP within a specific chart of accounts (KTOPL). However, the SAS Solutions Adapter for SAP can get only a financial statement (=balance sheet, used to build the hierarchy) for a single chart of accounts, so it uses SAKNR. SAKNR is shortened to show only the significant digits, controlled by the macro variable **&SAKLN** that is set in preprocparms12.sas. For example, if SAKNR contains the account number 0000012345 and the **&SAKLN** variable is 6, then the account number is 012345.

SAKNR is used for the leaves of the hierarchy but not the intermediate structure nodes. The structure nodes are the financial statement keys from RF011Z.ERGSO and RF011Z.ERGHB. Actual accounts become the leaves of the hierarchy.

You must understand and consider account hierarchies and naming structures. SAP places some accounts under different parent nodes depending on the sign on the summarized data such as the sum of all transactions for that account during the specified period. RF011Z.ERGSO and RF011Z.ERGHB mean that the child accounts (VONKT to BILKT) might be in two different places in the hierarchy. For example, a positive bank balance is an asset and an overdrawn account is a liability. The roll-up rules for the DDS specify that accounts must be unique and not appear in two places in the hierarchy. To avoid this situation, credit account numbers have a **C** suffix and debit accounts have a **D** suffix. One SAP account is split into two different SAS accounts, but only one of the accounts can be used within the same period. This requires summarizing the data from SAP R/3 before deciding what account transactions have. All transactions, both negative and positive, must post into the same account so that the totals in SAS are consistent with the totals on the balance sheet.

Profit and Loss Accounts

To distinguish between a profit/loss account and a balance account, the application uses the column XBILK in table SKA1. This scenario returns an **X** for balance sheet accounts. Note that SKA1 XBILK is used to create format **$ACCBALS**.

In SAP R/3, an account can belong to the profit-and-loss part of the balance sheet. The financial statement makes it possible to distinguish whether or not a balance sheet account is an asset or liability account. The application only handles a single financial statement or single value of VERSN. Therefore, companies that do not use the same chart of accounts on which the financial statement was based are excluded.

Other Transactions and Account Considerations

You can also use the SAP transaction OB53 – Define Retained Earnings Account to check that **retained_earnings_flg** is determined from table T030C. Most companies use one retained earnings account. For this reason, **X** is used as an indicator.

SAP represents the financial report as a collection of disjoint hierarchies or top-level items. All disjoint hierarchies are combined into a single hierarchy in the STAGE_GL_ACCOUNT_ASSOC output table. In this table, the ORDER_NO column reflects the order in which the items appear in the financial statement.

Accounts that are in the SKA1 table in R/3 but are not in the financial statement hierarchy are added to the UNASSIGNED branch.

The value of NORMAL_BALANCE_CD in STAGE_GL_ACCOUNT must be one of the predefined values for account type that are shipped with the common solution piece in the table SAS_GL_NORMAL_BALACE. These values are **D** for debit and **C** for credit. Also, the value of LANGUAGE_CD should be one of the predefined language values in the table SAS_LANGUAGE. This is accomplished by using the format **$SAPTOSAS_LANGUAGE** (SAP to SAS language code) that is based on SAP table T002.

T000100 Select Row in T011 According to Versn

This job selects a single row from table T011. This job helps to select only the relevant records that should contribute to the account dimension.

This job uses the following input:

- R3mirror.T011 (created in E00010 Extract R3 General and FI Tnnn Tables)
- **&_versn**

This job creates Intermed.T011 as output.

T000105 Create Accounts with Text

This job uses the following input:

- Intermed.T011 (created in T000100 Select Row in T011 According to Versn)
- R3mirror.SKA1 (created in E01400 Extract R3 Account Master)
- R3mirror.SKAT (created in E01400 Extract R3 Account Master)
- **&lang** (set in preprocessing step)

This job creates Intermed.T011 as output.

T000110 Account Item Mappings

Accounts can have only one parent in the GL account dimension hierarchy. SAP uses a methodology in which an account can appear in different positions on the financial statement according to the balance. For example, a bank account with a positive balance can be an asset, and a negative balance can be a liability. As a result, these accounts can appear in two different places in the hierarchy. In order to enable a fixed position in the hierarchy, the SAS Solutions Adapter for SAP creates two accounts. One account is a credit account, and the other is a debit account. The SAS Solutions Adapter for SAP, in turn, adjusts the original SAP account number by appending a corresponding **C** or **D**.

This job uses the following input:

- Intermed.T001 (created in T000100 Select Row in T011 According to Versn)
- Intermed.Accounts_with_text (created in T000105 Create Accounts with Text)
- R3mirror.RF011Z (created in E04010 Extract R/3 Balance sheet / Financial statement)
- **&sakln** (set in preprocessing step)

This job creates the following output:

- Intermed.One2many
- Intermed.AccItem
- Intermed.ACCITEM_TRANSFORMED
- Intermed.SAP_Account_plus_flag

The source **t000110_Add_debit_credit_suffix_to_accno.sas** reads the Intermed.One2many table. For all records where the Credit Item Key (ERGHB) is not equal to the Debit Item key (ERGSO), two records are output. These records include one negative value with a **D** appended at the end of Account interval lower limit (VONKT) and one positive value with a **C** suffix. All other records are simply rewritten.

T000120 Account Hier from Financial Statement

The hierarchy is based on financial statement items and associated text. Because extracted text items are designed to create a reporting template, they include formatting lines such as **--------** and **========**. As a result, the mapping process below excludes these items.

```
Clean '-----' lines, select each line according to preference
```

In order to create the correct parent/child structure, the subsequent processes perform the required manipulations. Additional transformations identify the account items that were not assigned in the financial statement and then add them to the hierarchy.

For hierarchy structure nodes, the account_id is the financial statement key. In the leaf nodes, which are real accounts, the number comes from SAKNR. A **C** or **D** based on the balance distinguishes the accounts that could appear in two places in the hierarchy. For example, a positive bank balance needs to appear under the assets sub-tree. A negative bank balance needs to appear as a liability.

The macro variable **&SAKLN** is used to shorten the account_id of leaf nodes by calling the macro **%adpt_adj_to_signif_account_len**.

This job uses the following input:

- R3mirror.RF011P (created in E04010 Extract R3 Balance Sheet and Financial Statement)
- R3mirror.RF011Q (created in E04010 Extract R3 Balance Sheet and Financial Statement)
- R3mirror.T011 (created in E00010 Extract R3 General and FI Tnnn Tables)
- ACCITEM_TRANSFORMED (created in T000110 Account Item Mappings)
- ACCOUNTS_WITH_TEXT (created in T000105 Create Accounts with Text)

This job creates the following output:

- Intermed.TRF011Q
- Intermed.TRF011P
- Intermed.NOTASS
- Intermed.ACCNT_ITEM_HIER
- Intermed.ACCHIER_WITH_ITEMS

- Intermed.ACCHIER_WITH_UNASSIGNED
- Intermed.ACCOUNT_HIERARCHY (main result)

This job uses the following user-written source:

- **t000120_account_hier_addroot.sas**
- **t000120_not_assigned_account.sas**
- **t000120_transform_rf011q_tr011q.sas**
- **t000120_add_number_for_sort.sas**

T000130 Account Dimension

This job collects and joins information from various sources that are useful in determining the account classification columns.

This job uses the following input:

- Intermed.ACCOUNT_HIERARCHY (created in T000120 Account Hier from Financial Statement)
- Intermed.T011 (created in T000100 Select Row in T011 According to Versn)
- R3mirror.T030C (created in E00060 Extraction of Account T Tables)
- Intermed.ACCITEM_TRANSFORMED (created in T000110 Account Item Mappings)
- **&_ktopl** (set in preprocessing step)
- **&_output**

This job creates intermed.GL_ACCOUNT_WITH_SAP_INFO as output.

The user-written source for this job is **t000130_sap_account_type.sas**.

T000135 Load Statistical Accounts - Site Specific

This job creates intermediate tables from statistical accounts. Because statistical accounts are generally site-specific, this job acts simply as an example. The output tables are also only examples and they do not flow into other tables.

If you use statistical accounts at your local installation, each table must be appended to its counterpart. For example, StageSAP.GL_ACCOUNT corresponds with StageSAP.GL_ACCOUNT_ASSOC.

This job has the following inputs:

- **&dds_source_system_id** (set in preprocessing step)
- **&_versn** (set in preprocessing step)

This job creates the following output:

- Intermed.GL_ACCOUNT_statistical
- Intermed.GL_ACCOUNT_ASSOC_statistical

The user-written source for this job is **t000135_stat_accounts.sas**.

T000140 GL Account Association Tables

This job creates the structure of the staging area tables for the GL account dimension hierarchy.

This job uses the following input:

- Intermed.ACCOUNT_HIERARCHY (created in T000120 Account Hier from Financial Statement)
- R3mirror.T011T (created in E00010 Extract R3 General and FI Tnnn Tables)

This job creates the following output:

- StageSAP.GL_ACCOUNT_ASSOC_TYPE
- StageSAP.GL_ACCOUNT_ASSOC

The user-written source is **`t000140_gl_account_add_dds_assoc_type.sas`**. This source executes the SAS macro **`%adpt_add_dds_assoc_type`** using GL_ACCOUNT as the dimension.

T000150 Create Stage GL Account with Site Specifics

This job serves as sample code. You must adapt this code to your specific site because generic code cannot accurately derive the GL account type for all SAP installations.

This job uses intermed.GL_ACCOUNT_WITH_SAP_INFO as input.

This job creates STAGESAP.GL_ACCOUNT as output.

The user-written source is **`t000150_site_specific_code.sas`**.

Job Group: T0002 Time Dimension

Overview

In the SAS Solutions Adapter for SAP default behavior, the time dimension splits time into financial years, financial years into quarters, and quarters into financial periods or weeks. If a detail week spans two quarters, the SAS Solutions Adapter for SAP determines that the week is in the quarter in which the week ends. If needed, you can customize this behavior. The financial period information is derived from the SAP T009x tables.

This job group includes common jobs that transform data for both SAS Financial Management and SAS Human Capital Management.

Inputs and Outputs (Tables, Formats, Macro Variables)

This job group uses the following inputs:

- Descriptions must be in the language that is specified by **&SPRAS** or **&LANG**, indicated in the global parameters. SPRAS contains quotation marks. LANG does not include quotation marks.
- The value of the source_system_id column is set to the value of **&dds_source_system_id**.
- **&KTOPL** (or **&_KTOPL**) subsets T001 to select the used financial period variant.
- Inputs from SAP R/3 tables include the following:
  - T001. The table needs the column PERIV (Financial Variant) for companies that have **ktopl=&KTOPL**.
  - T009 for information about fiscal period variants.
  - T009B for deriving real dates for non-calendar year financial periods.
  - T009T for names of fiscal period variants.

This job group creates the following outputs:

- StageSAP.TIME_PERIOD.
- StageSAP.TIME_PERIOD_ASSOC.
- StageSAP.TIME_PERIOD_ASSOC_TYPE. This includes important intermediate output tables. The start and end date of each financial year must be in a table that is in the intermediate library table. The intermediate library table helps build the time dimension tables such as in **intermed.FinYear_Start_and_End_dates**. Other processes consume this function.

Output SAS formats include the following:

- **$PERIOD_DESCRIPTION**
  - R3MIRROR.T009C(where spras="&lang")
  - start = trim(left(PERIV!!poper))
  - label = trim(left(LTEXT))
  - Time Period descriptions (long text)
- **$PERIOD_NAME**
  - R3MIRROR.T009C(where spras="&lang")
  - start = trim(left(PERIV!!poper))
  - end = trim(left(PERIV!!poper))
  - label = trim(left(KTEXT))
  - Time Period Name (short text)

This job group also uses the following SAS formats:

- **$period_description**
- **$period_name**

Administration Factors

Account Hierarchies and Numbering

One issue you must understand and consider is hierarchy values and structure. The PERIOD_TYPE_CD is one of the values in the SASHELP.TIME_PERIOD column PERIOD_TYPE_CD. There is also a separate hierarchy for each used fiscal year variant (PERIV in T001 for companies that have **KTOPL=&ktopl**). PERIV is the TIME_PERIOD_ASSOC_TYPE_CD and TIME_PERIOD_ASSOC_TYPE_DESC is the

fiscal year variant name (as in T009T). As a result, all the separate hierarchies for different PERIV values are concatenated into STAGE_TIME_PERIOD_ASSOC. If there is a fixed PERIV value, then no row is in the hierarchy more than once. Finally, the root of each hierarchy has an ALL value and PARENT_TIME_ID has an ALL value pointing back to itself. The root node is valid for all time.

The time periods in the leaf layer of each hierarchy must have no gaps and no overlapping periods except for special periods. The SAS Solutions Adapter for SAP defines SAP special periods (13 to 16) to start and end on the last day of the financial year. These special periods are for end-of-year closing. Because SAP does not represent periods in a hierarchy, the SAS Solutions Adapter for SAP creates a hierarchy by adding levels. The top (root) level is ALL and is assigned to be the parent of the next level that consists of years. A level for quarters is added below the years level. Actual periods are then assigned as children of one of the quarterly nodes.

Other Transactions and Account Considerations

The LANGUAGE_CD value must be one of the predefined language values shipped in the SAS_LANGUAGE table. You specify the language value as needed, use the format **`$SAPTOSAS_LANGUAGE`** (SAP to SAS language code), which is based on SAP table T002.

In the STAGE_TIME_PERIOD table, the PERIOD_TYPE_CD column uses the values that are in the PERIOD_TYPE_CD column of the SAS_PERIOD_TYPE table. Likewise, in the STAGE_TIME_PERIOD table, the SOURCE_SYSTEM_ID column must be the value of **`&dds_source_system_id`**.

The SAS Solutions Adapter for SAP also includes special handling for financial years and periods that do not start on the first day of a month. This enables it to handle the calendar year and fiscal years according to the various situations that can occur in SAP. For example, the SAS Solutions Adapter for SAP adjusts the dated terms as needed. If a financial year is shortened, then the last quarter is shortened appropriately. If the financial year is longer, then the final quarter is lengthened.

T000200 Transform Financial Period Info from R3

This job transforms SAP R/3 financial period tables into fiscal year periods by company. It handles financial years that are calendar years and non-calendar years. Typically, you only need to use one type.

This job uses the following input:

- R3mirror.T001 (created in E00010 Extract R3 General and FI Tnnn Tables)
- R3mirror.T009 (created in E00030 Extract Financial Period T Tables)
- R3mirror.T009B (created in E00030 Extract Financial Period T Tables)
- **`&_ktopl`** (set in preprocessing step)

This job creates the following output:

- Intermed.Used_Fiscal_Year_Variants
- Intermed.Fiscal_Year_Variants_unlike_cal
- Intermed.fiscal_periods_non_calendar
- Intermed.Fiscal_Year_Variants_like_cal
- Intermed.FISCAL_YEAR_LIKE_CALENDAR_YEAR

- Intermed.Fiscal_Year_Periods_with_dates

This job uses the following user-written source:

- **t000200_fiscal_year_non_cal.sas**
- **t000200_add_real_calendar_data.sas**

T000210 Load Staging Time Period

This job loads the Staging Period type from the values that SAS supplies. It also adds SAP special periods and sets up records for SAP special periods to be appended to the SASSupplied.PERIOD_TYPE data set. The following values are included:

- Special periods, record type SP
- Half months, record type HMO

This job uses SAS_Supplied.SAS_PERIOD_TYPE as input.

This job creates StageSAP.PERIOD_TYPE as output.

The user-written source is **t000210_add_sap_special_periods.sas**.

T000220 Determine FinYear Begin and End Dates

This job creates a table that has the start and end dates for each financial year. Several other jobs use this intermediate table. For example, other jobs use this table to establish the correct date parameters for extracting cost center hierarchies for each year.

This job uses Intermed.Fiscal_Year_Periods_with_dates (created in T000200 Transform Financial Period Info from R3) as input.

This job creates Intermed.FinYear_Start_and_End_dates as output.

The format **$FISCAL_START_MTH** converts the concatenation of Financial Year (Financial_Year) and Time Period Association Type Code (TIME_PERIOD_ASSOC_TYPE_CD) to Financial Year Start (Fin_year_start_dt). This stores the format in the Library.formats catalog.

The user-written source **t000220_determine_fy_begin_end_dates.sas** uses the Intermed.Fiscal_Year_Periods_with_dates table to determine the start date and end date for each financial year that is within each combination of Time Period association type, Financial Year, and Time Period ID.

T000230 Add Time Hierarchy Layers

This job adds top node ALL, year, and quarter layers to the time dimension.

This job uses the following input:

- Intermed.FinYear_Start_and_End_dates (created in T000220 Determine FinYear Begin and End Dates)
- Intermed.Fiscal_Year_Periods_with_dates (created in T000200 Transform Financial Period Info from R3)

This job creates the following output:

- Intermed.FinYear_Quarters

- Intermed.Fiscal_Year_Periods_with_parents
- Intermed.Time_period_dimension
- Uses format $company_to_fy_variant.

The user-written source is **`t000230_add_time_hierarchy_layers.sas`**.

T000240 TIME PERIOD Dimension Tables

This job loads the main Time Period dimension tables.

This job uses Intermed.Time_period_dimension (created in T000230 Add Time Hierarchy Layers) as input.

This job creates the following output:

- StageSAP.TIME_PERIOD_ASSOC
- StageSAP.TIME_PERIOD

T000250 TIME PERIOD ASSOC TYPE

This job loads the staging structure for STAGE_TIME_PERIOD_ASSOC_TYPE.

This job uses R3mirror.T009T (created in E00030 Extract Financial Period T Tables) as input.

This job creates StageSAP.TIME_PERIOD_ASSOC_TYPE as output.

T000260 Create Controlling Area Fin Year Combos

This job creates the Intermed.CONTROL_AREA_YEARS_COMBOS table. This table represents all the combinations of all controlling areas that are wanted with the start and end dates of each financial year.

Because cost center and profit center hierarchies change over time, the SAS Solutions Adapter for SAP extracts hierarchies for each financial year. The output table of this job is required to drive the queries to SAP. Both profit center and cost center dimensions depend on this job.

This job uses the following input:

- R3mirror.TKA01 (created in E00020 Extract TXXX General and FI Tables)
- Intermed.FinYear_Start_and_End_dates (created in T000220 Determine FinYear Begin and End Dates)

This job creates Intermed.CONTROL_AREA_YEARS_COMBOS as output.

Job Group: T0003 SAS Supplied Tables

Overview

This job group contains the jobs that read the tables that SAS supplies. These jobs then load the tables into the staging structure. You can also add records here. When the follow-up jobs use these tables, the staging tables merge again with the original tables that SAS supplied. Duplicates are then thrown away.

This job group includes common jobs that transform data for both SAS Financial Management and SAS Human Capital Management.

T000300 SAS+SAP Country + Currency + Language to Staging

This job uses the following input:

- R3mirror.T005T (created in E00010 Extract R3 General and FI Tnnn Tables)
- R3mirror.TCURT (created in E00050 Extract Currency T Tables)
- R3mirror.T002 (created in E00010 Extract R3 General and FI Tnnn Tables)
- R3mirror.T002T (created in E00010 Extract R3 General and FI Tnnn Tables)
- Format $SAPTOSAS_LANGUAGE.

This job creates the following output:

- StageSAP.COUNTRY
- StageSAP.CURRENCY
- StageSAP.LANGUAGE

This job uses the following formats:

- **$SAPTOSAS_COUNTRY**
- **$COUNTRY_NAME**

T000310 SAS+SAP Exrate Types to Staging

This job loads the staging structure of CURRENCY_EXCH_RATE_TYPE and creates a format **$SAPTOSAS_CURRENCY_EXCH_TYPE**. You must review this format at each installation to ensure that the mappings are correct. The exchange rate types cannot automatically convert in a 1:1 manner. As a result, the SAS Solutions Adapter for SAP does not use the format as such. Some installations might find the format useful in sub-setting the exchange rates in SAP.

This job uses SAS_Supplied.SAS_CURRENCY_EXCH_RATE_TYPE as input.

This job creates the following output:

- StageSAP.CURRENCY_EXCH_RATE_TYPE
- Intermed.SAPTOSAS_EXRATE_TYPE
- **$SAPTOSAS_CURRENCY_EXCH_TYPE.**

T000330 SAS to Stage GL_NORMAL_BAL

This job loads the values of GL_NORMAL_BAL. These values might not change, although additional language descriptions might be necessary.

This job uses SAS_Supplied.SAS_GL_NORMAL_BAL as input.

This job creates StageSAP.GL_NORMAL_BAL as output.

T000340 SAS to Stage Source System

This job loads the staging SAS_SOURCE_SYSTEM, which adds the value set by the global macro variable **`&dds_source_system_id`** into the list of valid values.

This job uses the following input:

- SAS_SUP.SAS_SOURCE_SYSTEM
- **`&dds_source_system_id`**

This job creates StageSAP.SOURCE_SYSTEM as output.

The user-written source for this job is **`t000340_add_sap_source.sas`**.

Job Group: T0004 Cost Center Dimension

Overview

Mapping SAP cost centers onto the DDS cost center dimension is relatively easy because both models include similar concepts. However, there are some issues with the hierarchy extraction that require special attention. Because the hierarchies change over time, a separate hierarchy is extracted for each financial year. As a result, the start of the financial year is stored in the VALID_FROM_DTM column and the end of the financial year in the VALID_TO_DTM column.

The hierarchy extraction part of the SAS Data Surveyor for SAP generates code on which the cost center hierarchy extraction is based. First, the extraction obtains the start and end date of each financial year from an intermediate table that was used to build the time dimension tables (in intermed.FinYear_Start_and_End_dates). In addition, the SAS Solutions Adapter for SAP extracts the cost centers per wanted controlling area in SAP, specified by the macro variable **`&wanted_controlling_areas`**.

In addition to understanding the cost center hierarchy, you must also consider several other table factors of this job group. These factors and characteristics include the following:

- The descriptions are created in the language that is specified by **`&SPRAS`** and **`&LANG`**.
- The value of the source_system_id is set to the value of **`&dds_source_system_id`**.
- The TIME_PERIOD_ASSOC_TYPE_CD for the controlling area is COST_CENTER_ASSOC_TYPE_CD.

- The macro **%adpt_cost_center_id_map** is used to set the cost center ID. The default version creates the cost center ID as a concatenation of the controlling area (KOKRS) and cost center (KOSTL). You might need to adapt this macro based on your organizational requirements.
- This macro is also used in the following jobs:
  - T000400 Cost Center Hierarchy Over Years Extraction (called within the compiled macro **%adpt_get_hier_overyears.sas**).
  - T000410 Load Stage Cost Center + Cost Center ASSOC Tables
  - T100100 Stage GL Journal
- The SAS Solutions Adapter for SAP creates multiple separate hierarchies for controlling areas and financial years. It then joins them into a single table. Within a controlling area (COST_CENTER_ASSOC_TYPE_CD) and date range, no row is in the hierarchy more than once. In other words, no node appears twice in the same time period, although you might find duplicates if the time period is ignored.
- The value of PARENT_COST_CENTER_ID references the COST_CENTER_ID of a row in the table.
- The RESPONSIBLE_EMPLOYEE_ID column cannot populate correctly from SAP. The name from the VERAK column in the R/3 table CSKS populates the column, but it is not an employee ID and therefore cannot be used in validation.
- The R/3 transaction code OKENN can compare the extracted data with the data in SAP R/3. OKENN only shows a hierarchy for a single controlling area but it can have a different start and end date.

This job group also includes common jobs that transform data for both SAS Financial Management and SAS Human Capital Management.

T000400 Cost Center Hierarchy Over Years Extraction

This job extracts one cost center hierarchy for each financial year and controlling area combination.

This job uses the following input:

- intermed.CONTROL_AREA_YEARS_COMBOS (created in T000010 Create Controlling Area Fin Year Combos)
- R3mirror.TKA01 (created in E00020 Extract TXXX General and FI Tables)
- Calls to ABAP functions

This job creates the following output:

- Intermed.cost_center_hierarchy
- StageSAP.COST_CENTER_ASSOC_TYPE

The user-written source for this job is **t000400_cost_center_hierarchy_extract.sas**.

T000410 Load Stage Cost Center + Cost Center ASSOC Tables

This job loads the staging structure for the main cost center dimension tables. In addition, this job creates the following two error-checking tables:

- Intermed.COST_CENTER_ONLY_HIER contains cost centers that are in the SAP cost center hierarchy but not in the master table extract for the relevant time period.
- Intermed.COST_CENTER_ONLY_MASTER represents records that are in the cost center master table but are not in the SAP hierarchy. You must review the SAP data if these tables are not blank.

This job uses the following input:

- intermed.FINYEAR_START_AND_END_DATES (created by T000220 Determine FinYear Begin and End Dates)
- intermed.COST_CENTER_HIERARCHY (created by T000400 Cost Center Hierarchy Over Years Extraction)
- R3mirror.CSKS (created by E01010 Extract R3 Cost Center Master)
- R3mirror.CSKT (created by E01010 Extract R3 Cost Center Master)
- **`&lang`**
- **`&wanted_controlling_areas`**

This job creates the following output:

- intermed.COST_CENTER_ONLY_HIER
- intermed.COST_CENTER_ONLY_MASTER
- StageSAP.COST_CENTER_ASSOC
- StageSAP.COST_CENTER

The user-written source for this job is **`t000410_sep_matches.sas`**.

T000420 Cost Center Add Standard Dim Values

The SAS Solutions Adapter for SAP adds fixed row records in dimension tables for the values OPENBAL, UNASSIGNED, and ALL keys. This fix ensures that the references in GL_JRNL and GL_JRNL_DETAIL always refer to existing rows. For more information about this relation, see the description in the GL_JRNL job group.

This job uses the following transformations:

- Add Standard Dimension ASSOC Rows
- Add Standard Dimension Rows

This job uses the following input:

- StageSAP.COST_CENTER_ASSOC_TYPE
- StageSAP.COST_CENTER_ASSOC (input and output)
- StageSAP.COST_CENTER (input and output)
- **`&dds_source_system_id`**

This job creates the following output:

- StageSAP.COST_CENTER_ASSOC (input and output)
- StageSAP.COST_CENTER (input and output)

Job Group: T0005 Profit Center Dimension

Overview

Mapping SAP profit centers onto the DDS profit center dimension is relatively easy because both models include similar concepts. The profit center processes are also similar to the cost center processes. However, there are some issues with the hierarchy extraction that require special attention. Because the hierarchies change over time, a separate hierarchy is extracted for each financial year. This records the dates for which the hierarchy nodes are valid. As a result, the start of the financial year is stored in the VALID_FROM_DTM column and the end of the financial year in the VALID_TO_DTM column.

The hierarchy extraction part of the SAS Data Surveyor for SAP generates code on which the profit center hierarchy extraction is based. First, the extraction obtains the start and end date of each financial year from an intermediate table that was used to build the time dimension tables in intermed.FinYear_Start_and_End_dates. In addition, the SAS Solutions Adapter for SAP extracts the profit centers per wanted controlling area in SAP, specified by the macro variable **&wanted_controlling_areas**.

In addition to understanding the profit center hierarchy, you must also consider several other table factors for this job group. These factors include the following characteristics:

- The descriptions are created in the language that is specified by **&SPRAS** and **&LANG**.
- The value of the source_system_id is set to the value of **&dds_source_system_id**.
- The TIME_PERIOD_ASSOC_TYPE_CD for the controlling area is the PROFIT_CENTER_ASSOC_TYPE_CD.
- The macro **%adpt_profit_center_id_map** sets the profit center ID. The default version creates the profit center ID as a concatenation of the controlling area (KOKRS) and profit center (PRCTR). You might need to adapt this macro based on your organizational requirements.
- This macro is also used in the following jobs:
  - T000500 Profit Center Hierarchy Over Years Extraction (called within the compiled macro **%adpt_get_hier_overyears.sas**).
  - T000510 Load Stage Profit Center + Profit Center ASSOC Tables
  - T100100 Stage GL Journal
- The SAS Solutions Adapter for SAP creates multiple separate hierarchies for controlling areas and financial years. It then joins them into a single table. Within the controlling area PROFIT_CENTER_ASSOC_TYPE_CD and date range, no row is in the hierarchy more than once. In other words, no node appears twice in the same time period, although you might find duplicates if the time period is ignored.
- The value of PARENT_PROFIT_CENTER_ID references the PROFIT_CENTER_ID of a row in the table.
- The RESPONSIBLE_EMPLOYEE_ID column cannot populate correctly from SAP. The name from the VERAK column in the R/3 table CEPC populates the column, but it is not an employee ID and therefore cannot be used in validation.
- The R/3 transaction code K6HCN can compare the extracted data with the data in SAP R/3. K6HCN only shows a hierarchy for a single controlling area but it can have a different start and end date.

This job group includes jobs that transform data for SAS Financial Management.

T000500 Profit Center Hierarchy Over Years Extraction

This job extracts one profit center hierarchy for each financial year and controlling area combination.

This job uses the following input:

- intermed.CONTROL_AREA_YEARS_COMBOS (created in T000010 Create Controlling Area Fin Year Combos)
- R3mirror.TKA01 (created in E00020 Extract TXXX General and FI Tables)
- Calls to ABAP functions

This job creates the following output:

- Intermed.profit_center_hierarchy
- StageSAP.PROFIT_CENTER_ASSOC_TYPE

The user-written source for this job is **`t000500_profit_center_hierarchy_extract.sas`**.

T000510 Load Stage Profit Center + Profit Center ASSOC Tables

This job loads the staging structure for the main Profit Center Dimension tables. It also creates the two error-checking tables below.

- Intermed.PROFIT_CENTER_ONLY_HIER contains profit centers that are in the SAP profit center hierarchy but not in the master table extract for the relevant time period.
- Intermed.PROFIT_CENTER_ONLY_MASTER represents records that are in the profit center master table but are not in the SAP hierarchy. You must review the SAP data if these tables are not blank.

This job uses the following input:

- intermed.FINYEAR_START_AND_END_DATES (created by T000220 Determine FinYear Begin and End Dates)
- intermed.PROFIT_CENTER_HIERARCHY (created by T000500 Profit Center Hierarchy Over Years Extraction)
- R3mirror.CEPC (created by E01000 Extract R3 Profit Center Master)
- R3mirror.CEPCT (created by E01000 Extract R3 Profit Center Master)
- **`&lang`**
- **`&wanted_controlling_areas`**

This job creates the following output:

- intermed.PROFIT_CENTER_ONLY_HIER
- intermed.PROFIT_CENTER_ONLY_MASTER
- StageSAP.PROFIT_CENTER_ASSOC
- StageSAP.PROFIT_CENTER

The user-written source for this job is **`t000510_separate_matches.sas`**.

T000520 Profit Center Add Standard Dim Values

The SAS Solutions Adapter for SAP adds fixed row records in dimension tables for values OPENBAL, UNASSIGNED, ALL keys. This fix ensures that the references in the GL_JRNL and GL_JRNL_DETAIL always reference existing rows. For more information about this relationship, see the description in the GL_JRNL job group.

This job uses the following transformations:

- Add Standard Dimension ASSOC Rows
- Add Standard Dimension Rows

This job uses the following input:

- StageSAP.PROFIT_CENTER_ASSOC_TYPE
- StageSAP.PROFIT_CENTER_ASSOC (input and output)
- StageSAP.PROFIT_CENTER (input and output)
- **`&dds_source_system_id`**

This job creates the following output:

- StageSAP.PROFIT_CENTER_ASSOC (input and output)
- StageSAP.PROFIT_CENTER (input and output)

Job Group: T0006 Analysis Dimension

Overview

This job group creates an analysis dimension that is not derived from SAP. The SAS Solutions Adapter for SAP extracts actual amounts that are related to actual dimensions so that the analysis dimension is hardcoded. Different installations might require adjustments to this extraction.

T000600 Analysis Dimension Population is the only job in this group, and it does not depend on any previous jobs.

This job group includes jobs that transform data for SAS Financial Management.

T000600 Analysis Dimension Population

This job creates the three minimum structures of the analysis dimension.

This job creates the following output:

- StageSAP.ANALYSIS
- StageSAP.ANALYSIS_ASSOC
- StageSAP.ANALYSIS_ASSOC_TYPE

This job uses the following user-written source:

- **`t000600_populate_analysis_table.sas`**
- **`t000600_populate_analysis_assoc_table.sas`**
- **`t000600_populate_analysis_assoc_type_table.sas`**

Job Group: T0007 Exchange Rates

Overview

This job group takes data from the exchange rate tables in R/3 (TCURR and TCURRT).

The exchange rate types must match those that are accepted as valid simple exchange rate types in the DDS. The historic (HIS) and derived (DER) exchange rate types form a group of complex exchange rate types. All the other exchange rate types are simple exchange rate types. The figure below shows a sample of exchange rate types.

View Data: SAS_CURRENCY_EXCH_RATE_TYPE

#	EXCHANGE_RATE_TYPE_CD	VALID_FROM_DTTM	LANGUAGE_CD	
1	HIS	01JAN1970	en	Historic market rate
2	NON	01JAN1970	en	No currency conve
3	PA	01JAN1970	en	Period average rate
4	PE	01JAN1970	en	Period close rate
5	C1	01JAN1970	en	User-defined Rate
6	C2	01JAN1970	en	User-defined Rate
7	DER	01JAN1970	en	Derived rate
8	PO	01JAN1970	en	Period open rate

By default, the SAS Solutions Adapter for SAP derives the daily rates from the exchange rate of SAP type M. It then distributes the exchange rates over all days for which it is valid. Then it loops over all time periods, looks up the start and end dates, and picks all the exchange rates that were valid on those days. For example, the SAS Solutions Adapter for SAP loads the rates for the DDS exchange rate types PO (period open rate) and PE (period end rate).

The SAS Solutions Adapter for SAP loads the exchange rates into the Stage Currency Exchange Rates table. Additional SAP exchange rates might need loading for types PA (period average) and the user-defined rates C1 and C2. If necessary, you must load these rates according to the installation preferences. The SAS Solutions Adapter for SAP does not complete this task.

The SAS Solutions Adapter for SAP also creates a demo analysis dimension because the exchange rate references the analysis dimensions. A customer-specific join must join the analysis table and the exchange rates. This is a simple join that provides an analysis that can be adapted for the customer.

This job group includes jobs that transform data for SAS Financial Management.

T000700 Exchange Rate Transform

This job creates StageSAP.CURRENCY_EXCH_RATE_SRC with a constant key that is equal to SAP.

This job creates StageSAP.CURRENCY_EXCH_RATE_SRC as output.

The user-written source for this job is **`t000350_populate_currency_exch_rate_src_table.sas`**.

T000720 Exchange Rate Population

This job implements the main logic that calculates the period opening and closing exchange rates.

This job uses the following input:

- StageSAP.TIME_PERIOD
- R3mirror.TCURR

This job creates the following output:

- intermed.DAILY_EXCHANGE_RATES
- StageSAP.CURRENCY_EXCH_RATE

This job uses the following user-written source:

- **t000720_identify_tcurr_end_dates.sas**
- **t000720_derive_periods_for_exrates.sas**

Job Group: T1000 Financial Documents

Overview

This job group transforms the data from the R/3 tables into the DDS staging format. This involves both the parent GL_JRNL and detail tables GL_JRNL_DETAIL.

The financial documents are based on an initial load of the SAP financial documents such as BKPF and BSEG. This load includes all data after the first start date in addition to subsequent Change-Data-Capture (CDC) loads.

Once the data loads, the SAS Solutions Adapter for SAP ensures that all the columns that reference dimensions or foreign keys use the correct keys. For example, cost_center_id is not simply equivalent to the SAP column KOSTL. It maps from KOKRS!!KOSTL.

This job group includes jobs that transform data for SAS Financial Management.

Additional Notes

Inputs and Outputs (Tables, Formats, Macro Variables)

The T1000 Financial Documents job group requires the following input formats:

- **$COMPANY_TO_FY_VARIANT** (based on R3mirror.T001)
- **$ACCBALS** (based on R3mirror.SKA1)
- **$XRANGE** (created and used, this format identifies internal document numbers)
- **$LOCALCR** (based on R3mirror.T001)

Note: Only the controlling areas specified by the global macro variable **&wanted_controlling_areas** include documentation. Document numbers identify the internal documents that are filtered out of what is passed into the target data model. Document numbers within certain ranges are internal. When R/3 allocates

document numbers, it chooses the next number in sequence from a number of pre-allocated number ranges. The ranges are stored in the R/3 table NRIV. You can also identify internal documents in that **`NRIV.OBJECT = 'RF_BELEG'`**. A format **`$XRANGE`** identifies the numbers accordingly, so that **`put(bukrs!!gjahr!!belnr,$xrange.) ='INTERNAL'`** identifies the document as being internal.

This job group also works with all the data in the table intermed.financial_documents. This includes all detailed transactions starting from the beginning of the **`&_startfinyear`** global macro variable or **`BKPF.GJAHR >= &_startfinyear`** and **`BSEG.GJAHR>=&_startfinyear`**. After successfully reading the data from the intermed file into the DDS, you should empty the intermed.financial_documents.

Dimension Foreign Keys

Here are rules for dimension foreign keys that relate to this job group:

- Dimension column values cannot be blank.
- Each value in each dimension column must be in the corresponding dimension table. If this is not the case, the records are rejected on import into the DDS.
- In all records coming from the standard financial documents (BKPS- BSEG derivative), all the dimension columns that do not have values in SAP must get the value UNASSIGNED. This implies that there must be a row with ID UNASSIGNED in all the dimensions. UNASSIGNED should not have a parent in the _ASSOC tables.

Opening Balance Considerations

Here are some characteristics of an opening balance that you must understand and consider for this job group:

- The opening balance for the initial period is included in order to ensure correct balances. These balances derive from the R/3 table GLT0. This means that the SAS Solutions Adapter for SAP must add a single row for each opening balance account with a time_period = period 1 in the financial year **`&_startfinyear`**.
- The SAS Solutions Adapter for SAP puts in a constant code OPENBL for the cost centers, profit centers, and other dimension columns in GLT0.
- The information used for opening balance should be used only for balance sheet accounts and not for profit-and-loss accounts. The format **`$ACCBALS.`** identifies balance sheet accounts. As a result, opening balance amounts are included only for accounts where **`put(glt0.raccnt,$accbals.)='X'`**. This implies that there must be a row with ID OPENBAL in all the relevant dimensions. OPENBAL has the parent ALL in the _ASSOC tables
- The GL account dimension IDs mostly derive from the SAP account number. However, some accounts appear in the financial statement in different positions according to the balance. Therefore, a single SAP account number maps onto two accounts: one with a **`C`** and one with a **`D`** appended. Because all transactions within the same financial year must be booked into the same account, the SAS Solutions Adapter for SAP uses the totals from GLT0 to decide whether to append a **`C`** or **`D`** onto the account number for each individual transaction.

  Note: Balances might change from positive to negative or vice versa during a financial year. This means that the C or D in transaction account IDs might

change over the course of the year. In these cases, information might need to be re-imported into the DDS and follow-up data marts that use the data.

T100010 Internal Number Ranges

This job reads the number range table in SAP and creates the format **$XRANGE** to identify internal document numbers.

This job uses R3mirror.NRIV as input.

This job creates the following output:

- intermed.INTERNAL_NUMBER_RANGES
- Format **$XRANGE**

T100020 Create Opening Balances

This job summarizes the GLT0 table to find the opening balance for all balance sheet accounts. This job creates a SAS view as a subset of R3mirror.GLT0. Using the SAS format **$ACCBALS.** on the account number (RACCT) keeps the balance sheet accounts or all records that translate to X. This job fills in default values of 'OPENBAL' for all unavailable columns. It also adds time dimension value to previous SAS view and outputs data set Opening_Balances.

For each financial period, this job concatenates the previous summary data set with the SAS view and then summarizes this data to a temporary data set. This temporary data set is then appended to the Opening_Balances data set.

This job uses the following input:

- Intermed.Financial_Documents (created in E02210 Extraction of Financial Documents BKPF and BSEG)
- R3mirror.GLT0 (created in E03000 Extraction of R3 GLT0 Account Transaction Summary)
- **$ACCBALS.** (created in E01400 Extract R3 Account Master)
- **$LOCALCR.** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **$COMPANY_TO_FY_VARIANT.** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **$XRANGE.** (created in T100010 Internal Number Ranges)

This job creates Intermed.Opening_Balances as output.

The user-written source for this job is **t100020_create_opening_balances.sas**.

T100030 Make GLT0 Vertical

This job creates a new version of data set GLT0, converting each row that contains 16 period values into 16 rows with each containing 1 period value. This job uses the data set Intermed.GLT0_Vertical to summarize the following parameters:

- Value of Posting (WRBTR) by Local Currency (WAERS)
- Company Code (BUKRS)
- Account number (HKONT)

- Credit/Debit Indicator (SHKZG)
- Fiscal Period (MONAT)

The job subsets this summary on the value of macro variable **&gjahr** and nonzero values of Fiscal Period (MONAT).

This job also reads the data set R3mirror.GLT0 and writes an output record for each of the 16 occurrences of nonzero total transactions in the period in local currency (HSLnn). This process writes a temporary data set that is then loaded to Intermed.GLT0_Vertical.

This job uses the following input:

- R3mirror.GLT0 (created in E03000 Extraction of R3 GLT0 Account Transaction Summary)
- **$SHKZG.** (created in E00100 Extraction of Additional Tables for Formats)
- **$COM2CAC.** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **$LOCALCR.** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **&ktopl**
- **&gjahr**

This job creates the following output:

- Intermed.GLT0_Vertical
- Intermed.GLT0_Vertical_Summary

This job uses the following user-written source:

- **t100030_make_glt0_vertical.sas**
- **t100030_summarise_glt0_vertical.sas**

T100040 Level to Account

This job decides whether an account must have a C or D appended to it by virtue of the balance. The job uses the data set Intermed.GLT0_Vertical to summarize Value of Posting (WRBTR) by Company Code (BUKRS), Account number (HKONT), and Fiscal Year (GJAHR).

This job also joins the columns Chart of Accounts (KTOPL) and Currency Key (WAERS) from the table R3mirror.T001.

This job uses the following input:

- R3mirror.T001 (created in E00010 Extract R3 General and FI Tnnn Tables)
- Intermed.GLT0_Vertical (created in T100030 Make GLT0 Vertical)
- Intermed.One2many (created in T000110 Account Item Mappings)

This job creates Intermed.Level_to_Account as output.

The user-written source for this job is **t100040_summary_on_level.sas**.

T100100 Stage GL Journal

This job populates the GL_JRNL and GL_JRNL_DETAILS tables in the staging area structure.

This job uses the following input:

- Intermed.Opening_Balances (created in T100020 Create Opening Balances)
- Intermed.Financial_Documents (created in E02210 Extraction of Financial Documents BKPF and BSEG)
- Intermed.Level_to_Account (created in T100040 Level to Account)
- **`$COMPANY_TO_FY_VARIANT.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$COM2CAC.`** (created in E00010 Extract R3 General and FI Tnnn Tables)
- **`$CC2CA.`** (created in E00020 Extract TXXX General and FI Tables)
- **`$XRANGE.`** (created in T100010 Internal Number Ranges)
- **`&_opening_balance_period`** (set in preprocessing step)
- **`&ktopl`** (set in preprocessing step)
- **`&SAKLN`** (set in preprocessing step)

This job creates the following output:

- Intermed.Financial_Fact_Table
- StageSAP.GL_JRNL
- StageSAP.GL_JRNL_DETAILS

If partial cost center IDs are used to create the internal organization dimension, then the column mappings in the financial documents need to be adjusted to correspond in the job T100100 Stage GL Journal. This process should be done in the load step of the table STAGE_GL_JRNL_DETAILS for the columns initiating_internal_org_id and affected_internal_org_id.

Because the macro variable **`&SAKLN`** is used to shorten the account_id of leaf nodes in the account dimension, the account number must be shortened in the transaction table STAGE_GL_JRNL_DETAILS. This is done by calling the macro **`%adpt_adj_to_signif_account_len`** when load step mapping the transaction table.

T100200 Summarize Financial Docs

The transformation jobs T100200 Summarize Financial Docs and T100210 Compare Financial Docs with GLT0 Summary work together to detect discrepancies in the financial document data that is extracted from SAP.

The job T100200 Summarize Financial Docs calculates the local currency totals for each financial period based on the columns BUKRS (Company code), BUSAREA (Business Area), HKONT (Account Number), and SHKZG (Debit/Credit Indicator). The input data is the financial document detail that is extracted from SAP. This input represents a join of data from SAP tables BKPF and BSEG.

This job uses Intermed.Financial_Documents (created in E02210 Extraction of Financial Documents BKPF and BSEG) as input.

This job creates Intermed.FI_Sum as output.

The user-written source for this job is **`t100200_summ_fi_docs.sas`**.

T100210 Compare Financial Docs with GLT0

The transformation jobs T100200 Summarize Financial Docs and T100210 Compare Financial Docs with GLT0 Summary work together to detect discrepancies in the financial document data that is extracted from SAP.

The job T100210 Compare Financial Docs with GLT0 Summary compares the arithmetical summary of the financial data (calculated in the job T100200 Summarize Financial Docs) with the summary data that is stored in the SAP table GLT0. The totals should always be identical.

A discrepancy between the totals can be a warning that the data is incorrect. If any discrepancies are found, then a discrepancy report is e-mailed to the user ID **`&emailerror`** that is initialized in the preprocparm12.sas file.

A discrepancy might occur because the mechanism that extracts changes in the financial document tables might have failed to detect the appropriate changes. This can happen if the triggering date stamps were not updated when a change occurred. If this happens, re-extract the data with a WHERE clause that selects the mismatched category. Then, use the re-extracted data to replace the data in the financial documents table (Intermed.Financial_Documents).

This job uses the following input:

- Intermed.FI_Sum (created in T1000200 Summarize Financial Docs)
- Intermed.GLT0_Vertical_Summary (created in T100030 Make GLT0 Vertical)
- Macro variable **`&emailerror`**

This job creates Intermed.FI_Docs_vs_GLT0__vert_SUM_Diffs as output.

The user-written source for this job is **`t100210_get_diffs.sas`**.

Job Group: T1100 External Org

Overview

The job group T1100 External Org is a collection of SAS Data Integration Studio jobs that transform a combination of customer and supplier data from R/3 tables into the DDS staging format. Customer external organization jobs take data from the customer master data in R/3 (such as KNA1) and load it into an intermediate structure. Then the data can join with Supplier External Data to form the DDS Stage External Organization Dimension.

Customers are those individuals or entities that buy goods or services from your organization. These customers might be general consumers in a retail store, or they might be other vendors and organizations about which your business maintains detailed records in SAP R/3.

Because there are several types of customers and complexities that identify them, SAP can maintain basic or complex customer records that reflect the specific customer and relationship with your organization. Additionally in SAP, you might also have created customer records for internal customers such as the departments or committees that use

company goods or services within your organization. For example, each site registers as a customer.

In SAP, a customer hierarchy provides a structure of customer nodes. These nodes define specific characteristics or parts of the customer that are tracked. For example, one customer node might be a specific customer's distribution center. In the customer hierarchy, the nodes are in a top-down structure in which the top-most customer node corresponds to the highest level of the hierarchy, and so forth.

In addition to traditional customers, vendors might also require representation in the customer and supplier hierarchies. Vendors are typically businesses or organizations that provide a service or product to your organization. You must consider your unique business environment to determine which, if any, businesses serve your organization as a vendor. You can also choose to register a vendor as a supplier if necessary.

In SAP, a supplier hierarchy provides a structure of supplier nodes. These nodes define specific characteristics or parts of the supplier that are tracked. For example, one supplier node might be a specific supplier's distribution center. In this hierarchy, the nodes are in a top-down structure in which the top-most node corresponds to the highest level of the hierarchy, and so forth.

This job group includes jobs that transform data for SAS Financial Management.

T110010 Customer External Org

This job prepares the customer part of the external organization dimension.

This job uses the following input:

- R3mirror.KNA1 (created in E01100 Extract R3 Customer Master)
- R3mirror.KNVH (created in E04070 Customer Hierarchy)
- R3mirror.THITT (created in E04070 Customer Hierarchy)

This job creates the following output:

- Intermed.Customer_External_Org
- Intermed.Customer_External_Org_Addr
- Intermed.Customer_External_Org_Assoc
- Intermed.Customer_External_Org_Assoc_Type
- Intermed.Customer_Org_Type

This job uses the following user-written source:

- **`t110010_add_order_no_to_kna1.sas`** adds a unique number for each Customer Number 1 (KUNNR) as Order Number (ORDER_NO). This is an incremented count for each new customer number that is required because the KUNNR field contains character values.
- **`t110010_load_customer_org_type.sas`**.

T110020 Supplier External Org

This job prepares the supplier part of the external organization dimension.

This job uses the following input:

- R3mirror.LFA1 (created in E01200 Extract R3 Vendor Master)
- R3mirror.LFMH (created in E04060 Supplier Hierarchy)
- R3mirror.TLHITT (created in E04060 Supplier Hierarchy)

This job creates the following output:

- Intermed.Supplier_External_Org
- Intermed.Supplier_External_Org_Addr
- Intermed.Supplier_External_Org_Assoc
- Intermed.Supplier_External_Org_Assoc_Type
- Intermed.Supplier_Org_Type

This job uses the following user-written source:

- **`t110010_add_order_no_to_lfa1.sas`** adds a unique number for each Account Number of Vendor or Creditor (LIFNR) as Order Number (ORDER_NO). This is an incremented count for each new supplier number that is required because the LIFNR field contains character values.
- **`t110010_load_supplier_org_type.sas`**.

T110030 Load EXTERNAL_ORG Group

This job populates the external_org dimension tables from the customer and supplier intermediate tables.

This job uses the following input:

- intermed.CUSTOMER_EXTERNAL_ORG
- intermed.SUPPLIER_EXTERNAL_ORG
- intermed.CUSTOMER_EXTERNAL_ORG_ASSOC
- intermed.SUPPLIER_EXTERNAL_ORG_ASSOC
- intermed.CUSTOMER_EXTERNAL_ORG_ASSOC_TYPE
- intermed.SUPPLIER_EXTERNAL_ORG_ASSOC_TYPE
- intermed.CUSTOMER_EXTERNAL_ORG_ADDR
- intermed.SUPPLIER_EXTERNAL_ORG_ADDR

This job creates the following output:

- StageSAP.EXTERNAL_ORG
- StageSAP.EXTERNAL_ORG_ASSOC
- StageSAP.EXTERNAL_ORG_ASSOC_TYPE
- StageSAP.EXTERNAL_ORG_ADDRESS

This job uses the following user-written source:

- **`t110030_add_c_and_s_nodes.sas`** adds a **`C`** to customer IDs and an **`S`** to supplier IDs so that the IDs are unique.
- **`t110030_add_c_and_s_assoc_nodes.sas`** adds a **`C`** to customer IDs and an **`S`** to supplier IDs so that the IDs are unique.

T110040 External Org Add Standard Dimension Values

This job adds rows for OPENBAL, UNASSIGNED, and ALL to the dimension.

This job uses the following input:

- StageSAP.EXTERNAL_ORG (input and output)
- StageSAP.EXTERNAL_ORG_ASSOC (input and output)

This job creates the following output:

- StageSAP.EXTERNAL_ORG (input and output)
- StageSAP.EXTERNAL_ORG_ASSOC (input and output)

Job Group: T1200 FM Internal Org

Overview

A combination of logical SAP data sources populates the Internal Org job group for SAS Financial Management.

The T1200 FM Internal Org job group is a collection of SAS Data Integration Studio jobs that transform data from R/3 tables into the DDS staging format. These jobs create a hierarchy by assigning the top level to be an operating concern, the second level to be controlling areas, and the other layers to be either a) a third layer for company code and fourth layer for business area, or b) the cost center hierarchy.

The internal organization dimension includes the following hierarchical layers:

operating concern (first level)

> is determined by the global macro variable **`&operating_concern`**. The SAS Solutions Adapter for SAP works with a single operating concern

controlling areas (second level)

> are assigned as parents of their respective companies according to the R/3 table tka02.

company code (third level)

> is the simplest identifier of a balancing or legal accounting entity. For example, if a large corporation or firm consists of several individual companies, a unique company code identifies each company within the larger business structure.

business areas (fourth level)

> are used to facilitate reporting across company codes, covering a company's main areas of operation. These areas might represent divisions, branches, product lines, and so forth. Also, not all companies have the same business areas. If your company has only one business area, you might adapt the logic to exclude the business area layer.

cost center hierarchies (additional layers if applicable)

> are included in the internal organization dimension if applicable. If an internal organization dimension includes cost center hierarchies, then the parents of the top cost center layers are controlling area layers.

You must create the cost center dimension before copying the information into the internal organization dimension tables.

Note: In most cases, only profit and loss (P&L) transactions that are obtained from BKPF/BSEG tables in SAP reference cost centers. Other transactions are associated with the business area.

Note: The Cost Center dimension does not include all the information that is required for the internal organization dimension. Review the source code **t120050_load_internal_org_from_cc.sas** for details.

Stage INTERNAL ORG is a simple concatenation of the various contributing tables and a set of standard values. The hierarchy for the ASSOC table is created by linking the operating concern (first layer) to the controlling area (second layer) by using **TKA01.KOKRS** and **TKA01.ERKRS**. Likewise, the controlling area (second layer) links to the company (third layer) by using **TKA02.BUKRS** and **TKA02.KOKRS**.

This job group includes jobs that transform data for SAS Financial Management.

T120010 Operating Concern Internal Org

This job creates the top layer of the hierarchy. This is the operating concern layer.

This job creates the format **$OP_CONCERN_CURRENCY** that the SAS Human Capital Management job T290080 Load HR INTERNAL ORG uses. The prefix **OPC** is added to the operating concern to form the internal_org_id. For example, an operating concern of **1000** becomes **OPC1000**. This ensures a unique value because the operating concern value can also be a valid value from one of the other layers.

This job uses the following input:

- R3mirror.TKEB (created in E00090 Extract Operating Concern T Tables)
- R3mirror.TKEBT (created in E00090 Extract Operating Concern T Tables)
- **&operating_concern_country_cd**
- **&dds_source_system_id**
- **$SAPTOSAS_COUNTRY.**

This job creates Intermed.Operating_Concern_Internal_Org as output.

This job uses the format **$op_concern_currency**.

T120020 Controlling Area Internal Org

This job creates the second layer of the hierarchy. This is the controlling area layer. The prefix **CAR** is added to the controlling area to form the internal_org_id to ensure a unique value. For example, a controlling area of **1000** becomes **CAR1000**.

This job uses the following input:

- R3mirror.TKA01 (created in E00020 Extract TXXX General and FI Tables)
- R3mirror.TKA00 (created in E00020 Extract TXXX General and FI Tables)
- **&operating_concern_country_cd**
- **&dds_source_system_id**
- **$SAPTOSAS_COUNTRY.**

This job creates Intermed.Controlling_Area_Internal_Org as output.

T120030 Company Internal Org

This job creates a third layer of the hierarchy. This is the company layer.

This job uses the following input:

- R3mirror.T001 (created in E00010 Extract R3 General and FI Tnnn Tables)
- R3mirror.T880 (created in E00010 Extract R3 General and FI Tnnn Tables)
- **&dds_source_system_id**
- **$SAPTOSAS_COUNTRY.**

This job creates Intermed.Company_Internal_Org as output.

T120040 Business Area Internal Org

This job creates a fourth layer of the internal organization hierarchy for the business area information.

Because your source data from SAP might not have a description for the business area, the initialization job I0050 Set Blank Business Area Name enables you to enter a description for a blank business area. You can also modify the setting for the Internal_Org_Id in the transaction data so that if the business area is blank then Internal_Org_Id defaults to the company code. The prefix **BA** is added to the business area to form the Internal_Org_Id. For example, a business area of **1000** becomes **BA1000**. This ensures a unique value because the business value can also be a valid value from one of the other layers.

This job uses the following input:

- R3mirror.TGSBT (created in E00020 Extract TXXX General and FI Tables)
- Intermed.TGSBT_Blank (created in I0050 Set Blank Business Area Name)
- Intermed.Company_Internal_Org (created in T120030 Company Internal Org)
- **&dds_source_system_id**
- **$SAPTOSAS_COUNTRY**

This job creates the following output:

- Intermed.Business_Area_Internal_Org
- Intermed.Busarea_Internal_Org_Assoc

T120050 INTERNAL_ORG Layer from Cost Center

This job creates internal organization records that can be used as alternative layers of the internal organization hierarchy if the job group T0004 Cost Center Dimension has already created the cost center dimension.

All cost centers in SAP are associated with a controlling area. As a result, the cost center hierarchy slots in the internal organization hierarchy are below the controlling area layer. This arrangement offers an alternative to the company and business area layers.

Note: This job has a site-dependent user exit (**t120050_load_internal_org_from_cc.sas**). Review this code to ensure proper administration of the job.

This job uses the following input:

- StageSAP.STAGE_COST_CENTER (created in job group T0004 Cost Center Dimension)
- **`&operating_concern_country_cd`**

This job creates Intermed.INTERNAL_ORG_from_Cost_Center as output.

T120060 INTERNAL_ORG_ASSOC Layer from Cost Center

This job creates records that can be used as alternative layers of the internal organization hierarchy if the job group T0004 Cost Center Dimension has already extracted the cost center hierarchy.

All cost centers in SAP are associated with a controlling area. As a result, the cost center hierarchy slots in the internal organization hierarchy are below the controlling area layer. This arrangement offers an alternative to the company and business area layers.

This job creates the Intermed.INTERNAL_ORG_ASSOC_from_Cost_C table from the corresponding cost center tables.

This job uses the following input:

- StageSAP.STAGE_COST_CENTER_ASSOC
- Intermed.Cost_Center_Hierarchy

This job creates Intermed.INTERNAL_ORG_ASSOC_from_Cost_C as output.

T120100 Load INTERNAL_ORG

This job appends the layers into a single table and loads STAGESAP.INTERNAL_ORG.

This job uses the following input:

- intermed.Operating_Concern_Internal_Org
- intermed.Company_Internal_Org
- intermed.Controlling_Area_Internal_Org
- intermed.Business_Area_Internal_Org
- intermed.Internal_Org_from_Cost_Center

This job creates StageSAP.INTERNAL_ORG as output.

T120110 Load FM INTERNAL_ORG_ASSOC

This job loads the staging INTERNAL_ORG_ASSOC table.

The logic for this job is in the **`t120110_load_staging_internal_org_assoc.sas`** program. This program creates the links between the various layers of the table by setting the appropriate values in the column PARENT_INTERNAL_ORG_ID.

This job uses the following input:

- R3mirror.TKEB
- R3mirror.TKA01

- R3mirror.TKA02
- Intermed.Busarea_Internal_Org_Assoc
- Intermed.Internal_Org_Assoc_from_Cost_C

This job creates StageSAP.INTERNAL_ORG_ASSOC as output

T120120 Internal Org Add Standard Dimension Values

This job adds standard rows such as OPENBAL, UNASSIGNED, ALL, and EXT to the internal organization dimension tables.

This job uses the following input:

- StageSAP.INTERNAL_ORG_ASSOC_TYPE
- StageSAP.INTERNAL_ORG (input and output)
- StageSAP.INTERNAL_ORG_ASSOC (input and output)

This job creates the following output:

- StageSAP.INTERNAL_ORG (input and output)
- StageSAP.INTERNAL_ORG_ASSOC (input and output)

Job Group: T2000 HR Common Reference Tables

Overview

The HR Common Reference Tables job group is a collection of SAS Data Integration Studio jobs that transform data from R/3 tables into the DDS staging format. These jobs create tables that enable the Pay Level, Pay Level Structure, and Exempt Status DDS Staging tables to populate.

This job group includes jobs that transform data for SAS Human Capital Management.

T200020A Get Base Wage Types from PA0008

This job provides the preferred method for populating the table Intermed.SAP Base Wage types. The intermediate table that is created is then used in the job group T2400 HR Employee Compensation History.

Not all payments made to employees are considered base salary. The payment information that is extracted from SAP can include a very large amount of data that is irrelevant depending on the extraction parameters. For example, the data might include values such as calculated values, cumulative annual values, and deductions. These values should not be included in the base salary amounts.

Using this job assumes that the base salary wage types that are recorded in infotype 8 are the desired values.

Note: The job T200020B also provides an alternative method of populating the table.

This job uses the following input:

- R3mirror.PA0008
- Intermed.Employee Country Association

This job creates Intermed.SAP_Base_Wage types as output.

T200020B Get Base Wage Types from Wage Type Groups

This job provides an alternative method to populate SAP Base Wage types. The intermediate table that is created is then used in the job group T2400 HR Employee Compensation History.

You can use this alternative job if the T52D7 table is maintained in SAP.

Note: This job is an alternative to job T200020A.

This job uses R3mirror.T52D7 as input.

This job creates Intermed.SAP_Base_Wage types as output.

T200030 SAP to SAS Wagetypes

This job creates the Staging Compensation Type table and a format that maps SAP wage types onto SAS wage types.

This job uses the SAS code that is in the t200030_sap2sas_wage types.sas source. It groups all wage types that have identical text descriptions and removes calculated wage types such as those starting with `/`.

This job uses the following input:

- R3mirror.T512T
- Intermed.SAP_Base_Wage types

This job creates Intermed.STAGE_COMPENSATION_TYPE as output.

This job uses the format **`$saptosas_wage type`**.

T200040 Create Inter T510 Label Texts

This job obtains the label texts that are related to the T510 table. These texts and descriptions are added later to the Staging Pay Level and Pay Level Structure data.

This job uses R3mirror.DD04T as input.

This job creates Intermed.T510_label_TextJoin as output.

T200045 Employee Pay Info

This job creates unique, pay level structures that are used in Infotype 8 (PA0008) and a version that contains descriptions derived from the column labels.

In addition, this job derives the payment information that is needed to populate employee actions.

This job uses the following input:

- R3 Mirror PA0008 - HR Infotype 0008 (Basic Pay)
- Intermed.Employee_Country_Association
- Intermed.T510_Label_Text_Join

This job creates the following output:

- Intermed.Employee_action_pay_info_PA0008
- Intermed.PAY_LEVEL_STRUCTURE_fromPA008
- Intermed.PAY_LEVEL_STRUCTURE_PA0008_hold

T200050 Load Stage Pay Level

This job creates the table Load the Stage Pay Level. In this case, the Pay Level is assumed to be the same as the Pay Level Structure, as populated in the job T200060 Load Stage Pay Level Structure. For example, Pay Level and Pay Level Structure are used synonymously. Therefore, very similar jobs are used. Some sites might choose a different behavior. In these cases, one of the two jobs and the references to the relevant columns must be modified.

Pay Level and Pay Level Structure are especially important where salary ranges are associated with the codes. Not all companies maintain salary range information, especially outside of North America. Therefore, it is possible that not all sites maintain this information in table T510N in SAP. In these cases, it might be necessary to populate the output table by using only information from the table Inter PAY_LEVEL_STRUCTUREs that is used in Infotype 8.

This job uses the following input:

- R3mirror.T510N
- Intermed.T510_Label_Text_Join
- R3Mirror.T510F
- Intermed.PAY_LEVEL_STRUCTURE_from_P0008
- **`&lang`**

This job creates StageSAP.PAY_LEVEL as output.

T200060 Load Stage Pay Level Structure

This job loads Stage Pay Level Structure. In this case, Pay Level Structure is assumed to be the same as Pay Level, as populated in the job T200050 Load Stage Pay Level.

Refer to the description job T200050 Load Stage Pay Level for more information.

This job uses the following input:

- R3mirror. T510N
- Intermed.T510 Label Text Join
- Intermed.PAY_LEVEL_STRUCTURE_from_P0008

This job creates StageSAP.PAY_LEVEL_STRUCTURE as output.

T200070 Load Stage Exempt Status

This job creates and loads the table Stage_Exempt_Status. The user-written code for this job creates a two-row table to record the values of Non-Exempt and Exempt.

This job uses user-written code t200070_populate_exempt_status.sas as input.

This job creates StageSAP.EXEMPT_STATUS as output.

T200080 Create SAS Format for Personnel Number Clean

This job creates a table that can be used to clean personnel numbers. It uses the HR Clean Personnel Numbers transformation.

Each individual has a single and unique Social Security Number or national identity number as shown in R/3 table PA0002.PERID. If multiple personnel numbers have been allocated to the same person in SAP, the related information might require consolidation. This format enables you to map the multiple numbers to a single number.

Note: Although this format is created, the SAS Solutions Adapter for SAP does not use it. However, if your environment requires such a consolidation, it is provided as a tool for that purpose.

This job uses the following input:

- R3mirror.PA0001
- R3Mirror.PA0002

This job creates Intermed.PA0002 Anomalies as output.

This job uses the format **`$R3_to_HRID`**.

Job Group: T2100 HR Employee General Information

Overview

This group of jobs selects processes and joins tables that are related to Employee General Information. The objective of these jobs is to populate the StageSAP.EMPLOYEE table. Many jobs carry out discrete stages of this process in order to enable customization. It would be difficult to troubleshoot a single job that carried out all the processing singularly.

The SAS Solutions Adapter for SAP classifies general information about employees into date-dependent and date-independent categories. An example of date-independent information is a person's birthdate. For example, it is possible that birthdate data in the SAP system was changed because it was initially entered incorrectly. In this case, the SAS Solutions Adapter for SAP takes the latest value and assumes that it is the best value.

Other information can also change logically where previous values are of no real interest. In these cases, the current version of the information is preferable and historical information is not collected. One example of this type of data is an employee's educational level. If you

decide that certain historical information is important, you need to adjust the jobs accordingly.

This job group includes jobs that transform data for SAS Human Capital Management.

Creating Consolidated and Subset Tables

The first set of transformations in this group create consolidated and subset tables in the intermediate (intermed) library. There, the names of the tables are the same as the names of the R3mirror tables unless stated otherwise.

The steps below describe the general process that occurs when creating the consolidated and subset tables in the intermediate library.

1. The Period Consolidation transformation is done on PA0008, on key PERNR, and the date range that is specified by BEGDA and ENDDA.
   - The columns for which differences are important: BSGRD.
   - The columns for which the latest value needs to be kept: <none>.
2. The Period Consolidation transformation is done on PA0016, on key PERNR, and the date range that is specified by BEGDA and ENDDA. This is the default behavior. The transformation uses WHERE cctype in (&Temporary_Contract_types).
   - The columns for which differences are important: CTTYP.
   - The columns for which the latest value needs to be kept: <none>.
3. The latest record per employee is used with PA0022, on key PERNR, and the date range that is specified by BEGDA and ENDDA.
   - The columns which are important: SLABS.
4. The Period Consolidation transformation is done on PA0057, on key PERNR, and the date range that is specified by BEGDA and ENDDA. This transformation uses a WHERE clause that selects only EMFSL in **`&labour_union_codes`**.
   - The columns for which differences are important: EMFSL.
   - The columns for which the latest value needs to be kept: <none>.
5. The Period Consolidation transformation is done on PA0077, on key PERNR, and the date range that is specified by BEGDA and ENDDA. This transformation assumes that disability status (DISAB) can change, but ethnicity cannot.
   - The columns for which differences are important: DISAB.
   - The columns for which the latest value needs to be kept: RACKY.

After the first set of preprocessing and consolidation is complete, all the date-independent tables are merged together.

The Date Join transformation is also used in this group. This transformation performs pair-wise joins on date-range tables. For these joins, it is more efficient to start with the tables that might not have records for all employees, such as consolidated PA0057 and consolidated PA0077, because each pairing might increase the resulting data volume.

The other consolidated date range tables that are joined are PA0002, latest PA0006, PA0008, and PA0016. Creating an interim table based on a date join of PA0002 and PA0006

(addresses), for example PA0002_PA0006, at an earlier stage is useful. Pair-wise joining can therefore follow the same order shown in the examples below.

- PA0057 and PA0077 join to form PA0057_PA0077
- PA0057_PA0077 and PA0002_PA0006 join to form PA0057_PA0077_PA0002_PA0006

The resulting tables that are date_independent are left joined to the date-dependent results.

T210010 Hire Date

This job creates internal hire date data. According to SAP help file descriptions, the hire date should be taken so that PA0000 is the preferential source of hire date. However, hiring actions are often entered only for employees that are hired after SAP HR was introduced into the company. For this reason, the SAS Solutions Adapter for SAP uses the lower of the two dates, PA0000.BEGDA and PA0001.BEGDA per employee.

This job uses the following input:

- R3mirror.PA0000
- R3Mirror.PA0001

This job creates Intermed.Hire_Date as output.

T210020 Service Date

This job creates internal service date data. Service_dt is calculated so that KONDT takes preference over HIREDATE. For example, HIREDATE is used only if KONDT is missing.

This job uses the following input:

- R3mirror.PA0016
- Intermed.Hire_Date

This job creates Intermed.Service_Date as output.

T210025 Populate_Stage_Position_Permanence

This job populates the STAGE_POSITION_PERMANENCE table from data in R3 Mirror T547S - Contract Type Texts.

This job uses R3mirror.T547S as input.

This job creates StageSAP.POSITION_PERMANENCE as output.

T210030 Employee Info from Infotype 2 PA0002

This job selects Employee info. A period consolidation is carried out on FAMST to identify marital status. The latest values of the following basic data are retained:

- forename
- middle

- name
- surname
- nationality
- gesch
- language
- perid
- birthdate

This job uses the following input:

- R3mirror.PA0002
- **$SAPTOSAS_country.**
- **$SAPTOSAS_language.**

This job creates Intermed.PA0002_employee_info as output.

T210040 Employee Address from PA0006

This job selects the current employee address. An address transformation enables specific SAP address fields to map to the Address Lines 1 through 4 under the SAS data model. Three SAP columns are selected by default: LOCAT, STRAS, and ADR04. A forth column can be added or the selection can be altered by changing the column options of the transformation and therefore enabling compliance with local address standards.

This job uses R3mirror.PA0006 as input.

This job creates Intermed.PA0006 Addresses as output.

T210050 Employee Contract Info from PA0016

This job consolidates Contract data by period.

This job uses R3mirror.PA0016 as input.

This job creates Intermed.PA0016_employee_contract _type as output.

T210070 Populate Staging Education Level from T519T

Certificate and Certificate Text fields populate the Education Level code and description fields.

This job uses the following input:

- R3mirror.T519T
- **$SAPTOSAS_Language.**

This job creates StageSAP.EDUCATION_LEVEL as output.

T210075 Load Stage Gender

This job populates the reference table STAGE GENDER with gender codes and text.

This job uses the following input:

- R3mirror.DD07T
- **$SAPTOSAS_Language.**

This job creates StageSAP.GENDER as output.

T210080 Load Stage Marital Status

This job populates the reference table STAGE MARITAL STATUS with codes and text.

This job uses the following input:

- R3mirror.T502T
- **$SAPTOSAS_Language.**

This job creates StageSAP.MARITAL_STATUS as output.

T210090 State Region

Concatenating the country code and the region code populates the reference table.

This job uses the following input:

- R3mirror.T005U
- **$SAPTOSAS_Language.**

This job creates StageSAP.STATE_REGION as output.

T210100 Load Stage Ethnicity

SAP Ethnicity data includes the classification of disabled. The disability flag for the StageSAP table comes from the PA0077.DISAB field. The PA0004 table for employee records provides details of the disability. This data can be a site-specific requirement as needed. For more information, refer to the descriptions of the disability types in the T523T table.

This job uses the following input:

- R3mirror.T505S
- **$SAPTOSAS_Language.**

This job creates Stage.STAGE_SAP_ETHNICITY as output.

T210110 Load Stage Employee Union

Employee Union affiliation can be derived when a company pays union dues for employees. The SAP HR infotype 57 (PA0057) captures this information. However, not all entries in PA0057 relate to unions. In addition, the related information that is needed to build the Employee Union reference in SAP tables T521B and T521C also contains information that does not relate to infotype 57. As a result, this job selects only the relevant information from the infotype 57 data.

This job uses the following input:

- R3mirror.T521B
- R3mirror.T521C
- **`&lang`**

This job creates StageSAP.EMPLOYEE_UNION as output.

T210115 Employee Email

The adapter in this job retrieves the employee e-mail information from SAP HR infotype 105 (R3mirror.PA0105). If this infotype is not maintained, then you might need to consider an alternative source in SAP infotype 1032 (r3mirror.HRP1032).

This job uses R3mirror.PA0105 as input.

This job creates Intermed.Employee email as output.

T210120 Infotype 2 and 6 Date Join

This job uses the date-join transformation to join data from infotypes 2 (r3mirror.PA0002) and 6 (and r3mirror.PA0006) on the personnel number.

This job uses the following input:

- Intermed.PA0006
- Intermed.PA0002

This job creates Intermed.PA0002_and_addresses as output.

T210130 Consolidate Full Time Equivalent

This job creates the full-time-equivalent column from the BSGRD column in infotype 8. A period consolidation is also performed because the value of BSGRD usually remains constant for most employees.

This job uses R3mirror.PA0008 as input.

This job creates Intermed.Full_Time_Equivalent as output.

T210140 Include Full Time Equivalent Data

This job date-joins the results of previous jobs such as infotype 2, infotype 6, and full-time-equivalent information.

This job uses the following input:

- Intermed.Full_Time_Equivalent
- R3Mirror.PA0002_and_addresses

This job creates Intermed.employee_and_fte as output.

T210150 Merge HireDate ServiceDate + Education

This job does a SAS DATA step merge on the results of previous jobs to merge date-independent information.

This job uses the following input:

- Intermed. Hire date
- Intermed. Service date
- R3mirror.PA0002

This job creates Intermed.join_date_independent_tabl as output.

T210160 DateJoin1 - Infotype 77 and 57 - PA0077 and PA0057

This job is the first of three jobs that date-join tables that have relevant date-dependent employee general information. These jobs could be consolidated into fewer jobs, but by having more intermediate results, it is easier to apply local modifications because they can be added as intermediate jobs.

This job uses the following input:

- R3mirror.PA0077
- R3mirror.PA0057

This job creates Intermed.Dj_join_77and57 as output.

T210170 DateJoin2 - 77_57 with PA0016_employee_contract_

This job is the second of three jobs that date-join tables that have relevant date-dependent employee general information. These jobs could be consolidated into fewer jobs, but by having more intermediate results, it is easier to apply local modifications because they can be added as intermediate jobs.

This job uses the following input:

- Intermed.Dj_join_77and57
- Intermed.PA0016_employee_contract_type

This job creates Intermed.Dj_join2_77_57and16 as output.

T210180 DateJoin3 - 77_57_16 with Date Dependant EE & FTE

This job is the third of three jobs that date-join tables that have relevant date-dependent employee general information. These jobs could be consolidated into fewer jobs, but by having more intermediate results, it is easier to apply local modifications because they can be added as intermediate jobs.

This job uses the following input:

- Intermed.Dj_join2_77_57and16
- Intermed.Employee_and_FTE

This job creates Intermed.DJ_77_57_16_EE_FTE as output.

T210190 Join Date Independent & Date Dependent

This job takes all the date-dependent and independent tables in order to populate the EMPLOYEE tables.

This job uses the following input:

- Intermed.DJ_77_57_16_EE_FTE
- Intermed.join Join_date_independent tabl
- **&dds_source_system_id**
- **$saptosas_country.**

This job creates StageSAP.EMPLOYEE as output.

Job Group: T2200 HR Employee Absences

Overview

Absences are periods where an employee is absent from work for vacation, illness, special leave, and so forth. This information is collected and categorized into absence types and descriptions in order to analyze the data.

The SAS Solutions Adapter for SAP extracts absences that are recorded. In SAP, absence information can be stored in infotype 2001 or in HR cluster tables such as the time recording tables SALDO and ZES. These absence types in infotype 2001 can differ in various personnel sub-areas. As a result, the SAS Solutions Adapter for SAP uses organizational assignments that recognize each unique absence type.

This job group includes jobs that transform data for SAS Human Capital Management.

T220010 Create DDS Staging Table Employee Absence Type

This job creates the main reference table for absences. This table provides descriptions of absence codes. The absence code is created by concatenating the SAP absence code column (AWART) and the personnel sub-area for the absences column (MOABW) to guarantee uniqueness. This procedure is important because the meaning of some AWART values is different in the various sub-areas.

Some companies might use only a single sub-area and as a result might consider simplifying the absence code to be derived directly from the AWART column only. This can reduce the complexity of some follow-up steps because the personnel sub-area is generally not stored in all tables that have the AWART transaction information. The personnel sub-area also must be derived from the sub-area that is associated with the employee. This is a complex arrangement because employees can move between sub-areas over time.

This job uses R3 Mirror T554T - Absence and Attendance Texts as input.

This job creates StageSAP.ABSENCE_TYPE as output.

T220020 Create DDS Table STAGE_EMPLOYEE_ABSENCE

This job creates the employee absence fact table from the 2001 infotype (PA2001). The personnel sub-area for absences (MOABW) is joined from infotype 1 (PA0001).

This job uses the following input:

- R3mirror.PA2001
- R3mirror.PA0001
- **`$personnel_subarea_absence.`**

This job creates StageSAP.EMPLOYEE_ABSENCE as output.

T220030 Load Staging Time_unit_of_measure

This job populates the time units for recording employee time. It also processes employee base compensation. The values are hardcoded in t220030_create_time_unit_of_measure.sas with the values DAY for days and H for hours.

This job uses t220030_create_time_unit_of_measure.sas as input.

This job creates StageSAP.STAGE_TIME_UNIT_OF_MEASURE as output.

Job Group: T2300 HR Employee Base Compensation

Overview

This group of jobs processes employee base compensation. It creates intermediate tables that are then used in the job group T3500 HR Employee Actions.

Employee base compensation is not always handled by SAP. In these circumstances, the output tables of these jobs should be populated from the alternative data source. Also, new SAS Data Integration Studio process flows should be created.

Note: There are distinguishing factors between base compensation current pay and compensation history. Both are extracted from SAP and handled separately.

Current pay is the theoretical base salary that an employee should receive, or should have received, at any given point in time. The employee's theoretical salary has a value that might be different from the amount that was actually paid. In general, the amounts from the past should be identical because the SAS Solutions Adapter for SAP always extracts the value that is considered to be the most recent or corrected. For the current month or pay period, the employee's base compensation has a value that represents an amount that is paid in the next payroll run. If an employee is new or will receive payment different from what was received in the past, it is not possible to use the compensation history to deduce the amount. The SAS Solutions Adapter for SAP uses

the current pay information to populate the payment information when an employee action occurs.

Compensation history is used to populate the DDS Staging Compensation table.

This job group includes jobs that transform data for SAS Human Capital Management.

T230000 Update Current Pay

This job uses the user-written transformation HR Current Pay Update. Refer to this transformation's description for more details about this job.

This job uses the following input:

- Intermed.Inter_Current_Pay_Extract_Log
- Intermed.Inter_Current_Pay_Results

This job creates the following output:

- R3Mirror.Current_Pay_Results
- Admin.Current_Pay_Extraction_History

T230010 Period Consolidation of Base Compensation

The input of this job contains multiple records for each employee. These records represent payment information for multiple occasions for each employee. This job consolidates the information so that date ranges are created for periods in which employees' payments remain unchanged.

This job uses R3mirror.Current_Pay_Results as input.

This job creates Intermed.Base_Compensation as output.

Job Group: T2400 HR Employee Compensation History

Overview

There are alternative methods for obtaining the input data for this transformation group. The job E08040A HR Extract Historical Compensation is the more common method, but E08040B HR Extract Historical Compensation can be used instead.

Each installation needs to populate a mapping table INTERMED.SAP_TO_SAS_WAGETYPES. This table maps SAP R/3 wage types onto SAS compensation types and identifies BASE salaries. In this context, the SAS compensation types are defined as the categories that need to be in the SAS reporting.

This job group includes jobs that transform data for SAS Human Capital Management.

T240020 HR Consolidate RT Delta

This job consolidates the historical payments that were extracted in one of the alternative jobs E08040A HR Extract Historical Compensation or E08040B HR Extract Historical Compensation. The SQL join maps SAP to SAS wage types and consolidates amounts. Rows that have a consolidated amount (BETRG) = 0, or not having a payment date, or not having a target wage type are discarded.

This job uses the following input:

- Intermed.Employee_Country_Associations
- Intermed.RT

This job creates StageSAP.COMPENSATION as output.

Job Group: T2500 HR Applicants General Information

Overview

Most data that is related to applicants is held in PBxxxx tables that are structurally identical to the corresponding PAxxxx tables used for employee general data. Not every PAxxxx table has a PBxxxx equivalent. For example, the PA0000 table relates only to employees.

The jobs within this group follow, where appropriate, the numbering scheme in the Employee General job group.

This job group includes jobs that transform data for SAS Human Capital Management.

T250010 PB0001 and PB0016 Join

This job uses the following input:

- R3mirror. PB0001
- R3Mirror.PB0016

This job creates Intermed.PB0001_and_ PB0016_join as output.

T250030 Applicant Info from PB0002

This job uses R3mirror.PB0002 as input.

This job creates Intermed.PB0002 as output.

T250040 Applicant Addresses from PB0006

This job uses R3mirror.PB0006 as input.

This job creates Intermed.Applicant_Addresses as output.

T250050 Info from PB0016

This job uses the R3mirror.PB0016 as input.

This job creates Inter.PB0016 as output.

T250080 Populate APPLICATION_STATUS

This job uses R3mirror.T751B as input.

This job creates StageSAP.APPLICATION_STATUS as output.

T250090 Populate Rejection Reason

This job uses R3mirror.T751C as input.

This job creates StageSAP.REJECTION_REASON as output.

T250100 Populate Recruitment Source

This job uses the following input:

- SAPSERVE.T750C
- SAPSERVE.T750H
- R3mirror.T750i

This job creates StageSAP.RECRUITMENT_SOURCE as output.

T250115 Inter PB4000 Applicant Status

This job uses the following input:

- R3mirror.PB4000
- R3Mirror.PB4001
- R3Mirror.PB4002

This job creates the following output:

- Intermed.PB4000
- Intermed.PB4001
- Intermed.PB4002

T250120 Inter PB0002 and Inter PB0006 Date Join

This job uses the following input:

- Intermed.PB0002
- Intermed.Applicant_Addresses

This job creates Intermed.PB0002_and_addresses as output.

T250130 Consolidate Applicant Basic Pay Information

This job uses R3mirror.PB0008 as input.

This job creates Intermed.PB0008 as output.

T250150 InterPB0001 InterPB000_PB10016 PB0022 Join

This job uses the following input:

- R3mirror.PB0022
- Intermed.PB0001_and_PB0016_join
- Intermed.PB0001

This job creates Intermed.merge_01_16__22_01 as output.

T250160 DateJoin1 - PB0077 and Inter PB0016

This job uses the following input:

- Intermed.PB0016
- R3Mirror.PB0077

This job creates Intermed.DJ_Infotypes_PB0077_PB0016 as output.

T250170 DateJoin2 - Inter Address with Date Dependent Join 1

This job uses the following input:

- Intermed.PB0002_and_addresses
- Intermed.DJ_Infotypes_PB0077_PB0016

This job creates Intermed.DJ_77_16 and address as output.

T250190 Stage Employment Application Population

This job uses the following input:

- Intermed.PB4000
- Intermed.PB4001
- Intermed.PB4002
- Intermed.DJ_77_16_and_address
- Intermed.merge_01_16__22_01

This job creates StageSAP.EMPLOYMENT_APPLICATION as output.

Job Group: T2600 HR Job

Overview

This job group processes tables that relate to HR job information. These include JOB tables plus the reference tables JOB_GROUP, and EEO_CLASS. Some of the information for these tables might not be available or relevant for all countries. The data processed by this job group includes job title and job grouping. For example, a secretary (job code 20) belongs to the clerical job group (job group 20).

This job group includes jobs that transform data for SAS Human Capital Management.

T260010 Populate Stage Job

This job populates the staging JOB table based on the SAP table T513S. The output table does not have a language code so only one description per job is supported even though SAP might have descriptions in multiple languages. It is also possible that SAP does not contain a description for each job in each language. For this reason, the preferred language transformation selects the description that is in the preferred language. If the description is not available in the preferred language, the transformation selects the description in an available language. The transformation always selects the latest description.

The job group code is based on a concatenation of the EEOCT and AAPCT columns from the SAP table T5U13. All jobs in T513S are used. If no matching record is found in T5U13, or if the EEOCT and AAPCT columns are blank, then the job group code is set to **UNKNOWN**. This practice is instated because the information for the concepts of EEO and AAP are generally maintained only at SAP installations in the United States.

This job uses the following input:

- R3mirror. T513S
- R3 Mirror.T5U13

This job creates StageSAP.JOB as output.

T260020 Populate Staging EEO_CLASS & JOB_GROUP

This job creates the reference tables for the JOB table. It takes the valid values for EEO and AAP from the SAP tables T5UEE and T5UAA respectively. In addition, this job uses the SAS source code t260020_Create_unknown_jobgroup.sas and t260020_Create_unknown_eeo_class.sas to add the value **UNKNOWN** as a valid code.

This job uses the following input:

- R3mirror.T5UEE
- R3Mirror.T5UAA
- Intermed.JOB_GROUP

This job creates the following output:

- StageSAP.EE_CLASS
- StageSAP.JOB_GROUP

Job Group: T2700 HR Job Position

Overview

The job group HR Job Position populates information that relates to employee positions. Employee positions can be considered to be instances of jobs, and many positions can be of the same job. As a result, employee positions need to reference a job. The data describes the permanence, pay level, and benefits associated with a given position.

This job group includes jobs that transform data for SAS Human Capital Management.

T270010 HR Employee x Position

This job creates tables that use the organizational assignments that are stored in infotype 1 (PA0001) to identify relationships. The cost centers associated with employees is populated as well as the Staging Employee_x_job table. This table identifies the position that each employee has.

Unfortunately, most SAP systems do not have perfect data for all employees for their entire employment history. This is especially the case for information that is relevant to time periods before SAP HR was implemented in the company. Therefore, many employees have data records that indicate that they held positions `00000000` or `99999999`, which are not real positions. The SAS Solutions Adapter for SAP removes false records, but you can also choose an alternative approach to add the false positions to the reference tables.

The Inter table Filled Positions is also populated by this job. The table is later combined with the defined positions and can be used to deduce when positions were filled or vacant.

This job uses the following input:

- R3mirror.PA0001
- R3mirror.PA0016

This job creates the following output:

- Intermed.Position X Cost Center
- StageSAP.EMPLOYEE_X_JOB
- Intermed.Filled_Positions

T270020 HR Position Title

This job populates its output table Intermed.position_title with the position title in the preferred language, if possible. If the preferred language is not available, the job uses an available description.

This job uses R3mirror.T528T as input.

This job creates Intermed.position title as output.

T270030 - Inter Pay Frequency

This job populates the output table Intermed.Pay_frequency as the most recent record for each position in the input table Intermed.Filled positions.

This job uses Intermed.Filled positions as input.

This job creates Intermed.Pay Frequency as output.

T270040 Load Stage Time Frequency

One of the attributes of an employee position is the amount of time that an employee needs to work to completely fill the position. For example, a position might be defined as a part-time position and could be filled by someone working part-time. SAP does not have direct information about the required working time per position. It provides information that specifies the number of hours (DIVGV) an employee works in each payment period in infotype 8 (PA0008).

This SAS Solutions Adapter for SAP job uses the unique values of working hours to create a reference table. This job also reduces the number of codes by rounding periods that are over 100 hours per pay period to a full hour. It also rounds periods that are under 100 hours to a tenth of an hour.

This job uses additional time frequency codes that are taken from the SAP table R3mirror.T549N. The two logically different time frequencies are combined into a single reference table StageSAP.TIME_FREQUENCY.

This job uses the following input:

- R3mirror.T548N
- R3Mirror.PA0008

This job creates the following output:

- Intermed.Time_Frequency_T548N
- Intermed.Time Frequency PA0008
- StageSAP.TIME_FREQUENCY

T270050 Inter Position Open and Close

This job derives the position opening date from the lowest date that is found in the input SAP table R3mirror.T528B. It also derives the position closing date from the highest date that is found in the table.

This job uses R3mirror.T528B as input.

This job creates the following output:

- Intermed.Position Close
- Intermed.Position Open

T270070 Load Inter Position x Cost Center

This job creates a table that identifies which cost centers are associated with each position.

This job uses R3mirror.PA0001 as input.

This job creates Intermed.Position X Cost Center as output.

T270080 Load Inter Cost Benefits

This job populates requisition cost and benefit information that is related to positions. The SAS source t270080_load_inter_cost_benefits.sas gets data from R3mirror HRP1015. This code is site-dependent and needs to be reviewed to ensure that it meets the needs of your specific environment. It is possible that the information on position requisition and position benefits is not directly stored in SAP. An alternative that is not supported by the SAS Solutions Adapter for SAP is to extract the benefits information that is associated with employees from SAP and deduce that the benefits are associated with the position.

Some benefits are expressed as percentages. To ensure that the output table is populated correctly, the benefit should be defined in the **`benefit_percent`** macro variable of preprocparms12.sas. When a match to the macro variable is made, the user-written code in this job ensures that a value is provided in the BENEFIT_PERCENT_NO field.

All non-percentage values are totaled and populate the REQUISITION_COST_AMT field.

This job uses R3mirror.HRP1015 as input.

This job creates Intermed.Cost Benefits as output.

T270090 Load Inter Position x Int Org

This job populates a table that relates positions to internal organizations.

This job uses R3mirror.HRP1001 as input.

This job creates Intermed.Position x Int Org as output.

T270100 Load Inter Filled Position + Consolidated 52

This job derives the position status as filled or open. For best results, you should ensure that the following mappings are correct in the LOADER step of the output table Intermed.Position Status Info:

- **`ACTUALPERIOD_BEGIN`** maps to **`coalesce(BEGDA ,VALID_FROM_DTTM )`**
- **`ACTUALPERIOD_END`** maps to **`coalesce(ENDDA ,VALID_TO_DTTM )`**

This job includes an example of how anomalies can be reported via e-mail. For this action to work correctly, the SAS server has to be configured to send e-mail.

This job uses the following input:

- R3mirror.T528B
- Intermed.Filled Positions

This job creates Intermed.Position Status Info as output.

T270110 Load Inter Job Position

This job populates an intermediate table that contains information required in the final staging JOB_POSITION table. It merges EEO and AAP information from the SAP table R3mirror.T5U29. Because EEO and AAP information is generally maintained by United States companies, you might choose to simplify the job by removing the merge step. However, this is not strictly necessary because if the table R3mirror.T5U29 is empty, the result is not affected.

This job uses the following input:

- R3mirror.T528B
- R3Mirror.T5U29

This job creates Intermed.JOB_POSITION as output.

T270120 Populate Inter Position Stage 2

This job merges all the intermediate tables that are date-independent with Intermed.JOB_POSITION. The Intermed.JOB_POSITION table contains date range columns. These tables need to contribute to the final JOB_POSITION table.

This job uses the following input:

- Intermed.Position x Cost Center
- Intermed.Position Open
- Intermed.Position Close
- Intermed.Position title
- Intermed.Cost benefits
- Intermed.JOB_POSITION
- Intermed.Pay Frequency

This job creates Intermed.JOB_POSITION_STAGE2 as output.

T270130 Load Stage Job Position

This is the last job in the sequence of jobs that create StageSAP.JOB_POSITION. This job date-joins the date-dependent results of the previous steps. This job also populates the reference table StageSAP.POSITION_STATUS with constant values using the SAS source code in t270130_load_position_status.sas.

This job uses the following input:

- Intermed.JOB_POSITION_STAGE2
- Intermed.Position Status Info

This job creates the following output:

- StageSAP.JOB_POSITION
- StageSAP.POSITION_STATUS

Job Group: T2800 HR Military Experience

Overview

This job group populates tables that relate to military experience. Military experience information can be stored in different, country-dependent locations in SAP HR. The SAS Solutions Adapter for SAP provides a few alternatives. You can select the most appropriate version for your environment.

This job group includes jobs that transform data for SAS Human Capital Management.

T280000A Load Stage Military Experience and Type

This job populates the reference table Stage Military Experience Type. The format **`$SAPTOSAS_LANGUAGE`** needs to be used in the mapping step to convert one-character language in SAP to the equivalent SAS value.

This job uses the following input:

- R3mirror.PA0081
- R3mirrir.T591S
- **`$SAPTOSAS_Language.`**

This job creates StageSAP.MILITARY_EXPERIENCE_TYPE as output.

T280000B Load Stage Military Experience Type

This job is an alternative to T280000A Load Stage Military Experience and Type. It populates the reference table Stage Military Experience Type.

This job uses the following input:

- R3mirror.T505N
- **$SAPTOSAS_Language.**

This job creates StageSAP.MILITARY_EXPERIENCE_TYPE as output.

T2800010B Military Experience from PA0001and PA0077

This job must be used in conjunction with the job T280000B Load Stage Military Experience Type. The military experience code is based on the MILSA column in infotype 77 (R3mirror.PA0077).

This job uses the following input:

- R3mirror.PA0001
- R3mirror.PA0077
- **$company_to_country.**
- **$country_to_molga.**

This job creates Intermed.Inter_Military_Experience as output.

Job Group: T2900 HR Internal Org

Overview

This job group populates the tables that relate to the Internal Org Assoc information from SAP HR sources. These tables include EMPLOYEE_X_INTERNAL_ORG, INTERNAL_ORG, INTERNAL_ORG_ASSOC, and other related reference tables.

Not every site has current data with which to record this information. For example, data about work-group manager positions might not be maintained. In these cases, the recommended practice is to leave the process and code intact for when the data is maintained later.

The HR organization hierarchy in SAP is represented differently from financial dimensional data, and it is difficult to relate financial data to the HR hierarchy. The tables that this job group populates are therefore populated for SAS Financial Management from different sources. In addition, the two approaches used to load the internal_org dimension co-exist and can be used in isolation.

The main input SAP tables that are used are those that describe HR objects and the relationships among them. Here are the tables that are used:

- HRP1000 contains text descriptions of HR objects.
- HRP1001 describes relationships between HR objects such as employees, organizational entities, positions, and so forth.
- R3mirror.HRP1028 contains addresses of HR objects.
- T777V contains text descriptions of possible relationships.

This job group includes jobs that transform data for SAS Human Capital Management.

T290010 HR Work Group Hierarchy Join SAP HR Hier Tables

The core logic of this job is an SQL join of the two SAP tables R3mirror.HRP1000 and R3mirror.HRP1001. Because these tables can both contain non-relevant information, the SQL join conditions select the information that is required. The relationship that is used to select objects is 002 ("reports to").

This job uses the following input:

- R3mirror.HRP1000
- R3Mirror.HRP1001

This job creates Intermed.HR Org Structure as output.

T290020 Select HR Employee Work Group Manager

The position_ids for managers of groups is extracted from the HR relationships table R3mirror.HRP1001 using the relationship 012 ("manages").

This job uses R3Mirror.HRP1001 as input.

This job creates Intermed.Workgroup Manager Positions as output.

T290030 Get Manager Employee_id

This job uses the result of the previous job (Intermed.Workgroup Manager Positions) and combines it with the StageSAP.EMPLOYEE_X_JOB information to derive the employee ID of workgroup managers.

This job uses the following input:

- Intermed.Workgroup Manager Positions
- StageSAP.EMPLOYEE_X_JOB

This job creates Intermed.Workgroup Manager Employee_ids as output.

T290040 Load EMPLOYEE_INVOLVEMENT_TYPE

This job loads the StageSAP.EMPLOYEE_INVOLVEMENT_TYPE reference table. It loads this table from the SAP table that contains descriptions of the relationships that can be had among HR objects in the table HRP1001.

This job uses R3mirror.T777V as input.

This job creates StageSAP.EMPLOYMENT_INVOLVEMENT_TYPE as output.

T290060A Load EMPLOYEE_X_INTERNAL_ORG Method A

This job is one of two alternative approaches to populate the output table StageSAP.EMPLOYEE_X_INTERNAL_ORG. This job uses HR infotype 1 (PA0001) as its source.

This job uses R3mirror.PA0001 as input.

This job creates StageSAP.EMPLOYEE_X_INTERNAL_ORG as output.

T290060B Load EMPLOYEE_X_INTERNAL_ORG Method B

This alternative approach uses the HR object table HRP1001. It also uses StageSAP.EMPLOYEE_X_JOB which is derived from PA0001, the source used in the first alternative.

This job uses the following input:

- R3mirror.HRP1001
- StageSAP.EMPLOYEE_X_JOB

This job creates StageSAP.EMPLOYEE_X_INTERNAL_ORG as output.

T290070 Load HR Org Address

This job loads the latest address for HR organizations.

This job uses R3mirror.HRP1028 as input.

This job creates Intermed.Org Address as output.

T290080 Load HR INTERNAL ORG

This job combines the results of previous jobs in the job group to populate the output table StageSAP.INTERNAL_ORG.

Note: This job uses a format **`$OP_CONCERN_CURRENCY`** that the T120010A Operating Concern Internal Org job creates. The T120010A Operating Concern Internal Org job must be run prior to the T290080 Load HR INTERNAL ORG job.

This job uses the following input:

- Intermed.Org Address
- Intermed.HR Org Strucure
- Intermed.Workgroup Manager Employee_ids

This job creates StageSAP.INTERNAL_ORG as output.

T290090 Load HR INTERNAL_ORG_ASSOC

This job combines the results of previous jobs in the job group to populate the output table StageSAP.INTERNAL_ORG_ASSOC.

This job uses Intermed.HR Org Strucure as input.

This job creates StageSAP.INTERNAL_ORG_ASSOC as output.

T290100 Internal Org Add Standard Dimension Values

This job adds integer order numbers to the output table StageSAP.INTERNAL_ORG_ASSOC and populates the table StageSAP.INTERNAL_ORG.

This job uses the following input:

- StageSAP.INTERNAL_ORG_ASSOC_TYPE
- Transformation Add Standard Dimension Rows

This job creates the following output:

- StageSAP.INTERNAL_ORG_ASSOC
- StageSAP.INTERNAL_ORG

Job Group: T3500 HR Employee Actions

Overview

This job group populates tables that relate to actions and events in an employee's career. The main table is EMPLOYEE_ACTION, in addition to other related reference tables.

The EMPLOYEE_ACTION table defines current and historical action information that pertains to employees. This table defines types of employee actions such as pay increases, probations, suspensions, and so forth. This table also represents valid combinations of action type and reason type, as well as status information that was valid at the time of the action or event.

There is a specific logic flow in this job group. First, the job group creates a table that contains status information and date-joins all relevant information. Then, the jobs join all action information with the action date that is in the begin and end date range of the status table. If the action date is not at the beginning of the date range, a derived action is used to indicate that the status was changed. For example, if salary data changed but there is no explicit action for this occurrence, then an action that means "change in salary" is created.

This job group includes jobs that transform data for SAS Human Capital Management.

T350010 Load Stage Employee Status

This job loads the StageSAP.EMPLOYEE_STATUS reference table from the SAP table that contains data element texts in SAP such as DD07T.

This job uses R3mirror.DD07T as input.

This job creates StageSAP.EMPLOYEE_STATUS as output.

T350020 Load Stage Employee Type

This job uses R3mirror.T503T as input.

This job creates StageSAP.EMPLOYEE_TYPE as output.

T350030 Load Stage Action Type

The action type reference table is populated from the SAP action type text tables and actions that are added to represent implicitly detected actions.

This job uses the following input:

- R3mirror.T530F
- R3mirror.T529T
- Admin.Additional ACTION_TYPES

This job creates StageSAP.ACTION_TYPE as output.

T350040 Employee Working Time

The working time is derived from the column WOSTD (weekly worked hours) in infotype 7 (PA0007).

This job uses R3mirror.PA0007 as input.

This job creates Intermed.Working Times as output.

T350060 Load Stage Action Reason

Action reasons are a combination of the reasons for employee actions and a related type reference table. The related type reference table that is used is populated from the SAP action type text tables and actions that are added to represent implicitly detected actions.

This job uses the following input:

- R3mirror.T530T
- R3Mirror.T530F

This job creates StageSAP.ACTION_REASON as output.

T350070 DateJoin PayInfo

This job combines tables that are created in other job groups to prepare the payment status information at the time of actions or events.

This job uses the following input:

- Intermed.Base Compensation
- Intermed.Employee action payinfo

This job creates Intermed.Payinfo_for_employee_actions as output.

T350080 DateJoin PayInfo and Working Time for Actions

This job date-joins working time and payment information to create a table that shows both sets of information with date ranges.

This job uses the following input:

- Intermed.Payinfo_for_employee_actions
- Intermed.Working_times

This job creates Intermed.Pay_info_and_work_time as output.

T350090 DateJoin PayInfor-Worktime with PA001

This job is another in the sequence of date-joins that join the previous intermediate results to create a table about the employee at the time of an action.

This job uses the following input:

- Intermed.Pay_info_and_work_time
- R3Mirror.PA0001

This job creates Intermed.Status info for employee actions as output.

T350100 Actions from Infotype 0

This job creates two output tables with the information that is needed from infotype 0 (PA0000).

This job uses R3mirror.PA0000 as input.

This job creates the following output:

- Intermed.emp_actions_what_when_why
- Intermed.Employee status

T350105B Employee Appraisal Groupings for Infotype 25

SAP supports two different mechanisms that record information about employee appraisals. The SAS Solutions Adapter for SAP extracts the data from both possible data sources. This job must be used in conjunction with job T350110B Appraisal (Evaluation Result) Using Infotype 25. You can choose the appropriate method for your local site.

This job creates an intermediate table that stores employee appraisal groupings that are needed to derive the appraisal criteria weightings for appraisals in PA0025.

This job uses the following input:

- R3mirror.PA0001
- Format **`$personnel_subarea_mobur`**
- Format **`$employee_group_appraisal_subgrp`**

This job creates Intermed.EMPLOYEE_APPRAISAL_GROUPINGS as output.

T350110A Appraisal (Evaluation Result) Using HRP1045

This job is an alternative to the jobs T350105B Employee Appraisal Groupings for Infotype 25 and T350110B Appraisal (Evaluation Result) Using Infotype 25. This job also creates an intermediate table that stores employee appraisal groupings that are needed to derive the appraisal criteria weightings for appraisals in PA0025.

This job uses the following input:

- R3mirror.HRP1045
- R3mirror.HRT1045
- R3mirror.HRP1001

This job creates Intermed.Evaluation_results as output.

T350110B Appraisal (Evaluation Result) Using Infotype 25

This job creates the appraisal results. The job must be used in conjunction with job T350105B Employee Appraisal Groupings for Infotype 25.

This job uses the following input:

- R3mirror.PA0025
- Intermed.EMPLOYEE_APPRAISAL_GROUPINGS

This job creates Intermed.Evaluation_results as output.

T350120 Append Additional Actions from Infotype 302

This job combines results from previous jobs and appends additional actions from infotype 302 (PA0302). Infotype 302 is used as an overflow table when more than one action occurs on the same day, because that would cause problems with the date ranges (specified by the BEGDA and ENDDA columns) in HR infotypes.

Note: Not all SAP sites activate infotype 302. If a site does activate infotype 302, the table can be removed from the job.

The source code that performs the combining process is **`t350120_combine.sas`**.

This job uses the following input:

- R3mirror.PA0302
- Intermed.Evaluation_results
- Intermed.emp_actions_what_when_why

This job creates Intermed.emp_actions_what_when_why2 as output.

T350130 EEO Exemption (USA) Status

This job creates a table that contains EEO exemption information that is relevant for United States employees. Only exempt employees are kept in the output table.

Note: This job should be run for all countries at least once to create the correct structure.

This job uses R3mirror.PA0077 as input.

This job creates Intermed.USA_EEO_Exemption as output.

T350200 Join Action Status with Action Dates

This job combines the results of previous jobs to create the final employee action table.

This job uses the following input:

- Intermed.emp_actions_what_when_why2
- Intermed.State_info_employee_actions
- Intermed.employee_status

This job creates SAPStage.SAP_EMPLOYEE_ACTION as output.

T350220 Update Employee Actions with Exempt Status

This job updates the employee_action table with USA EEO exempt status. This job is optional for companies that are not based in the United States.

Because this job uses an update load step, the output table is actually used as both input and output.

This job uses the following input:

- SAPStage.SAP_EMPLOYEE_ACTION
- Intermed.USA_EEO_Exemption

This job creates SAPStage.SAP_EMPLOYEE_ACTION as output.

CHAPTER 8

Integrating into the Detail Data Store

Integrating Data into the Detail Data Store

Overview

The SAS Solutions Adapter for SAP loads staging area tables for SAS Financial Management and SAS Human Capital Management solutions. Later releases of the SAS Solutions Adapter for SAP populates other subject-area tables for other SAS solutions.

The SAS Solutions Adapter for SAP creates tables that must be incorporated into the overall data architecture. These staging tables that are created as output have the same names and structure as the input tables of the jobs in the SAS Detail Data Store repository. After you run the SAS Solutions Adapter for SAP jobs, you can run the SAS Detail Data Store jobs that use the adapter's output tables as input.

Note: For more information about how to incorporate data into the DDS for SAS Data Integration Studio jobs, refer to the *SAS Solutions Services: Data Administration Guide*.

There are two basic ways to integrate the SAP staging tables into SAS Detail Data Store jobs.

1. Use one physical location for stage libraries.
2. Use multiple sources.

Method 1: Use One Physical Location for Stage Libraries

To use one physical location for the stage libraries, you must define the STAGESAP and STAGEDSS libraries to represent the same physical location.

Note: This method might not be applicable if the data that needs to be loaded comes from various sources such as multiple SAP systems and other operational sources. In this case, consider Method 2.

Method 2: Multiple Sources

To accommodate multiple data sources, you can build additional jobs that combine the multiple data sources into a new library in the STAGEDDS structure. These new jobs might consist of a simple step that appends all the data from the various input sources.

In the case that a simple append of data from different sources does not produce the required results, you must resolve the issue with site-specific code that replaces or augments the append step.

Customizing the Validation Steps in the Load DDS Jobs

Certain data checks must be removed in order to successfully integrate SAP data into the DDS. For example, some items might need to be removed from the data validation step or the lookup steps for FACT tables might need to be switched off.

The table below details actions that might be required for successful data loading to the DDS for your site. This table provides a general guide, but it is possible that it does not provide an all-inclusive list for your specific site. This table might not provide all of the actions that are required for your specific site because each individual SAP site can provide some data by default and some data by customization within the SAS Solutions Adapter for SAP jobs that create or bring in data from additional tables.

Note: The SAS Solutions Adapter for SAP creates stage tables where relevant data is available in SAP. You must create alternative jobs to populate other tables if other tables are required.

The following table describes customizations that you might consider. You do not need to run all jobs in order to populate the DDS for a particular solution.

Note: The table below includes only the jobs that typically require modification. The table does not include a comprehensive list of all available jobs.

Table 8.1 Actions That Are Required for Successful Data Loading to the DDS

Job Group	Job Name	Action
Reference Table Jobs	103800 Load DDS GL_ACCOUNT_TYPE Table	Delete the APPEND step because the SAS table provides the data.
Dimension Table Job	102400 Load DDS CURRENCY Table	Delete the input tables and APPEND step that is above the STAGE_CURRENCY table. The input table is the STAGE_CURRENCY table that is prepared in the SAS Solutions Adapter for SAP.
Reference Table Jobs	102700 Load DDS CURRENCY_EXCH_RATE_TYPE Table	Delete the APPEND step. The input table should be SAS_ CURRENCY_EXCH_RATE_TYPE
	1003900 Load DDS GL_NORMAL_BAL Table	Delete the APPEND step because the SAS table provides the data.

Job Group	Job Name	Action
	100500 Load DDS PERIOD_TYPE Table	Get the data from SAPStage library. Delete the APPEND step because SAP includes all entries that are held in the SAS table.
		See the job T000210 Load Staging Time Period.
	100600 Load DDS SOURCE_SYSTEM Table	Delete the APPEND step because the SAP table includes all entries that are held in the SAS table.
	104500 Load DDS MEASURE Table	Delete the APPEND step because the SAS table provides the data.
	10700 0 Load DDS CURRENCY_EXCH_RATE Table	Switch off the lookup for CURRENCY_EXCH_RATE_SET.
	106300 Load DDS JOB_POSITION Table	Remove data validation for POSITION_STATUS_CD, FLSA_STATUS_CD, EXEMPT_STATUS_CD, PAY_FREQUENCY_CD, and CURRENCY_CD.
		Switch off the lookup for ALLOCATED_TO_COST_CENTER_ID.
Dimension Table Jobs	105810 Load DDS TIME_PERIOD_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_TIME_PERIOD_ID.
	106100 Load DDS COST_CENTER Table	Switch off the lookup for RESPONSIBLE_EMPLOYEE_ID.
	106110 Load DDS COST_CENTER_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_COST_CENTER_ID.
	106410 Load DDS ANALYSIS_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_ANALYSIS_ID.
	106700 Load DDS EXTERNAL_ORG Table	Delete the INDUSTRY_CD data validation.
	106710 Load DDS EXTERNAL_ORG_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_EXTERNAL_ORG_ID.
	106200 Load DDS INTERNAL_ORG Table	Delete the STATE_REGION_CD data validation.
		Switch off the lookup for EMPLOYEE.
	106210 Load DDS INTERNAL_ORG_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_INTERNAL_ORG_ID.
	106500 Load DDS GL_ACCOUNT Table	Allow blanks for ACCOUNT_TYPE_CD, EXCHANGE_RATE_TYPE_CD, and NORMAL_BALANCE_CD.
		Switch off the lookup for INTERNAL_ORG_ID.
	106510 Load DDS GL_ACCOUNT_ASSOC_TYPE Table	Switch off the FACT table lookup for DEFAULT_GL_ACCOUNT_ID.
	106900 Load DDS PROFIT_CENTER Table	Switch off the lookup for DEFAULT_RESPONSIBLE_EMPLOYEE_ID.
	106910 Load DDS PROFIT_CENTER_ASSOC_TYPE Table	Switch off the lookup for DEFAULT_PROFIT_CENTER_ID.

Job Group	Job Name	Action
Fact Table Jobs	107910 Load DDS GL_JRNL_DETAILS Table	Remove data validation for ITEM_CATEGORY_CD.
	106740 Load DDS EXTERNAL_ORG_ADDRESS Table	Remove data validation for STATE_REGION_CD.
	107700 Load DDS EMPLOYMENT_APPLICATION Table	Remove data validation for STATE_REGION_CD, EMPLOYEE_ID, GENDER_CD, REJECTION_REASON_CD, COUNTY_NM, COUNTRY_CD, and ETHNICITY_CD.
	107600 Load DDS EMPLOYEE_ACTION Table	Remove data validation for PAY_LEVEL_CD, CURRENCY_CD, EMPLOYEE_ACTION_REASON_CD, EMPLOYEE_STATUS_CD, EMPLOYEE_TYPE_CD, ACTION_TYPE_CD, PAY_LEVEL_STRUCTURE_CD, PAY_FREQUENCY_CD, and EMPLOYEE_PAID_FREQUENCY_CD.

Here are some examples of job customizations that might be required to ensure that the checks do not reject valid records:

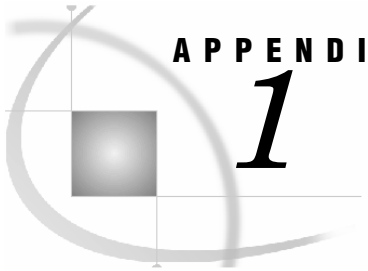
- In the job Load DDS Cost Center Table and the job Load DDS Profit Center Table, remove the lookup checks for the employee table that is removed. This customization is necessary for the following reasons:
  - The employee number is not in the relevant SAP tables. These tables simply include an unchecked string that often contains a name or initials of the responsible employee.
  - The SAS Solutions Adapter for SAP might not load the employee dimension table.
- In the job Load DDS GL_JRNL_DETAILS Table, remove the check against the Item_category table because the SAS Solutions Adapter for SAP does not load it.
- In the job Load DDS INTERNAL_ORG Table, remove the checks against the County, State_Region, and employee columns. This information is not loaded into the INTERNAL_ORG table. Alternatively, you might consider populating the County and State_Region columns in an additional user-exit step.
- The job Load DDS EXTERNAL_ORG Table should not validate against Industry because the SAS Solutions Adapter for SAP does not load Industry.
- The job Load External_ORG_ADDRESS Table should not validate against State_Region for non-US data because this column is not populated for other countries.
- Do not reject blank address types. Replace them with a standard value. The SAS Solutions Adapter for SAP does not populate the Address_type reference table. As a result, add an address type table and enter the appropriate values.

Implementation Notes for SAS Human Capital Management Users

After loading the DDS, you must load the SAS Human Capital Management solution data mart if that is the application that you are using. Preparation for this task includes reviewing the SAS source code **prebuild.sas**.

In order to enable a smooth interaction with the SAS Solutions Adapter for SAP, be sure to replace the corresponding macro variable code with the following statements:

- **%let number_of_hiers = 1;**
- **%let hiers = HRG;**



APPENDIX 1

Solution Package Deployment Parameters

Solution Package Deployment Parameters

To deploy the SAS Solutions Adapter for SAP, you can use the Solution Package Deployment Wizard that is available from SAS Management Console. This wizard prompts you to specify parameter values based on the components that you choose to deploy.

The Parameters page of the wizard includes a list of parameters that varies based on your deployment environment and selected components. The list below describes the possible parameters that you might need to specify on this page.

Enter path for storing SAS Source Code

Enter the SAS server folder where the delivered source for the SAS Solutions Adapter for SAP is stored.

Path to location where HTML error reports will be published

Enter the SAS server folder that is used for storing HTML reports.

Provide email address for receipt of error reports

Enter the e-mail address of the recipient who receives error reports.

Enter server path for creation of FORMATS Library

Enter the location where the SAS Solutions Adapter for SAP stores SAS formats.

Enter server path for creation of INTERMED Library

Enter the location where the SAS Solutions Adapter for SAP creates and stores Intermediate tables.

Enter server path for creation of R3MIRROR Library

Enter the location where the SAS Solutions Adapter for SAP creates and stores R3Mirror tables.

Enter server path for creation of STAGESAP Library

Enter the location where the SAS Solutions Adapter for SAP creates and stores STAGESAP tables.

Enter server path for creating of SAS SUPPLIED Tables Library

Enter the location where the SAS Solutions Adapter for SAP creates and stores SAS SUPPLIED tables.

Enter server path for creation of ADMIN Library

Enter the location where the SAS Solutions Adapter for SAP creates and stores ADMIN tables.

Select SAP Server

Select the server that corresponds to the desired SAP server.

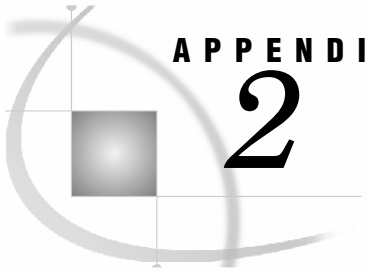
Select SAPSERVE connection

Select the connection that corresponds to the desired SAP server.

Select SAPSERVE library

Select the library that corresponds to the SAP server.

Note: The data that is extracted and transformed from an SAP system is confidential. Financial Management data is confidential to protect the business and HR data and protect employees' privacy. In addition to using the authenticated data access that is provided by SAS Management Console and SAS Data Integration Studio, ensure that the appropriate data protection at the operating level is instated on the selected server locations.



APPENDIX 2

Tables Extracted from SAP

Tables Extracted from SAP

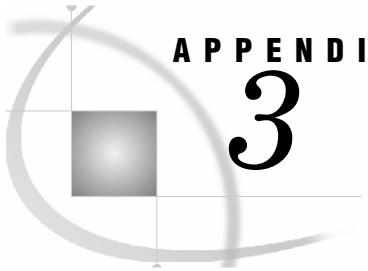
Table	Description
AUFK	Order master data
BKPF	Accounting Document Header
BSEG	Accounting Document Segment
CEPC	Profit Center Master Data Table
CEPCT	Texts for Profit Center Master Data
COBK	CO Object: Document Header
COEP	CO Object: Line Items (by Period)
CSKS	Cost Center Master Data
CSKT	Cost Center Texts
CURRENT_PAY_RESULTS	Employees' current compensation information, as extracted by the ABAP program from SAP R/3.
DD03T	DD: Texts for fields (language dependent)
DD04T	DD: Data element texts
DD07T	DD: Texts for Domain Fixed Values (Language-Dependent)
DD07T_GESCH	DD: Texts for Domain GESCH (Gender) (Language-Dependent)
DD07T_SHKZG	DD: Texts for Domain Fixed Values (Language-Dependent)
DD07T_STATN	DD: Texts for Domain STATN (Employee Status) (Language-Dependent)
GLT0	G/L account master record transaction figures
HRP1000	Infotype 1000 DB Table
HRP1001	Infotype 1001 DB Table
HRP1015	Infotype 1015 DB Table
HRP1028	Infotype 1028 Object Address
HRP1032	Infotype 1032 Email address
HRP1045	Infotype 1045 DB Table
HRT1045	Infotype 1045 Table Section
KNA1	General Data in Customer Master
KNVH	Customer Hierarchies

Table	Description
LFA1	Vendor Master (General Section)
LFMH	Vendor hierarchy
MAKT	Material Descriptions
MARA	General Material Data
NRIV	Number Range Intervals
PA0000	HR Master Record: Infotype 0000 (Actions)
PA0001	HR Master Record: Infotype 0001 (Org. Assignment)
PA0002	HR Master Record: Infotype 0002 (Personal Data)
PA0006	HR Master Record: Infotype 0006 (Addresses)
PA0007	HR Master Record: Infotype 0007 (Planned Working Time)
PA0008	HR Master Record: Infotype 0008 (Basic Pay)
PA0016	HR Master Record: Infotype 0016 (Contract Elements)
PA0022	HR Master Record: Infotype 0022 (Education)
PA0025	HR Master Record: Infotype 0025 (Appraisals)
PA0029	HR Master Record: Infotype 0029 (Workers' Compensation)
PA0041	HR Master Record: Infotype 0041 (Date Specifications)
PA0057	HR Master Record: Infotype 0057 (Membership Fees)
PA0077	HR Master Record: Infotype 0077 (Additional Personal Data)
PA0080	HR Master Record: Infotype 0080 (Maternity Protection)
PA0081	HR Master Record: Infotype 0081 (Military Service)
PA0105	HR Master Record: Infotype 0105 (Communications)
PA0302	HR Master Record: Infotype 0302 (Additional Actions)
PA0521	HR Master Record: Infotype 0521 (Semiretirement D)
PA0597	HR Master Record: Infotype 0597 (Teilzeitarbeit waehrend Elternzeit)
PA2001	HR Time Record: Infotype 2001 (Absences)
PB0001	HR Master Record: Infotype 0001 (Org. Assignment)
PB0002	HR Master Record: Infotype 0002 (Personal Data)
PB0006	HR Master Record: Infotype 0006 (Addresses)
PB0007	HR Master Record: Infotype 0007 (Planned Working Time)
PB0008	HR Master Record: Infotype 0008 (Basic Pay)
PB0016	HR Master Record: Infotype 0016 (Contract Elements)
PB0022	HR Master Record: Infotype 0022 (Education)
PB0077	HR Master Record: Infotype 0077 (Additional Personal Data)
PB4000	HR Master Record: Infotype 4000 (Applicant Events)
PB4001	HR Master Record: Infotype 4001 (Applications)
PB4002	HR Master Record: Infotype 4002 (Vacancy Assignment)
PRPS	WBS (Work Breakdown Structure) Element Master Data

Table	Description
RF011P	Items in Financial statement for a selected balance sheet report
RF011Q	Financial statement text
RF011Z	Balance sheet line assignment
SALDO	HR time recording cluster table SALDO - Monthly aggregates of employee time balances.
SKA1	G/L Account Master Record (Chart of Accounts)
SKAT	G/L Account Master Record (Chart of Accounts: Description)
SKB1	G/L Account master (company code)
T001	Company Codes
T001P	Personnel Area/Subarea
T002	Language Keys
T002T	Language Key Texts
T003	Document Types
T003T	Document Type Texts
T004	Directory of Charts of Accounts
T004T	Chart of Account Names
T005T	Country Names
T005U	State, Region, County
T009	Fiscal Year Variants
T009B	Fiscal Year Variant Periods
T009C	Period names
T009T	Fiscal year variant names
T009Y	Shortened fiscal years in Asset Accounting
T011	Financial Statement Versions
T011T	Financial statement version names
T030	Standard Accounts Table
T030C	Global Standard Account Table (contains retained earnings specification)
T030D	Acct Determ.for Open Item Exch.Rate Differences
T030H	Acct Determ.for Open Item Exch.Rate Differences
T030R	Rules for Determination of Standard Accounts
T030W	Transaction Key Names
T077X	Account Group Names (Table T077D)
T179	Materials: Product Hierarchies
T179T	Materials: Product hierarchies Texts
T500L	Personnel Country Grouping
T500P	Personnel Areas
T502T	Marital Status
T503	Employee Group/Subgroup

Table	Description
T503T	Employee Subgroup Names
T503Z	Country Assignment for Employee Group/Subgroup
T505N	Type of Military Experience
T505S	Ethnic Origin Texts
T510F	Assign Pay Scale to Currency
T510N	Pay Scales for Annual Salaries (NA)
T512T	Wage Type Texts
T513H	Appraisal Criteria
T513S	Job Titles
T519T	Education Final Certificates
T521B	Payee Keys
T521C	Check Table for Payees
T523T	Challenge Types
T528B	Positions
T528C	Wage Type Catalog
T528T	Position Texts
T529T	Personnel Action Texts
T52D6	Wage Type Group Texts
T52D7	Assign Wage Types to base Wage Type Groups
T530F	Reasons for Changes
T530T	Reason for Action Texts
T536B	Communication Keys
T538T	Units of Time/Measurement Texts
T542T	Employment Contracts
T547S	Contract Type Texts
T548T	Date Types
T549A	Payroll Areas
T549N	Period Modifiers
T549Q	Payroll Periods
T549R	Period Parameters
T554T	Absence and Attendance Texts
T591S	Subtype Texts
T5U13	Jobs
T5U29	EEO/AAP: Position Attributes for US
T5UAA	AAP Occupational Categories
T5UEE	EEO Occupational Categories
T750B	Recruitment Advertisement
T750C	Recruitment instrument

Table	Description
T750H	Text for recruitment medium
T750I	Text for unsolicited application group
T751B	Text for applicant status
T751C	Applicant status reason
T751F	Text for applicant event type
T777Q	Proficiency Texts
T777T	Relationship Infotypes
T777U	Relationship Subtype Texts
T777V	Relationship Texts
T880	Global Company Data (for KONS Ledger)
TBSLT	Posting Key Names
TCURR	Exchange Rates
TCURT	Currency Code Names
TCURW	Usage of Exchange Rate Types
TGSBT	Business Area Names
THITT	Texts for Customer Hierarchy Types
TKA00	Control parameters for controlling areas
TKA01	Controlling Areas
TKA02	Controlling area assignment
TKEB	Management for Operating Concerns (Client-Specific)
TKEBT	Description of operating concern
TLHITT	Vendor Hierarchy Category (Description)
ZES	HR detailed time recording cluster table ZES



APPENDIX 3

ABAP Function Module SAS/HR_EVALUATE_WAGES

ABAP Function Module SAS/HR_EVALUATE_WAGES Overview

The ABAP function module SAS/HR_EVALUATE_WAGES indirectly evaluates wages for several persons employed on a given date. This module returns the wages for each person that was employed at the specified date for the requested infotype. The specified country grouping is also required to 1) check which wage types are indirectly evaluated, and 2) retrieve the associated evaluation module.

Note: Currently, only infotypes 0008, 0014, 0015, and 0052 are supported.

You can use the table parameters LGART_INCL and LGART_IGNORE to limit the result set of wage types retrieved from table T511. Here are three specific ways that you can limit the result set:

- Specify a range of personnel numbers to process, including minimum and maximum values for personnel numbers.
- Specify a list of wage types that should be included in the result. Types that are not listed are not evaluated or returned. This action is associated with the table parameter LGART_INCL.
- Specify a list of wage types that should be excluded from the result set. In this case, only the types that are not listed are evaluated and returned. This action is associated with the table parameter LGART_IGNORE, used only if LGART_INCL is the initial parameter.

Note: The result table lists the wages per personnel number, infotype, and wage type, whether or not the wage has been computed. It is also possible for wages to be overwritten during indirect valuation of master data maintenance. In SAP, you can see the overwrite indicator flag by using the SAP transaction OH13. This corresponds to the behavior of the SAP transaction PA30.

The figure below provides an example of a call from SAS that retrieves all indirectly evaluated base wages (infotype 0008) for employees with personnel numbers less than or equal to 5000.

```
     PROC CALLRFC libref=SAPSERVE;
   CALL FUNCTION '/SAS/HR_EVALUATE_WAGES '
      EXPORTING pevaldate = '20060731'
                P_MAX_PERNR = '00005000'
                pinftype = '0008'
                tables result = results;
QUIT;
```

After this call, you can process the results to determine which of the wages have been computed (INDBW = 'I').

APPENDIX 4

Job Groups and SAS Solution Data Marts

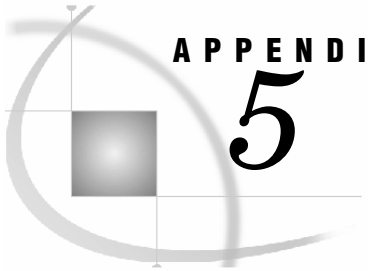
Job Groups and SAS Solution Data Marts

The job groups that are described in this document extract, transform, and load data from standard SAP tables into standard SAS tables in the SAS Detail Data Store. The data is then available in a usable format for SAS Solution Data Marts such as SAS Financial Management and SAS Human Capital Management.

Each job group, in turn, provides ETL functions that are required for or relate to one or more SAS Solution Data Marts. The table below provides a list of each job group and the SAS Solution Data Mart(s) for which it extracts, transforms, and loads data.

Job Group	Corresponding SAS Solution Data Mart(s)
Initialization Job Group	
0 Data Environment Initialization	SAS Human Capital Management
Extraction Job Groups	
E00 Extraction of T Tables	SAS Financial Management
	SAS Human Capital Management
E01 Extraction of Master Tables	SAS Financial Management
	SAS Human Capital Management
E02 Extraction of Transaction Tables	SAS Financial Management
E03 Extraction of Summary Tables	SAS Financial Management
E04 Extraction of Hierarchies and Structures	SAS Financial Management
E05 Extraction of Miscellaneous Other Tables	SAS Financial Management
E06 Extraction of HR PA Infotype Tables	SAS Human Capital Management
E07 Extraction of HR T Lookup Tables	SAS Human Capital Management
E08 Extraction of HR Tables Using ABAP Calls	SAS Human Capital Management
E09 Extraction of HR Applicant Tables	SAS Human Capital Management
E10 Extraction of HR HRP Infotype Tables	SAS Human Capital Management

Job Group	Corresponding SAS Solution Data Mart(s)
Transformation Job Groups	
T0000 Transformations for Common Tables	SAS Financial Management
	SAS Human Capital Management
T0001 Account Dimension	SAS Financial Management
T0002 Time Dimension	SAS Financial Management
	SAS Human Capital Management
T0003 SAS Supplied Tables	SAS Financial Management
	SAS Human Capital Management
T0004 Cost Center Dimension	SAS Financial Management
	SAS Human Capital Management
T0005 Profit Center Dimension	SAS Financial Management
T0006 Analysis Dimension	SAS Financial Management
T0007 Exchange Rates	SAS Financial Management
T1000 Financial Documents	SAS Financial Management
T1100 External Org	SAS Financial Management
T1200 FM Internal Org	SAS Financial Management
T2000 HR Common Reference Tables	SAS Human Capital Management
T2100 HR Employee General Information	SAS Human Capital Management
T2200 HR Employee Absences	SAS Human Capital Management
T2300 HR Employee Base Compensation	SAS Human Capital Management
T2400 HR Employee Compensation History	SAS Human Capital Management
T2500 HR Applicants General Information	SAS Human Capital Management
T2600 HR Job	SAS Human Capital Management
T2700 HR Job Position	SAS Human Capital Management
T2800 HR Military Experience	SAS Human Capital Management
T2900 HR Internal Org	SAS Human Capital Management
T3500 HR Employee Actions	SAS Human Capital Management



APPENDIX 5

Recommended Reading

Recommended Reading

Here is the recommended reading list for this title:

- *Customer Data Integration: Reaching a Single Version of the Truth*
- *SAS Data Integration Studio: User's Guide*
- *SAS Data Surveyor for SAP: Help*
- *SAS Management Console: User's Guide*
- *SAS Solutions Services: Data Administration Guide*
- *SAS Solutions Services: Data Model Reference*

For a complete list of SAS publications, see the current SAS Publishing Catalog. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at the following address:

SAS Publishing Sales

SAS Campus Drive

Cary, NC 27513

Telephone: (800) 727-3228*

Fax: (919) 677-4444

E-mail: **sasbook@sas.com**

Web address: **support.sas.com/pubs**

* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Your Turn

If you have comments or suggestions about *SAS® Solutions Adapter 1.3 for SAP: User's Guide*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: **yourturn@sas.com**

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: **suggest@sas.com**