

# **SAS<sup>®</sup> Financial Management Adapter 5.3 for SAP User's Guide**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Financial Management Adapter 5.3 for SAP: User's Guide*. Cary, NC: SAS Institute Inc.

**SAS® Financial Management Adapter 5.3 for SAP: User's Guide**

Copyright © 2012, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

**For a hard-copy book:** No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**For a Web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

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

October 2012

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/publishing](http://support.sas.com/publishing) or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

---

# Contents

<b>Chapter 1 • Introduction</b>	<b>1</b>
Introduction	1
Job Organization and Names	2
References	2
<b>Chapter 2 • SAP Administration Tasks</b>	<b>3</b>
Configure SAS/ACCESS Interface to R/3 Software	3
Define User Permissions	3
<b>Chapter 3 • Installation and Configuration</b>	<b>5</b>
Overview	5
Create Directories	6
Define Shortcuts	6
Set Up for Internationalization	6
Define the Environment	7
Modify the Preprocessing Files	11
<b>Chapter 4 • Initialization Jobs</b>	<b>19</b>
Introduction to Initialization Jobs	19
fmsadpt_I0050_Initialize_Blank_Business_Area_Table	19
<b>Chapter 5 • Extraction Jobs</b>	<b>21</b>
Extraction Jobs	22
Job Group: E00 Extraction of T Tables	22
Job Group: E01 Extraction of Master Tables	27
Job Group: E02 Extraction of Transaction Tables	29
Job Group: E03 Extraction of Summary Tables	32
Job Group: E04 Extraction of Hierarchies and Structures	33
Job Group: E05 Extraction of Miscellaneous Tables	34
<b>Chapter 6 • Transformation Jobs</b>	<b>37</b>
Introduction to Transformation Jobs	39
Job Group: T0000 SAS Supplied Tables	39
Job Group: T0001 Transformations for Common Tables	40
Job Group: T0002 Time Dimension	42
Job Group: T0003 Exchange Rates	46
Job Group: T0004 Analysis Dimension	47
Job Group: T0005 Account Dimension	48
Job Group: T0006 Cost Center Dimension	56
Job Group: T0007 Profit Center Dimension	58
Job Group: T0008 Currency Dimension	61
Job Group: T0009 Financial Documents	61
Job Group: T1000 External Org Dimension	66
Job Group: T1100 Internal Org Dimension	68
Job Group: T1200 Data Validation	73
<b>Appendix 1 • Tables Extracted from SAP</b>	<b>75</b>
<b>Appendix 2 • Transformations for SAP</b>	<b>77</b>
Overview of Transformations Provided for SAP	77

Add Standard Dimension Rows and Add Standard Dimension	
ASSOC Rows Transformations . . . . .	78
Add Parent to Hierarchy Table Based on Level and Position Transformation . . . . .	80
Format Generator . . . . .	82
User Written Code for Multiple Input/Output Tables Transformation . . . . .	87
Changed Data Extraction Using Date or Time-Stamp and Overlap Transformation . . .	88
Date Join Transformation . . . . .	89
Period Consolidation Transformation . . . . .	90
Repeat Record for Date Range Overlap Transformation . . . . .	91
Split the NLS Data Transformation . . . . .	93
Convert Number Order to Integer Value Transformation . . . . .	94

## Chapter 1

# Introduction

---

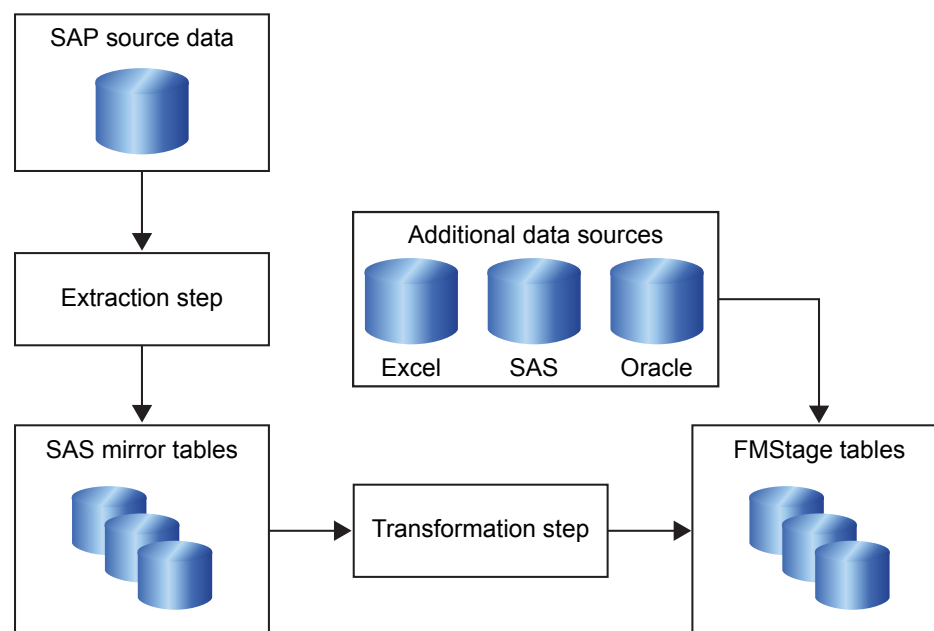
<b>Introduction</b> .....	<b>1</b>
<b>Job Organization and Names</b> .....	<b>2</b>
<b>References</b> .....	<b>2</b>

---

## Introduction

The SAS Financial Management Adapter for SAP enables you to extract data from your source SAP system, transform the data, and load it to the staging area of SAS Financial Management. The extraction and transformation jobs are run from SAS Data Integration Studio. The figure below illustrates the data flow.

**Figure 1.1** Data Flow Diagram



After the data is in the staging area, it can be processed using SAS Data Integration Studio jobs as described in the *SAS Financial Management: Data Administration Guide*.

---

## Job Organization and Names

The SAS Financial Management Adapter for SAP includes a collection of ETL jobs that extract, transform, and load data from standard SAP tables into standard SAS tables. Jobs are organized by type and job group.

- extraction jobs

Extraction job names begin with the letter E and a two-digit number that indicates the job group.

- transformation jobs

Transformation job names begin with the letter T and a four-digit number that indicates the job group.

Extraction jobs should be run before transformation jobs.

Within a group, job numbers typically imply the order of execution.

---

## References

Refer to the following documents to assist with your installation and configuration of the SAS Financial Management Adapter for SAP:

- SAS Financial Management documentation, available at the following site:  
<http://support.sas.com/documentation/onlinedoc/fm>.

References include the *SAS Financial Management: System Administrator's Guide* and the *SAS Financial Management: Data Administrator's Guide*.

*Note:* This site is password-restricted. You can find the user name and password in the preinstallation checklist or by calling Technical Support.

- SAS Intelligence Platform documentation, available at <http://support.sas.com/93administration>.

References include the *SAS Intelligence Platform: Security Administration Guide*.

## Chapter 2

# SAP Administration Tasks

---

<b>Configure SAS/ACCESS Interface to R/3 Software . . . . .</b>	<b>3</b>
<b>Define User Permissions . . . . .</b>	<b>3</b>

---

## Configure SAS/ACCESS Interface to R/3 Software

The SAS Financial Management Adapter for SAP uses SAS/ACCESS Interface to R/3. This software requires configuration before use. See the *Post-Installation Instructions for SAS/ACCESS Interface to R/3* in your installation package.

---

## Define User Permissions

The SAP administrator must define and provide appropriate authorizations to SAP users so that they can log on and extract data using the SAS Financial Management Adapter for SAP.

The SAS administrator must configure the SAS Financial Management Adapter for SAP so that only the desired SAP users can log on. See [“Define Users” on page 7](#).





## Chapter 3

# Installation and Configuration

---

<b>Overview</b> .....	<b>5</b>
<b>Create Directories</b> .....	<b>6</b>
<b>Define Shortcuts</b> .....	<b>6</b>
<b>Set Up for Internationalization</b> .....	<b>6</b>
<b>Define the Environment</b> .....	<b>7</b>
Prepare to Import the Metadata for SAS Financial Management Adapter for SAP .....	7
Define Users .....	7
Define an SAP Server, If Necessary .....	8
Test the Connection to the SAP Server .....	9
Import the Adapter Metadata .....	9
Modify the fmsadpt_T110080_Load_stage_Internal_Org_Assoc_Table Job Properties .....	9
(UNIX Only) Modify the fmsadpt_I0050_Initialize_Blank_Business_Area_Table Job Properties .....	10
(UNIX Only) Modify the fmsadpt_E02000_Extract_Financial_Documents_from_BK PF_BSEG Job Properties .....	10
<b>Modify the Preprocessing Files</b> .....	<b>11</b>
Overview .....	11
The preprocglobal.sas File .....	11
The preprocparms12.sas File .....	11
The preprocparms13.sas File .....	17

---

## Overview

This chapter explains how to install the SAS Financial Management Adapter for SAP.

*Note:* The SAS Financial Management Adapter for SAP uses SAS/ACCESS Interface to R/3. This software requires configuration before use. For detailed installation and configuration instructions, see the *SAS/ACCESS Interface to R/3: User's Guide* that is included in your installation package.

---

## Create Directories

On the data tier of your installation, create the following physical directories under the *SAS-config-dir*\Config\Lev1\SASApp\Data directory:

- FinancialMgmtAdapterSAP
- FinancialMgmtAdapterSAP\Formats
- FinancialMgmtAdapterSAP\IntermediateData
- FinancialMgmtAdapterSAP\R3Mirror
- FinancialMgmtAdapterSAP\Reports
- FinancialMgmtAdapterSAP\AdminData

---

## Define Shortcuts

In the sasv9.cfg file in the *SAS-config-dir*\Config\Lev1\SASApp directory, define the following shortcuts:

- ADPTFMSINSTALL: shortcut to the **SASHOME** \SASFinancialManagementAdapterforSAPData\5.3\CustomSASCode directory
- ADPTFMSDATA: shortcut to the *SAS-config-dir*\Config\Lev1\SASApp\Data\FinancialMgmtAdapterSAP directory.

For example, on a Windows machine with **Program Files** and the SAS configuration directories on the C drive, you would append the following lines to the sasv9.cfg file:

```
-SET ADPTFMSINSTALL "C:\Program Files\SASHome\
SASFinancialManagementAdapterforSAPData\5.3\CustomSASCode\"
-SET ADPTFMSDATA "C:\SAS\Config\Lev1\SASApp\Data\FinancialMgmtAdapterSAP"
```

*Note:* Line break added for readability. *SAS-config-dir* is the path to the SAS configuration directory. On Windows, a typical path is **C:\SAS\Config**. On UNIX, a typical path is **/usr/local/sas/config**.

---

## Set Up for Internationalization

Configure the SAS environment so that the encoding matches what is on your SAP server.

Perform the following steps:

*Note:* The examples below assume that the SAP server is a Unicode server. Make the appropriate substitutions if your SAP server uses different encoding.

1. (Windows only) Modify the sasv9.cfg file to specify the SAP encoding.

For example, on a Windows system with SASHome installed in **C:\Program Files**, do the following:

- a. Open the sasv9.cfg file in **C:\Program Files\SASHome\SASFoundation\9.3** for editing.

- b. Find this line:

```
-config "C:\Program Files\SASHome\SASFoundation\9.3\nls\en\sasv9.cfg"
```

- c. Replace it with the following line:

```
-config "C:\Program Files\SASHome\SASFoundation\9.3\nls\u8\sasv9.cfg"
```

2. (UNIX only) Create a symbolic link from **!sasroot\sas** to **\bin\sas\_u8**.
3. Restart the SAS Services.
4. Use the appropriate tables in the SAS Supplied FM library (libref SASSUPFM) for your environment.

For example, if you use the Adapter with euc-cn encoding to load the data from SAP to SAS Financial Management, ensure that the tables supplied by SAS, such as SAS\_CURRENCY, contain Chinese text. This is necessary because the Adapter jobs load the staging tables with the currency text as defined in the table SAS\_CURRENCY. If you use the SAS\_CURRENCY table with English text, then the stage currency table will contain currency descriptions in English text.

This guideline also applies to SAS\_COUNTRY\_ISO3166, another table in the SAS Supplied FM library.

The Adapter is packaged with predefined SAP metadata that is suitable for both Unicode and non-Unicode systems. As a result, you might see warning messages that are related to the base and data length mismatch in the Adapter jobs. This depends on the encoding option that you have used. However, warning messages do not appear if you use Unicode encoding or another double-byte encoding such as euc-cn.

---

## Define the Environment

### ***Prepare to Import the Metadata for SAS Financial Management Adapter for SAP***

Before importing the metadata for the SAS Financial Management Adapter for SAP, you must prepare your system environment by completing the following tasks:

- In SAS Management Console, define user and SAP login details.
- In SAS Management Console, define an SAP server.
- In SAS Data Integration Studio, test and verify the resulting connection.

### ***Define Users***

To define users for SAS Financial Management Adapter for SAP, complete the following steps:

1. Log on to SAS Management Console.
2. In the **User Manager** plug-in, define a new group called **SAP Group**.

3. Add SAS Trusted User to this group.
4. On the **Accounts** tab, create a login for this group, with these properties:

Property	Description
<b>User ID</b>	User ID for the SAP Server.
<b>Password</b>	Password for the SAP Server.
<b>Authentication Domain</b>	SAPAuth If necessary, click <b>New</b> to create this domain.

These users are necessary for installing and testing the software. After installation and testing, add any other users who will use the SAP adapter to this group.

All users of the SAP adapter must belong to the SASSDM MySQL Users Group in addition to the SAP Group. On Windows, users need the local **Log on as a batch job** right on the machine where SAS Data Integration Studio resides. For more information about users and groups, see [“References” on page 2](#).

*Note:* Users also need Read/Write/Modify permission for the **SASApp\Data** directory and its subdirectories. For more information, see “Secure Your Installation” in the *SAS Financial Management: System Administrator’s Guide*.

### Define an SAP Server, If Necessary

If an SAP Server is already defined in SAS Management Console, skip to the next section.

Otherwise, follow these steps:

1. Right-click the **Server Manager** plug-in and select **New Server**.
2. Select **SAP server**, under the **Enterprise applications servers** folder.
3. On the server properties page, enter the version numbers for your SAP software. These values are optional.
4. On the connection properties page, enter the following values:

Field	Description
<b>Authentication type</b>	Select <b>User/Password</b> .
<b>Authentication domain</b>	<b>SAPAuth</b>
<b>Client</b>	Enter the three-digit SAP identification number for the SAP system with which you are connecting.
<b>Language</b>	Specify the SAP language.
<b>Connection Type</b>	Select the connection type <b>Application server</b> that corresponds to a normal SAP GUI connection for your system.  Click <b>Options</b> . Enter the server name and system number for the SAP application server.

Field	Description
<b>Advanced Options</b>	If you have defined batch servers or processes, click <b>Advanced Options</b> and specify those options.

For more information, see the online Help for the wizard.

### ***Test the Connection to the SAP Server***

Log on to SAS Management Console as an SAP user. Select **Tools** ⇒ **Extract from R/3**.

In the window that opens, click **Test SAP Server Connection** and follow the instructions.

### ***Import the Adapter Metadata***

The adapter's metadata packages are located in the **SASHome\SASFinancialManagementAdapterforSAPData\5.3\SASPackages** directory. SASHOME is the name of the folder in which you installed SAS. On Windows, the typical location is **C:\Program Files\SASHOME**. On UNIX, the typical location is **/usr/local/sas**.

On the **Folders** tab of SAS Management Console or SAS Data Integration Studio, import the packages in the following order:

1. Import the SASFMAdapterforSAP\_DIS\_Transformations.spk package into the **/System/Applications** folder.
2. Import the SASFMAdapterforSAP\_DIS\_Tables.spk package into the **/Products** folder.
3. Import the SASFMAdapterforSAP\_DIS\_Jobs.spk into the **/Products** folder.

### ***Modify the fmsadpt\_T110080\_Load\_stage\_Internal\_Org\_Assoc\_Table Job Properties***

Follow these steps:

1. In SAS Data Integration Studio, open the following folder: **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/2 Transformation Jobs - R3Mirror to StageFM/T1100 Internal Org Dimension**.
2. Right-click the fmsadpt\_T110080\_Load\_stage\_Internal\_Org\_Assoc\_Table job and choose **Properties**.
3. Double-click the Data Validation transformation to open its properties.
4. Select the **Error and Exception tables** tab.
5. Select the **Create error table** and **Create exception table** check boxes.
6. Click **OK**.

**(UNIX Only) Modify the  
fmsadpt\_I0050\_Initialize\_Blank\_Business\_Area\_Table Job  
Properties**

For UNIX systems only, follow these steps:

1. In SAS Data Integration Studio, open the following folder: **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/0 Data Environment Initialization.**
2. Right-click the fmsadpt\_I0050\_Initialize\_Blank\_Business\_Area\_Table DIStudio job and choose **Properties**.
3. Select the **Precode and Postcode** tab.
4. Search **Precode** for the following line:

```
libname fmamacro "!SASROOT\fmadaptsap\cmacros";
```

Change the line as follows:

```
libname fmamacro "!SASROOT/cmacros/fmadaptsap";
```

5. Click **OK**.

**(UNIX Only) Modify the  
fmsadpt\_E02000\_Extract\_Financial\_Documents\_from\_BKPF\_BSEG  
Job Properties**

For UNIX systems only, follow these steps:

1. In SAS Data Integration Studio, open the following metadata folder: **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/1 Extraction Jobs from SAP to R3Mirror/E02 Extraction of Transaction Tables.**
2. Right-click the fmsadpt\_E02000\_Extract\_Financial\_Documents\_from\_BKPF\_BSEG DIStudio job and choose **Properties**.
3. Select the **Precode and Postcode** tab.
4. In the **Precode** area, click the Open (Open folder) icon
5. Select the **Attach to Code** radio button.
6. Add this path if it is not already listed: **!ADPTFMSINSTALL/preprocglobal.sas.**
7. Click **OK**.
8. Click **OK** for the properties window.

Repeat steps 1-8 for the following jobs:

- In the **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/2 Transformation Jobs - R3Mirror to StageFM/T0000 SAS Supplied Tables** folder: the fmsadpt\_T000030\_Load\_stage\_country\_table job
- In the **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/2 Transformation Jobs - R3Mirror to StageFM/T0006 Cost**

**Center Dimension** folder: the

fmsadpt\_T000600\_Transform\_Cost\_Center\_Hierarchy\_Over\_Years job

- In the **/Products/SAS Financial Management Adapter for SAP/5.3 Jobs/2 Transformation Jobs - R3Mirror to StageFM/T1000 External Org Dimension** folder: the fmsadpt\_T100020\_Load\_stage\_External\_Org job

*Note:* StageFM is the SAS Financial Management staging library.

---

## Modify the Preprocessing Files

### Overview

The SAS Financial Management Adapter for SAP uses the following preprocessing files:

- preprocglobal.sas
- preprocparms12.sas
- preprocparms13.sas

On Windows, these files are located in the **\SASHome**

**\SASFinancialManagementAdapterforSAPData\5.3\CustomSASCode** directory.

On UNIX, they are located in the **/SASHome/**

**SASFinancialManagementAdapterforSAPData/5.3/CustomSASCode** directory.

*Note:* Be sure to make backup copies of these files before changing them.

### The preprocglobal.sas File

The preprocglobal.sas code serves as the preprocessing step in each job. It includes both the preprocparms12.sas and preprocparms13.sas programs. These programs initialize global macro variables that are used by SAS Financial Management Adapter for SAP jobs.

### The preprocparms12.sas File

Modify the preprocparms12.sas file, using the following table and the file comments as guidelines.

**Table 3.1** Modifications to *preprocparms12.sas*

Macro Variable or Statement	Description
DDS_SOURCE_SYSTEM_ID	<p>The macro variable DDS_SOURCE_SYSTEM_ID is a three-character ID that tells the SAS Financial Management staging area 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 system ID.</p> <p>This variable populates the SOURCE_SYSTEM_CD column that occurs in multiple tables of the SAS Financial Management staging area. If data originates from multiple SAP systems, multiple extracts are necessary. Each extract has its own preprocparms12.sas file and unique value set for the variable DDS_SOURCE_SYSTEM_ID.</p>
<code>options debug=dbms_select;</code>	<p>Uncomment this statement to log detailed information about queries that are passed directly to SAP. Generally, the SAP queries section of the log is not needed unless you think that SAP server-side joins are taking longer than expected to run. This section provides additional tracking that might help solve problems.</p>
SPRAS LANG SAS_LANG_FOR_SDM DEFAULT_SPRAS DEFAULT_LANG	<p>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. Use single quotation marks around the value for SPRAS. Quoted and unquoted versions exist for easy inclusion into code and for making the code easy to understand and read. Use <a href="#">Table 3.2 on page 16</a> to determine the valid language value for your local environment.</p> <p>The macro variable SAS_LANG_FOR_SDM sets the main language that SAS uses. The two-character language code must be lowercase and in the table SASHELP.LANGUAGE that is supplied by SAS.</p> <p>DEFAULT_LANG is a two-digit ISO code that sets the default language flag in the STAGE_CODE_LANGUAGE table.</p>
CLIENT	<p>The value of the CLIENT macro variable is the SAP client (column MANDT) that is part of the SAP R/3 logon in most cases.</p>
DEFAULT_CURR CURRENCY_ASSOC_TYPE_CD DEFAULT_DATE DEF_N_FLAG DEFAULT_COUNTRY_CD	<p>Update as needed. Follow the comments in the file.</p>
EMAILERROR	<p>The e-mail contact for the error reports section of preprocparms12.sas includes the e-mail address of the contact who receives error notifications. These error notifications might be sent for extractions or transformation flows that use a publish-to-e-mail transformation. The initial value is obtained from the parameters that are supplied in the import steps.</p>
COST_CENTER_HIER_ROOT PROFIT_CENTER_HIER_ROOT	<p>By default, the SAS Financial Management Adapter for SAP extracts the standard hierarchies for the cost center and the profit center. You can select a different 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.</p>



Macro Variable or Statement	Description
WANTED_COMPANY_CODES	<p>To extract data from SAP for only selected companies, specify the WANTED_COMPANY_CODES macro variable to select company codes. The list of valid company code values is in the SAP R/3 table TKA02. The companies that you specify need to belong to controlling areas that are selected by the macro variable WANTED_CONTROLLING_AREAS.</p>
CONVERT_CURRENCY_TO_EURO PRE_EURO_CURRENCY	<p>The SAS Financial Management Adapter for SAP can convert employee compensation amounts that are paid to employees in pre-euro currencies, so that all 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.</p> <p>The macro variable CONVERT_CURRENCY_TO_EURO must have a value of YES or NO.</p> <p>The macro PRE_EURO_CURRENCY must be set to one of the standard international currency codes. These codes are three characters in length. The macro PRE_EURO_CURRENCY is used only when the macro CONVERT_CURRENCY_TO_EURO is set to YES.</p> <p><i>Note:</i> The SAS Financial Management Adapter for SAP does not use the euro currency conversion macro variables in areas other than the calculation of employee compensation. However, you can use these macros elsewhere.</p>
KTOPL _KTOPL	<p>The SAS Financial Management Adapter for SAP uses one chart of accounts at a time. You specify this chart of accounts in the macro variables KTOPL and _KTOPL.</p> <p>A chart of accounts in SAP is a list of all general ledger accounts that one or more company codes use. For each general ledger account, the chart of accounts contains the account number and the account name. It also contains information that controls how an account functions, and how the account is created in a company code.</p> <p>To specify a correct value for the chart of accounts, see 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 use a WHERE clause to select the appropriate language. The KTOPL value must appear in the KTOPL column of the T001 table.</p> <p>After you have determined the correct variable for the chart of accounts, enter the value. Enclose the KTOPL value in single quotation marks. Do not use quotation marks for the _KTOPL value.</p> <p>If you need more than one chart of accounts for your environment, set up additional instances of the SAS Financial Management Adapter for SAP and specify the relevant values. Be sure to select unique values for the data source macro variable DDS_SOURCE_SYSTEM_ID.</p>
SAKLN	<p>The value for SAKLN determines the number of digits in the account number.</p> <p>If you do not know this value, see the contents of the SAP R/3 table T004. The SAKLN column in this table corresponds to the chart of accounts KTOPL value previously chosen.</p> <p>Remember that valid account numbers often contain leading zeros. You can delete these leading zeros from the account digit value.</p>

Macro Variable or Statement	Description
OPERATING_CONCERN OPERATING_CONCERN_COUNTRY_CD	<p>Specify the operating concern in the macro variable OPERATING_CONCERN and a corresponding country code in the macro variable .</p> <p>To specify a correct value for the operating concern and country code, see the contents of the SAP R/3 table TKEB. Descriptions based on language dependencies are available in the TKEBT table.</p>
EXT_ORG_ASSOC_TYPE_CD	<p>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.</p> <p>The default SAP value for EXT_ORG_ASSOC_TYPE_CD is <b>A</b>. For other valid values, see the contents of the SAP R/3 tables, THITT and TLHITT.</p>
WANTED_CONTROLLING_AREAS CONTROLLING_AREA_COUNTRIES	<p>You can use controlling areas to form the second level of the internal organizational hierarchy. You can use them to select cost-center and profit-center standard hierarchies. You must specify the corresponding country for each controlling area.</p> <p>To set the variables for the 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 specifying the list, make sure that the controlling areas are all in the same operating concern. For example, select values for KOKRS only where ERKRs='&amp;OPERATING_CONCERN'.</p> <p>After you specify the required controlling areas, specify the corresponding country for each controlling area. Each country is identified with a three-digit ISO code in the SASHELP.SAS_COUNTRY table.</p> <p>Use the macro variable CONTROLLING_AREA_COUNTRIES to specify the corresponding country for each controlling area. The first country code must correspond to the first controlling area. The second country code must correspond with the second controlling area, and so on.</p>
RLDNR RRCTY RVERS	<p>Use the macro variables RLDNR, RRCTY, and RVERS to subset the data that is extracted from the SAP R/3 table GLT0. The GLT0 table contains figures that are summed by transaction.</p> <p>Although the default settings are generally acceptable, check them against valid values.</p> <ul style="list-style-type: none"> <li>For valid ledger values, see the contents of the SAP R/3 tables T881 and T881T.</li> <li>For the record type value, 0 is the typical value for actual data.</li> <li>For valid version values, see the contents of the SAP R/3 table T894. The version descriptions are in the T894T table.</li> </ul>
PERIOD_TYPE_CD_FOR_EXRATES	<p>Use the macro variable PERIOD_TYPE_CD_FOR_EXRATES to specify the period type that is associated with exchange rates. A list of valid values is in the SASHELP.SAS_PERIOD_TYPE table that is provided by SAS.</p>

Macro Variable or Statement	Description
<b>VERSN</b> <b>_VERSN</b>	<p>The SAS Financial Management Adapter for SAP handles one financial statement at a time. You must specify this financial statement in the macro variables <b>VERSN</b> and <b>_VERSN</b>.</p> <p>To specify a correct value for the financial statement, see the contents of the SAP R/3 table T011. You can use the T011 table for text descriptions as well. You can use a WHERE clause in the T011 table to select the KTOPL value. Using these references, choose and enter one of the matching <b>VERSN</b> values.</p> <p>Enclose the <b>VERSN</b> value in single quotation marks. Do not use quotation marks around the <b>_VERSN</b> value.</p> <p>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. You can add a common root by specifying a description for it in the macro variable <b>FINANCIAL_STATEMENT_ROOT_TXT</b>. However, we recommend that you leave this macro variable blank.</p>
<b>USE_FISCAL_YR_VARIANT_IN_TIME</b>	<p>SAP enables companies to have multiple fiscal year variants, although most companies use only one variant. 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 period IDs and descriptions that do not include the variant information.</p> <p>The macro variable <b>USE_FISCAL_YR_VARIANT_IN_TIME</b> controls behavior based on the following two values:</p> <ul style="list-style-type: none"> <li>• <b>N</b>: Do not use the variant.</li> <li>• <b>Y</b>: Include the variant information in the period IDs and descriptions.</li> </ul>
<b>FINYEAR_STARTMONTH</b> <b>_START</b> <b>_STARTFINYEAR</b>	<p>Use the macro variables <b>_START</b> and <b>_STARTFINYEAR</b> to set the starting date for the extraction financial transactions. These values help reduce the amount of data that is extracted in the initial load by not extracting financial transactions with dates before the specified date.</p> <p>Set the extraction variables based on your environment:</p> <ul style="list-style-type: none"> <li>• <b>FINYEAR_STARTMONTH</b>: the number of the month in which the financial year starts. January is represented by 1, February by 2, and so on.</li> <li>• <b>_START</b>: the first day of your organization's financial year. This value must be in SAS DATE9 format.</li> <li>• <b>_STARTFINYEAR</b>: your organization's financial year. For example, if your organization's financial year for 2012 begins on 01 December 2011, set <b>_STARTFINYEAR</b> to 2012. Choose a value that reflects the year after which detailed transaction data is to be extracted from SAP.</li> </ul>
<b>today=today();</b>	<p>The current fiscal year and time period are derived from the current date.</p> <p>For testing purposes, you can set <b>today</b> to a fixed date.</p>

Macro Variable or Statement	Description
OVERLAP	<p>The macro variable OVERLAP enables you to re-extract data that has changed during the overlap since the last extraction of financial transactions.</p> <p>The OVERLAP value is specified in number of days. Do not use a value less than 2. That is the minimum time needed to handle overnight processing and time zone differences. The default value is 4 days.</p>

The following table contains the language codes for preprocparms12.sas.

**Table 3.2** Language Codes

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 preprocparms13.sas File***

The preprocparms13.sas program enables you to specify parameters that identify leaf nodes and non-leaf nodes in the cost center and profit center standard hierarchies for your site. This program is included in the preprocglobal.sas program that is specified in the preprocessing section of every SAS Financial Management Adapter for SAP extract and transformation.

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



## Chapter 4

# Initialization Jobs

---

Introduction to Initialization Jobs . . . . .	19
fmsadpt_I0050_Initialize_Blank_Business_Area_Table . . . . .	19

---

## Introduction to Initialization Jobs

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

Currently, there is only one initialization job.

---

## fmsadpt\_I0050\_Initialize\_Blank\_Business\_Area\_Table

The SAS Financial Management Adapter for SAP requires business area descriptions when creating default internal organization dimensions. However, descriptions might be missing in the transaction data from SAP, because SAP allows blank descriptions in business area text tables.

The fmsadpt\_I0050\_Initialize\_Blank\_Business\_Area\_Table initialization job assigns text descriptions to business areas that do not have descriptions in the transaction data from SAP. When a business area description is missing in the transaction data from SAP, the initialization job assigns a default text description **Business Area not specified**. The source code for this job is in i0050\_set\_blank\_busarea.sas.

*Note:* This initialization job is site-dependent and must be reviewed for each local implementation. You can substitute wording or use another language, as needed.





## Chapter 5

# Extraction Jobs

---

<b>Extraction Jobs</b>	<b>22</b>
<b>Job Group: E00 Extraction of T Tables</b>	<b>22</b>
Overview	22
fmsadpt_E00010_Extract_SAP_General_T_Tables	22
fmsadpt_E00011_Create_Formats_from_SAP_FI_Tables	23
fmsadpt_E00020_Extract_SAP_FI_T_Tables	24
fmsadpt_E00030_Extract_SAP_Fiscal_Period_T_Tables	24
fmsadpt_E00031_Create_Formats_from_SAP_Fiscal_Period_Tables	25
fmsadpt_E00040_Extract_Currency_T_Tables	25
fmsadpt_E00050_Extract_Account_T_Tables	25
fmsadpt_E00060_Extract_Operating_Concern_T_Tables	26
fmsadpt_E00070_Extract_Addl_Tables_for_Formats	26
fmsadpt_E00080_Extract_New_GL_Control_Table	26
<b>Job Group: E01 Extraction of Master Tables</b>	<b>27</b>
Overview	27
fmsadpt_E01000_Extract_SAP_Master_Code_Tables	27
fmsadpt_E01010_Extract_SAP_Master_Description_Tables	27
fmsadpt_E01050_Create_Formats_from_SAP_Tables	28
<b>Job Group: E02 Extraction of Transaction Tables</b>	<b>29</b>
Introduction	29
fmsadpt_E02000_Extract_Financial_Documents_from_BKPF_BSEG (Initial)	29
fmsadpt_E02005_Extract_Intermediate_Table_merging_BKPF_BSEG	30
fmsadpt_E02010_Extract_Intermediate_Table_from_BKPF_BSEG (Changes)	30
<b>Job Group: E03 Extraction of Summary Tables</b>	<b>32</b>
Introduction	32
fmsadpt_E03000_Extract_GL_Balance_from_New_GL_Tables (New GL)	32
fmsadpt_E03010_Extract_Split_Docs_from_FAGLFLEXA (New GL)	32
fmsadpt_E03020_Extract_GL_Balances_from_OLd_GL_Tables	33
<b>Job Group: E04 Extraction of Hierarchies and Structures</b>	<b>33</b>
Introduction	33
fmsadpt_E04010_Extract_Balance_Sheet_FSV_Structures	33
fmsadpt_E04020_Extract_Supplier_Hierarchy	33
fmsadpt_E04030_Extract_Customer_Hierarchy	34
<b>Job Group: E05 Extraction of Miscellaneous Tables</b>	<b>34</b>
Introduction	34
fmsadpt_E05000_Extract_Number_Ranges_from_NRIV	34
fmsadpt_E05010_Extract_State_Region_County_Data_from_T005U	34
fmsadpt_E05020_Extract_Cost_Elements_from_CSKB	35

---

## Extraction Jobs

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

**Simple complete extraction**

extract and load all rows of a table from the SAP server into a mirror SAS table.

Rerunning these extractions re-extracts the entire table and automatically retrieves all new information.

**Simple partial extraction**

extract and load a subset (not all rows and not all columns) of an SAP table into a partial mirror SAS table. Running these extractions again re-extracts the same logical parts of the table and automatically retrieves all relevant new information.

**Change data capture extraction**

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 extraction or a simple partial extraction. Change data capture extractions are important for large transaction tables. This extraction's complexity is not needed for small amounts of data.

Extraction done through SAP remote-callable function  
is the easiest way to extract SAP data in some cases.

---

## Job Group: E00 Extraction of T Tables

### Overview

This job group includes common jobs that extract data for SAS Financial Management. A mirror SAS table is created in the R3mirror library for each job given below.

### ***fmsadpt\_E00010\_Extract\_SAP\_General\_T\_Tables***

This job extracts general and financial tables from SAP R/3 that are small and are named with a T prefix. For each table, a mirror SAS table is created in the R3mirror library. Individual extractions are independent of each other.

This job extracts the following tables from SAP:

- T001: Company Codes
- T002: Language Keys (Component BC - 118)
- T002T: Language Key Texts
- 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 tables from the SAP server:

- T001
- T002
- T002T
- T005T
- T011
- T011T
- T880

This job creates the following output:

- Fmsr3mir.T001
- Fmsr3mir.T002
- Fmsr3mir.T002T
- Fmsr3mir.T005T
- Fmsr3mir.T011
- Fmsr3mir.T011T
- Fmsr3mir.T880

### ***fmsadpt\_E00011\_Create\_Formats\_from\_SAP\_FI\_Tables***

This job creates formats from the tables that are extracted in the E00010 job. Run this job immediately after the job E00010, so that the formats are as current as the tables on which they are based.

This job uses the following input tables from SAS:

- Fmsr3mir.T001: Company Codes
- Fmsr3mir.T002: Language Keys
- Fmsr3mir.T011: Financial Statement Versions

This job creates the following output formats:

- \$SAPTOSAS\_LANGUAGE
- \$company\_to\_country
- \$LOCALCR
- \$unassigned\_accounts\_parent
- \$COMPANY\_TO\_FY\_VARIANT
- \$BUKRS
- \$TRAD\_PTNR\_TO\_BUKRS
- \$COM2CAC

### **fmsadpt\_E00020\_Extract\_SAP\_FI\_T\_Tables**

This job extracts general and financial tables from SAP R/3 that are named with a T prefix. These tables are generally text description tables. Individual extractions are independent of each other.

This job extracts the following tables:

- T8G21: Splitting rule: (Item to be processed)
- TGSBT: Business Area Names
- TKA01: Controlling Areas
- TKA02: Controlling area assignment
- TKA00: Control parameters for controlling areas

This job uses the following input tables from the SAP server:

- T8G21
- TGSBT
- TKA00
- TKA01
- TKA02

This job creates the following output:

- Fmsr3mir.T8G21
- Fmsr3mir.TGSBT
- Fmsr3mir.TKA00
- Fmsr3mir.TKA01
- Fmsr3mir.TKA02
- \$CC2CA converts Company Code (BUKRS) to Controlling Area (KOKRS)

### **fmsadpt\_E00030\_Extract\_SAP\_Fiscal\_Period\_T\_Tables**

This job extracts financial tables from SAP R/3 that are named with a T009 prefix.

This job extracts the following tables:

- T009: Fiscal Year Variants
- T009B - Fiscal year variant periods
- T009C - Period names
- T009T - Fiscal year variant names

This job uses the following input:

- T009
- T009B
- T009C
- T009T

- &lang (set in the preprocessing step)

This job creates the following output:

- Fmsr3mir.T009
- Fmsr3mir.T009T
- Fmsr3mir.T009B
- Fmsr3mir.T009C

### ***fmsadpt\_E00031\_Create\_Formats\_from\_SAP\_Fiscal\_Period\_Tables***

This job creates formats from the tables that are extracted in the job E00030. Run this job immediately after the job E00030, so that the formats are as current as the tables on which they are based.

This job uses the following input:

- Fmsr3mir.T009C - Period names
- &lang (set in the preprocessing step)

This job creates the following output:

- \$PERIOD\_DESCRIPTION converts concatenated Fiscal Year Variant (PERIV) and Posting Period (POPER) to Period Name Long Text (LTEXT)
- \$PERIOD\_NAME output converts concatenated Fiscal Year Variant (PERIV) and Posting Period (POPER) to Period Name Short Text (KTEXT)

### ***fmsadpt\_E00040\_Extract\_Currency\_T\_Tables***

This job extracts the currency tables from SAP R/3 that are named with a TCUR prefix.

This job extracts the following tables:

- TCURR - Exchange Rates
- TCURT - Currency Code Names

This job uses the following input tables from the SAP server:

- TCURR
- TCURT

This job creates the following output:

- Fmsr3mir.TCURR
- Fmsr3mir.TCURT

### ***fmsadpt\_E00050\_Extract\_Account\_T\_Tables***

This job extracts general and financial tables from SAP R/3 that are named with a T030 prefix, and are related to accounts.

This job extracts the following tables:

- T030 - Standard Accounts Table
- T030C - Global Standard Account Table

This job uses the following input tables from the SAP server:

- T030
- T030C

This job creates the following output:

- Fmsr3mir.T030
- Fmsr3mir.T030C

### ***fmsadpt\_E00060\_Extract\_Operating\_Concern\_T\_Tables***

This job extracts operating concern tables from SAP R/3 that are named with a T prefix.

This job extracts the following tables:

- TKEB - Management for Operating Concerns
- TKEBT - Description of operating concern

This job uses the following input tables from the SAP server:

- TKEB
- TKEBT

This job creates the following output:

- Fmsr3mir.TKEB
- Fmsr3mir.TKEBT

### ***fmsadpt\_E00070\_Extract\_Addl\_Tables\_for\_Formats***

This job extracts miscellaneous tables from SAP R/3 that are required for formats, and have not already been extracted.

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

This job uses the following input:

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

This job creates the following output:

- Fmsr3mir.DD07T
- \$SHKZG uses Fmsr3mir.DD07T to create the format \$SHKZG. It converts values for domains. It converts single value or lower limit (DOMVALUE\_L) to short text for fixed values (DDTEXT). \$SHKZG is stored in the LIBRARY.FMT&LANG catalog, which is language dependent.

### ***fmsadpt\_E00080\_Extract\_New\_GL\_Control\_Table***

This job extracts the new GL Control table from SAP ECC.

This job extracts the following table:

- FAGL\_ACTIVEC - Activation of New General Ledger

This job uses the following input table from the SAP server:

- FAGL\_ACTIVEC

This job creates the following output:

- Fmsr3mir.FAGL\_ACTIVEC

---

## Job Group: E01 Extraction of Master Tables

### Overview

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

#### ***fmsadpt\_E01000\_Extract\_SAP\_Master\_Code\_Tables***

This job extracts all required master data tables from SAP R/3. For each table, a mirror SAS table is created in the R3mirror library.

This job extracts the following tables:

- CEPC - Profit Center Master Data Table
- CSKS - Cost Center Master Data
- SKA1 - G\_L Account Master (Chart of Accounts)
- KNA1 - General Data in Customer Master
- LFA1 - Vendor Master (General Section).

This job uses the following input tables from the SAP server:

- CEPC
- CSKS
- SKA1
- KNA1
- LFA1

This job creates the following output:

- Fmsr3mir.CEPC
- Fmsr3mir.CSKS
- Fmsr3mir.SKA1
- Fmsr3mir.KNA1
- Fmsr3mir.LFA1

#### ***fmsadpt\_E01010\_Extract\_SAP\_Master\_Description\_Tables***

This job extracts all required master data description tables from SAP R/3. For each table, a mirror SAS table is created in the R3mirror library.

This job extracts the following tables:

- CEPCT - Texts for Profit Center Master Data
- CSKT - Cost Center Texts
- SKAT - GL Chart of Accounts Description

This job uses the following input tables from the SAP server:

- CEPCT
- CSKT
- SKAT

This job creates the following output:

- Fmsr3mir.CEPCT
- Fmsr3mir.CSKT
- Fmsr3mir.SKAT

### ***fmsadpt\_E01050\_Create\_Formats\_from\_SAP\_Tables***

This job creates formats from the tables that are extracted in the previous jobs.

This job uses the following input:

- Fmsr3mir.DD07T
- Fmsr3mir.SKA1
- Fmsr3mir.T001
- Fmsr3mir.T002
- Fmsr3mir.T009C
- Fmsr3mir.T011
- Fmsr3mir.TKA02

This job creates the following output formats:

- \$LOCALCR.
- \$COMPANY\_TO\_FY\_VARIANT.
- \$company\_to\_country.
- \$TRAD\_PTNR\_TO\_BUKRS.
- \$BUKRS.
- \$COM2CAC.
- \$unassigned\_accounts\_parent.
- \$SAPTOSAS\_LANGUAGE.
- \$PERIOD\_DESCRIPTION.
- \$PERIOD\_NAME.
- \$responsible\_employee\_id\_to\_name.
- \$CC2CA. Converts Company Code (BUKRS) to Controlling Area (KOKRS).
- \$SHKZG uses the Fmsr3mir.DD07T table to create the format \$SHKZG. It converts values for domains: Single Value or Lower Limit (DOMVALUE\_L) to short text for



fixed values (DDTEXT). \$SHKZG is stored in the LIBRARY.FMT&LANG catalog, which is language-dependent.

- \$ACCBALS output uses the Fmsr3mir.SKA1 table to create the format \$ACCBALS. It also converts Values for General Ledger Account Number (SAKNR) to Indicator: Account is a Balance Sheet Account? (XBILK). This stores the format in the Library.fmt&\_ktopl&lang catalog that is for the chart of accounts, and is language-dependent.

This job calls the user-written code e01010\_responsible\_employee\_to\_id\_mapping.sas to create the format \$responsible\_employee\_id\_to\_name. Edit this program to include all responsible employees. Data should be based on the R3mirror CSKS data provided in this extract.

---

## Job Group: E02 Extraction of Transaction Tables

### Introduction

This job group includes jobs that extract transactional data for SAS Financial Management from SAP R/3.

### ***fmsadpt\_E02000\_Extract\_Financial\_Documents\_from\_BKPF\_BSEG (Initial)***

#### **Overview**

This job extracts financial documents from SAP R/3. The E00011 job must execute before the E02000 job. The E02005 job must run immediately after this job in order to create a consolidated table of financial document data.

This job extracts the following tables:

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

This job uses the following input:

- BKPF from the SAP server
- BSEG from the SAP server
- \$LOCALCR (created in the E00010 job)
- \$COM2CAC (created in the E00010 job)
- \$BUKRS (created in the E00010 job)
- \$TRAD\_PTNR\_TO\_BUKRS (created in the E00010 job)
- &\_STARTFINYEAR (set in preprocessing step)
- &OVERLAP (set in preprocessing step)

This job creates the following output:

- Fmsr3mir.BKPF
- Fmsr3mir.BSEG

**Additional Notes**

This job does a complete extraction. In addition, it executes the SAS macro %ADPT\_UPDATE\_BKPF\_DELTA to update the delta control date record for the BKPF table.

In general, this job should be run during the first run of the job suite. However, you might need to refresh data later if you think that there is incomplete data.

You must run the E02010 job in all subsequent cases. The initial extraction by the E02000 job completely extracts the financial document data from the BKPF and BSEG tables. The extraction depends on the date in the macro variable &\_STARTFINYEAR. However, in the subsequent run, the E02010 job saves time by extracting only changes made to the tables.

The initial extraction by the E02000 job works in two stages:

- The first stage extracts data from SAP to SAS data sets. This stage fully extracts the BSEG table, but only partially extracts the BKPF table. From the BKPF table, extraction is restricted to 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 in the job E2005. Because the BKPF table extraction is restricted by date, an SQL LEFT join is run. Information from the BSEG table is joined with information from the BKPF table.

**fmsadpt\_E02005\_Extract\_Intermediate\_Table\_merging\_BKPF\_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 E02000Initial 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 the output Intermed.Financial\_Documents.

**fmsadpt\_E02010\_Extract\_Intermediate\_Table\_from\_BKPF\_BSEG (Changes)****Overview**

This job extracts changes made to the financial document tables. See the related E02000 job for more information.

The following jobs must run before this job:

- E00011
- E02000

This job extracts the following tables:

- BKPF
- BSEG

This job uses the following input:

- BKPF from SAP server
- BSEG from SAP server
- \$LOCALCR (created in the E00010 job)
- \$COM2CAC (created in the E00010 job)
- \$BUKRS (created in the E00010 job)
- \$TRAD\_PTNR\_TO\_BUKRS (created in the E00010 job)
- &\_STARTFINYEAR (set in preprocessing step)
- &OVERLAP (set in preprocessing step)

This job creates the following output:

- Fmsr3mir.BKPF
- Fmsr3mir.BSEG
- Intermed.Financial\_Documents

### **Additional Notes**

This job includes E02010\_CHECK\_DELTA\_CONTROL\_DATE.SAS. In addition, it executes the SAS macro %ADPT\_DELTACTL2 to get the latest delta control date record for the BKPF table. This job includes E02010\_UPDATE\_BKPF\_DELTA\_CONTROL\_DATE.SAS, which calls the macro %ADPT\_UPDATE\_BKPF\_DELTA to update the delta control date record for the BKPF table.

If you require a complete refresh of financial document data, you can run the E02000 job. The E02010 job extracts only 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 Financial Management Adapter for SAP. Because a complete extraction takes many hours and uses a significant amount of SAP R/3 and network resources, it should not be done frequently.

There is no perfect approach to this task because SAP does not use a perfect process to create timestamps for the tables. However, you can use the following columns in the BKPF table to identify changes:

- CPUDT (Accounting Document Entry Date)
- AEDAT (Date of Last Document Change by Transaction)
- UPDDT (Date of Last Document Update)

In the same way, the BSEG table has no date-and-time stamps that are useful for extraction. The SAS Financial Management Adapter for SAP must identify the new and changed records in the heading table BKPF. It then extracts the matching records from the BSEG table. A server-side join extracts the new and changed records in the BKPF and BSEG tables.

This process restricts the extraction by checking the three dates in the columns previously listed against an extract date. The extract date is stored in a SAS data set named DELTACONTROL in the Admin library. This extract date helps identify

changed records. The SAS Financial Management Adapter for SAP uses this date to subset the extraction. It updates the latest extract date in a postprocessing step. The macro %ADPT\_DELTACTL updates DELTACONTROL.

You can use the macro variable &OVERLAP to specify a value of overlap days. This variable subtracts the specified number of days from the last extract date, and thus, increases the amount of data extracted.

---

## Job Group: E03 Extraction of Summary Tables

### Introduction

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

If SAP New General Ledger accounting is active on your SAP server, use the jobs E03000 and E03010. Otherwise, use E03020. The activation of the new general ledger can be seen from the R3 Mirror FAGL\_ACTIVEC - Activation of New General Ledger table in SAS Data Integration Studio.

*Note:* Run job fmsadpt\_E00080\_Extract\_New\_GL\_Control\_Table first.

### **fmsadpt\_E03000\_Extract\_GL\_Balance\_from\_New\_GL\_Tables (New GL)**

This job extracts account transaction summary tables from SAP R/3. For each table, a mirror SAS table is created in the R3mirror library.

This job extracts the table GLT0 GL account master record transaction.

This job uses the following input:

- FAGLFLEXT - General Ledger: Totals
- FAGLFLEXP - General Ledger: Plan Line Item
- FAGLFLEXA - General Ledger: Actual Line It

This job creates the following output:

- Fmsr3mir.FAGLFLEXT
- Fmsr3mir.FAGLFLEXP
- Fmsr3mir.FAGLFLEXA

### **fmsadpt\_E03010\_Extract\_Split\_Docs\_from\_FAGLFLEXA (New GL)**

This job extracts the split documents from SAP R/3 if the new GL is implemented. For each table, a mirror SAS table is created in the R3mirror library.

The output table Inter\_Split\_Documents is blank if the customer SAP system does not have split documents.

This job uses the following input:

- Fmsr3mir.FAGLFLEXA
- Fmsr3mir.T8G21

This job creates the output FMS\_Inter.Inter\_Split\_Documents.

### ***fmsadpt\_E03020\_Extract\_GL\_Balances\_from\_OLd\_GL\_Tables***

This job extracts account transaction summary tables from SAP R/3. For each table, a mirror SAS table is created in the R3mirror library.

This job extracts the table GLT0 - GL account master record transaction.

It uses the input table GLT0 from the SAP server.

It creates the output Fmsr3mir.GLT0.

## **Job Group: E04 Extraction of Hierarchies and Structures**

### ***Introduction***

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

### ***fmsadpt\_E04010\_Extract\_Balance\_Sheet\_FSV\_Structures***

This job extracts financial statement data from SAP R/3. The easiest way to approach this data is by calling the remote-enabled SAP functions. However, the SAP functions that produce this data are not remote-enabled. The SAS Financial Management Adapter for SAP provides functions that are remote enabled, and can call the relevant SAP functions.

This job uses the following input:

- The macro variables listed below:
  - \_VERSN Financial statement version from T011 without quotation marks
  - SPRAS Language code

This job creates the following output:

- Fmsr3mir.RF011P
- Fmsr3mir.RF011Q
- Fmsr3mir.RF011Z

This job calls E04010\_CALLRFC\_IMPORT\_BALANCE\_SHEET.SAS and submits a PROCCALLRFC to execute the following SAP functions:

- /SAS/FI\_IMPORT\_BAL\_SHEET\_POS
- /SAS/FI\_IMPORT\_BAL\_SHEET\_TEXT

### ***fmsadpt\_E04020\_Extract\_Supplier\_Hierarchy***

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

This job extracts the following tables:

- LFMH - Vendor hierarchy
- TLHITT - Vendor hierarchy category (description)

This job uses the following input tables from the SAP server:

- LFMH
- TLHITT

This job creates the following output:

- Fmsr3mir.LFMH
- Fmsr3mir.TLHITT

### ***fmsadpt\_E04030\_Extract\_Customer\_Hierarchy***

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

This job extracts the following tables:

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

This job uses the following input tables from the SAP server:

- KNVH
- THITT

This job creates the following output:

- Fmsr3mir.KNVH
- Fmsr3mir.THITT

---

## **Job Group: E05 Extraction of Miscellaneous Tables**

### ***Introduction***

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

### ***fmsadpt\_E05000\_Extract\_Number\_Ranges\_from\_NRIV***

This job extracts the table NRIV - Number Range Intervals.

This job uses the input table NRIV from the SAP server.

This job creates the output Fmsr3mir.NRIV.

### ***fmsadpt\_E05010\_Extract\_State\_Region\_County\_Data\_from\_T005U***

This job extracts the table T005U - Taxes: Region Key: Texts.

This job uses the input table T005U from the SAP server.

This job creates the output Fmsr3mir.T005U.

***fmsadpt\_E05020\_Extract\_Cost\_Elements\_from\_CSKB***

This job extracts the table CSKB - Cost Elements.

This job uses the input table CSKB from the SAP server.

This job creates the output Fmsr3mir.CSKB.





## Chapter 6

# Transformation Jobs

---

<b>Introduction to Transformation Jobs</b>	<b>39</b>
<b>Job Group: T0000 SAS Supplied Tables</b>	<b>39</b>
Overview	39
fmsadpt_T000010_Load_stage_gl_account_type_Table	39
fmsadpt_T000020_Load_stage_period_type_Table	39
fmsadpt_T000030_Load_stage_country_and_language_Tables	39
fmsadpt_T000040_Load_stage_gl_normal_bal_Table	40
fmsadpt_T000050_Load_stage_source_system_Table	40
<b>Job Group:T0001 Transformations for Common Tables</b>	<b>40</b>
Overview	40
fmsadpt_T000100_Load_stage_org_type_table	41
fmsadpt_T000110_Load_stage_code_language_table	41
fmsadpt_T000120_Load_stage_state_region_table	41
fmsadpt_T000140_Load_Inter_GL_Balances (New GL)	41
<b>Job Group: T0002 Time Dimension</b>	<b>42</b>
Overview	42
Input and Output (Tables, Formats, and Macro Variables)	42
Administration Factors	43
fmsadpt_T000200_Transform_Financial_Period_Info_from_SAP	44
fmsadpt_T000210_create_Inter_time_period_dimension_Table	45
fmsadpt_T000220_Load_stage_time_period_dimension_Tables	45
fmsadpt_T000230_create_Inter_controlling_area_fin_year_combo	45
<b>Job Group: T0003 Exchange Rates</b>	<b>46</b>
Overview	46
fmsadpt_T000300_Load_stage_currency_exch_rate_type_Table	46
fmsadpt_T000310_Load_stage_currency_exch_rate_set_table	46
fmsadpt_T000320_Load_stage_currency_exch_rate_src_Table	47
fmsadpt_T000330_Load_stage_currency_exch_rate_Table	47
<b>Job Group: T0004 Analysis Dimension</b>	<b>47</b>
Overview	47
fmsadpt_T000400_Load_stage_analysis_dimension_tables	48
<b>Job Group:T0005 Account Dimension</b>	<b>48</b>
Overview	48
Inputs and Outputs (Tables, Formats, and Macro Variables)	48
Financial Statement Nodes	50
Account Hierarchies and Numbering	50
Profit and Loss Accounts	50
Other Transactions and Account Considerations	51

fmsadpt_T000500_create_Inter_T011 .....	51
fmsadpt_T000505_create_Inter_Accounts_with_Text .....	51
fmsadpt_T000510_create_Inter_Account_Items_Transformed .....	51
fmsadpt_T000520_create_Inter_Account_Hierarchy .....	52
fmsadpt_T000525_create_Inter_Account_Balances .....	53
fmsadpt_T000528_create_Inter_Account_Type .....	53
fmsadpt_T000530A_create_Inter_GL_Account_with_SAP_Info .....	53
fmsadpt_T000530B_create_Inter_GL_Account_with_SAP_Info .....	54
fmsadpt_T000535_create_Inter_Account_statistic_Site_Specific .....	54
fmsadpt_T000540_Load_stage_gl_account_association_Tables .....	55
fmsadpt_T000550A_Load_stage_gl_account_tables_Site_Specific .....	55
fmsadpt_T000550B_Load_stage_gl_account_tables_Site_Specific .....	55
fmsadpt_T000560_Load_stage_source_gl_account_Table .....	56
<b>Job Group: T0006 Cost Center Dimension .....</b>	<b>56</b>
Overview .....	56
fmsadpt_T000600_Transform_Cost_Center_Hierarchy_Over_Years .....	57
fmsadpt_T000610_Load_stage_cost_center_dimension_Tables .....	57
fmsadpt_T000620_Transform_Add_Std_Dim_Values_cost_center .....	58
<b>Job Group: T0007 Profit Center Dimension .....</b>	<b>58</b>
Overview .....	58
fmsadpt_T000700_Transform_Profit_Center_Hierarchy_Over_Years .....	59
fmsadpt_T000710_Load_stage_profit_center_dimension_Tables .....	60
fmsadpt_T000720_Transform_Add_Std_Dim_Values_profit_center .....	60
<b>Job Group: T0008 Currency Dimension .....</b>	<b>61</b>
Overview .....	61
fmsadpt_T000800_Load_stage_currency_dimension_tables .....	61
<b>Job Group: T0009 Financial Documents .....</b>	<b>61</b>
Overview .....	61
Input and Output (Tables, Formats, and Macro Variables) .....	62
fmsadpt_T000900_create_Inter_Opening_Balances_and_Num_Ranges .....	63
fmsadpt_T000910_create_Inter_GLT0_Vertical_Summary .....	64
fmsadpt_T000920_create_Inter_Level_to_Account .....	64
fmsadpt_T000930_Load_stage_transaction_sum .....	65
<b>Job Group:T1000 External Org Dimension .....</b>	<b>66</b>
Overview .....	66
fmsadpt_T100000_Load_stage_External_Org_Assoc_Type .....	66
fmsadpt_T100010_Load_stage_External_Org_Assoc .....	67
Fmsadpt_T100020_Load_stage_External_Org .....	67
Fmsadpt_T100030_Load_stage_External_Org_Address .....	68
<b>Job Group:T1100 Internal Org Dimension .....</b>	<b>68</b>
Overview .....	68
fmsadpt_T110000_Load_stage_internal_org_assoc_type_tableJob .....	69
fmsadpt_T110010_create_Inter_Org_from_Operating_Concern .....	70
fmsadpt_T110020_create_Inter_Org_from_Controlling_Area .....	70
fmsadpt_T110030_create_Inter_Org_from_Company_Code .....	70
fmsadpt_T110040_create_Inter_Org_from_Business_Area .....	71
fmsadpt_T110050_create_Inter_Org_from_Cost_Center .....	71
fmsadpt_T110060_create_Inter_Org_Assoc_from_Cost_Center .....	72
fmsadpt_T110070_Load_stage_Internal_Org_Table .....	72
fmsadpt_T110080_Load_stage_Internal_Org_Assoc_Table .....	72
<b>Job Group: T1200 Data Validation .....</b>	<b>73</b>
Overview .....	73

fmsadpt_T120000_create_Inter_Financial_Summary .....	73
fmsadpt_T120010_compare_Financial_Docs_with_GLT0_Summary .....	73

---

## Introduction to Transformation Jobs

The SAS Financial Management Adapter for SAP uses several transformation jobs that transform data extracted from SAP R/3 into the structure that is needed for the staging area.

---

## Job Group: T0000 SAS Supplied Tables

### Overview

This job group includes the jobs that read the tables supplied by SAS. Then, the jobs load the tables into the staging area structure. You can also use this job to load records. When the follow-up jobs use these tables, the staging tables merge again with the original tables that are provided by SAS. Duplicate tables are discarded.

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

### ***fmsadpt\_T000010\_Load\_stage\_gl\_account\_type\_Table***

This job creates the structure of the staging area tables for the general ledger account types.

This job uses the input SAS\_GL\_ACCOUNT\_TYPE from SAS.

This job creates the output STAGEFM.GL\_ACCOUNT\_TYPE.

### ***fmsadpt\_T000020\_Load\_stage\_period\_type\_Table***

This job loads the STAGE\_PERIOD\_TYPE from the values supplied by SAS. The job adds SAP special periods and sets up records for SAP special periods to be appended to the SASSUPFM.PERIOD\_TYPE data set.

*Note:* SASSUPFM is the libref for the SAS Supplied FM library.

This job includes the following values:

- special periods, record type SP
- half months, record type HMO

This job uses the input SASSUPFM.SAS\_PERIOD\_TYPE.

This job creates the output STAGEFM.PERIOD\_TYPE.

This job uses the user-written source T000020\_ADD\_SAP\_SPECIAL\_PERIODS.SAS.

### ***fmsadpt\_T000030\_Load\_stage\_country\_and\_language\_Tables***

This job uses the following input:

- Fmsr3mir.T002
- Fmsr3mir.T002T
- Fmsr3mir.T005T
- sas\_country\_iso3166
- \$SAPTOSAS\_LANGUAGE
- SAS\_LANGUAGE\_ISO0639

This job uses the following formats:

- \$SAPTOSAS\_COUNTRY
- \$COUNTRY\_NAME

This job creates the following output:

- STAGEFM.COUNTRY
- STAGEFM.LANGUAGE

### ***fmsadpt\_T000040\_Load\_stage\_gl\_normal\_bal\_Table***

This job loads the values of GL\_NORMAL\_BAL. These values might not change, but additional language descriptions might be needed.

This job uses the input SASSUPFM.SAS\_GL\_NORMAL\_BAL.

This job creates the output STAGEFM.GL\_NORMAL\_BAL.

### ***fmsadpt\_T000050\_Load\_stage\_source\_system\_Table***

This job loads the stage table STAGE\_SOURCE\_SYSTEM from SAS\_SOURCE\_SYSTEM, which adds the value set by the global macro variable &DDS\_SOURCE\_SYSTEM\_ID to the list of valid values.

This job uses the following input:

- SASSUPFM.SAS\_SOURCE\_SYSTEM
- &DDS\_SOURCE\_SYSTEM\_ID

This job creates the output STAGEFM.SOURCE\_SYSTEM.

This job uses the user-written source t000050\_add\_sap\_source.sas.

---

## **Job Group:T0001 Transformations for Common Tables**

### ***Overview***

This job group includes jobs that transform data for SAS Financial Management. It also includes jobs that logically fit into more than one job group.

**fmsadpt\_T000100\_Load\_stage\_org\_type\_table**

This job loads the STAGE\_ORG\_TYPE table with constant values. The table below shows the STAGEFM.ORG\_TYPE values that are required for the internal and external organization dimensions.

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

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

The actual descriptions are in English. For descriptions in other languages, modify the source code in t000100\_populate\_org\_type.sas.

This job uses the following input:

- &LANG
- \$SAPTOSAS\_LANGUAGE

This job creates the output STAGEFM.ORG\_TYPE.

**fmsadpt\_T000110\_Load\_stage\_code\_language\_table**

This job uses the following input:

- Fmsr3mir.T002
- Fmsr3mir.T002T
- SAS\_LANGUAGE\_ISO0639

This job creates the output STAGEFM.CODE\_LANGUAGE.

The STAGEFM.CODE\_LANGUAGE table is used to split the non-default language texts into NLS dimension tables.

**fmsadpt\_T000120\_Load\_stage\_state\_region\_table**

This job populates the reference table STAGE\_STATE\_REGION by concatenating the country code and the region code.

This job uses the following input:

- Fmsr3mir.T005U
- \$SAPTOSAS\_LANGUAGE

This job creates the output STAGEFM.STATE\_REGION.

**fmsadpt\_T000140\_Load\_Inter\_GL\_Balances (New GL)**

This job loads the Inter\_GL\_Balances\_new table with old and new GL total tables. This job has been designed to get the GL totals from old and new GL tables.

This job uses the following input:

- FMA\_R3\_Mirror.GLTO
- FMA\_R3\_Mirror.FAGLFLEXT

This job creates the output Inter\_GL\_Balances\_new.

If you get an error stating that table GLT0 does not exist, and your server has New GL active, follow these steps in SAS Data Integration Studio:

1. Right-click the FMA\_R3\_Mirror GLT0 - GL account master record transaction table and select **Open**.
2. When you receive a prompt asking if you want to create the table, select **Yes**.
3. Rerun the job.

---

## Job Group: T0002 Time Dimension

### Overview

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

### Input and Output (Tables, Formats, and Macro Variables)

This job group uses the following input:

- Descriptions must be in the language that is specified by SPRAS or LANG, which are specified in the global parameters. SPRAS includes 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 the T001 table to select the financial period variant.

This job group uses the following SAP data sources as input:

- T001 - PERIV column contains the financial period variant
- T009 - Fiscal Year Variants
- T009B - Fiscal year variant periods
- T009T - Fiscal year variant names

This job group creates the following output:

- STAGEFM.TIME\_PERIOD
- STAGEFM.TIME\_PERIOD\_ASSOC
- STAGEFM.TIME\_PERIOD\_ASSOC\_TYPE
- This job group 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.

This job group includes the following SAS output formats:

- \$PERIOD\_DESCRIPTION
- Fmsr3mir.T009C (where SPRAS="&LANG")
- START=TRIM (left (PERIV!!poper))
- LABEL=TRIM (left (LTEXT))
- Time Period Descriptions (long text)
- \$PERIOD\_NAME
- Fmsr3mir.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 uses the following SAS formats:

- \$PERIOD\_DESCRIPTION
- \$PERIOD\_NAME

## **Administration Factors**

### **Account Hierarchies and Numbering**

It is important to understand hierarchy values and structure. PERIOD\_TYPE\_CD is one of the values in the PERIOD\_TYPE\_CD column of the SAS\_PERIOD\_TYPE table. There is 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, the separate hierarchies for different PERIV values are concatenated into STAGE\_TIME\_PERIOD\_ASSOC. If there is a fixed PERIV value, then no row exists 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 periods.

The time periods in the leaf layer of each hierarchy must have no gaps and no overlapping periods, except for special periods. The SAS Financial Management 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 closing the end of the year. Because SAP does not represent periods in a hierarchy, the SAS Financial Management 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, which consists of years. A level for quarters is added below the year level. Actual periods are assigned as children of one of the quarter levels.

### **Other Transactions and Account Considerations**

The LANGUAGE\_CD value must be set to one of the predefined language values in the SAS\_LANGUAGE table. You can specify the language value as needed. To do this, use the format \$SAPTOSAS\_LANGUAGE it is based on the T002 table in SAP.

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, the SOURCE\_SYSTEM\_ID column must be set to the value of &DDS\_SOURCE\_SYSTEM\_ID in the STAGE\_TIME\_PERIOD table.

The SAS Financial Management Adapter for SAP includes a special handling feature for financial years and periods that do not start on the first day of a month. This feature enables it to handle the calendar year and fiscal year, depending on the various situations that can occur in SAP data. For example, the SAS Financial Management 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 lengthened, then the final quarter is lengthened.

### ***fmsadpt\_T000200\_Transform\_Financial\_Period\_Info\_from\_SAP***

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 need to use only one type.

This job also 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 the following input:

- Fmsr3mir.T001 (created in E00010 Extract R3 General and FI Tnnn Tables)
- Fmsr3mir.T009 (created in E00030 Extract Financial Period T Tables)
- Fmsr3mir.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
- Intermed.FinYear\_Start\_and\_End\_dates
- \$FISCAL\_START\_MTH

*Note:* 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). The format is stored in the Library.formatscatalog.

This job uses the following user-written source:

- t000200\_fiscal\_year\_non\_cal.sas
- t000200\_add\_real\_calendar\_data.sas
- t000200\_determine\_fy\_begin\_end\_dates.sas
- The user-written source t000200\_determine\_fy\_begin\_end\_dates.sas uses Intermed.Fiscal\_Year\_Periods\_with\_dates table to determine the start date and end date for each financial year that are within each combination of time period association type, financial year, and time period ID.



**fmsadpt\_T000210\_create\_Inter\_time\_period\_dimension\_Table**

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
- Intermed.Fiscal\_Year\_Periods\_with\_dates

This job creates the following output:

- Intermed.FinYear\_Quarters
- Intermed.Fiscal\_Year\_Periods\_with\_parents
- Intermed.Time\_period\_dimension

This job uses the \$COMPANY\_TO\_FY\_VARIANT.

This job uses the user-written source

t000210\_ADD\_TIME\_HIERARCHY\_LAYERS.SAS.

**fmsadpt\_T000220\_Load\_stage\_time\_period\_dimension\_Tables**

This job loads the main Time Period dimension tables and staging area structure for STAGE\_TIME\_PERIOD\_ASSOC\_TYPE.

This job uses the following input:

- Intermed.Time\_period\_dimension (created in the T000210 job)
- Intermed.Used\_Fiscal\_Year\_Variants
- Fmsr3mir.T009T

This job creates the following output:

- STAGEFM.TIME\_PERIOD\_ASSOC
- STAGEFM.TIME\_PERIOD
- STAGEFM.TIME\_PERIOD\_ASSOC\_TYPE

**fmsadpt\_T000230\_create\_Inter\_controlling\_area\_fin\_year\_combo**

This job creates the output Intermed.CONTROL\_AREA\_YEARS\_COMBOS. This table represents all combinations of all controlling areas that are required with the start and end dates of each financial year.

Because cost center and profit center hierarchies change over time, the SAS Financial Management Adapter for SAP extracts hierarchies for each financial year. The output table of this job is required to determine the queries to SAP. Profit center and cost center dimensions depend on this job.

This job uses the following input:

- Fmsr3mir.TKA01
- Intermed.FinYear\_Start\_and\_End\_dates

This job creates the output Intermed.CONTROL\_AREA\_YEARS\_COMBOS.

This job uses the user-written source t000230\_set\_hier\_root.sas.

## Job Group: T0003 Exchange Rates

### Overview

This job group includes jobs that transform data for SAS Financial Management. The job group takes data from the exchange rate tables in SAP R/3 (TCURR and TCURRT).

The exchange rate types must match the types that are accepted as valid simple exchange rate types in the SAS Financial Management staging area. The historic (HIS) and derived (DER) exchange rate types form a group of complex exchange rate types. All of the other types are valid, simple exchange rate types.

By default, the SAS Financial Management Adapter for SAP derives the daily rates from the exchange rate of type M in SAP. It distributes the exchange rates to all days for which it is valid. It loops through all time periods, looks up the start and end dates, and selects all exchange rates that were valid on those days.

For example, the SAS Financial Management Adapter for SAP loads the rates for the staging area exchange rate types PO (period open) and PE (period end). The SAS Financial Management Adapter for SAP loads the exchange rates into the Stage Currency Exchange Rates table. Additional exchange rates might need to be loaded for exchange rate type PA (period average), and for the user-defined exchange rates C1 and C2. You must manually load these exchange rates; the SAS Financial Management Adapter for SAP does not complete this task.

The SAS Financial Management Adapter for SAP creates a sample analysis dimension because the exchange rate references the analysis dimension. A user-specific join must join the analysis table and the exchange rates. This is a simple join that provides an analysis dimension, which can be adapted.

### ***fmsadpt\_T000300\_Load\_stage\_currency\_exch\_rate\_type\_Table***

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 using a 1:1 approach. As a result, the SAS Financial Management Adapter for SAP does not use this format in most cases. However, some installations might find the format useful in subsetting the exchange rates in SAP.

This job uses SASSUPFM.SAS\_CURRENCY\_EXCH\_RATE\_TYPE as input.

This job creates the following output:

- STAGEFM.CURRENCY\_EXCH\_RATE\_TYPE
- Intermed.SAPTOSAS\_EXRATE\_TYPE
- \$SAPTOSAS\_CURRENCY\_EXCH\_TYPE

The user-written source for this job is t000300\_saptosas\_exchrates\_type\_mapping.sas.

### ***fmsadpt\_T000310\_Load\_stage\_currency\_exch\_rate\_set\_table***

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. For example, this is helpful when exchange rates are obtained or maintained in an external system.

This job uses the input t000310\_set\_exch\_rate\_set.sas.

This job creates the output STAGEFM.CURRENCY\_EXCH\_RATE\_SET.

### **fmsadpt\_T000320\_Load\_stage\_currency\_exch\_rate\_src\_Table**

This job creates STAGEFM.CURRENCY\_EXCH\_RATE\_SRC with a constant key that is equal to SAP.

This job creates the output STAGEFM.CURRENCY\_EXCH\_RATE\_SRC.

The user-written source for this job is  
t000320\_populate\_currency\_exch\_rate\_src\_table.sas.

### **fmsadpt\_T000330\_Load\_stage\_currency\_exch\_rate\_Table**

This job implements the main logic that calculates the period opening and closing exchange rates.

This job uses the following input:

- STAGEFM.TIME\_PERIOD
- Fmsr3mir.TCURR

This job creates the following output:

- intermed.DAILY\_EXCHANGE\_RATES
- STAGEFM.CURRENCY\_EXCH\_RATE

This job uses the following user-written source:

- t000330\_identify\_tcurr\_end\_dates.sas
- t000330\_derive\_periods\_for\_exrates.sas
- t000330\_expand\_to\_all\_days.sas

---

## **Job Group: T0004 Analysis Dimension**

### **Overview**

This job group creates an analysis dimension that is not derived from SAP. The SAS Financial Management Adapter for SAP extracts actual amounts that are related to actual dimensions. Therefore, the analysis dimension is hard-coded. The extraction might require adjustments, depending on your installation.

The fmsadpt\_T000400\_Load\_stage\_analysis\_dimension\_tables is the only job in this group. It does not depend on any of the previous jobs.

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

**fmsadpt\_T000400\_Load\_stage\_analysis\_dimension\_tables**

This job creates the minimum four structures of the analysis dimension.

This job creates the following output:

- STAGEFM.ANALYSIS
- STAGEFM.ANALYSIS\_ASSOC
- STAGEFM.ANALYSIS\_ASSOC\_TYPE
- STAGEFM.STAGE\_ANALYSIS\_NLS

This job uses the following user-written source:

- t000400\_populate\_analysis\_table.sas
- t000400\_populate\_analysis\_assoc\_table.sas
- t000400\_populate\_analysis\_assoc\_type\_table.sas
- t000410\_populate\_analysis\_nls\_table.sas

---

## **Job Group:T0005 Account Dimension**

### **Overview**

The general ledger 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 depends on the language and the chart of accounts.

There is no standard SAP hierarchy to display account structure. As a result, the financial statement is extracted using the RFC version of the SAP function modules FI\_IMPORT\_BALANCE\_SHEET\_POS and FI\_IMPORT\_BALANCE\_SHEET\_TEXT.

There is no generic approach to accurately populate some of the classification columns (such as ACCOUNT\_TYPE\_CD) for all rows of the STAGE\_GL\_ACCOUNT table. To prepare the account dimension data for import into the staging area, nonstandard and user-specific code must incorporate the additional attributes.

The SAS Financial Management Adapter for SAP provides sample code that you can customize. If you do not customize the code for your environment, an error message is returned.

*Note:* The value of ACCOUNT\_TYPE\_CD (in the STAGE\_GL\_ACCOUNT table) must be a predefined value in the SAS\_GL\_ACCOUNT\_TYPE table.

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

### **Inputs and Outputs (Tables, Formats, and Macro Variables)**

The following list describes general inputs to this job group:

- Descriptions 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 is the financial statement version. &\_VERSN is the financial statement version enclosed in quotation marks.

This job group uses the following SAP data sources as input:

- SKA1 - G/L Account Master Chart of Accounts. This source is used directly as a table and is used to create the format \$ACCBALS.
- SKAT - GL 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 Key that uses the format \$SAPTOSAS\_LANGUAGE.

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

```
PROC CALLRFC libref=sapserve;
call function Z_SAS_FI_IMPORT_BAL_SHEET_POS
EXPORTING version=&_versn
OUTTABLES X011P=Fmsr3mir.RF011P
I011Z=Fmsr3mir.RF011Z;
call function Z_SAS_FI_IMPORT_BAL_SHEET_TEXT
EXPORTING sprache=&spras version=&_versn
OUTTABLES X011Q=Fmsr3mir.RF011Q;
run;
```

This job group uses the following input:

- Fmsr3mir.RF011P
- Fmsr3mir.RF011Q
- Fmsr3mir.RF011Z

This job group creates the following output:

- STAGEFM.GL\_ACCOUNT\_ASSOC\_TYPE  
The T000140 job creates this output table. This table lists the codes that identify the types of hierarchies that are represented in the association table.
- STAGEFM.GL\_ACCOUNT\_ASSOC  
The T000140 job creates this output table. This table lists the hierarchical parent and child relationships for the GL\_ACCOUNT member IDs. The GL\_ACCOUNT\_ASSOC\_TYPE\_CD table distinguishes each unique hierarchy type.
- STAGEFM.GL\_ACCOUNT  
The T000150 job creates this output table. This table lists the general ledger account members. The table includes the parent and child general ledger account members that are used in traditional hierarchies or dimensions. Examples include cash, accounts receivable, current assets, and assets.

## **Financial Statement Nodes**

The SAP financial statement consists of a series of nodes. There are predefined root nodes, such as asset and liability. The child nodes of the asset root node are also assets. At the lowest level, the node has a list of accounts or intervals of accounts. At the highest level, there is a node named UNASSIGNED. The UNASSIGNED node contains accounts that are not assigned to other parts of the financial statement.

## **Account Hierarchies and Numbering**

It is important to understand account hierarchies and naming structures. SAP puts some accounts into different parent nodes, depending on the sign on the summarized data.

For example, a sign can be the sum of all transactions for that account during the specified period. RF011Z.ERGSO and RF011Z.ERGHB indicate that the child accounts (VONKT to BILKT) might be in two different nodes 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 staging area specify that accounts must be unique, and that an account cannot appear in two different nodes in the hierarchy. To avoid this, credit account numbers have a C suffix, and debit account numbers have a D suffix. One SAP account is split into two different SAS accounts. However, only one of the accounts can be used within the same period. This requires summarizing the data from SAP before deciding what account a transaction has. 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.

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

In an SAP R/3 table, the base column is SKA1.SAKNR. This base column is used during mapping to create ACCOUNT\_ID. This is unique in SAP within a chart of accounts. The SAS Financial Management Adapter for SAP can get a financial statement (the same as a balance sheet; it is used to build the hierarchy) for a single chart of accounts, so it uses SAKNR. SAKNR is shortened to show only the significant digits. The macro variable &SAKLN (set in PREPROCPARMS12.SAS) controls this presentation or setting of the digits. For example, if SAKNR contains the account number 0000012345 and the &SAKLN variable is 6, then the account number is 012345.

## **Profit and Loss Accounts**

To distinguish between a profit and loss account, and a balance account, the SAS Financial Management Adapter for SAP uses the column XBILK in the SKA1 table. If an X is returned, it is a balance account. SKA1 XBILK is used to create the format \$ACCBALS.

In SAP, an account can belong to the profit-and-loss part of the balance sheet. The financial statement makes it possible to determine whether a balance sheet is an asset or a liability account. The SAS Financial Management Adapter for SAP handles only a single financial statement or a 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 use the SAP transaction OB53 - Define Retained Earnings Account to check that the RETAINED\_EARNINGS\_FLG is defined from the T030C table. Most companies use one retained earnings account. For this reason, X is used to indicate that the flag is defined.

SAP represents the financial statement report as a collection of disjointed hierarchies or top-level items. All disjointed 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 SAP, and not in the financial statement hierarchy, are added to the UNASSIGNED node.

The value of NORMAL\_BALANCE\_CD in STAGE\_GL\_ACCOUNT must be one of the predefined account type values in the SAS\_GL\_NORMAL\_BALANCE table. These predefined values are D for debit and C for credit. The value of LANGUAGE\_CD must be one of the predefined language values in the SAS\_LANGUAGE table. This is accomplished by using the format \$SAPTOSAS\_LANGUAGE based on the T002 table.

### **fmsadpt\_T000500\_create\_Inter\_T011**

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

This job uses the following input:

- Fmsr3mir.T011 (created in E00010 Extract R3 General and FI Tnnn Tables)
- &\_VERSN

This job creates the output INTERMED.T011.

### **fmsadpt\_T000505\_create\_Inter\_Accounts\_with\_Text**

This job uses the following input:

- Intermed.T011 (created in T000100 Select Row in T011 according to the macro variable &\_VERSN)
- Fmsr3mir.SKA1 (created in E01400 Extract R3 Account Master)
- Fmsr3mir.SKAT (created in E01400 Extract R3 Account Master)
- &lang (set in preprocessing step)

This job creates the output INTERMED.ACCOUNTS\_WITH\_TEXT.

### **fmsadpt\_T000510\_create\_Inter\_Account\_Items\_Transformed**

Accounts can have only one parent in the general ledger account dimension hierarchy. SAP uses a methodology in which an account can appear in different places in the financial statement based on the balance. For example, a bank account with a positive balance can be an asset, and a bank account with a negative balance can be a liability. As a result, these accounts can appear in two different places in the hierarchy. To enable a fixed place in the hierarchy, the SAS Financial Management Adapter for SAP creates two accounts. One account is a credit account, and the other is a debit account. The SAS

Financial Management Adapter for SAP, in turn, adjusts the original SAP account number by appending a C or D suffix.

This job uses the following input:

- Intermed.T001 (created in T000500 Select Row in T011 according to the macro variable &\_VERSN)
- Intermed.Accounts\_with\_text (created in T000505 Create Accounts with Text)
- Fmsr3mir.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 in which the credit item key (ERGHB) does not equal the debit item key (ERGSO), two output records are created:

- One record includes a negative value that has a D suffix at the end of the lower limit of the account interval (VONKT).
- The other record has a positive value with a C suffix.

All other records are rewritten.

### **fmsadpt\_T000520\_create\_Inter\_Account\_Hierarchy**

The hierarchy is based on financial statement items and their associated text. Because the extracted text creates a reporting template, the text includes formatting lines, such as----- and =====. As a result, the mapping process excludes these lines with the following command:

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

To create the correct parent and child structure, subsequent processes perform the required customizations. 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 actual accounts), the key is from SAKNR. Based on the account balance, a C or D suffix, distinguishes the accounts into two different sets that appear in two different places in the hierarchy. For example, a positive bank balance needs to appear as an asset. A negative bank balance needs to appear as a liability.

The macro variable &SAKLN shortens the ACCOUNT\_ID of leaf nodes by calling the macro %ADPT\_ADJ\_TO\_SIGNIF\_ACCOUNT\_LEN.

This job uses the following input:

- Fmsr3mir.RF011P (created in E04010\_Extract\_Balance\_Sheet\_FSV\_Structures)
- Fmsr3mir.RF011Q (created in E04010\_Extract\_Balance\_Sheet\_FSV\_Structures)
- Fmsr3mir.T011 (created in E00010\_Extract\_SAP\_General\_T\_Tables)
- ACCITEM\_TRANSFORMED (created in T000510 Accounts Item Transferred)



- ACCOUNTS\_WITH\_TEXT (created in T000505 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 (the main result)

This job uses the following user-written source:

- t000120\_account\_hier\_addroot.sas
- t000520\_not\_assigned\_account.sas
- t000520\_transform\_rf011q\_tr011q.sas
- t000520\_add\_number\_for\_sort.sas

### ***fmsadpt\_T000525\_create\_Inter\_Account\_Balances***

This job uses the following input:

- Inter\_GL\_Balances\_new (created in fmsadpt\_T000140\_Load\_Inter\_GL\_Balances(New GL))
- FMA\_R3\_Mirror SKA1 - G\_L Account Master (Chart of Accounts)

This job creates the output Inter\_GL\_Account\_Balances.

This job uses the user-written source t000525\_summarise\_gl\_account\_balances.sas.

### ***fmsadpt\_T000528\_create\_Inter\_Account\_Type***

This job uses the following input:

- Inter\_GL\_Account\_Balances (created in fmsadpt\_T000525\_create\_Inter\_Account\_Balances)
- FMA\_R3\_Mirror CSKB - Cost Elements

This job creates the output Inter\_GL\_Account\_type\_co\_balances.

### ***fmsadpt\_T000530A\_create\_Inter\_GL\_Account\_with\_SAP\_Info***

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 the macro variable &\_VERSN)
- Fmsr3mir.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 the output Intermed.GL\_ACCOUNT\_WITH\_SAP\_INFO.

This job uses the following user-written source:

- t000530\_sap\_account\_type\_site\_specific.sas
- t000530\_sap\_account\_type.sas

### **fmsadpt\_T000530B\_create\_Inter\_GL\_Account\_with\_SAP\_Info**

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)
- Fmsr3mir.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

- t000530b\_sap\_account\_type\_site\_specific.sas
- t000530b\_sap\_account\_type.sas

*Note:* The above two jobs are alternative jobs that have to be exclusively executed. The T000530A job has to be executed when the T030C table is correctly configured, so as to reflect the nodes in the financial statement version (FSV). If the T030C table is not configured as per the nodes in the FSV, then the user-written code in the job T000530B has to be customized as per site specifications. Also, the job T000530B has to be executed instead of T000530A.

### **fmsadpt\_T000535\_create\_Inter\_Account\_statistic\_Site\_Specific**

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

If you use statistical accounts at your local installation, append each table to its counterpart. For example, STAGEFM.GL\_ACCOUNT corresponds with STAGEFM.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 t000535\_stat\_accounts.sas.

### **fmsadpt\_T000540\_Load\_state\_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)
- Fmsr3mir.T011T (created in E00010 Extract R3 General and FI Tnnn Tables)

This job creates the following output:

- STAGEFM.GL\_ACCOUNT\_ASSOC\_TYPE
- STAGEFM.GL\_ACCOUNT\_ASSOC

The user-written source for this job is t000540\_gl\_account\_add\_ddc\_assoc\_type.sas. This source executes the SAS macro %ADPT\_ADD\_DDS\_ASSOC\_TYPE using GL\_ACCOUNT as the dimension.

### **fmsadpt\_T000550A\_Load\_stage\_gl\_account\_tables\_Site\_Specific**

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 STAGEFM.GL\_ACCOUNT as output.

This job uses the following user-written source:

- t000550a\_standard\_settings
- t000550a\_site\_specific\_code

### **fmsadpt\_T000550B\_Load\_stage\_gl\_account\_tables\_Site\_Specific**

This job is an alternative for the job mentioned in the previous section. 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.

*Note:* This is an alternative job to T000550A. Do not run both jobs! If you need to determine the account type for revenue and expense accounts from the SAP controlling module, run T000550B only. Otherwise, run T000550A.

This job uses the input intermed.GL\_ACCOUNT\_WITH\_SAP\_INFO.

This job creates the output STAGEFM.GL\_ACCOUNT.

The user-written source for this job is:

- t000550b\_standard\_settings

- t000550b\_site\_specific\_code

### **fmsadpt\_T000560\_Load\_stage\_source\_gl\_account\_Table**

This job uses the input Inter.ACCNT\_ITEM\_HIER.

This job creates the output SOURCE\_GL\_ACCOUNT.

This job uses the user-written source  
t000560\_source\_gl\_account\_site\_specific\_code.sas.

---

## **Job Group: T0006 Cost Center Dimension**

### **Overview**

Mapping SAP cost centers to the staging area cost center dimension is relatively easy because both models include similar concepts. However, the hierarchy extraction requires 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\_DTTM column. The end of the financial year is stored in the VALID\_TO\_DTTM column.

The hierarchy extraction in the SAS Data Surveyor for SAP generates code on which the cost center hierarchy extraction is based. First, the extraction gets the start date 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). The SAS Financial Management Adapter for SAP extracts the cost centers on the basis of wanted controlling areas in SAP, as specified by the macro variable &WANTED\_CONTROLLING\_AREAS.

It is important to understand the table factors of this job group. The following factors affect this job group:

- Descriptions must be in the language that is specified by SPRAS or LANG.
- The value of the SOURCE\_SYSTEM\_ID column 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 behavior creates the cost center ID as a concatenation of the controlling area (KOKRS) and cost center (KOSTL). You can change this macro based on your organization requirements.

This macro is used in the following jobs:

- T000400 (called within the compiled macro %ADPT\_GET\_HIER\_OVERYEARS.SAS)
- T000410
- T100100
- The SAS Financial Management Adapter for SAP creates separate hierarchies for controlling areas and financial years. It joins them into a single table. Within a controlling area (COST\_CENTER\_ASSOC\_TYPE\_CD) and date range, no row

appears 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 does not populate correctly from SAP. The name from the VERAK column in the SAP R/3 table CSKS populates the column. However, the name is not an employee ID and cannot be used for validation.
- The SAP R/3 transaction code OKENN can be used to compare the extracted data with the data in SAP R/3. OKENN shows only a hierarchy for a single controlling area, but there can be different start and end dates.

### **fmsadpt\_T000600\_Transform\_Cost\_Center\_Hierarchy\_Over\_Years**

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 T000230 Create Controlling Area Fin Year Combos)
- Fmsr3mir.TKA01 (created in E00020 Extract TXXX General and FI Tables)
- Calls to ABAP functions

This job creates the following output:

- Intermed.cost\_center\_hierarchy
- STAGEFM.COST\_CENTER\_ASSOC\_TYPE

The user-written source for this job is t000600\_cost\_center\_hierarchy\_extract.sas.

### **fmsadpt\_T000610\_Load\_stage\_cost\_center\_dimension\_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 T000200 Determine FinYear Begin and End Dates)
- intermed.COST\_CENTER\_HIERARCHY (created by T000600 Cost Center Hierarchy Over Years Extraction)
- Fmsr3mir.CSKS (created by E01000\_Extract\_SAP\_Master\_Code\_Tables)
- Fmsr3mir.CSKT (created by E01000\_Extract\_SAP\_Master\_Code\_Tables)
- Fmsr3mir.T002 (created by E00010 Extract\_SAP\_General\_T\_Tables)
- &lang

- &wanted\_controlling\_areas

This job creates the following output:

- intermed.COST\_CENTER\_ONLY\_HIER
- intermed.COST\_CENTER\_ONLY\_MASTER
- STAGEFM.COST\_CENTER\_ASSOC
- STAGEFM.COST\_CENTER
- STAGEFM.COST\_CENTER\_NLS

The user-written source for this job is t000610\_sep\_matches.sas.

### **fmsadpt\_T000620\_Transform\_Add\_Std\_Dim\_Values\_cost\_center**

The SAS Financial Management Adapter for SAP adds fixed-row records in dimension tables for the OPENBAL, UNASSIGNED, and ALL keys. This ensures that the records in GL\_TRANSACTION\_SUM table always reference existing rows.

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

This job uses the following transformations:

- Add Standard Dimension ASSOC Rows
- Add Standard Dimension Rows

This job uses the following input:

- STAGEFM.COST\_CENTER\_ASSOC\_TYPE
- STAGEFM.COST\_CENTER (input and output)
- &dds\_source\_system\_id

This job creates the following output:

- STAGEFM.COST\_CENTER\_ASSOC (input and output)
- STAGEFM.COST\_CENTER (input and output)

---

## **Job Group: T0007 Profit Center Dimension**

### **Overview**

Mapping SAP profit centers to the staging area profit center dimension is relatively easy because both models include similar concepts. The profit center processes are similar to the cost center processes. However, the hierarchy extraction requires special attention. A separate hierarchy is extracted for each financial year that records the dates for which the hierarchy nodes are valid. This is done because the hierarchies change over time. As a result, the start of the financial year is stored in the VALID\_FROM\_DTTM column. The end of the financial year is stored in the VALID\_TO\_DTTM column.

The hierarchy extraction in the SAS Data Surveyor for SAP generates the code that forms the basis for the profit center hierarchy extraction. First, the extraction gets the start date 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`). The SAS Financial Management Adapter for SAP extracts the profit centers on the basis of wanted controlling areas in SAP, as specified by the macro variable `&WANTED_CONTROLLING_AREAS`.

It is important to understand the table factors of this job group. This job group includes the following factors:

- Descriptions must be in the language that is specified by `SPRAS` or `LANG`.
- The value of the column `SOURCE_SYSTEM_ID` is set to the value of `&DDS_SOURCE_SYSTEM_ID`.
- The value of the column `TIME_PERIOD_ASSOC_TYPE_CD` for the given controlling area is the same as the value of the column `PROFIT_CENTER_ASSOC_TYPE_CD`.
- The macro `%ADPT_PROFIT_CENTER_ID_MAP` is used to set the profit center ID. The default behavior creates the profit center ID as a concatenation of the controlling area (`KOKRS`) and profit center (`PRCTR`). You can change this macro depending on the needs of your organization.

This macro is used in the following jobs:

- T000500 (called within the compiled macro `%ADPT_GET_HIER_OVERYEARS.SAS`)
- T000510
- T100100
- The SAS Financial Management Adapter for SAP creates separate hierarchies for controlling areas and financial years. It joins them into a single table. Within a controlling area (`PROFIT_CENTER_ASSOC_TYPE_CD`) and date range, no row appears 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 does not populate correctly from SAP. The name from the `VERAK` column in the SAP R/3 table `CEPC` populates the column, but the name is not an employee ID and cannot be used for validation.
- The SAP R/3 transaction code `K6HCN` can be used to compare the extracted data with the data in SAP R/3. `K6HCN` shows only a hierarchy for a single controlling area, but there can be different start and end dates.

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

### ***fmsadpt\_T000700\_Transform\_Profit\_Center\_Hierarchy\_Over\_Years***

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 T000230 Create Controlling Area Fin Year Combos)
- `Fmsr3mir.TKA01` (created in E00020 Extract TXXX General and FI Tables)
- Calls to ABAP functions

This job creates the following output:

- Intermed.profit\_center\_hierarchy
- STAGEFM.PROFIT\_CENTER\_ASSOC\_TYPE

The user-written source for this job is t000700\_profit\_center\_hierarchy\_extract.sas.

### **fmsadpt\_T000710\_Load\_stage\_profit\_center\_dimension\_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 T000200 Determine FinYear Begin and End Dates)
- intermed.PROFIT\_CENTER\_HIERARCHY (created by T000700 Transform Profit Center Hierarchy Over Years Extraction)
- Fmsr3mir.CEPC (created by E01000 Extract\_SAP\_Master\_Code\_Tables)
- Fmsr3mir.CEPCT (created by E01000Extract\_SAP\_Master\_Code\_Tables)
- Fmsr3mir.T002 (created by E00010 Extract\_SAP\_General\_T\_Tables)
- &lang
- &wanted\_controlling\_areas

This job creates the following output:

- intermed.PROFIT\_CENTER\_ONLY\_HIER
- intermed.PROFIT\_CENTER\_ONLY\_MASTER
- STAGEFM.PROFIT\_CENTER\_ASSOC
- STAGEFM.PROFIT\_CENTER
- STAGEFM.PROFIT\_CENTER\_NLS

The user-written source for this job is t000710\_separate\_matches.sas.

### **fmsadpt\_T000720\_Transform\_Add\_Std\_Dim\_Values\_profit\_center**

The SAS Financial Management Adapter for SAP adds fixed row records in dimension tables for the OPENBAL, UNASSIGNED, and ALL keys. This ensures that the records in the GL\_TRANSACTION\_SUM table always reference existing rows.

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

This job uses the following transformations:

- Add Standard Dimension ASSOC Rows



- Add Standard Dimension Rows

This job uses the following input:

- STAGEFM.PROFIT\_CENTER\_ASSOC\_TYPE
- STAGEFM.PROFIT\_CENTER\_ASSOC (input and output)
- STAGEFM.PROFIT\_CENTER (input and output)
- &dds\_source\_system\_id

This job creates the following output:

- STAGEFM.PROFIT\_CENTER\_ASSOC (input and output)
- STAGEFM.PROFIT\_CENTER (input and output)

---

## Job Group: T0008 Currency Dimension

### Overview

This job group creates currency dimension reference tables. These jobs are loaded into the tables and the staging area structure. When the follow-up jobs use these tables, the staging tables merge again with the original tables that are provided by SAS. Records can be added to these tables.

### ***fmsadpt\_T000800\_Load\_stage\_currency\_dimension\_tables***

This job creates the currency association-related tables.

This job uses the following input:

- FMA\_R3\_Mirror.TCURT - Currency Code Names
- FMA\_R3\_Mirror.T002 - Language Keys
- SASSUPFM.SAS\_CURRENCY

This job creates the following output:

- STAGEFM.CURRENCY\_ASSOC\_TYPE
- STAGEFM.CURRENCY\_ASSOC

This job uses the user-written source  
t000800\_populate\_STAGE\_CURRENCY\_ASSOC\_TYPE\_table.sas.

---

## Job Group: T0009 Financial Documents

### Overview

This job group transforms the data from the SAP R/3 tables into the staging area staging format.

The financial documents are based on an initial load of the SAP financial documents, such as BKPF and BSEG. This initial load includes all data after the first start date and subsequent change-data-capture (CDC) loads.

After the data loads, the SAS Financial Management Adapter for SAP ensures that all columns, that reference dimensions or foreign keys use the correct keys. For example, the COST\_CENTER\_ID column is not equal to the KOSTL column in SAP. The COST\_CENTER\_ID column maps from KOKRS.

## **Input and Output (Tables, Formats, and Macro Variables)**

### **Input**

This job uses the following input:

- \$COMPANY\_TO\_FY\_VARIANT
- \$ACCBALS
- \$XRANGE
- \$LOCALCR

*Note:* Only the controlling areas that are 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 to the target data model. Document numbers within certain ranges are internal. When SAP R/3 allocates document numbers, it chooses the next number in sequence from pre-allocated ranges. The ranges are stored in the SAP R/3 table NRIV. You can identify internal documents using NRIV.OBJECT='RF\_BELEG'. The \$XRANGE format identifies the document numbers accordingly, so PUT(BUKRS!!GJAHR!!BELNR,\$XRANGE.)='INTERNAL' identifies the document as internal.

This job group works with all of the data in the Intermed.Financial\_Documents. This data includes all detailed transactions starting from the beginning of the &STARTFINYEAR global macro variable, or BKPF.GJAHR>=&\_STARTFINYEAR and BSEG.GJAHR>=&\_STARTFINYEAR. After reading the data from the Intermed.Financial\_Documents file into the staging area, you should empty the file.

### **Dimension Foreign Keys**

This job group has the following rules for foreign keys:

- 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 during import into the staging area.
- A dimension column that does not have a value in SAP must contain the value UNASSIGNED. This must be true for all the records from the standard financial documents with a BKPS or BSEG derivative. This implies that there must be a row with ID UNASSIGNED in all of the dimensions. UNASSIGNED should not have a parent in the \_ASSOC tables.

### **Opening Balance Considerations**

It is important to understand the characteristics of an opening balance:

- The opening balance for the initial period is included to ensure correct balances. These balances are derived from the SAP R/3 table GLT0. The SAS Financial Management 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 Financial Management Adapter for SAP inserts a constant value OPENBAL for the cost centers, profit centers, and other dimension columns in GLT0.
- The information used for the opening balance should be used only for balance sheet accounts, and not for profit-and-loss accounts. The format \$ACCBALS identifies balance sheet accounts. Opening balance amounts are included only for accounts where the following condition is true: PUT(GLT0.RACCNT,\$ACCBALS.)='X'. This implies that there must be a row with ID OPENBAL in all of the relevant dimensions. OPENBAL has the parent ALL in the \_ASSOC tables.
- The general ledger account dimension IDs mostly derive from the SAP account number. However, some accounts appear in the financial statement in different places, based on the account balance. Therefore, a single SAP account number maps to two accounts: one with a C suffix and one with a D suffix. All transactions within the same financial year must be booked in the same account. For this reason, the SAS Financial Management Adapter for SAP uses the totals from GLT0 to decide whether to append a C or a D to the account number for each transaction.

*Note:* Balances might change from positive to negative, or vice versa, during a financial year. This means that the C or D in a transaction account ID might change over the course of the financial year. If this is the case, data might need to be re-imported into the staging area and follow-up data marts that use the data.

### **fmsadpt\_T000900\_create\_Inter\_Opening\_Balances\_ and\_Num\_Ranges**

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 Fmsr3mir.GLT0. Using the format \$ACCBALS on the account number (RACCT) gets the balance sheet accounts or all records that translate to X. This job fills in default values of OPENBAL for all unavailable columns. It adds time dimension values to previous SAS views. It also outputs the data set Intermed.Opening\_Balances.

For each financial period, this job concatenates the previous summary data set with the SAS view. It summarizes this data and creates a temporary data set. This temporary data set is appended to the Intermed.Opening\_Balances data set.

This job uses the following input:

- Intermed.Financial\_Documents (created in E02005\_Extract\_Intermediate\_Table\_merging\_BKPF\_BSEG)
- Inter\_GL\_Balances\_new (created in T000140\_Load\_Inter\_GL\_Balances(New GL))
- \$ACCBALS. (created in E01050\_Create\_Formats\_from\_SAP\_Tables)
- \$LOCALCR (created in E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$COMPANY\_TO\_FY\_VARIANT (created in E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$XRANGE (created in this job only)
- Fmsr3mir.NRIV

This job creates the following output:

- intermed.INTERNAL\_NUMBER\_RANGES

- Format \$XRANGE
- Intermed.Opening\_Balances

The user-written source for this job is t000900\_create\_opening\_balances.sas.

This job also reads the number range table in SAP and creates the format \$XRANGE to identify internal document numbers. The \$XRANGE format is used to create the Intermed.Opening\_Balances.

### **fmsadpt\_T000910\_create\_Inter\_GLT0\_Vertical\_Summary**

This job creates a new version of the data set GLT0. It converts each row that contains 16 period values into 16 rows, where each row contains one 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 and Debit Indicator (SHKZG)
- Fiscal Period (MONAT)

The job subsets this summary on the value of the macro variable &gjahr and nonzero values of the Fiscal Period (MONAT). It reads the data set Inter\_GL\_Balances\_new.

In addition, this job writes an output record for each of the 16 occurrences of the total nonzero transactions for 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:

- Inter\_GL\_Balances\_new (created in T000140\_Load\_Inter\_GL\_Balances(New GL))
- \$SHKZG. (created in E00070\_Extract\_Addl\_Tables\_for\_Formats)
- \$COM2CAC. (created in E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$LOCALCR. (created in E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- &KTOPL
- &GJAHR

This job creates the following output:

- Intermed.GLT0\_Vertical
- Intermed.GLT0\_Vertical\_Summary

This job uses the following user-written source:

- t000910\_make\_glt0\_vertical.sas
- t000910\_summarise\_glt0\_vertical.sas

### **fmsadpt\_T000920\_create\_Inter\_Level\_to\_Account**

This job decides whether an account must have a C or D appended, based on the account balance. The job uses the data set Intermed.GLT0\_Vertical to summarize the Value of Posting (WRBTR) by Company Code (BUKRS), Account Number (HKONT), and Fiscal Year (GJAHR).

This job joins the KTOPL and WAERS columns from the Fmsr3mir.T001 table.

This job uses the following input:

- Fmsr3mir.T001 (created in E00010 Extract SAP General and T Tables)
- Intermed.GLT0\_Vertical (created in T100030 Make GLT0 Vertical)
- Intermed.One2many (created in T000510\_create\_Inter\_Account\_Items\_Transformed)

This job creates Intermed.Level\_to\_Account as output.

The user-written source for this job is t000920\_summary\_on\_level.sas.

### **fmsadpt\_T000930\_Load\_stage\_transaction\_sum**

This job populates the GL\_TRANSACTION\_SUM table in the staging area.

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 fmsadpt\_E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$COM2CAC. (created in fmsadpt\_E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$CC2CA. (created in fmsadpt\_E00011\_Create\_Formats\_from\_SAP\_FI\_Tables)
- \$XRANGE. (created in fmsadpt\_T000900\_create\_Inter\_Opening\_Balances\_and\_Num\_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
- STAGEFM.GL\_TRANSACTION\_SUM

If partial cost center IDs are used to create the internal organization dimension, then column mappings in the financial documents need to be adjusted to correspond, in the T100100 job. These adjustments should be made in the load step of the STAGE\_GL\_JRNL\_DETAILS table 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 action is performed by calling the macro %ADPT\_ADJ\_TO\_SIGNIF\_ACCOUNT\_LEN in the load step of the transaction table.

---

## Job Group:T1000 External Org Dimension

### Overview

This job group is a collection of SAS Data Integration Studio jobs that transform a combination of customer and supplier data from SAP R/3 tables into the staging area format.

Customers are 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 for whom you maintain detailed records in SAP R/3.

SAP can maintain basic or complex customer records that reflect the specific customers and their relationships with your organization. This flexibility covers the various types of customers and their complexities. In SAP, you might have created customer records for internal customers, such as a department or committee that uses 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 organization that are tracked. For example, one customer node might be a specific customer's distribution center. In the customer hierarchy, nodes are in a top-down structure. In this type of structure, the topmost customer node corresponds to the highest level of the hierarchy, and so on.

In addition to traditional customers, you might need to represent vendors 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 environment to determine which businesses serve you as a vendor. If necessary, you can choose to register a vendor as a supplier.

In SAP, a supplier hierarchy provides a structure of supplier nodes. These nodes define specific characteristics or parts of the supplier's organization that are tracked. For example, one supplier node might be a specific supplier's distribution center. In the supplier hierarchy, nodes are in a top-down structure. In this type of structure, the topmost node corresponds to the highest level of the hierarchy, and so on.

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

### ***fmsadpt\_T100000\_Load\_stage\_External\_Org\_Assoc\_Type***

This job prepares the external organization dimension Assoc type tables.

This job uses the following input:

- Fmsr3mir.THITT
- Fmsr3mir.TLHITT
- This job creates the following output:
- Intermed.Customer\_External\_Org\_Assoc\_Type
- Intermed.Supplier\_External\_Org\_Assoc\_Type
- STAGEFM.EXTERNAL\_ORG\_ASSOC\_TYPE

**fmsadpt\_T100010\_Load\_stage\_External\_Org\_Assoc**

This job populates the customer and supplier part of the external organization dimension Assoc tables.

This job also adds rows for OPENBAL, UNASSIGNED, and ALL to the dimension using the Add Standard Dimension Rows transformation.

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

This job uses the following input:

- Fmsr3mir.KNA1
- Fmsr3mir.KNVH
- Fmsr3mir.LFA1
- Fmsr3mir.LFMH

This job creates the following output:

- Intermed.Customer\_External\_Org\_Assoc
- Intermed.Supplier\_External\_Org\_Assoc
- STAGEFM.STAGE\_EXTERNAL\_ORG\_ASSOC

This job uses the following user-written sources:

- t100010\_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. It is required because the KUNNR field contains character values.
- t100010\_add\_c\_and\_s\_assoc\_nodes adds a C to a customer ID, and an S to a supplier ID in the STAGE\_EXTERNAL\_ORG\_ASSOC table. This ensures that the IDs are unique.
- t100010\_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. It is required because the LIFNR field contains character values.
- t100010\_internal\_number\_ranges.sas

**Fmsadpt\_T100020\_Load\_stage\_External\_Org**

This job populates the external organization tables from the customer and supplier intermediate tables.

This job also adds rows for OPENBAL, UNASSIGNED, and ALL to the dimension using the Add Standard Dimension Rows transformation.

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

This job uses the following input:

- Inter.LFA1

- Inter.KNA1
- Fmsr3mir.T002

This job creates the output STAGEFM.EXTERNAL\_ORG.

This job uses the user-written source T100020\_add\_c\_and\_s\_nodes.sas that adds a C to a customer ID, and an S to a supplier ID in the STAGE\_EXTERNAL\_ORG table. This ensures that the IDs are unique.

### ***Fmsadpt\_T100030\_Load\_stage\_External\_Org\_Address***

This job populates the external organization Address table from the Customer and Supplier intermediate tables.

This job uses the following input:

- Inter.LFA1
- Inter.KNA1

This job creates the following output:

- STAGEFM.STAGE\_EXTERNAL\_ORG\_ADDRESS.

---

## **Job Group:T1100 Internal Org Dimension**

### **Overview**

A combination of logical SAP data sources populates the internal organization job group for SAS Financial Management.

This job group is a collection of SAS Data Integration Studio jobs that transform data from SAP R/3 tables into the staging area format. These jobs create a hierarchy by assigning the first level for an operating concern, the second level for controlling areas, and the other levels in the following ways:

- a third level for company codes
- a fourth level for business areas
- other levels as cost center hierarchies

The internal organization dimension includes the following hierarchical levels:

operating concern (first level)

determined by the global macro variable &OPERATING\_CONCERN. The SAS Financial Management Adapter for SAP works with a single operating concern.

controlling areas (second level)

assigned as parents of their respective companies, based on the SAP R/3 table TKA02.

company codes (third level)

the simplest identifier of a balancing or legal accounting entity. For example, if a large corporation consists of several individual companies, a unique company code identifies each company within the larger corporate structure.



**business areas (fourth level)**

facilitates reporting across company codes, covering a company's main areas of operation. These main areas might represent divisions, branches, product lines, and so on. Not all companies have the same business areas. If your company has only one business area, you can exclude the business area level.

**cost center hierarchies (additional levels)**

included in the internal organization dimension if applicable. If an internal organization dimension includes cost center hierarchies, then the parents of the higher cost center levels are controlling area levels.

You must create the cost center hierarchy before copying the data into the internal organization dimension tables.

*Note:* In most cases, only profit and loss transactions that are obtained from BKPF or BSEG tables in SAP reference cost centers. Other transactions reference business areas.

*Note:* The cost center dimension does not include all of the information that is required for the internal organization dimension. See the source code in T110050\_LOAD\_INTERNAL\_ORG\_FROM\_CC.SAS for more information.

The STAGE\_INTERNAL\_ORG table is a simple concatenation of the contributing tables and a set of standard values. The hierarchy for the ASSOC table is created by linking the operating concern (first level) to the controlling areas (second level) by using the columns TKA01.KOKRS and TKA01.ERKRS. The controlling areas (second level) link to the company codes (third level) by using the columns TKA02.BUKRS and TKA02.KOKRS.

**fmsadpt\_T110000\_Load\_stage\_internal\_org\_assoc\_type\_tableJob**

This job loads the STAGE\_INTERNAL\_ORG\_ASSOC\_TYPE table with constant values. The table below shows the values that are required for the internal organization dimensions.

Internal Organization Association Type Code	Language Code	Internal Organization Association Type Description
OCC	en	Op Concern, Controlling Area, Company, Cost Center
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 t110000\_load\_staging\_internal\_org\_assoc\_type.sas accordingly.

This job uses the following input:

- Macro variable &lang
- Format \$SAPTOSAS\_LANGUAGE (created by E00010 Extract SAP General and T Tables)

This job creates the output table  
STAGEFM.STAGE\_INTERNAL\_ORG\_ASSOC\_TYPE.

**fmsadpt\_T110010\_create\_Inter\_Org\_from\_Operating\_Concern**

This job creates the first level of the hierarchy, the operating concern level.

This job creates the \$OP\_CONCERN\_CURRENCY format that the SAS Financial Management Adapter job uses. The prefix OPC is added to the operating concern to form the INTERNAL\_ORG\_ID. For example, an operating concern with controlling area of 1000 becomes OPC1000. This ID ensures a unique value.

This job uses the following input:

- Fmsr3mir.TKEB (created in \_E00060\_Extract\_Operating\_Concern\_T\_Tables)
- Fmsr3mir.TKEBT (created in \_E00060\_Extract\_Operating\_Concern\_T\_Tables)
- &operating\_concern\_country\_cd
- &dds\_source\_system\_id
- \$SAPTOSAS\_COUNTRY

This job creates the output Intermed.Operating\_Concern\_Internal\_Org.

This job uses the format \$op\_concern\_currency.

**fmsadpt\_T110020\_create\_Inter\_Org\_from\_Controlling\_Area**

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 that ensures a unique value. For example, a controlling area of 1000 becomes CAR1000.

This job uses the following input:

- Fmsr3mir.TKA01 (created in \_E00020\_Extract\_SAP\_FI\_T\_Tables)
- Fmsr3mir.TKA00 (created in \_E00020\_Extract\_SAP\_FI\_T\_Tables)
- &operating\_concern\_country\_cd
- &dds\_source\_system\_id
- \$SAPTOSAS\_COUNTRY

This job creates Intermed.Controlling\_Area\_Internal\_Org as output.

This job uses the user-written code t120020\_add\_control\_area\_countries.sas.

**fmsadpt\_T110030\_create\_Inter\_Org\_from\_Company\_Code**

This job creates a third layer of the hierarchy. This is the company layer.

This job uses the following input:

- Fmsr3mir.T001 (created in E00010 Extract SAP General and T Tables)
- Fmsr3mir.T880 (created in E00010 Extract SAP General and T Tables)
- Fmsr3mir.TKA01 (created in \_E00020\_Extract\_SAP\_FI\_T\_Tables)
- Fmsr3mir.TKA02 (created in t\_E00020\_Extract\_SAP\_FI\_T\_Tables)
- Fmsr3mir.T002 (created in E00010 Extract SAP General and T Tables)
- &dds\_source\_system\_id
- \$SAPTOSAS\_COUNTRY

- SASSUPFM.SAS\_COUNTRY\_ISO3166

This job creates the output Intermed.Company\_Internal\_Org.

This job uses user-written code t110030\_get\_used\_companies.sas.

### **fmsadpt\_T110040\_create\_Inter\_Org\_from\_Business\_Area**

This job creates the fourth level of hierarchy, the business areas level.

Because source data from SAP might not have a description for the business area, the initialization job I0050 enables you to enter a description. You can modify the setting for 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 ID ensures a unique value.

This job uses the following input:

- Fmsr3mir.TGSBT (created in E00020\_Extract\_SAP\_FI\_T\_Tables)
- Intermed.TGSBT\_Blank (created in I0050 Set Blank Business Area Name)
- Intermed.Company\_Internal\_Org (created in T110030\_create\_Inter\_Org\_from\_Company\_Code)
- &dds\_source\_system\_id
- \$SAPTOSAS\_COUNTRY

This job creates the following output:

- Intermed.Business\_Area\_Internal\_Org
- Intermed.Busarea\_Internal\_Org\_Assoc

### **fmsadpt\_T110050\_create\_Inter\_Org\_from\_Cost\_Center**

This job creates internal organization records that can be used as alternative levels of the internal organization hierarchy, if the T0004 job group 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 under the controlling areas level. This structure offers an alternative to the company codes and business areas levels.

*Note:* This job has a site-dependent user exit (T110050\_LOAD\_INTERNAL\_ORG\_FROM\_CC.SAS). Review the code in this file to ensure the proper administration of the job. After you make any necessary changes, comment out the error message.

This job uses the following input:

- Intermed.Inter\_stage\_cost\_center (created in job group T0006 Cost Center Dimension (T000610\_Load\_stage\_cost\_center\_dimension\_Tables))
- &operating\_concern\_country\_cd

This job creates the output Intermed.INTERNAL\_ORG\_from\_Cost\_Center.

**fmsadpt\_T110060\_create\_Inter\_Org\_Assoc\_from\_Cost\_Center**

This job creates records that can be used as alternative layers of the internal organization hierarchy. This can be done if the job group T0006 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:

- STAGEFM.STAGE\_COST\_CENTER\_ASSOC
- Intermed.Cost\_Center\_Hierarchy

This job creates the output Intermed.INTERNAL\_ORG\_ASSOC\_from\_Cost\_C.

**fmsadpt\_T110070\_Load\_stage\_Internal\_Org\_Table**

This job merges the levels into a single table and loads the STAGE\_INTERNAL\_ORG.

This job uses the transform Add Standard Dimension ASSOC Rows to add standard rows such as OPENBAL, UNASSIGNED, ALL, and EXT to the internal organization dimension tables.

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

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 the output STAGEFM.INTERNAL\_ORG.

**fmsadpt\_T110080\_Load\_stage\_Internal\_Org\_Assoc\_Table**

This job loads the staging INTERNAL\_ORG\_ASSOC table.

The logic for this job is in the t110080\_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 transform Add Standard Dimension ASSOC Rows to add standard rows such as OPENBAL, UNASSIGNED, ALL, and EXT to the internal organization dimension tables

*Note:* When you run this job, you might receive a notification that there are incomplete or out-of-order transformations in the job. You are asked if you want to run the job anyway. Respond **Yes**.

This job uses the following input:

- Fmsr3mir.TKEB
- Fmsr3mir.TKA01
- Fmsr3mir.TKA02
- Intermed.Busarea\_Internal\_Org\_Assoc
- Intermed.Internal\_Org\_Assoc\_from\_Cost\_C

---

## Job Group: T1200 Data Validation

### Overview

The transformation jobs T120000 and T120010 work together to detect discrepancies in the financial document data that is extracted from SAP.

#### ***fmsadpt\_T120000\_create\_Inter\_Financial\_Summary***

The job fmsadpt\_T120000\_create\_Inter\_Financial\_Summary calculates the local currency totals for each financial period. The calculations are based on the columns BUKRS (Company code), BUSAREA (Business Area), HKONT (Account Number), and SHKZG (Debit or 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 the input Intermed.Financial\_Documents (created in E02005\_Extract\_Intermediate\_Table\_merging\_BKPF\_BSEG).

This job creates the output Intermed.FI\_Sum.

The user-written source for this job is t120000\_summ\_fi\_docs.sas.

#### ***fmsadpt\_T120010\_compare\_Financial\_Docs\_with\_GLT0\_Summary***

The job T120010\_compare\_Financial\_Docs\_with\_GLT0\_Summary compares the arithmetical summary of the financial data (calculated in the job T120000\_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 the output Intermed.FI\_Docs\_vs\_GLT0\_\_vert\_SUM\_Diffs.

The user-written source for this job is t120010\_get\_diffs.sas.

## Appendix 1

# Tables Extracted from SAP

Table	Description
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)
CSKB	Cost Elements
CSKS	Cost Center Master Data
CSKT	Cost Center Texts
DD07T	DD: Texts For Domain Fixed Values (Language-Dependent)
FAGL_ACTIVEC	Activation of New General Ledger
FAGLFLEXA	General Ledger: Actual Line Items
FAGLFLEXP	General Ledger: Plan Line Items
FAGLFLEXT	General Ledger: Totals
GLT0GL	account master record transaction figure
KNA1	General Data in Customer Master
KNVH	Customer Hierarchies
LFA1	Vendor Master (General Section)
LFMH	Vendor hierarchy
NRIV	Number Range Intervals
RF011P	Items in financial statement for a selected balance sheet report
RF011Q	Financial statement text
RF011Z	Balance sheet line assignment

Table	Description
SKA1	GL Account Master (Chart of Accounts)
SKAT	GL Chart of Accounts Description
T001	Company Codes
T002	Language Keys (Component BC 118)
T002T	Language Key Texts
T005T	Country Names
T005U	Taxes: Region Key: Texts
T009	Fiscal Year Variants
T009B	Fiscal year variant periods
T009C	Period names
T009T	Fiscal year variant names
T011	Financial Statement Versions
T011T	Financial statement version names
T030	Standard Accounts Table
T030C	Global Standard Account Table
T549A	Payroll areas
T880	Global Company Data (for KONS Ledger)
T8G21	Splitting rule: item to be processed
TCURR	Exchange Rates
TCURT	Currency Code Names
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
TKEBT	Description of operating concern
TLHITT	Vendor hierarchy category (description)



## Appendix 2

# Transformations for SAP

---

<b>Overview of Transformations Provided for SAP</b> . . . . .	<b>77</b>
<b>Add Standard Dimension Rows and Add Standard Dimension ASSOC Rows Transformations</b> . . . . .	<b>78</b>
Introduction . . . . .	78
Using the Add Standard Dimension Rows Transformation . . . . .	78
Using the Add Standard Dimension ASSOC Rows . . . . .	79
<b>Add Parent to Hierarchy Table Based on Level and Position Transformation</b> . . . . .	<b>80</b>
Introduction . . . . .	80
Using the Add Parent to Hierarchy Table Based on Level and Position Transformation . . . . .	81
<b>Format Generator</b> . . . . .	<b>82</b>
Introduction . . . . .	82
Properties Handled by the Format Generator Transformation . . . . .	82
Using the Format Generator Transformation . . . . .	84
<b>User Written Code for Multiple Input/Output Tables Transformation</b> . . . . .	<b>87</b>
Introduction . . . . .	87
Using the User Written Code for Multiple Input/Output Tables Transformation . . . . .	87
<b>Changed Data Extraction Using Date or Time-Stamp and Overlap Transformation</b> . . . . .	<b>88</b>
<b>Date Join Transformation</b> . . . . .	<b>89</b>
<b>Period Consolidation Transformation</b> . . . . .	<b>90</b>
<b>Repeat Record for Date Range Overlap Transformation</b> . . . . .	<b>91</b>
<b>Split the NLS Data Transformation</b> . . . . .	<b>93</b>
Introduction . . . . .	93
Using the Split NLS Data Transformation . . . . .	94
<b>Convert Number Order to Integer Value Transformation</b> . . . . .	<b>94</b>

---

## Overview of Transformations Provided for SAP

This appendix describes the transformations that are provided with the SAS Financial Management Adapter for SAP. This information can help you better understand the jobs that use these transformations. You can also use this information to determine whether the transformation can be used for other jobs that you might create.

## Add Standard Dimension Rows and Add Standard Dimension ASSOC Rows Transformations

### Introduction

The Add Standard Dimension Rows transformation and the Add Standard Dimension ASSOC Rows transformation add standard row values to dimension and dimension ASSOC tables. These two transformations work with each other.

These transformations are necessary, because the staging area 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 value is converted to **UNASSIGNED**.

The following table lists the standard row values and their corresponding meanings or instances:

Row Value	Meaning or Instance
ALL	The root node in the hierarchy of certain dimensions. The hierarchy is stored in the _ASSOC table.
UNASSIGNED	Signifies that the original value in the transaction was blank.
OPENBAL	Signifies that the transaction was obtained from the opening balance summary information, for which there is no detail information for all dimensions.
EXT	A required row for the internal organization dimension.

### Using the Add Standard Dimension Rows Transformation

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

*Note:* This section explains, in detail, the steps that you must complete to perform the corresponding tasks. Other sections describe transformations that are provided by the SAS Financial Management Adapter for SAP, but they do not explain each step. Use this section as a reference and generalize the steps for use with other transformations.

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

1. Add the transformation to a job.
2. Add an output table by dragging and dropping a dimension table.  
The dimension table must have a key column ending in **\_ID**.
3. Verify that all columns in the output table are in the input table, as follows:
  - a. Select the **Mapping** tab.

- b. Right-click the **Column** box under **Target table** and select **Import Columns**.
- c. In the Import Columns window, select all the columns from the output table that you added to the job.

The result should look similar to this image:

4. Select 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** option value adds the corresponding row to the target table.

6. From the **Model table for Target** menu, select the output table that you added to the job.

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

7. Select the **Column Options** tab.
8. From the list of options, select the table columns and map these columns to their predefined roles.

Column	Ends In	Role
Dimension Id	ID	Primary key of the dimension table
Dimension Description	DESC	Describes the dimension
Dimension Name	NM	Name of the dimension

The generated code checks for invalid selections, but not immediately.

9. Select the **Load Technique** tab to review the load step.
 

The generated code creates a table with the requested rows in the correct structure. Add the rows in the created table to the dimension table.
10. To enable the job to run multiple times, modify the default settings for the load step, as follows:
  - a. Set **Load Technique** to **Update**.
  - b. Select an update type.
  - c. Move the **ID** column option from **Available Columns** to **Selected Keys**.

The **ID** column is the only column that must be moved.

### Using the Add Standard Dimension ASSOC Rows

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

1. Drag and drop the Add Standard Dimension ASSOC Rows transformation onto a job.
2. Drag and drop a dimension \_ASSOC\_TYPE table as the input table.
3. Drag and drop a dimension \_ASSOC table as the output table.
4. On the **Mapping** tab, verify that all columns in the output table are in the input table.
5. Select the **Options** tab.
6. Enter **Yes** or **No** as the option value for the following option names:
  - **Add OPENBAL**
  - **Add UNASSIGNED**
  - **Add ALL**
  - **Add EXT**

A **Yes** option value adds the corresponding row to the target table.
7. Enter **Yes** or **No** for the **Convert blank parents to ALL** option.
 

A **Yes** option value changes blank parent IDs to **ALL**. The SAS Financial Management staging area requires that a parent ID is not blank. This requirement includes the root node. In that case, the root becomes its own parent.
8. From the **Target ASSOC table** menu, select the \_ASSOC table that you added to the job.
 

*Note:* Remember that the model table must physically exist before you run the job.
9. On the **Column Options** tab, select available columns and 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.
10. Select the **Load Technique** tab and verify that **Load Technique** is set to **Update**.

---

## Add Parent to Hierarchy Table Based on Level and Position Transformation

### Introduction

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 use the root node as the first record. To establish the parent of a row, a preceding row with a lower level is used. Therefore, preceding rows with identical levels are siblings.

The macro %ADPT\_LEVEL\_TO\_PARENT\_CHILD represents 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

You can 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 Parent to 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 a single output table.

2. Drag and drop an input table and an output table.

The input table must include an ID or key column that contains unique values. The input table must also include a column that stores the level. This level column can contain either numeric- or character-based content. However, character-based content must contain numeric strings. The names of the columns must be associated with the roles in the column options.

The output table must include an ID or key column and a PARENT\_ID column. The PARENT\_ID column must be the same type and length as the ID column. The parent of the root node can be blank.

3. On the **Mapping** tab, verify that all columns in the output table are in the input table. The generated code copies all columns that are in both the input table and the output table. Although mapping is not required in this transformation, resolve any inconsistencies so that the impact analysis tool has more information, if needed.

In the example, four of the original five columns are propagated to the output table. Two new columns, Nr (numeric column) and Parent (character column), are added.

*Note:* Do not add target table columns that do not exist in the input table or that are not assigned a parent role or sequence number. For more information, review the **Column Options** tab.

4. Select the **Options** tab.
5. Enter option values for the following option names:

Option	Description
Output type	<p>Select <b>VIEW</b> if the output table is loaded through a standard load step.</p> <p>Select <b>DATA</b> if the output table is loaded through a load step with code generation turned off (a null loader).</p>

Option	Description
Root level	Use the default value. If the input table contains multiple, disjointed hierarchies (many root nodes), then the root level forces the parent to be blank.

6. Select the **Column Options** tab.
7. From the **Option** menu, select column options and map the columns to their predefined roles.

Column	Description
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 or import this column to the target table 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 table and output table. It should 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 the same type and length as the node ID. You must add or import this column to the target table before you can select it.

## Format Generator

### Introduction

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

You can access information about this transformation's options on the **Options** and **Column Options** tabs.

### Properties Handled by the Format Generator Transformation

#### Table Properties

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.

### ***Format Issues and Properties***

The generated code must address these format issues and properties:

- 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 check out the library.
- The format does not have to be saved in 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.

### ***Code Properties and Requirements***

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 this default library cannot be used. Only libraries that are available in the current metadata repository can be checked. In this case, a message notifies the user to define and use the `LIBNAME LIBRARY`.

### ***Issues for CNTLIN Table Columns***

The Format Generator transformation creates a temporary format `CNTLIN` table from the underlying table. (For a description of the `CNTLIN` table, see 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 address the following issues for the `CNTLIN` table columns:

- The format must have a name (with a maximum of 30 characters) that is unique within the format catalog. Because the Format Generator transformation does not verify uniqueness, a new format overwrites an existing format with the same name.
- The `START` column must be an expression of the columns in the underlying table.
- The `END` column is optional. It 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.
- You can add a row to map missing values into a specified value. The specified value can be any expression. For example, the `START` column can be set to a missing value such as a blank value for characters or a `.` value for numerics. `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 to specify the label for `MISSING`. 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. The `CNTLIN` table column `HLO` must be `o`. `START` can be set to a missing value. `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. It might be 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 consider removing duplicate values. The generated CNTLIN table can contain duplicate START values or overlapping ranges. The START value is inside the range of a START-END combination of another row. The following variables eliminate duplicates in the START value:

- The **NODUP** option generates a **PROC SORT NODUP** by START value. Because this might remove the wrong rows, consider using more controlled 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 rows are best to keep. These additional columns are kept in the temporary CNTLIN table. The subsequent SORT variable has these additional columns in the BY statement. A second DATA step 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

Complete the following 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. That way, any changes to, or reloading of the table automatically regenerates the formats based on the table.

2. Drag and drop the table as the input table.
3. On the **General** tab, change the name of the transformation to **\$FORMATNAME Format Generator** or to a similar name.
4. On the **Options** tab, use the information below to verify and edit your Format Generator transformation properties.

Label	Macro Variable	Description	Additional Information
System Options	OPTIONS	Options on a SAS OPTIONS statement	Type: String Required: N
Format Name	FORMATNAME	Name of the format	Type: String Required: Y
Description	FORMATDESCRIPTION	Description of the format	Type: String Required: N
Format Type (C/I/J/N)	FORMATTYPE	Type of format. Possible values are: <ul style="list-style-type: none"> <li>• <b>C</b>: character format</li> <li>• <b>I</b>: numeric informat</li> <li>• <b>J</b>: character informat</li> <li>• <b>N</b>: numeric format, excluding pictures. Picture formats are not supported</li> </ul>	Type: String Required: Y Valid values: <b>C, N, J, I</b>



Label	Macro Variable	Description	Additional Information
Target Library	TARGETLIBRARY	Target library for the generated format	Type: Metadata library Required: Y
Target Catalog	TARGETCATALOG	Name of the target catalog for the format	Type: String Required: Y Default: FORMATS Constraints: Minimum string length is 1; maximum string length is 32
From Expression	FROMEXPRESSION	Expression to be used for the range's starting value. Use this field to enter an expression, or use the <b>From Column(s)</b> option on the <b>Column Options</b> tab to select columns.	Type: String Required: N
To Expression	TOEXPRESSION	Expression to be used for the range's ENDING value. Use this field to enter an expression, or use the <b>To Column(s)</b> option on the <b>Column Options</b> tab to select columns.	Type: String Required: N
Label Expression	LABELEXPRESSION	Expression to build the informatted or formatted value. Use this field to enter an expression, or use the <b>Label Column(s)</b> option on the <b>Column Options</b> tab to select columns.	Type: String Required: 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.	Type: String Required: N
Label for Missing Values	MISSINGLABEL	String or expression to be used as the label for missing values.	Type: String Required: N
Strategy for Removing Duplicates	DUPSTRATEGY	Specify <b>NODUP</b> to remove duplicates (unordered), <b>FIRST</b> to keep the first duplicate, and <b>LAST</b> to keep the last duplicate. Used with the <b>Sort By Column(s) to Remove Dups</b> option on the <b>Column Options</b> tab.	Type: String Required: N Default: <b>FIRST</b> Valid values: <b>FIRST</b> , <b>LAST</b> , <b>NODUP</b>

Label	Macro Variable	Description	Additional Information
<b>WHERE Clause</b>	WHERECLAUSE	WHERE clause to be applied to the input data set.	Type: String Required: N

5. Select the **Column Options** tab.

You can specify the starting and ending values for the range and the formatted or informatted values in two different ways:

- If the values are a concatenation of columns in the input table, you can use the **From Column(s)**, **To Column(s)**, and **Label Column(s)** options on 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 on the **Column Options** tab. If specified, these values take precedence over the selected columns.

The following information describes the option names and options values that are available. Use this information to verify and edit your Format Generator transformation properties.

Label	Macro Variable	Description	Constraint
<b>From Column(s)</b>	FROMCOLUMNS	Selects columns for the format's starting value. Multiple columns are concatenated. Specify either <b>From Column(s)</b> or <b>From Expression</b> .	No limit on number of selectable columns.
<b>To Column(s)</b>	TOCOLUMNS	Selects columns for the format's ending value. Multiple columns are concatenated. Specify either <b>To Column(s)</b> or <b>To Expression</b> .	No limit on number of selectable columns.
<b>Label Column(s)</b>	LABELCOLUMNS	Selects columns for the informatted or formatted value. Multiple columns are stripped and concatenated with a blank between the columns. Specify either <b>Label Column(s)</b> or <b>Label Expression</b> .	
<b>Sort By Cols To Remove Dups</b>	SORTBYCOLUMNS	Sorts the control data set by the selected columns and removes duplicate keys according to the Strategy for Removing Duplicates option.	

---

## User Written Code for Multiple Input/Output Tables Transformation

### Introduction

The User Written Code for Multiple Input/Output Tables transformation is a generic, user-written code transformation. Use this transformation, rather than the standard user-written code transformation that SAS Data Integration Studio provides, in the following two cases:

- when there is more than one input table or output table
- when the user-written code needs additional LIBNAME 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 the transformation and select the following options:
  - **Add Input** to add drop zones for the input tables
  - **Add Output** for each required output table
3. Drag and drop tables or transformations into the new drop zone.
4. On the **General** tab, replace the default transformation name with a more specific name.
5. Select the **Process** tab and complete the following steps:
  - a. Do not modify the default settings for **Code Generation**.
  - b. Make sure that **Automatically create source code** is selected.
  - c. Review the registered target tables. Specify 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 **&\_OUTPUT** or **&\_OUTPUT0**, then the default target table names can remain unchanged.  
  
Make sure that the load steps of the output tables load the output tables from these temporary target tables.
    - If the source code creates tables with more logical names, register these tables as target tables.
    - If the source code directly loads the output tables that you dragged and dropped onto the transformation, then specify the physical name of the target table and select the library accordingly. When you finish reviewing the properties, change the load step to not automatically generate code.

6. Select the **Mapping** tab.

For the subsequent load step to automatically generate code for the load, import all columns in the output tables into the corresponding target tables.

Although mapping in this transformation is not required to generate code, we recommend that you use the mapping assignments to document the mapping that is made in the source code. This provides the impact analysis tool with more information if needed.

7. Select the **Options** tab.

8. Replace the default **Source code** option value `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.

Four additional parameters can force library assignments to occur before your source code executes:

- PREDEFINED LIBRARY 1
- PREDEFINED LIBRARY 2
- PREDEFINED LIBRARY 3
- PREDEFINED LIBRARY 4

The automatic code generation assigns libraries for all input tables and all libraries that are entered in the Data Location window 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 Transformation

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 a source table. Then, it extracts the data and stores the results in a target table that is conceptually a mirror image of the source table. The target table can have fewer columns than the source table.

This transformation is often required when extracting all of the data from the source table would be a very time-consuming task. For example, extracting 20 million rows of data across an entire network might take many hours. In comparison, extracting only the changed data and merging it with 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 extraction occurs for all records with a key or timestamp that is greater than the largest value that was previously extracted. The resulting data is then appended to the previous data.

The timestamp or key does not need to exclusively increase to work in this transformation. 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. But, the operational system does not create a real datestamp or timestamp for the changed financial transactions. In this case, there is an overlap when two or more financial periods are open, and the data can change. To

manage this overlap, an extraction needs to re-extract data from all open periods. The new records can replace the previous records that have the same keys. If records can be deleted, or if records have keys that are modified, then it is not always possible to match the new records with the previous records. 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 to or merged with the target table, the record must be updated. In addition, a record of previous extractions must be kept to create the next extraction. If the extract, append, or merge step fails, then the data should remain unchanged. This ensures that the next extraction gets the data that would have been extracted if the previous extraction had not failed.

---

## Date Join Transformation

The Date Join transformation creates a SAS Data Integration Studio transformation that works with tables that have date range columns. This transformation joins 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 past periods and extend into future periods. 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 date, including the HR data that is extracted from SAP.

To join tables that have effective dates, consider that the dates are basically keys. A logical match can occur when periods overlap, even if they overlap only partially. To join 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, considering that the date ranges might partially overlap. 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 10 years.

The Date Join transformation and the underlying %DATEJOIN macro provide an efficient alternative. You can join tables accurately by introducing a time dimension. A list of dates is used. Each date selects a record from the contributing tables. This is effective if you want to capture a snapshot of results, such as employee head count at the beginning of each month. However, the drawback is 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. Unless you need to represent only a few fixed dates, use the Date Join transformation and %DATEJOIN macro for exploitation time.

The following example shows how two input tables with a common key value are joined into a single output table. Notice the new begin dates and end dates in the tables, and how the values in Var1 (from Input Table 1) and Var2 (from Input Table 2) are respectively populated.

**Table A2.1** *Input Table 1*

Key	Begin Date	End Date	Var1
1	01Jan06	15Feb06	1
1	16Feb06	20Apr06	2

Key	Begin Date	End Date	Var1
1	01Jun06	31Jul06	3
1	01Aug06	31Aug06	4

**Table A2.2** Input Table 2

Key	Begin Date	End Date	Var2
1	01Feb06	31Mar06	A
1	01Apr06	31Jul06	B
1	01Aug06	30Sep06	C

**Table A2.3** 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 Transformation

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 in which there are no changes in the corresponding, non-key columns that are specified. The similar and sequential data is consolidated into a single date range. The transformation uses the generic transformation generator.

The Period Consolidation transformation uses the %CONSOLIDATE\_PERIOD macro. This macro consolidates into a single record with consecutive date ranges for a key that is passed as a parameter.

The following example shows a data table before and after the Period Consolidation transformation. The key values are represented by 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 and consolidated date ranges.

**Table A2.4** Input Table

	Key1	Key2	Begin Date	End Date	non_key1	non_key 2	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

**Table A2.5** 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

Row 10 is not concatenated to row 9, because there is a gap between the end date of row 9 and the begin date of row 10.

---

## Repeat Record for Date Range Overlap Transformation

The Repeat Record for Date Range Overlap 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 and is adjusted 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 on the later of two dates: either the current date as adjusted with the interval value or the specified date 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 determine the begin date for a time range.

**OVERLAP**

specifies the number of intervals to go back (negative value) or to go forward (positive value), when determining begin date and end date for a time range.

**INTERVAL**

specifies the calendar interval (month, day, or year) to go back, when determining begin date and end date for a time range. This parameter specifies the calendar interval that is used for adding records within a time range. For example, if the INTERVAL parameter is monthly, one record is added for each month between the begin date and end date.

**ALIGNMENT**

specifies if new date values are aligned at the beginning, midpoint, or end of the interval. The default value is BEGINNING.

**INITIAL START**

specifies the begin 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.

**Table A2.6** Input Table

DATECOL	
A	
B	01JAN2006
C	01MAR2006
D	01JUL2006

**Table A2.7** Output Table

DATECOL	
A	01JAN2006
A	01FEB2006



	DATECOL
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

---

## Split the NLS Data Transformation

### Introduction

The Split the NLS Data transformation splits the data of the dimension member table into two tables. One table stores the default language-specific data, and the other table stores the non-default, language-specific data.

You can store names and descriptions in multiple languages and locales. The staging area has a three-character language code that differs from the locale that is used by the SAS Financial Management data mart and the operating system.

The staging area language code can identify differences in the languages and their descriptions in the staging area. Different procedures are used, depending on the type of table. Language code is part of the primary key in a staging area `_ASSOC_TYPE` table. However, there can be multiple records for the same `_ASSOC_TYPE` code in some `_ASSOC_TYPE` tables. The records are differentiated by the language code.

For a dimension member table such as `StageFM.GL_ACCOUNT`, it is more complicated. The language code is not represented in a dimension member table. For example, a dimension member table such as `GL_ACCOUNT` should contain the name and description of the default language code in the `StageFM.CODE_LANGUAGE` table.

If more than one language code is required, then the names and descriptions are added to the GL\_ACCOUNT\_NLS table.

Here is a sample StageFM.CODE\_LANGUAGE table:

LANGUAGE_CD	LOCALE_LANGU GE_CD	LOCALE_COUNTR Y_CD	LOCALE_VARIANT _CD	DEFAULT_LANGU AGE_FLG
zh	Zh	CN		Y
zhx	Zh	TW		N
en	En			N

In the table, names and descriptions for account members for language code zh are added to the GL\_ACCOUNT table. Language codes zhx and en are added to the GL\_ACCOUNT\_NLS table.

Based on this action, the transformation splits the data in the staging area.

### Using the Split NLS Data Transformation

Complete the following steps to use the Split the NLS Data transformation:

1. Drag and drop the Split the NLS Data transformation onto a job.
2. Drag and drop one input table and two output tables.

The input table stores data that corresponds to all of the languages. The two output tables store the default language-specific data and the non-default language-specific data.

3. On the **Options** tab, provide specifications for the input table and output tables.

The following types of tables are required:

- An input table with the prompt **Place the Input Source table**. This table has data that corresponds to all of the languages. In most scenarios, this input table is an intermediate table that contains the dimension member data in all languages.
- An output table with the prompt **Place the NLS table here**. This table specifies the NLS table. This table stores the non-default language-specific data. (Non-default languages are languages that have DEFAULT\_LANGUAGE\_FLG set to blank in the table STAGE\_CODE\_LANGUAGE.)
- An output table with the prompt **Place the Stage Table here**. This table specifies the stage dimension table. This table stores the default language-specific data. (The default language is the language that has DEFAULT\_LANGUAGE\_FLG set to Y in the table STAGE\_CODE\_LANGUAGE.)

---

## Convert Number Order to Integer Value Transformation

Some columns depict numerical order using floating-point numbers. In this case, it might be necessary to convert the numbers to unique integer values so that they are

accepted in a subsequent data model. The Convert Number Order to Integer Value transformation accepts a single table as input.

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

This transformation does not have an output table because all work is performed on the input table.

