SAS® Financial Management Adapter 5.1 for SAP
User’s Guide
# Contents

Contents .................................................................................................................................. iii

**Chapter 1 Introduction to SAS Financial Management Adapter for SAP** ........................................ 1

Introduction............................................................................................................................... 1

Data Flow Architecture ........................................................................................................... 1

Benefits.................................................................................................................................... 3

Configuration and Administration .......................................................................................... 3

Components............................................................................................................................... 3

Job Organization and Names.................................................................................................... 4

Accessibility............................................................................................................................... 5

What’s New in the SAS Financial Management Adapter for SAP ............................................ 5

**Chapter 2 SAP Administration Tasks** .................................................................................. 7

Overview: Customizing for the Installation Environment..................................................... 7

Configuring SAS/ACCESS Interface to R/3 Software ........................................................... 7

**Chapter 3 Customizing the SAS Financial Management Adapter for SAP** ......................... 9

Installation Prerequisites ...................................................................................................... 10

Prepare to Import the SAS Financial Management Adapter for SAP ................................ 10

Define the Environment ......................................................................................................... 10

Deploy SAS Financial Management Adapter for SAP Metadata ........................................ 29

Customizing SAS Data Integration Studio Jobs and Programs ........................................... 30

Customizing the preprocglobal.sas Program ...................................................................... 30

Customizing Non-Leaf Text in the preprocparms13.sas Program ........................................ 41

Setting Up for Internationalization (I18N) ........................................................................... 42

Additional General Information ............................................................................................ 43

**Chapter 4 Transformations Provided for SAP** ......................................................................... 45

Overview of Transformations Provided for SAP ................................................................... 45

Add Standard Dimension Rows and Add Standard Dimension ASSOC Rows
Transformations ...................................................................................................................... 46

Format Generator .................................................................................................................. 59

User Written Code for Multiple Input Output Tables Transformation ................................... 65

Changed Data Extraction Using Date or Time-Stamp and Overlap Transformation ................ 70

Date Join Transformation ...................................................................................................... 70

Period Consolidation Transformation ................................................................................. 72

Repeat Record for Date Range Overlap Transformation ...................................................... 73

Split the NLS Data Transformation ..................................................................................... 75

Additional Generic Transformations ..................................................................................... 77

**Chapter 5 Initialization Jobs** .................................................................................................. 79

Introduction to Initialization Jobs .......................................................................................... 79

fmsadpt_I0050_Initialize_Blank_Business_Area_Table.......................................................... 79
# Chapter 6 Extraction Jobs

Extraction Jobs ................................................................. 81
Job Group: E00 Extraction of T Tables ........................................ 82
Job Group: E01 Extraction of Master Tables ................................. 87
Job Group: E02 Extraction of Transaction Tables ......................... 89
Job Group: E03 Extraction of Summary Tables ............................ 92
Job Group: E04 Extraction of Hierarchies and Structures ............. 93
Job Group: E05 Extraction of Miscellaneous Tables ..................... 94

# Chapter 7 Transformation Jobs

Introduction to Transformation Jobs ......................................... 98
Job Group: T0000 SAS Supplied Tables .................................... 99
Job Group: T0001 Transformations for Common Tables ................ 100
Job Group: T0002 Time Dimension .......................................... 102
Job Group: T0003 Exchange Rates ........................................... 106
Job Group: T0004 Analysis Dimension ...................................... 109
Job Group: T0005 Account Dimension ...................................... 109
Job Group: T0006 Cost Center Dimension ................................. 118
Job Group: T0007 Profit Center Dimension ............................... 120
Job Group: T0008 Currency Dimension ..................................... 123
Job Group: T0009 Financial Documents ..................................... 123
Job Group: T1000 External Org Dimension ............................... 128
Job Group: T1100 Internal Org Dimension ............................... 130
Job Group: T1200 Data Validation .......................................... 135

# Chapter 8 Integrating into the SAS Detail Data Store

Integrating Data into the SAS Detail Data Store ......................... 137
Customizing the Validation Steps in the Load SAS Detail Data Store Jobs ........................................... 138

# Appendix 1 Tables Extracted from SAP

Tables Extracted from SAP ................................................... 141

# Appendix 2 Job Groups and SAS Solution Data Marts

Job Groups and SAS Solution Data Marts ............................... 143

# Appendix 3 Recommended Reading

Recommended Reading ....................................................... 145
Introduction

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

Data Flow Architecture

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

Source system data must flow through a series of ETL steps to be converted into a usable format for SAS solution data marts such as SAS Financial Management. The figure details the basic architecture of this data flow. The SAS Financial Management Adapter for SAP acts in the first step.
The following steps describe the data flow:

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

   Once the source data is loaded into the staging area, it is ready for ETL processing that loads it into the DDS. ETL processing includes these actions:
   - validating the data
   - creating surrogate keys instead of operational system IDs to identify the data
   - introducing date-and-time stamps to reflect data validity if differences in dimensional arise

2. The transformed data is stored in the SAS Detail Data Store, a lightly denormalized, relational data model that provides storage flexibility. The SAS Detail Data Store is a specific detail data model that represents a standard business function and supports SAS solution data marts. Some of the data that the SAS Detail Data Store captures contains current and historical information. This includes temporal data (event data that occurs at a particular date and time, such as an account inquiry) and nontemporal data (non-event data such as a customer or a financial account).
Once data is cleansed, validated, and stored in the SAS Detail Data Store, it is ready for transforming and loading to a data mart model.

### Benefits

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

The SAS Financial Management Adapter for SAP includes SAP knowledge that streamlines the data conversion process. This built-in knowledge reduces your dependence on SAP resources that help you understand how SAP works. It saves you time and valuable SAP resources, so that the resources are available to work on other projects.


### Configuration and Administration

The SAS Financial Management Adapter for SAP is configured and managed using the SAS Deployment Wizard and SAS Data Integration Studio:

- The SAS Deployment Wizard enables you to specify the adapter configurations that are required for your unique business environment. You can use this tool to specify parameters during initial installation or import.
- After you configure the SAS Financial Management Adapter for SAP, you can use SAS Data Integration Studio to manage and customize the adapter.

### Components

The SAS Financial Management Adapter for SAP includes components that you can manage and configure using the SAS Deployment Wizard. This wizard enables you to configure and manage the following components that you need to import a SAS Financial Management Adapter for SAP environment:

**SAS Data Integration Studio jobs**

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

**Transformation jobs**

- create intermediate tables or populate tables in the SAP staging area.
Job Organization and Names

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

- extraction jobs
- transformation jobs

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

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

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

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

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

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

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

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

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

What’s New in the SAS Financial Management Adapter for SAP

Overview

SAS Financial Management Adapter for SAP leverages the SAS 9.2 intelligence platform. The new features and enhancements in the SAS 9.2 intelligence platform improve the deployment process of the SAS Financial Management Adapter for SAP. With this release, SAS Financial Management Adapter for SAP provides additional platform support. Some new jobs are also included to suit specific requirements.

General Enhancements

SAS Financial Management Adapter for SAP has the following general enhancements:

- SAS Financial Management Adapter for now SAP supports SAS Financial Management 5.1 and 5.2.
- SAS Financial Management Adapter for SAP supports these SAP versions:
  - SAP R/3 4.7 (SAP Kernel 620)
  - SAP - ECC 5.0 (SAP Kernel 710)
  - SAP - ECC 6.0 (ERP 6.0) (SAP Kernel 710)
- Installation and Configuration tasks are automated by the SAS Deployment Wizard. For more information about the SAS Deployment Wizard, see the SAS 9.2 Intelligence Platform documentation at http://support.sas.com.
Platform Support

The adapter supports the platforms that are supported by the SAS Financial Management solution. The following platforms are supported:

- **Windows**
  - WIN (Windows 32 bit on x86)
  - WX6 (Windows 64 bit on x86)
- **UNIX**
  - S64 (Solaris on SPARC)
  - R64 (AIX on PowerPC)

For more details, see the Platform Support Matrix for the SAS Financial Management solution.

New Jobs

SAS Financial Management Adapter for SAP contains the following new jobs:

- fmsadpt_E00080_Extract_New/GL_Control_Table
- fmsadpt_E03000_Extract/GL_Balance_from_New/GL_Tables (New GL)
- fmsadpt_E03010_Extract/Split_Docs_from_FAGLFLEXA (New GL)

These jobs are designed to support implementation of new GL accounts. For more details see “Job Group: E03 Extraction of Summary Tables” in the chapter “Extraction Jobs”.
Overview: Customizing for the Installation Environment

The SAS Financial Management Adapter for SAP requires customization to work in your unique installation environment. When you make these changes for your site, you must configure the SAS Data Surveyor for SAP software, and also customize the SAS Financial Management Adapter for SAP for the assigned libraries in your installation environment.

Configuring SAS/ACCESS Interface to R/3 Software

Install Transports and Customize the Installation Environment

The SAS Financial Management Adapter for SAP uses SAS/ACCESS Interface to R/3. This software requires extensive installation setup before use. For detailed installation instructions and configuration information, see the SAS/ACCESS Interface to R/3 Installation documentation that is included in your installation package.

In addition to configuring SAS/ACCESS Interface to R/3 software, the following transport must be copied and installed on the SAP server to accommodate the SAS Financial Management Adapter for SAP:

Transport: SAPKD92010INSAS
Object: Development class /SAS/ADDDS
Applied to: SAP R/3 Systems
Objects: Function group /SAS/ADDDS1 includes function modules /SAS/FI_IMPORT_BAL_SHEET_POS and /SAS/FI_IMPORT_BAL_SHEET_TEXT
Purpose: ADDDS1 retrieves balance sheet information

You can locate this transport in the \fmadaptsap\sasmisc directory that is relative to the location where your SAS Foundation software is installed (for example, x:\Program Files\SAS\SASFoundation\9.2\fmadaptsap).
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 needs to configure the SAS Financial Management Adapter for SAP so that only authorized SAP users can log on.
### Customizing the SAS Financial Management Adapter for SAP

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Prerequisites</td>
<td>10</td>
</tr>
<tr>
<td>Prepare to Import the SAS Financial Management Adapter for SAP</td>
<td>10</td>
</tr>
<tr>
<td>Define the Environment</td>
<td>10</td>
</tr>
<tr>
<td>Define a User</td>
<td>10</td>
</tr>
<tr>
<td>Define an SAP Library and an SAP Server</td>
<td>15</td>
</tr>
<tr>
<td>Test the SAP Connection</td>
<td>25</td>
</tr>
<tr>
<td>Configure the Properties File</td>
<td>27</td>
</tr>
<tr>
<td>Set Install Folder Authorizations</td>
<td>28</td>
</tr>
<tr>
<td>Deploy SAS Financial Management Adapter for SAP Jobs and Tables Metadata</td>
<td>29</td>
</tr>
<tr>
<td>Overview of Install Process</td>
<td>29</td>
</tr>
<tr>
<td>Incremental Install of the SAS Financial Management Adapter for SAP</td>
<td>29</td>
</tr>
<tr>
<td>Customizing SAS Data Integration Studio Jobs and Programs</td>
<td>30</td>
</tr>
<tr>
<td>Overview of the preproglobal.sas Program</td>
<td>30</td>
</tr>
<tr>
<td>Locating Files</td>
<td>31</td>
</tr>
<tr>
<td>SAP System Parameters</td>
<td>32</td>
</tr>
<tr>
<td>Introduction</td>
<td>32</td>
</tr>
<tr>
<td>Data Source</td>
<td>33</td>
</tr>
<tr>
<td>SAP Queries</td>
<td>33</td>
</tr>
<tr>
<td>SAP Language Codes</td>
<td>33</td>
</tr>
<tr>
<td>SAP Client</td>
<td>34</td>
</tr>
<tr>
<td>Common Extraction Parameters</td>
<td>35</td>
</tr>
<tr>
<td>E-Mail Address for the Recipient of Error Reports</td>
<td>35</td>
</tr>
<tr>
<td>Cost Center and Profit Center Hierarchies</td>
<td>35</td>
</tr>
<tr>
<td>Company Codes</td>
<td>35</td>
</tr>
<tr>
<td>Euro Currency Conversion</td>
<td>35</td>
</tr>
<tr>
<td>SAS Financial Management Parameters</td>
<td>36</td>
</tr>
<tr>
<td>Chart of Accounts</td>
<td>36</td>
</tr>
<tr>
<td>Digits in the Account Number</td>
<td>37</td>
</tr>
<tr>
<td>Operating Concern</td>
<td>37</td>
</tr>
<tr>
<td>External Organizations</td>
<td>37</td>
</tr>
<tr>
<td>Controlling Areas</td>
<td>37</td>
</tr>
<tr>
<td>SAP R/3 Table GLT0 Data Subsets</td>
<td>38</td>
</tr>
<tr>
<td>Period Type for Exchange Rates</td>
<td>38</td>
</tr>
<tr>
<td>Financial Statement</td>
<td>39</td>
</tr>
<tr>
<td>Fiscal Year Variant</td>
<td>39</td>
</tr>
<tr>
<td>Set the Extraction Starting Date</td>
<td>40</td>
</tr>
<tr>
<td>Determine Current Fiscal Year</td>
<td>41</td>
</tr>
<tr>
<td>Buffer for Re-Extraction</td>
<td>41</td>
</tr>
<tr>
<td>Customizing Non-Leaf Text in the preproparms13.sas Program</td>
<td>41</td>
</tr>
<tr>
<td>Setting Up for Internationalization (I18N)</td>
<td>42</td>
</tr>
<tr>
<td>Steps for Setting Up for Internationalization</td>
<td>42</td>
</tr>
<tr>
<td>Warning Messages</td>
<td>43</td>
</tr>
<tr>
<td>Additional General Information</td>
<td>43</td>
</tr>
</tbody>
</table>
Installation Prerequisites

As part of the installation of SAS Financial Management Adapter for SAP, first install the dependent (required) SAS software listed below. Refer to the installation instructions of the specific product for more information.

- second maintenance release for SAS 9.2
- SAS Data Surveyor for SAP 4.3 (includes SAS/ACCESS to R/3)

If you are using the second maintenance release for SAS 9.2, then update the SAS Platform from the second maintenance release to the third maintenance release, before you begin to install the SAS Financial Management Adapter for SAP.

After you update SAS 9.2 with updates from the third maintenance release, run the setup.exe with the nosasupdate option. SAS 9.2 is already updated with the third maintenance release as part of the above prerequisite task.

Prepare to Import the SAS Financial Management Adapter for SAP

After the install process is complete, the next step is to prepare your system environment. Perform the following tasks before you access the SAS Deployment Wizard to configure the SAS Financial Management Adapter for SAP:

- Define user and SAP Login details in SAS Management Console in the Foundation repository.
- Define an SAP library and an SAP server in SAS Management Console.
- Test and verify the resulting connection in SAS Data Integration Studio.
- Configure the properties file for SAS Financial Management Adapter for SAP.

These prerequisite tasks are required to ensure that the SAS Financial Management Adapter for SAP works properly when it is configured. The remaining sections in this chapter provide instructions on how to complete the required tasks.

Define the Environment

Define a User

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

1. Open SAS Management Console.

2. Right-click User Manager in the hierarchy tree in the left pane of the SAS Management Console window, and select New→Group.
A New Group Properties dialog box appears.
3 Enter **SAP Group** in the **Name** field.

4 Click the **Members** tab.
5 Select sasdemo and SAS Trusted User in Available Identities.

6 Click the arrow button that points to the Current Members pane.

   Your selections appear in the Current Members pane. You can add other users to the SAP User Group, as required.

7 Click the Accounts tab.
8 Click **New**. The New Login Properties dialog box appears.

![New Login Properties dialog box](image)

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

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

**Define an SAP Library and an SAP Server**

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

1. Open SAS Management Console.

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

3. Right-click **Libraries** and select **New Library**. A New Library Wizard dialog box appears.
4 Select **SAP Library** in the **Enterprise Applications Data** folder.
5  Click **Next** to name the new SAP library.

6  Enter a name and description for the SAP library and click **Next**.
7 Select **SASApp** as the SAS server. Click **Next** to specify the library properties.
8. Enter **FMSSAP** in the **Libref** field.

**FMSSAP** is the default value that the imported metadata uses.

9. Click **Next** to select a database server.
10 In the **Database Server** field, select a database server that contains parameters that connect to the SAP server.

11 Complete the following steps to create an appropriate database server if one is not already available in the field list:

   a Click **New** to open the New SAP Server Wizard.
b Enter a name and description for the server.

c Click Next to enter information about your SAP software.
Define the Environment  Chapter 3

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

![New Server Wizard]

- Click **Next** to enter connection properties.
- Select **SAPAuth** in the **Authentication Domain** field.
Specify the SAP client number and SAP language.

Select the connection type (Application Server) that corresponds to a normal SAP GUI connection for your system.

Click the **Options** button that corresponds to your selection and enter the required values.
12 Scroll down the page and click the Advanced Options button. If you have defined batch servers or processes, you might need to specify some advanced options. Click OK.

13 Click Next to display a summary page that lists your new specifications.
14 Review the set parameters and click **Finish** to save the new SAP server and return to the New Library Wizard.

Based on your system specifications up to this point, the wizard might prompt you to select a SAS server where the library is to be assigned. If this page appears, select **SASMain** and click **Next** to continue.

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

16 Click **Finish** to save.

## Test the SAP Connection

To test the SAP connection, complete the following steps:

1. Open SAS Management Console.
2. Open the connection profile for the user. Select Tools → Extract from R/3. The Source Designer dialog box opens.
3 In the Configured SAP Server field, select the name of the SAP server that you defined in SAS Management Console.

4 Click Test SAP Server Connection.

5 If the connection is successful, then the following message appears

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

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

![SAS Log Output]

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

8 Click **Cancel** to exit the Register Tables.

---

**Configure the Properties File**

To configure the properties file for SAS Financial Management Adapter for SAP, complete the following steps:

1 Open the properties file “fmadaptdata.properties” located in the default product configuration folder. If the installation is done at the default location, then the properties file is located at “C:\Program Files\SAS\SASFinancialManagementAdapterforSAPData\5.1\Config”.

2 At the end of the file, replace the value of the property “fmadaptdata.simserver.name” from “TBD” to the name of the SAP Server created in the previous section.

The default property value is given as “TBD”. The value must be changed to match the SAP Server that you have created.
Set Install Folder Authorizations

To avoid deployment errors, perform the steps below in the order in which they are listed:

1. Grant full permissions to the Install folder CustomSASCode, as the deployment process writes the custom SAS code into this folder.

*Note:* This is a mandatory step. If you ignore the above step and proceed with the deployment, the following load content failed error message appears.
When you import the metadata for the SAS Financial Management Adapter for SAP using the SAS Deployment Wizard in the SAS Financial Management Adapter for SAP Dependencies window, the parameter **SAS Application Server Context** has the value **SASMeta** selected by default. Change it to **SASApp**.

**Note:** The screen above appears only when you install the SAS Financial Management Adapter for SAP on the SAS9.2M3 environment.

---

### Deploy SAS Financial Management Adapter for SAP Metadata

#### Overview of Install Process

In this release, the deployment is done automatically by the SAS Deployment Wizard. In the previous version, the metadata for objects (in this case, jobs and tables) for the SAS Financial Management Adapter for SAP was deployed manually.

#### Incremental Install of the SAS Financial Management Adapter for SAP

If you have installed the SAS Financial Management and plan to do an incremental install of the SAS Financial Management Adapter for SAP, there are specific steps that you must follow:

1. After you install and configure SAS Financial Management, stop all SAS services.
2. Set the **Startup** to **Manual** mode and restart the server.
4. On the Select Deployment Task page, select **Install SAS Software**.

5. On the Select Deployment Type page, make sure that the **Install SAS Software** option is checked and the **Configure SAS Software** option is unchecked.

6. On the Select Products to Install page, select **SAS Foundation and SAS Financial Management Adapter for SAP**.

7. Specify the SAS Foundation components through the interactive install prompt. Select **Base SAS** and **SAS Financial Management Adapter for SAP**.

   *Note*: If Base SAS update fails to update, then select **Yes** when prompted to continue and complete the installation.

8. After the installation is complete, restart your machine. Also, start all services and change the **Startup** from **Manual** to **Automatic**.

   *Note*: If the OLAP server, Table server, and SAS Shared Server fail to start, use a SAS Installation Data (SID) file that contains SAS Financial Management, SAS Financial Management Adapter for SAP, and other required components.

9. After you implement the new SID file, ensure that all SAS services have started successfully.

10. Restart the SAS Deployment Wizard. Check the **Configure SAS Software** option on the Select Deployment Type page, and proceed with the configuration tasks.

11. On the Select Products to Configure page, select **SAS Add-on for SAP** and **SAS Financial Management Adapter for SAP Data**.

---

### Customizing SAS Data Integration Studio Jobs and Programs

To customize the SAS Financial Management Adapter for SAP to fit into your unique environment, you must review and change several jobs in SAS Data Integration Studio. Several SAS programs within these jobs also require your review and changes. Although many of the jobs and programs might not require changes, you must review each one to ensure the results that you want. This section provides information about each job and program that requires your review and changes. It describes each job in SAS Data Integration Studio and explains whether the job or program requires changes to work successfully.

---

### Customizing the preprocglobal.sas Program

#### Overview of the preprocglobal.sas Program

SAS Financial Management Adapter for SAP jobs use several global macro variables that are initialized in the SAS program `preprocparms12.sas`, which is subsequently...
called in the program `preprocglobal.sas`. This program serves as the preprocessing step in each job.

Make appropriate changes in the following section of the program to suit your environment.

```
libname fmamacro 'C:\Program Files\SAS\SASFoundation\9.2\fmadaptsap\cmacros';
options MSTORED SASMSTORE=fmamacro;
```

Use this section of the program given above to change the path to the Adapter macro catalog as per your host requirement.

*Note:* Be sure to make a backup copy of the program before changing it.

---

**Locating Files**

The `preprocparms12.sas` program is located in the install directory. This file can be found at the location that you specified in the field *Enter the path for SAS source code*, on the Enter Parameters page of the SAS Deployment Wizard.

The `preprocparms12.sas` program file is called by the `preprocglobal.sas` program. The `preprocglobal.sas` program must be edited to suit the local environment.
Customizing the preprocglobal.sas Program

Chapter 3

After you open preprocparms12.sas, you can see the banner for the SAS program file. Edit this file to suit your local environment.

SAP System Parameters

Introduction

The subtopics below discuss sections of the preprocparms12.sas file.
Data Source

Below the banner, the first section of `preprocparms12.sas` designates the data source location.

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

The macro variable `DDS_SOURCE_SYSTEM_ID` is a three-character ID that tells the SAS Detail Data Store 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.

This variable populates the `SOURCE_SYSTEM_CD` column that occurs in multiple tables of the SAS Detail Data Store. If data originates from multiple SAP systems, multiple extracts are necessary, with each extract having its own `preprocparms12.sas` file and unique value set for the variable `DDS_SOURCE_SYSTEM_ID`.

SAP Queries

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

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

The statement `options debug=dbms_select` shows detailed information about queries that are passed directly to SAP. Generally, the SAP queries section 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.

SAP Language Codes

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

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

Set the macro variables `spras` and `lang` to the single-character language code that corresponds to the main language that SAP uses. This is also the language that the program uses to maintain texts. Quoted and unquoted versions exist for easy inclusion into code and for making the code easy to understand and read.
Use the following language code table to determine the valid language value for your local environment.

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

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.

The new macro variable DEFAULT_LANG is a two-digit ISO code that sets the default language flag in the STAGE_CODE_LANGUAGE table.

**SAP Client**

The value of the LET CLIENT macro variable is the SAP client (column MANDT) that is part of the SAP R/3 logon in most cases.

```sas
%let client = 800 ; * = MANDT ;
```
Common Extraction Parameters

E-Mail Address for the Recipient of Error Reports

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.

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

Cost Center and Profit Center Hierarchies

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.

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

Company Codes

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.

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

Euro Currency Conversion

The SAS 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.
The macro variable CONVERT_CURRENCY_TO_EURO needs to have a value of **YES** or **NO**. 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**.

*Note: 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.*

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

---

**SAS Financial Management Parameters**

**Chart of Accounts**

The SAS Financial Management Adapter for SAP uses one chart of accounts at a time. You must specify this chart of accounts in the macro variables KTOPL and _KTOPL.

A chart of accounts in SAP is a list of all general ledger accounts that one or more company codes 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.

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

If you need more than one chart of accounts for your environment, you must set up additional instances of the SAS 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, as shown in the data source section of the preprocparms12.sas file.

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. Also, the KTOPL value must appear in the KTOPL column of the T001 table.

After you have determined the correct variable for the chart of accounts, enter the value. You must enclose the KTOPL value in single quotation marks. Do not enclose the _KTOPL value in double quotation marks.
Digits in the Account Number

The value for SAKLN determines the number of digits in the account number. If you do not know this value, 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.

Remember that valid account numbers often contain leading zeros. You can delete these leading zeros from the account digit value.

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

Operating Concern

The preprocparms12.sas program file enables you to specify the operating concern in the macro variable OPERATING_CONCERN and a corresponding country code in the macro variable OPERATING_CONCERN_COUNTRY_CD.

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

To 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.

External Organizations

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

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

The default SAP value for EXT_ORG_ASSOC_TYPE_CD is A. For other valid values, see the contents of the SAP R/3 tables, THITT and TLHITT.

Controlling Areas

You can use controlling areas to form the second level of the internal organizational hierarchy. You can use them to select cost-center and profit-center standard hierarchies. You must specify the corresponding country for each controlling area.
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="&OPERATING_CONCERN".

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.

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.

**SAP R/3 Table GLTO Data Subsets**

Use the macro variables RLDNR, RRCTY, and RVERS to subset the data that is extracted from the SAP R/3 table GLTO. The GLTO table contains figures that are summed by transaction.

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

  * (apply WHERE clause where rldnr="&rldnr");

Although the default settings are generally acceptable, check them against valid values.

- For valid ledger values, see the contents of the SAP R/3 tables T881 and T881T.
- For the record type value, 0 is the typical value for actual data.
- For valid version values, see the contents of the SAP R/3 table T894. The version descriptions are in the T894T table.

**Period Type for Exchange Rates**

Use the macro variable PERIOD_TYPE_CD_FOR_EXRATES to specify the period type that will be associated with exchange rates. A list of valid values is in the SASHELP.SAS_PERIOD_TYPE table that is provided by SAS.

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

The SAS Financial Management Adapter for SAP handles one financial statement at a time. You must specify this financial statement in the macro variables VERSN and _VERSN as shown:

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

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 VERSN values. You must enclose the VERSN value in single quotation marks. Do not enclose the _VERSN value in double quotation marks.

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 FINANCIAL_STATEMENT_ROOT_TXT.

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

Fiscal Year Variant

SAP enables companies to have multiple fiscal year variants, although most companies 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.

The macro variable USE_FISCAL_YR_VARIANT_IN_TIME can control behavior based on the following two values:

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

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

1 Use the macro variables START and STARTFINYEAR 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.

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

* The following macro variables determine from which date financial transactions should be extracted from SAP. This is IMPORTANT when initializing the data during the initial load phase. In subsequent extracts, only new/changed data is extracted;
%let _start = 01JAN2000;   * Start date - First extraction to take transactions after this date;
%let _startfinyear = 2000;
%let _opening_balance_period=200001; * Opening balance period.
   * This must be in the format YYYYMM - It should usually be the period of _start macro variable above.
   * If it is set to blank, then No opening balance is extracted.

2 Set the extraction variables based on your environment.

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

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

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

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

Buffer for Re-Extraction

The macro variable OVERLAP enables you to re-extract data that has changed during the overlap since the last extraction of financial transactions.

The OVERLAP value is specified in number of days. Do not set this value to less than 2 because that is the minimum time needed to handle overnight processing and time zone differences. The default value is 4 days.

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

Customizing Non-Leaf Text in the preprocparms13.sas Program

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 called by the preprocgloba1.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.

## Setting Up for Internationalization (I18N)

### Steps for Setting Up for Internationalization

If your SAP system is a Unicode system, configure the SAS environment to point to the SAS Unicode Server. SAP data is on the Unicode pages, and you must use the appropriate transcoding while extracting the SAP data using the Data Surveyor.

To configure the SAS Financial Management Adapter for SAP in an Internationalization environment, perform the following steps:

1. Modify the SASV9.CFG file to access the SAS Unicode Server. The default path for the SASV9.CFG file is `C:\Program Files\SAS\SASFoundation\9.2`.

2. Open the SASV9.CFG file to make changes. Replace `en` with `u8` in the code below.

   ```
   -CONFIG "$C:\Program Files\SAS\SASFoundation\9.2\nls\u8\SASV9.CFG"
   ```

3. Restart the SAS Services after the code changes are complete.

4. Change the above SASV9.CFG file to point to the appropriate encoding format that suits your encoding requirement.

   *Note:* The example given above applies in a Windows environment only. You need to apply the above changes appropriately, as per your site host.

5. Use the appropriate SAS supplied tables suitable to your environment.
For example, if you use the Adapter with `euc-cn` encoding to load the data from SAP into the solution. Ensure that the tables supplied by SAS, such as `SAS_CURRENCY`, contain Chinese text. This is because the Adapter jobs load the stage 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.

*Note:* This guideline also applies to `SAS_COUNTRY_ISO3166`, another table supplied by SAS.

---

**Warning Messages**

The Adapter is packaged with predefined SAP metadata that is suitable for both Unicode and non-Unicode systems. Due to this, you might see warning messages that are related to the base and data length mismatch in the Adapter jobs, depending on the encoding option that you have used.

For example, when you run the Adapter jobs in `W LATIN1` encoding, you see a warning message as shown below. However, warning messages do not appear if you use Unicode encoding or another double-byte encoding like `euc-cn`.

These warning messages are displayed for your information only, and can be safely ignored. They have no impact on the data extraction or loading.

---

**Additional General Information**

Some Adapter jobs use a new feature from the latest version of the SAS Data Integration Studio that enables you to reuse a table multiple times in a job data flow. This capability was not available in the previous version of SAS Data Integration Studio. Because of this feature, you might see control flow warnings when you try to execute a job. These warning messages appear due to the default behavior of the table loader transform.
The warning messages are displayed for your information only and can be safely ignored. They have no impact on the data extraction or loading.
Overview of Transformations Provided for SAP

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

There are two types of transformations that are provided and used by the SAS Financial Management Adapter for SAP:

- transformations that are specific to applications related to SAP
- generic transformations that are not specific or exclusive to applications related to SAP
This chapter describes these transformations. This information can help you better understand the existing SAS Financial Management Adapter for SAP jobs that use these transformations. You can 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 SAS Detail Data Store 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 value.

The following list describes the standard row values and their corresponding meanings or instances:

- **ALL**: represents the root node in the hierarchy of certain dimensions. The hierarchy is stored in the _ASSOC table.
- **UNASSIGNED**: implies that the original value in the transaction was blank.
- **OPENBAL**: implies that the transaction was obtained from the opening balance summary information, for which there is no detail information for all dimensions.
- **EXT**: is 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 the specific 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 the specific steps. 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 input and an output table.

To add an input and an output table, drag and drop a dimension table. The dimension table must have a key column ending in _ID.
3. Select the **Mappings** tab.

Make sure that all columns in the output table are in the input table. Complete the following steps to check the tables:
- Right-click Column under Target tables.
- Select Import Columns.

4. Select all of the columns from the output table that you just dragged and dropped.

The result should look similar to this image:
5 Click 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 From the **Model table for Target** menu, select the table that you dragged and dropped.

*Note:* 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.

8 Select the **Options** tab.
9. From the list of options, select the table columns and map these columns to their predefined roles.
   - Dimension Id column - ends in ID and is the primary key of the dimension table
   - Dimension Name Column - ends in NM and is the name of the dimension
   - Dimension Description column - ends in DESC and describes the dimension
   The generated code checks for invalid selections, but this checking does not happen immediately.

10. Right-click the Update Loader transformation and select Properties.
    This transformation is located below the Add Standard Dimension Rows transformation.

11. In the Upload Loader Properties window, 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 these rows to the dimension table.

12 Enable the job so that it can run multiple times.
   To do this, modify the default settings for the load step.

13 Set Load Style to Update/Insert.

14 Select an update type.

15 Select the ID column option from Available column(s) and click the right arrow to move it to Column(s) to match.
   The ID column is the only column that you must move to the Column(s) to match area.
Using the Add Standard Dimension ASSOC Rows

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

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. Select the **Mappings** tab.

Make sure 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 Convert blank parents to ALL.

A **Yes** option value changes blank parent IDs to **ALL**. The DDS 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 dragged and dropped.

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

10 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.

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

12 Set **Load Style** to **Update/Insert**.

13 Add Parent to Hierarchy Table Based on Level and Position.

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 format.

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. It returns 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 Select the **Mappings** tab.

Make sure that all columns in the output table (**Source table**) are in the input table (**Target 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 **Options** tab.

4 Select the **Options** tab.

5 Enter option values for the following option names:

**Output type** Select **VIEW** if the output table is loaded through a standard load step. Select **DATA** if the output table is loaded through a load step with code generation turned off (a null loader).
**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 **Options** tab.

From the **Options** menu, select column options and map the columns to their predefined roles.

- **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 tab.

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 and tasks:

- specifications such as format, and library catalogs
- tasks such as 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 does not have to be saved in an existing catalog. 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 server can be used. 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 transformation 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 (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.
- A row can be added 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 . (period) value for numerics. LABEL can be set to MISSING, UNKNOWN, or an expression such as PUT(‘MISSING’, $FORMAT_WORDS). In this case, 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 0. The START column can be set to a missing value. The LABEL column can be set to OTHER, UNKNOWN, or an expression such as PUT(‘OTHER’, $FORMAT_WORDS). In this case, enter the expression to specify the LABEL for MISSING.
- The rows in the CNTLIN: The 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 Select the **General** tab.

Change the name of the transformation to $FORMATNAME Format Generator or to a similar name.

4 Select the **Options** tab.

The option names and option values are listed below. Use this information to verify and edit your Format Generator transformation properties.

- **Label**: System Options
- **Macro Variable**: OPTIONS
- **Description**: Options on a SAS OPTIONS statement.
Type: String
Required? N

Label: Format Name
Macro Variable: FORMATNAME
Description: Name of the format.
Type: String
Required? Y

Label: Description
Macro Variable: FORMATDESCRIPTION
Description: Description of the format.
Type: String
Required? N

Label: Format Type (C/I/J/N)
Macro Variable: FORMATTYPE
Description: Type of format. Possible values are C for character format, I for numeric format, J for character format, and N for numeric format (excluding pictures). Picture formats are not supported.
Type: String
Required? Y
Default: C
Valid Values: C N J I

Label: Target Library
Macro Variable: TARGETLIBRARY
Description: Target library for the generated format.
Type: Metadata Library
Required? Y

Label: Target Catalog
Macro Variable: TARGETCATALOG
Description: 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.

Label: From Expression
Macro Variable: FROMEXPRESSION
Description: Expression to be used for the starting value of the range. Use this field to enter an expression or use the From Column(s) option on the Options tab to select columns.
Type: String
Required? N

Label: To Expression
Macro Variable: TOEXPRESSION
Description: Expression to be used for the ending value of the range. Use this field to enter an expression or use the To Column(s) option on the Options tab to select columns.

Type: String
Required? N

Label: Label Expression
Macro Variable: LABELEXPRESSION
Description: Expression to build the unformatted or formatted value. Use this field to enter an expression or use the Label Column(s) option on the Options tab to select columns.

Type: String
Required? N

Label: Label for Other Values
Macro Variable: OTHERLABEL
Description: 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: Label for Missing Values
Macro Variable: MISSINGLABEL
Description: String or expression to be used as the label for missing values.

Type: String
Required? N

Label: Strategy for Removing Duplicates
Macro Variable: DUPSTRATEGY
Description: Specify NODUP to remove duplicates (unordered), FIRST to keep the first duplicate, and LAST to keep the last duplicate. Used with the Sort By Column(s) to Remove Dups option on the Options tab.

Type: String
Required? N
Default: FIRST
Valid Values: FIRST LAST NODUP

Label: WHERE Clause
Macro Variable: WHERECLAUSE
Description: WHERE clause to be applied to the input data set.

Type: String
Required? N

Select the Options tab.

You can specify the starting and ending values for the range and the formatted or unformatted 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 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 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.

<table>
<thead>
<tr>
<th>Label</th>
<th>From Column(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro Variable</td>
<td>FROMCOLUMNS</td>
</tr>
<tr>
<td>Description</td>
<td>Selects columns for the starting value of the format. Multiple columns are concatenated. Specify either From Column(s) or From Expression.</td>
</tr>
<tr>
<td>Constraint</td>
<td>No limit on number of selectable columns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Label</th>
<th>To Column(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro Variable</td>
<td>TOCOLUMNS</td>
</tr>
</tbody>
</table>

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.
Description: Selects columns for the ending value of the format. Multiple columns are concatenated. Specify either To Column(s) or To Expression.
Constraint: No limit on number of selectable columns
Label: Label Column(s)
Macro Variable: LABELCOLUMNS
Description: Selects columns for the unformatted or formatted value. Multiple columns are stripped and concatenated with a blank between the columns. Specify either Label Column(s) or Label Expression.
Constraint: No limit on number of selectable columns
Label: Sort By Cols to Remove Dups
Macro Variable: SORTBYCOLUMNS
Description: 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 in the following two cases, not the standard user-written code transformation that SAS Data Integration Studio provides:

- 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 on the transformation and select the following ports:
- **Add Input Port** to add drop zones for the input tables
- **Add Output Port** for each required output table

3 Drag and drop tables or transformations onto the new drop zone.

4 Select the **General** tab. Replace the default transformation name with a more specific name.

5 Select the **Code** tab.

6 Check the **Code generation mode** field with these two important points in mind:
   - Do not modify the default settings for Code generation mode.
   - Make sure that **Automatic** is selected for the Code generation mode.
7  Review the registered target tables. Specify parameters for the target tables based on the following three scenarios:

- **Scenario 1. Source code refers to tables that are using a macro variable**

  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. Right-click on the transformation, select Properties and view the Physical Storage tab.
Scenario 2. Source code creates tables with more logical names

If the source code creates tables with more logical names, register these tables as target tables. In the following example, my_logical_name replaces the default target table name W5UYV7RN.

![Image showing the registration of tables](image)

Scenario 3. Source code loads the output tables that you dropped onto the transformation

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. Also, select the library accordingly.

Select the **Mappings** 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, you can 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.
9 Select the **Options** tab.

10 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 dialog box for the target tables on the **Code** tab. However, some libraries, such as SAP server libraries, need to be assigned by using one of these four parameters to specify them.
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 a few 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 information types 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. The increase in volume of data is based on the validity of records that might span from one to ten 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, with each date selecting 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. As a result, use the Date Join transformation and %DATEJOIN macro for exploitation time unless you need to represent only a few fixed dates.

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

Figure 4.1 Input and Output Tables Using Date Join Transformation

**Input Table 1**

<table>
<thead>
<tr>
<th>Key</th>
<th>Begin Date</th>
<th>End Date</th>
<th>Var1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01JAN06</td>
<td>15FEB06</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>16FEB06</td>
<td>20APR06</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>01JUN06</td>
<td>31JUL06</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>01AUG06</td>
<td>31AUG06</td>
<td>4</td>
</tr>
</tbody>
</table>

**Input Table 2**

<table>
<thead>
<tr>
<th>Key</th>
<th>Begin Date</th>
<th>End Date</th>
<th>Var2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01FEB06</td>
<td>31MAR06</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>01APR06</td>
<td>31JUL06</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>01AUG06</td>
<td>30SEP06</td>
<td>C</td>
</tr>
</tbody>
</table>
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.
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 latter 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, one record is added for each month between the begin date and end date if the INTERVAL parameter is monthly.

**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 April 19, 2006.

---

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

**Input Table**

<table>
<thead>
<tr>
<th>DATECOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B 01JAN2006</td>
</tr>
<tr>
<td>C 01MAR2006</td>
</tr>
<tr>
<td>D 01JUL2006</td>
</tr>
</tbody>
</table>
### 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 SAS Detail Data Store has a three-character language code that differs based on the locale used by the solution data mart and the operating system.

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

A dimension member table such as DDS.GL_ACCOUNT is more complex. 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 DDS.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 DDS.CODE_LANGUAGE table:

<table>
<thead>
<tr>
<th>LANGUAGE_CD</th>
<th>LOCALE_LANGUAGE_CD</th>
<th>LOCALE_COUNTRY_CD</th>
<th>LOCALE_VARIANT_CD</th>
<th>DEFAULT_LANGUAGE_FLG</th>
</tr>
</thead>
<tbody>
<tr>
<td>zh</td>
<td>zh</td>
<td>CN</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>zhx</td>
<td>zh</td>
<td>TW</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>en</td>
<td>En</td>
<td></td>
<td></td>
<td>N</td>
</tr>
</tbody>
</table>

In the above table names and descriptions of the language codes for account members are added. For example, the language code *zh* of account member table GL_ACCOUNT is added in the table above.

Based on this action, the transformation splits the data in the SAS Detail Data Store.

**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 Select 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.

---

**Additional Generic Transformations**

**Introduction**

A group of transformations called data transforms are based on the generic transformation generator. These transformations process data for the SAS Financial Management Adapter for SAP. However, the processing capabilities of the
transformations are not restricted to SAP data and functionality. These transformations can also be used in jobs outside of the SAS Financial Management Adapter for SAP.

---

**Keep First or Last Record in Group Transformation**

This transformation keeps the first record or last record in a group. It differs from SORT NODUPKEY in that it can control which record is kept.

---

**DATA Step Merge Transformation**

This transformation includes a DATA step that merges a master table (left table) with one or more additional tables. A single parse goes through all the tables, which is not possible using an SQL left join.

---

**Convert Number Order to Integer Value Transformation**

Some columns depict a 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 **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 processing is performed on the input table.

---

**Map SAP Address Lines to SAS Address Lines Transformation**

SAS data models use the field names ADDRESS_LINE_1 through ADDRESS_LINE_4 to store address data that appears after the addressee or company name but before the city and ZIP code. SAP has many fields that are available to store address data. The Map SAP Address Lines to SAS Address Lines transformation enables you to select between one and four relevant address columns from SAP data and map them to the SAS columns ADDRESS_LINE_1 to ADDRESS_LINE_4.

On the **Options** tab, select the SAP address columns that correspond to the required SAS address columns. For example, you can define SAP columns CNAME, STRAS, and STR2 to feed to SAS as ADDRESS_LINE_1, ADDRESS_LINE_2, and ADDRESS_LINE_3, respectively. If a particular data record has missing STRAS data, then the STR2 value populates the ADDRESS_LINE_2 column, and the ADDRESS_LINE_3 column remains blank.
Introduction to Initialization Jobs

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

These jobs initialize your environment and provide data for SAS Financial Management.

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.
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 Job**

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
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 Job**

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 Job**

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 Job**

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
Extraction Jobs  Job Group: E00 Extraction of T Tables  85

- &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 Job**

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 Job**

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

This job extracts the following tables:
Job Group: E01 Extraction of Master Tables

Chapter 6

- 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 Job

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 Job

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 Job

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 Job

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 Job**

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.
- $SAPOTOSAS_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) Job

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 E02000 Initial Extraction Financial Documents BKPF and BSEG must execute before this job. This job joins the following R3mirror tables:

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

This job uses the following input:

- BKPF from R3mirror
- BSEG from R3mirror

This job creates the output Intermed.Financial_Documents.

**fmsadpt_E02010_Extract_Intermediate_Table_from_BKPF_BSEG (Changes)**

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)
Extraction Jobs  Job Group: E03 Extraction of Summary Tables

- $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.

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:
- GLT0 - GL account master record transaction
- FAGLFLEXT - General Ledger: Totals
- FAGLFLEXP - General Ledger: Plan Line Item
- FAGLFLEXA - General Ledger: Actual Line It

This job creates the following output:
- Fmsr3mir.GLTO
- 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
This job uses the input table GLT0 from the SAP server.

This job creates the output Fmsr3mir.GL0.

The first two jobs in this group need to be run only if New GL is implemented in your SAP system. These jobs should not be run if your SAP system does not have New GL capabilities.

In case your system does not have New GL capabilities, then run the third job.

---

**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 PROC CALLRFC 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)
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 Job

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 Job

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 Job

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 7

Transformation Jobs

Introduction to Transformation Jobs ............................................................................................................. 98
Job Group: T0000 SAS Supplied Tables ........................................................................................................... 99
Overview ...................................................................................................................................................... 99
fmsadpt_T000010_Load_stage_gl_account_type_Table Job ................................................................. 99
fmsadpt_T000020_Load_stage_period_type_Table Job ........................................................................ 99
fmsadpt_T000030_Load_stage_currency_and_language_Tables Job ...................................................... 99
fmsadpt_T000040_Load_stage_gl_normal_bal_Table Job ...................................................................... 100
fmsadpt_T000050_Load_stage_source_system_Table Job .................................................................... 100
Job Group: T0001 Transformations for Common Tables ........................................................................ 100
Overview .................................................................................................................................................. 100
fmsadpt_T000100_Load_stage_org_type_table Job ............................................................................... 101
fmsadpt_T000110_Load_stage_code_language_table Job ...................................................................... 101
fmsadpt_T000120_Load_stage_state_region_table Job ........................................................................ 101
fmsadpt_T000130_Load_stage_Address_Type Job .................................................................................. 102
fmsadpt_T000140_Load_Inter_GL_Balances (New GL) Job ................................................................. 102
fmsadpt_T000150_Load_Inter_GL_Balances (New GL) Job ................................................................. 102
Job Group: T0002 Time Dimension .................................................................................................... 102
Overview ................................................................................................................................................ 102
Input and Output (Tables, Formats, and Macro Variables) ..................................................................... 103
Administration Factors ............................................................................................................................ 104
Account Hierarchies and Numbering ........................................................................................................ 104
Other Transactions and Account Considerations .................................................................................. 104
fmsadpt_T000200_Transform_Financial_Period_Info_from_SAP Job .................................................. 104
fmsadpt_T000210_Create_Inter_time_period_dimension_Table_Job .................................................. 105
fmsadpt_T000220_Load_stage_time_period_dimension_Tables Job ...................................................... 106
fmsadpt_T000230_Create_Inter_controlling_area_fin_year_combo Job .................................................. 106
Job Group: T0003 Exchange Rates ........................................................................................................ 106
Overview ................................................................................................................................................ 106
fmsadpt_T000300_Load_stage_currency_exch_rate_type_Table Job .................................................... 107
fmsadpt_T000310_Load_stage_currency_exch_rate_set_table Job ....................................................... 107
fmsadpt_T000320_Load_stage_currency_exch_rate_src_Table Job ...................................................... 108
fmsadpt_T000330_Load_stage_currency_exch_rate_table Job ............................................................. 108
Job Group: T0004 Analysis Dimension .............................................................................................. 109
Overview ................................................................................................................................................ 109
fmsadpt_T000400_Load_stage_analysis_dimension_tables Job .......................................................... 109
Job Group: T0005 Account Dimension ............................................................................................... 109
Overview ................................................................................................................................................ 109
Inputs and Outputs (Tables, Formats, and Macro Variables) ................................................................. 110
Financial Statement Nodes ....................................................................................................................... 111
Account Hierarchies and Numbering ........................................................................................................ 111
Profit and Loss Accounts .......................................................................................................................... 112
Other Transactions and Account Considerations .................................................................................. 112
fmsadpt_T000500_Create_Inter_T011 Job ......................................................................................... 112
fmsadpt_T000505_Create_Inter_Accounts_with_Text Job ................................................................. 113
fmsadpt_T000510_Create_Inter_Account_Items_Transformed Job .................................................... 113
fmsadpt_T000520_Create_Inter_Account_Hierarchy ............................................................................ 114
fmsadpt_T000525_Create_Inter_Account_Balances Job ...................................................................... 115
fmsadpt_T000528_Create_Inter_Account_Type Job ............................................................................ 115
fmsadpt_T000530A_Create_Inter_GL_account_with_SAP_Info Job .................................................. 115
fmsadpt_T000530B_Create_Inter_GL_Account_with_SAP_Info Job .................................................... 115
fmsadpt_T000535_Create_Inter_Account_statistic_Site_Specific Job .................................................. 116
fmsadpt_T000540_Load_state_gl_account_association_Tables Job .................................................... 116
fmsadpt_T000550A_Load_stage_gl_account_tables_Site_Specific Job .................................................. 117
fmsadpt_T000550B_Load_stage_gl_account_tables_Site_Specific Job .................................................. 117
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.
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 Job

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 FMS_STAGESAP.GL_ACCOUNT_TYPE.

fmsadpt_T000020_Load_stage_period_type_Table Job

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 SASSupplied.PERIOD_TYPE data set.

This job includes the following values:
- special periods, record type SP
- half months, record type HMO

This job uses the input SAS_Supplied.SAS_PERIOD_TYPE.

This job creates the output FMS_STAGESAP.PERIOD_TYPE.

This job uses the user-written source T000020_ADD_SAP_SPECIAL_PERIODS.SAS.

fmsadpt_T000030_Load_stage_country_and_language_Tables Job

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:
- FMS_STAGESAP.COUNTRY
- FMS_STAGESAP.LANGUAGE

**fmsadpt_T000040_Load_stage_gl_normal_bal_Table Job**

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 SAS_Supplied.SAS_GL_NORMAL_BAL.

This job creates the output FMS_STAGESAP.GL_NORMAL_BAL.

**fmsadpt_T000050_Load_stage_source_system_Table Job**

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:
- SAS_SUP.SAS_SOURCE_SYSTEM
- &DDS_SOURCE_SYSTEM_ID

This job creates the output FMS_STAGESAP.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 Job

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

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

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

```sas
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 FMS_STAGESAP.ORG_TYPE.

fmsadpt_T000110_Load_stage_code_language_table Job

This job uses the following input:
- `Fmsr3mir.T002`
- `Fmsr3mir.T002T`
- `SAS_LANGUAGE_ISO0639`

This job creates the output FMS_STAGESAP.CODE_LANGUAGE.

The FMS_STAGESAP.CODE_LANGUAGE table is used to split the non-default language texts into NLS dimension tables.

fmsadpt_T000120_Load_stage_state_region_table Job

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 FMS_STAGESAP.STATE_REGION.

**fmsadpt_T000130_Load_stage_Address_Type Job**

This job loads the STAGE_ADDRESS>Type table with constant values. The figure below shows FMS_STAGESAP.ADDRESS TYPE table values that are required for both the internal and external organization dimensions.

<table>
<thead>
<tr>
<th>#</th>
<th>ADDRESS_TYPE_CD</th>
<th>LANGUAGE_CD</th>
<th>VALID_FROM_DTTM</th>
<th>VALID_TO_DTTM</th>
<th>ADDRESS_TYPE_DESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CUS</td>
<td>en</td>
<td>21JAN1990:30:00:00</td>
<td></td>
<td>Customer</td>
</tr>
<tr>
<td>2</td>
<td>SUP</td>
<td>en</td>
<td>21JAN1990:30:00:00</td>
<td></td>
<td>Supplier</td>
</tr>
</tbody>
</table>

The actual descriptions are in English. For descriptions in other languages, you can modify the source code in t000130_add_additional_types.sas accordingly.

This job creates the output FMS_STAGESAP.ADDRESS_TYPE.

**fmsadpt_T000140_Load_Inter_GL_Balances (New GL) Job**

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:
- R3 Mirror.GLT0
- R3 Mirror.FAGLFLExT

This job creates the output Inter_GL_Balances_new.

---

**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:

- FMS_STAGESAP.TIME_PERIOD
- FMS_STAGESAP.TIME_PERIOD_ASSOC
- FMS_STAGESAP.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 Job

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_YEARLIKECALENDAR_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 Job**

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 Job**

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:
- FMS_STAGESAP.TIME_PERIOD_ASSOC
- FMS_STAGESAP.TIME_PERIOD
- FMS_STAGESAP.TIME_PERIOD_ASSOC_TYPE

---

**fmsadpt_T000230_create_Inter_controlling_area_fin_year_combo Job**

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 Detail Data Store. 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. The following figure shows a sample of exchange rate types:

<table>
<thead>
<tr>
<th>#</th>
<th>EXCHANGE_RATE_TYPE_CD</th>
<th>VALID_FROM_DTTM</th>
<th>LANGUAGE_CD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HIS</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>Historic market rate</td>
</tr>
<tr>
<td>2</td>
<td>NCO</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>No currency</td>
</tr>
<tr>
<td>3</td>
<td>PA</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>Period average rate</td>
</tr>
<tr>
<td>4</td>
<td>PE</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>Period close rate</td>
</tr>
<tr>
<td>5</td>
<td>C1</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>User-defined Rate</td>
</tr>
<tr>
<td>6</td>
<td>C2</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>User-defined Rate</td>
</tr>
<tr>
<td>7</td>
<td>DER</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>Derived rate</td>
</tr>
<tr>
<td>8</td>
<td>PO</td>
<td>01 JAN 1970</td>
<td>en</td>
<td>Period open rate</td>
</tr>
</tbody>
</table>

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 picks all exchange rates that were valid on those days.

For example, the SAS Financial Management Adapter for SAP loads the rates for the SAS Detail Data Store 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 Job**

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 sub-setting the exchange rates in SAP.

This job uses SAS_Supplied.SAS_CURRENCY_EXCH_RATE_TYPE as input.

This job creates the following output:
- FMS_STAGESAP.CURRENCY_EXCH_RATE_TYPE
- Intermed.SAPTOSAS_EXRATE_TYPE
- $SAPTOSAS_CURRENCY_EXCH_TYPE

The user-written source for this job is t000300_saptosas_exchrate_type_mapping.sas.
fmsadpt_T000310_Load_stage_currency_exch_rate_set_table Job

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

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

This job uses the input t000310_set_exch_rate_set.sas.

This job creates the output FMS_STAGESAP.CURRENCY_EXCH_RATE_SET.

fmsadpt_T000320_Load_stage_currency_exch_rate_src_Table Job

This job creates FMS_STAGESAP.CURRENCY_EXCH_RATE_SRC with a constant key that is equal to SAP.

This job creates the output FMS_STAGESAP.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 Job

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

This job uses the following input:
- FMS_STAGESAP.TIME_PERIOD
- Fmsr3mir.TCURR

This job creates the following output:
- intermed.DAILY_EXCHANGE_RATES
- FMS_STAGESAP.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 hardcoded. 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 Job

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

This job creates the following output:
- FMS_STAGESAP.ANALYSIS
- FMS_STAGESAP.ANALYSIS_ASSOC
- FMS_STAGESAP.ANALYSIS_ASSOC_TYPE
- FMS_STAGESAP.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 SAS Detail Data Store, 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.
- 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:

```sas
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:

- **FMS_STAGESAP.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.

- **FMS_STAGESAP.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.

- **FMS_STAGESAP.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 SAS Detail Data Store 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, but 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_BALACE 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 Job```

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

This job uses the following input:

- 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 Job**

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 Job**

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 Job

This job uses the following input:
- Inter_GL_Balances_new (created in fmsadpt_T000140_Load_Inter_GL_Balances(New GL))
- 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 Job

This job uses the following input:
- Inter_GL_Account_Balances (created in fmsadpt_T000525_create_Inter_Account_Balances)
- R3 Mirror CSKB - Cost Elements

This job creates the output Inter_GL_Account_type_cobalances.

fmsadpt_T000530A_create_Inter_GL_Account_with_SAP_Info Job

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 Job

This job collects and joins information from various sources that are useful in determining the account classification columns.
This job uses the following inputs:
- 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 Job**

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, FMS_STAGESAP.GL_ACCOUNT corresponds with FMS_STAGESAP.GL_ACCOUNT_ASSOC.

This job has the following inputs:
- &dds_source_system_id (set in preprocessing step)
- &_VERSN (set in preprocessing step)

This job creates the following output:
- Intermed.GL_ACCOUNT_statistical
- Intermed.GL_ACCOUNT_ASSOC_statistical

The user-written source for this job is t000535_stat_accounts.sas.

---

**fmsadpt_T000540_Load_state_gl_account_association_Tables Job**

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:

- FMS_STAGESAP.GL_ACCOUNT_ASSOC_TYPE
- FMS_STAGESAP.GL_ACCOUNT_ASSOC

The user-written source for this job is t000540_gl_account_add.dds_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 Job**

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 FMS_STAGESAP.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 Job**

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.

This job uses the input intermed.GL_ACCOUNT_WITH_SAP_INFO.

This job creates the output FMS_STAGESAP.GL_ACCOUNT.

The user-written source for this job is:

- t000550b_standard_settings
- t000550b_site_specific_code

*Note:* This is an alternative job to T000550A. This job must be run only when you need to determine the account type for revenue and expense accounts from the SAP controlling module and not by using only the user-written code.

**fmsadpt_T000560_Load_stage_source_gl_account_Table Job**

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 SAS Detail Data Store 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 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 Job**

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
- FMS_STAGESAP.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 Job**

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
Job Group: T0007 Profit Center Dimension

Overview

Mapping SAP profit centers to the SAS Detail Data Store 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_CONTROLLINGAREAS.
Transformation Jobs  Job Group: T0007 Profit Center Dimension

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 Job**

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
- FMS_STAGESAP.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 Job**

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_CENTERONLY_HIER
- intermed.PROFIT_CENTERONLY_MASTER
- FMS_STAGESAP.PROFIT_CENTER_ASSOC
- FMS_STAGESAP.PROFIT_CENTER
- FMS_STAGESAP.PROFIT_CENTER_NLS

The user-written source for this job is t000710_separate_matches.sas.

---

**fmsadpt_T000720_Transform_Add_Std_Dim_Values_profit_center Job**

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.

This job uses the following transformations:

- Add Standard Dimension ASSOC Rows
- Add Standard Dimension Rows
This job uses the following input:
- FMS_STAGESAP.PROFIT_CENTER_ASSOC_TYPE
- FMS_STAGESAP.PROFIT_CENTER_ASSOC (input and output)
- FMS_STAGESAP.PROFIT_CENTER (input and output)
- &dds_source_system_id

This job creates the following output:
- FMS_STAGESAP.PROFIT_CENTER_ASSOC (input and output)
- FMS_STAGESAP.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 Job

This job creates the currency association-related tables.

This job uses the following input:
- R3 Mirror.TCURT - Currency Code Names
- R3 Mirror.T002 - Language Keys
- SAS_SUP.SAS_CURRENCY

This job creates the following output:
- FMS_STAGESAP.CURRENCY_ASSOC_TYPE
- FMS_STAGESAP.CURRENCY_ASSOC

This job uses the user-written source
f000800_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 SAS Detail Data Store 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) Job

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 SAS Detail Data Store, 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 SAS Detail Data Store.
- 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 $ACCBALS identifies balance sheet accounts. Opening balance amounts are included only for accounts where the following condition is true: PUT(GLT0.RACCT,$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 SAS Detail Data Store and follow-up data marts that use the data.

---

fmsadpt_T000900_create_Inter_Opening_Balances_and_Num_Ranges Job

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.GLTO. 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_GLTO_Vertical_Summary Job

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

&GJAHŘ

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 Job**

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 (GJAHŘ).

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 Job**

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)
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 SAS Detail Data Store staging area format. Customer external organization jobs take data from the customer master data in SAP R/3 (such as KNA1) and load it into an intermediate structure. Then, the data can be joined with the supplier external data to form the SAS Detail Data Store stage external organization dimension.

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 top-most 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 suppliers’ 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 top-most 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 Job**

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
- FMS_STAGESAP.EXTERNAL_ORG_ASSOC_TYPE

---

**fmsadpt_T100010_Load_stage_External_Org_Assoc Job**

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.

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
- FMS_STAGESAP.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.
Job Group: T1100 Internal Org Dimension

Chapter 7

- 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 Job**

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.

This job uses the following input:
- Inter.LFA1
- Inter.KNA1
- Fmsr3mir.T002

This job creates the output FMS_STAGESAP.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 Job**

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:
- FMS_STAGESAP.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 SAS Detail Data Store 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:

1. 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.

2. Controlling areas (second level)
   assigned as parents of their respective companies, based on the SAP R/3 table TKA02.

3. 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.

4. 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.

5. 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 T120050_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_table Job

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

<table>
<thead>
<tr>
<th>#</th>
<th>Internal Organization Association Type Code</th>
<th>Language Code</th>
<th>Internal Organization Association Type Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OCC</td>
<td>en</td>
<td>Op Concern, Controlling Area, Company, Cost Center</td>
</tr>
<tr>
<td>2</td>
<td>HRC</td>
<td>en</td>
<td>HR Work Groups</td>
</tr>
</tbody>
</table>

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
FMS_STAGESAP.STAGE_INTERNAL_ORG_ASSOC_TYPE.

fmsadpt_T110010_create_Inter_Org_from_Operating_Conscience Job

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_Conscience_T_Tables)
- Fmsr3mir.TKEBT (created in _E00060_Extract_Operating_Conscience_T_Tables)
- &operating_concern_country_cd
- &dds_source_system_id
- $SAPTOSAS_COUNTRY

This job creates the output Intermed.Operating_Conscience_Internal_Org.

This job uses the format $op_concern_currency.
Transformation Jobs  Job Group: T1100 Internal Org Dimension 133

fmsadpt_T110020_create_Inter_Org_from_Con trolling_Area Job

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 Job

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 _E00020_Extract_SAP_FI_T_Tables)
- Fmsr3mir.T002 (created in E00010_Extract_SAP_General_and_T_Tables)
- &dds_source_system_id
- $SAPTOSAS_COUNTRY
- SASSUP.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 Job

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 Job**

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 (T120050_LOAD_INTERNAL_ORG_FROM_CC.SAS). Review the code in this file to ensure the proper administration of the job.

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 Job**

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:

- FMS_STAGESAP.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 Job**

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.

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 FMS_STAGESAP.INTERNAL_ORG.

**fmsadpt_T110080_Load_stage_Internal_Org_Assoc_Table Job**

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

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 Job

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_GLTO_Summary Job

The job T120010_compare_Financial_Docs_with_GLTO_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.GLTO_Vertical_Summary (created in T100030 Make GLT0 Vertical)
- Macro variable &emailerror

This job creates the output Intermed.FI_Docs_vs_GLTO__vert_SUM_Diffs.

The user-written source for this job is t120010_get_diffs.sas.
Integrating Data into the SAS Detail Data Store

Overview

The SAS Financial Management Adapter for SAP loads staging area tables for the SAS Financial Management solution. Later releases of the SAS Financial Management Adapter for SAP are designed to load different subject-area tables for other SAS solutions, as well.

The SAS Financial Management Adapter for SAP creates tables that must be incorporated into the overall data architecture. The staging area tables that are created as output have the same names and structure as the input tables of the jobs in the SAS Detail Data Store repository. After you run the SAS Financial Management Adapter for SAP jobs, you can run the SAS Detail Data Store jobs. These jobs use the SAS Financial Management Adapter for SAP output tables as input.

Note: For more information about how to incorporate data into the SAS Detail Data Store for SAS Data Integration Studio jobs, see the SAS Solutions Services: Data Administration Guide.

There are two basic ways to incorporate the SAP staging area tables into the SAS Detail Data Store jobs.

1. Use one physical location for stage libraries.
2. Use multiple sources.

Method 1: Use One Physical Location for Stage Libraries

To use one physical location for stage libraries, you must define the FMS_StageSAP and stagedds libraries so as to represent the same physical location.
Note: This method might not apply if the data that needs to be loaded is from various sources; for example: multiple SAP systems and other operational sources. In case the data needs to be loaded from various sources, consider the second method.

Method 2: Multiple Sources

To use multiple sources, you can build additional jobs that combine the multiple sources into a new library in the stagedds structure. Each new job might consist of a simple step that appends all data from the various sources.

If the simple approach to append data from different sources does not produce the required results, you must resolve the issue with site-specific code that replaces or augments the Appendstep.

Customizing the Validation Steps in the Load SAS Detail Data Store Jobs

To successfully incorporate SAP data into the SAS Detail Data Store, certain data checks must be removed. For example, you might need to remove some checks from the data validation step. In some cases, you might need to turn off the lookup steps for fact tables.

The following table details the actions that might be required for successful loading to the SAS Detail Data Store. This table provides a general guide, but it is not an all-inclusive list for your specific site. Your site might have some data by default and some data by customization within the SAS Financial Management Adapter for SAP jobs that create or get data from additional tables. You do not need to run all the jobs to populate the SAS Detail Data Store for a particular solution.

Note: The SAS Financial Management Adapter for SAP creates stage tables when relevant data is available in SAP. You must create alternative jobs to populate the other tables if the other tables are required.

<table>
<thead>
<tr>
<th>Job Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>cind_dds_103800_load_gl_account_type_table</td>
<td>Delete the APPEND step. The input table should be STAGE_GL_ACCOUNT_TYPE that is created in the SAS Financial Management Adapter for SAP.</td>
</tr>
<tr>
<td>cind_dds_102400_load_currency_table</td>
<td>Delete the input tables and APPEND step above the STAGE_CURRENCY table. The input table is the STAGE_CURRENCY table that is created in the SAS Financial Management Adapter for SAP.</td>
</tr>
<tr>
<td>Job Name</td>
<td>Action</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cind_dds_102700_load_currency_exch_rate_type_table</td>
<td>Delete the APPEND step. The input table should be STAGE_CURRENCY_EXCH_RATE_TYPE that is created in the SAS Financial Management for SAP.</td>
</tr>
<tr>
<td>cind_dds_103900_load_gl_normal_bal_table</td>
<td>Delete the APPEND step. The input table should be STAGE_GL_NORMAL_BAL that is created in the SAS Financial Management Adapter for SAP.</td>
</tr>
<tr>
<td>cind_dds_100500_load_period_type_table</td>
<td>Get the data from the FMS_STAGESAP library. Delete the APPEND step because the SAP table includes all entries that are held in the SAS table. See the T000020 job.</td>
</tr>
<tr>
<td>cind_dds_100600_load_source_system_table</td>
<td>Delete the APPEND step because the SAP table includes all entries that are held in the SAS table. The input table should be STAGE_SOURCE_SYSTEM that is created in the SAS Financial Management Adapter for SAP.</td>
</tr>
<tr>
<td>cind_dds_106100_load_cost_center_table</td>
<td>Change the data validation for RESPONSIBLE_EMPLOYEE_ID to allow blanks.</td>
</tr>
<tr>
<td>cind_dds_106700_load_external_org_table</td>
<td>Delete the INDUSTRY_CD data validation.</td>
</tr>
<tr>
<td>cind_dds_106200_load_internal_org_table</td>
<td>Delete the STATE_REGION_CD data validation. Turn off the lookup for EMPLOYEE. Turn off the lookup for COUNTY.</td>
</tr>
<tr>
<td>cind_dds_106500_load_gl_account_table</td>
<td>Turn off the lookup for INTERNAL_ORG_ID.</td>
</tr>
<tr>
<td>cind_dds_108000_load_gl_transaction_sum_table</td>
<td>Remove data validation for ITEM_CATEGORY_CD.</td>
</tr>
<tr>
<td>cind_dds_106740_load_external_org_address_table</td>
<td>Change the data validation for STATE_REGION_CD to allow blanks. Delete the APPEND step because the SAP table includes all entries that are held in the SAS table. The input table should be STAGE_COUNTRY that is created in the SAS Financial Management Adapter for SAP</td>
</tr>
<tr>
<td>cind_dds_100100_load_country_table</td>
<td>Delete the APPEND step because the SAP table includes all entries that are held in the SAS table. The input table should be STAGE_COUNTRY that is created in the SAS Financial Management Adapter for SAP</td>
</tr>
<tr>
<td>cind_dds_100600_load_source_system_table</td>
<td>Delete the APPEND step because the SAP table includes all entries that are held in the SAS table. The input table should be STAGE_SOURCE_SYSTEM that is created in the SAS Financial Management Adapter for SAP.</td>
</tr>
<tr>
<td>cind_dds_106200_load_internal_org_table</td>
<td>Delete the STATE_REGION_CD data validation. Turn off the lookup for EMPLOYEE. Turn off the lookup for COUNTY.</td>
</tr>
<tr>
<td>cind_dds_106500_load_gl_account_table</td>
<td>Turn off the lookup for INTERNAL_ORG_ID.</td>
</tr>
<tr>
<td>cind_dds_108000_load_gl_transaction_sum_table</td>
<td>Remove data validation for ITEM_CATEGORY_CD.</td>
</tr>
</tbody>
</table>
Job Name | Action
--- | ---
cind_dds_107050_load_currency_complex_exch_rate_table | Remove data validation for ITEM_CATEGORY_CD1

Below are examples of customizations that might be required to ensure that the data checks do not reject valid records:

- In the 106100 job and the 106900 job, turn off the lookup for the employee table that is removed. This customization is necessary for the following reasons:
  - The employee number is not in the relevant SAP table. These tables include an unchecked string that often contains a name or initials of the relevant employee for the Cost Center or Profit Center.
  - The SAS Financial Management Adapter for SAP might not load the employee dimension table.
- In the 107910 job, remove the data check against the ITEM_CATEGORY table because the SAS Financial Management Adapter for SAP does not load it.
- In the 106200 job, remove the data checks against the COUNTY, STATE_REGION, and EMPLOYEE columns. This information is not loaded into the INTERNAL_ORG table. You might consider populating the COUNTY and STATE_REGION columns in an additional step.
- The 106700 job should not validate against the SAS Detail Data Store table INDUSTRY because the SAS Financial Management Adapter for SAP does not load Stage INDUSTRY data.
- The 106740 job should not validate against the table STATE_REGION for data records of any locale outside the United States, because this column is not populated for other countries.
- Do not reject blank address types. Replace them with a standard value. The SAS Financial Management Adapter for SAP does not populate the ADDRESS_TYPE reference table. Add an ADDRESS_TYPE table and enter the appropriate values.

Note: In order to load the solution data mart successfully with data from the jobs included in the SAS Financial Management Adapter for SAP, modify the following table supplied by SAS:

- SAS_LANGUAGE_ISO0639 – The Adapter jobs load only those language texts that are defined in the SAS_LANGUAGE_ISO0639 table supplied by SAS. Hence if you have customized the Adapter to extract specific language text, then ensure that there is a record for the required language in the table supplied by SAS: SAS_LANGUAGE_ISO0639

Note: In order to load the solution data mart successfully with data from the jobs included in the SAS Financial Management Adapter for SAP, the following table structures (with zero observations) need to be present in the stagedds library:

- TIME_PERIOD_NLS
- EXTERNAL_ORG_NLS
- GL_JRNL
- GL_JRNLDETAILS
### Tables Extracted from SAP

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKPF</td>
<td>Accounting Document Header</td>
</tr>
<tr>
<td>BSEG</td>
<td>Accounting Document Segment</td>
</tr>
<tr>
<td>CEPCT</td>
<td>Profit Center Master Data Table</td>
</tr>
<tr>
<td>CEFC</td>
<td>Texts For Profit Center Master Data</td>
</tr>
<tr>
<td>COB</td>
<td>CO Object: Document Header</td>
</tr>
<tr>
<td>COEP</td>
<td>CO Object: Line Items (by Period)</td>
</tr>
<tr>
<td>CSKB</td>
<td>Cost Elements</td>
</tr>
<tr>
<td>CSKS</td>
<td>Cost Center Master Data</td>
</tr>
<tr>
<td>CSKT</td>
<td>Cost Center Texts</td>
</tr>
<tr>
<td>DD07T</td>
<td>DD: Texts For Domain Fixed Values (Language-Dependent)</td>
</tr>
<tr>
<td>FAGL_ACTIVEC</td>
<td>Activation of New General Ledger</td>
</tr>
<tr>
<td>FAGL_FLEXA</td>
<td>General Ledger: Actual Line Items</td>
</tr>
<tr>
<td>FAGL_FLEXP</td>
<td>General Ledger: Plan Line Items</td>
</tr>
<tr>
<td>FAGL_FLEXT</td>
<td>General Ledger: Totals</td>
</tr>
<tr>
<td>GLT0GL</td>
<td>account master record transaction figure</td>
</tr>
<tr>
<td>KNA1</td>
<td>General Data in Customer Master</td>
</tr>
<tr>
<td>KNVH</td>
<td>Customer Hierarchies</td>
</tr>
<tr>
<td>LFA1</td>
<td>Vendor Master (General Section)</td>
</tr>
<tr>
<td>LFHM</td>
<td>Vendor hierarchy</td>
</tr>
<tr>
<td>NRIV</td>
<td>Number Range Intervals</td>
</tr>
<tr>
<td>RF011P</td>
<td>Items in financial statement for a selected balance sheet report</td>
</tr>
<tr>
<td>RF011Q</td>
<td>Financial statement text</td>
</tr>
<tr>
<td>RF011Z</td>
<td>Balance sheet line assignment</td>
</tr>
<tr>
<td>SKA1</td>
<td>GL Account Master (Chart of Accounts)</td>
</tr>
<tr>
<td>SKAT</td>
<td>GL Chart of Accounts Description</td>
</tr>
<tr>
<td>T001</td>
<td>Company Codes</td>
</tr>
<tr>
<td>T002</td>
<td>Language Keys (Component BC – 118)</td>
</tr>
<tr>
<td>T002T</td>
<td>Language Key Texts</td>
</tr>
<tr>
<td>T005T</td>
<td>Country Names</td>
</tr>
<tr>
<td>T005U</td>
<td>Taxes: Region Key: Texts</td>
</tr>
<tr>
<td>T009</td>
<td>Fiscal Year Variants</td>
</tr>
<tr>
<td>T009B</td>
<td>Fiscal year variant periods</td>
</tr>
<tr>
<td>T009C</td>
<td>Period names</td>
</tr>
<tr>
<td>T009T</td>
<td>Fiscal year variant names</td>
</tr>
<tr>
<td>T011</td>
<td>Financial Statement Versions</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>T030</td>
<td>Standard Accounts Table</td>
</tr>
<tr>
<td>T030C</td>
<td>Global Standard Account Table</td>
</tr>
<tr>
<td>T549A</td>
<td>Payroll areas</td>
</tr>
<tr>
<td>T880</td>
<td>Global Company Data (for KONS Ledger)</td>
</tr>
<tr>
<td>T8G21</td>
<td>Splitting rule: item to be processed</td>
</tr>
<tr>
<td>TCURR</td>
<td>Exchange Rates</td>
</tr>
<tr>
<td>TCURT</td>
<td>Currency Code Names</td>
</tr>
<tr>
<td>TGSBT</td>
<td>Business Area Names</td>
</tr>
<tr>
<td>THITT</td>
<td>Texts for Customer Hierarchy Types</td>
</tr>
<tr>
<td>TKA00</td>
<td>Control parameters for controlling areas</td>
</tr>
<tr>
<td>TKA01</td>
<td>Controlling Areas</td>
</tr>
<tr>
<td>TKA02</td>
<td>Controlling area assignment</td>
</tr>
<tr>
<td>TKEB</td>
<td>Management for Operating Concerns</td>
</tr>
<tr>
<td>TKEBT</td>
<td>Description of operating concern</td>
</tr>
<tr>
<td>TLHITT</td>
<td>Vendor hierarchy category (description)</td>
</tr>
</tbody>
</table>
Job Groups and SAS Solution Data Marts

The job groups that are described in this document extract, transform, and load data from standard SAP tables into standard SAS tables in the SAS Detail Data Store. The data is then available in a usable format for solution data marts such as SAS Financial Management.

Each job group provides ETL functions that are required for, or relate to, one or more SAS solution data marts. The table below lists each job group and the SAS solution data marts for which it extracts, transforms, and loads data. Currently, SAS Financial Management solution data mart is supported.

<table>
<thead>
<tr>
<th>Job Group</th>
<th>Corresponding SAS Solution Data Mart(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization Job Group</strong></td>
<td></td>
</tr>
<tr>
<td>0 Data Environment Initialization</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td><strong>Extraction Job Groups</strong></td>
<td></td>
</tr>
<tr>
<td>E00 Extraction of T Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>E01 Extraction of Master Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>E02 Extraction of Transaction Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>E03 Extraction of Summary Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>E04 Extraction of Hierarchies and Structures</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>E05 Extraction of Miscellaneous Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td><strong>Transformation Job Groups</strong></td>
<td></td>
</tr>
<tr>
<td>T0000 SAS-Supplied Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0001 Transformations for Common Tables</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0002 Time Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0003 Exchange Rates</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0004 Analysis Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0005 Account Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0006 Cost Center Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0007 Profit Center Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0008 Currency Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T0009 Financial Documents (Transaction Data)</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T1000 External Org Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T1100 Internal Org Dimension</td>
<td>SAS Financial Management</td>
</tr>
<tr>
<td>T1200 Data Validation</td>
<td>SAS Financial Management</td>
</tr>
</tbody>
</table>
Recommended Reading

Here is the recommended reading list for this title:

- Customer Data Integration: Reaching a Single Version of the Truth
- SAS Data Integration Studio: User’s Guide
- SAS Data Surveyor for SAP: Help
- SAS Management Console: User’s Guide
- SAS Solutions Services: Data Administration Guide
- SAS Solutions Services: Data Model Reference

For a complete list of SAS publications, see the current SAS Publishing Catalog. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at the following address:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: (800) 727-3228*
Fax: (919) 677-4444
E-mail: sasbook@sas.com
Web address: support.sas.com/pubs

* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.
Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to yourturn@sas.com. Include the full title and page numbers (if applicable).

- If you have comments about the software, please send them to suggest@sas.com.
SAS® Publishing Delivers!

SAS Publishing provides you with a wide range of resources to help you develop your SAS software expertise. Visit us online at support.sas.com/bookstore.

**SAS® PRESS**
SAS Press titles deliver expert advice from SAS® users worldwide. Written by experienced SAS professionals, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

**SAS® DOCUMENTATION**
We produce a full range of primary documentation:

- Online help built into the software
- Tutorials integrated into the product
- Reference documentation delivered in HTML and PDF formats—free on the Web
- Hard-copy books

support.sas.com/documentation

**SAS® PUBLISHING NEWS**
Subscribe to SAS Publishing News to receive up-to-date information via e-mail about all new SAS titles, product news, special offers and promotions, and Web site features.

support.sas.com/spn

**SOCIAL MEDIA: JOIN THE CONVERSATION!**
Connect with SAS Publishing through social media. Visit our Web site for links to our pages on Facebook, Twitter, and LinkedIn. Learn about our blogs, author podcasts, and RSS feeds, too.

support.sas.com/socialmedia