SAS® Customer Analytics for Communications 5.4

Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *SAS® Customer Analytics for Communications 5.4: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Customer Analytics for Communications 5.4: Administrator's Guide

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

Printing 1, January 2013

Contents

About This Book

Audience

This document is primarily intended for administrators who will perform the initial installation and configuration of SAS Customer Analytics for Communications and also perform the administrative tasks on a regular basis. The document gives an overview of the SAS Customer Analytics for Communications architecture. It also explains various architecture components and the interactions between them. In addition, this document provides the data flow diagram. This diagram explains the data-processing infrastructure of SAS Customer Analytics for Communications. The solution-specific administrative tasks are detailed in the respective chapters.

Prerequisites

Before you administer SAS Offer Optimization for Communications, make sure that you are familiar with the following concepts:

basic concepts and components of the SAS Intelligence Platform

For details, see *SAS Intelligence Platform: Overview*, which is located at `http://support.sas.com/documentation/cdl/en/biov/63145/PDF/default/biov.pdf`.

the SAS environment

For details, see *SAS Intelligence Platform: System Administration Guide*, which is located at `http://support.sas.com/documentation/cdl/en/bisag/60945/PDF/default/bisag.pdf`.

the SAS applications servers that are required for particular content

For details, see *SAS Intelligence Platform: Application Server Administrative Guide*, which is located at `http://support.sas.com/documentation/cdl/en/biasag/61237/PDF/default/biasag.pdf`.

security concepts

You should be familiar with the authentication and authorization concepts. You should also know how to manage access in the metadata layer. In addition, you should know how to create and manage user and group definitions in metadata. For details, see *SAS Intelligence Platform: Security Administration Guide*, which is located at `http://support.sas.com/documentation/cdl/en/bisecag/61133/PDF/default/bisecag.pdf`.

the middle-tier environment

You should know how to configure the Application server.

SAS products

You should know the basic procedures for using the applications that you plan to administer. For example, if you are responsible for administering SAS Web Report Studio, then you should know how to log on, navigate, and create reports in SAS Web Report Studio.

server context

You should have complete information about the SAS Application Server context. A SAS Application Server knows its server context (the context in which it is being used) and makes decisions based on that knowledge. For example, a client such as SAS Data Integration Studio is assigned a default SAS Application Server. When the client generates code, it submits the code to that application server. The application server determines what type of code is being submitted and directs it to the correct server. That is, if the code is a typical SAS code that can be run in SAS Display Manager, the code is executed by the application server's workspace server. In addition, data-related objects such as SAS libraries, database libraries, and OLAP schemas can be assigned to a SAS Application Server. After this assignment, a client might need to access data in a particular library or OLAP schema. The client then uses a server component that belongs to the application server to which the library or schema is assigned.

Document Conventions

The following table lists the conventions that are used in this document:

Convention	Description
`<SAS Home>`	Represents the path to the folder where SAS is installed. For example, on a Windows computer, this can be **`C:/Program Files/SASHome`**.
`<SAS configuration directory>`	Represents the path to the folder where SAS configuration data is stored. For example, on a Windows computer, this can be **`C:/SAS/Config`**.

Recommended Reading

When you refer to this guide, make sure that you read the following documents in the sequence mentioned below:

- *SAS Communications Analytics Architecture: Data Reference Guide*
- *SAS Communications Analytics Architecture: Administrator's Guide*
- *SAS Customer Analytics for Communications: Data Reference Guide*

For a complete list of SAS publications, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Part 1

Installing and Configuring SAS Customer Analytics for Communications

Chapter 1
Architecture Overview

SAS Customer Analytics for Communications Architecture

Overview

The SAS Customer Analytics for Communications architecture is designed to efficiently process a large volume of data. At the same time, the architecture enables the solution to use this data to support user-driven workflow through the application user interface. SAS Customer Analytics for Communications has an n-tier architecture that separates the workflow-related activities from data-intensive process routines and distributes functionality across computer resources that are most suitable for these tasks.

You can scale the architecture to meet the demands of your workload. For a large organization, the tiers can be installed across a multitude of machines with different operating systems. For tasks such as developing prototypes and presenting demonstrations, all the tiers can be installed on a single machine. Similarly, if you are implementing SAS Customer Analytics for Communications for small enterprises, then you can install all the tiers on a single machine.

The SAS Customer Analytics for Communications architecture consists of the following four tiers:

Data Tier
: stores your enterprise data. This data is stored in SAS tables in appropriate libraries and is used in various analysis-through-client programs.

Server Tier
: consists of data routines and SAS servers that process your enterprise data based on requests from client programs (through middle-tier services) and other programs.

Middle Tier
: provides an environment in which the SAS Customer Analytics for Communications client, along with other business intelligence web applications, such as SAS Web

Report Studio and SAS Information Delivery Portal, can execute in an integrated environment. These applications run in a web application server and communicate with the user by sending and receiving data from the user's web browser. The middle-tier applications depend on the servers that are deployed on the server tier to process, query on, and analyze data.

Client Tier
: includes web-based and desktop user interface content and applications. These clients provide access to content, appropriate query and reporting interfaces, and business intelligence functionality, including advanced design and analysis tasks for all information consumers in your enterprise.

Application Architecture

The following diagram describes the components in the SAS Customer Analytics for Communications architecture.

Figure 1.1 SAS Customer Analytics for Communications Architecture

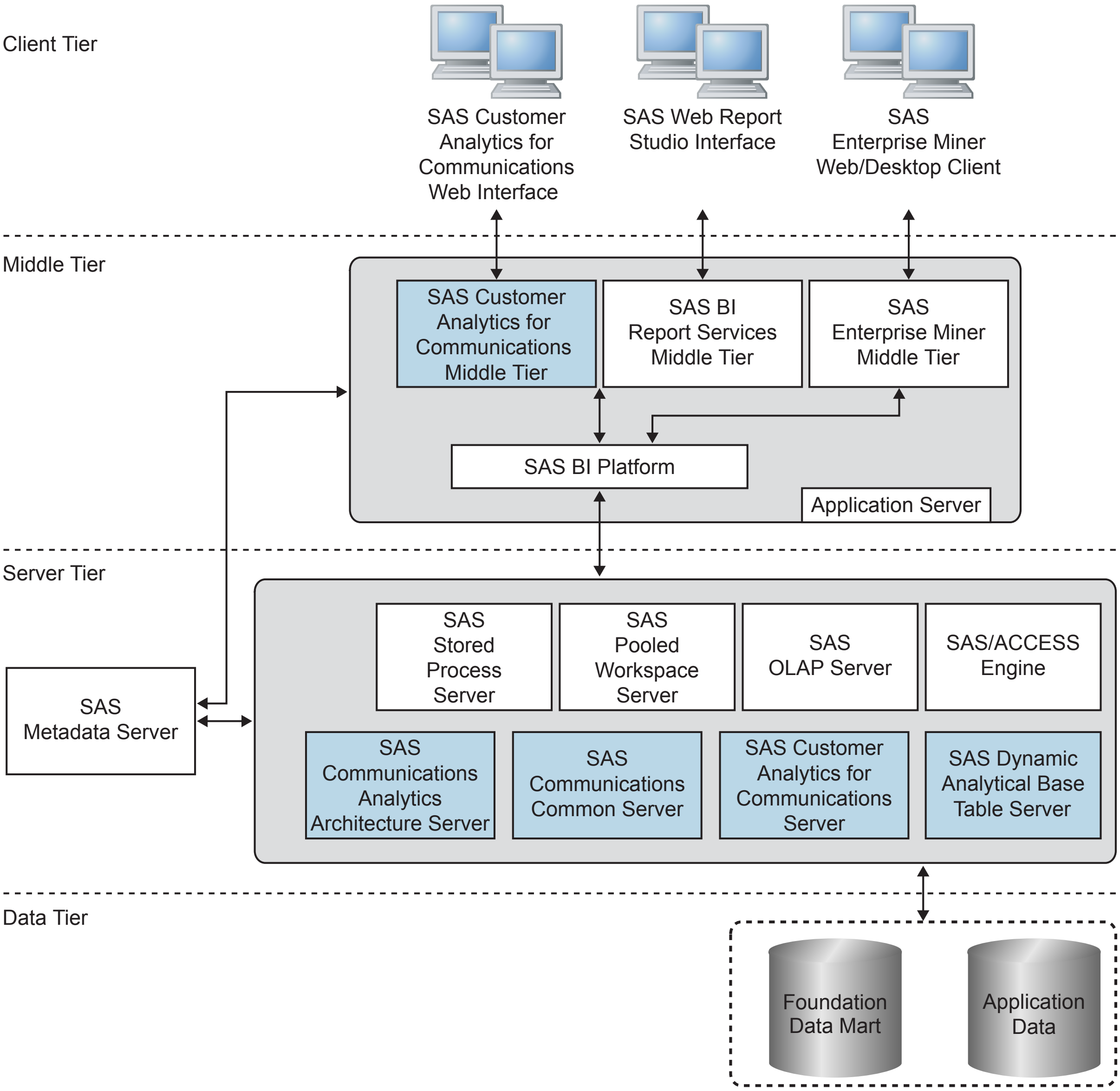

Client Tier
SAS Customer Analytics for Communications Web Interface
SAS Web Report Studio Interface
SAS Enterprise Miner Web/Desktop Client
Middle Tier
SAS Customer Analytics for Communications Middle Tier
SAS BI Report Services Middle Tier
SAS Enterprise Miner Middle Tier
SAS BI Platform
Application Server
Server Tier
SAS Metadata Server
SAS Stored Process Server
SAS Pooled Workspace Server
SAS OLAP Server
SAS/ACCESS Engine
SAS Communications Analytics Architecture Server
SAS Communications Common Server
SAS Customer Analytics for Communications Server
SAS Dynamic Analytical Base Table Server
Data Tier
Foundation Data Mart
Application Data

Table 1.1 *Data Tier Components*

Component	Function
Foundation Data Mart	Stores your enterprise data (and history) that is classified into reference data, dimensions, and transaction summary facts. This data is used by the business solutions to support analysis at customer, product, and offer level. This data is also used to generate business intelligence reports and analyze the best offer recommendations. For information about the entities and attributes, see *SAS Communications Analytics Architecture: Data Reference Guide*.
Application Data	Consists of base data structures that are used to build analytical models. These models are used for scoring or segmenting customers. The solution has a defined set of derived, behavior, and descriptive variables. These variables are initially used to configure the model and later as an input to the scoring code that is provided by the model. For information about the entities and attributes, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Table 1.2 *Server Tier Components*

Component	Function
SAS Communications Analytics Architecture Server	Represent the data processing SAS routines that are packaged as a part of the solution. These routines perform distinct data operations based on client requests and other SAS routines (in batch mode).
SAS Communications Common Server	
SAS Dynamic Analytical Base Table Server	
SAS Customer Analytics for Communications Server	
SAS Metadata Server	Is a multi-user-centralized resource for storing, managing, and delivering metadata for all SAS applications across your enterprise.
SAS Stored Process Server	Responds to client requests to execute solution-specific stored processes.
SAS Pooled Workspace Server	Provides access to SAS software features such as SAS language, SAS libraries, the server file system, results content, and formatting services-execution environments for solution data routines.

Component	Function
SAS OLAP Server	Provides interfaces to a wide range of relational, hierarchical, and network model databases. With this product, SAS Customer Analytics for Communications can read, write, and update data regardless of which database and platform the data is stored on. SAS/ACCESS interfaces provide fast, efficient data loading and enable SAS applications to work directly from your data sources without making a copy.
SAS/ACCESS Engine	Provides interfaces to a wide range of relational, hierarchical, and network model databases. With this product,SAS Customer Analytics for Communications can read, write, and update data regardless of which database and platform the data is stored on. SAS/ACCESS interfaces provide fast, efficient data loading and enable SAS applications to work directly from your data sources without making a copy.

Table 1.3 *Middle-Tier Components*

Component	Function
SAS BI Platform and Foundation Services	Consists of SAS Shared Services, SAS Remote Services, Java Platform Services, and SAS Web Infrastructure Platform. For details, see *SAS Intelligence Platform: Overview*, which is located at **http://support.sas.com/documentation/cdl/en/biov/63145/PDF/default/biov.pdf**.
SAS BI Report Services Middle Tier	Provides an execution environment for the following business intelligence applications: • SAS Web Report Studio • SAS Information Delivery Portal • SAS BI Dashboard • SAS BI Portlets
SAS Enterprise Miner Middle Tier	Provides an application framework for SAS Enterprise Miner. For details, see *Administrator's Guide for SAS Analytics Platform*, which is located at **http://support.sas.com/documentation/onlinedoc/apcore/admin15.pdf**.
SAS Customer Analytics for Communications Middle Tier	Consists of solution-specific services that interact with the client interface to accept user requests (query analysis or data processing) and respond to them with the help of the server tier.

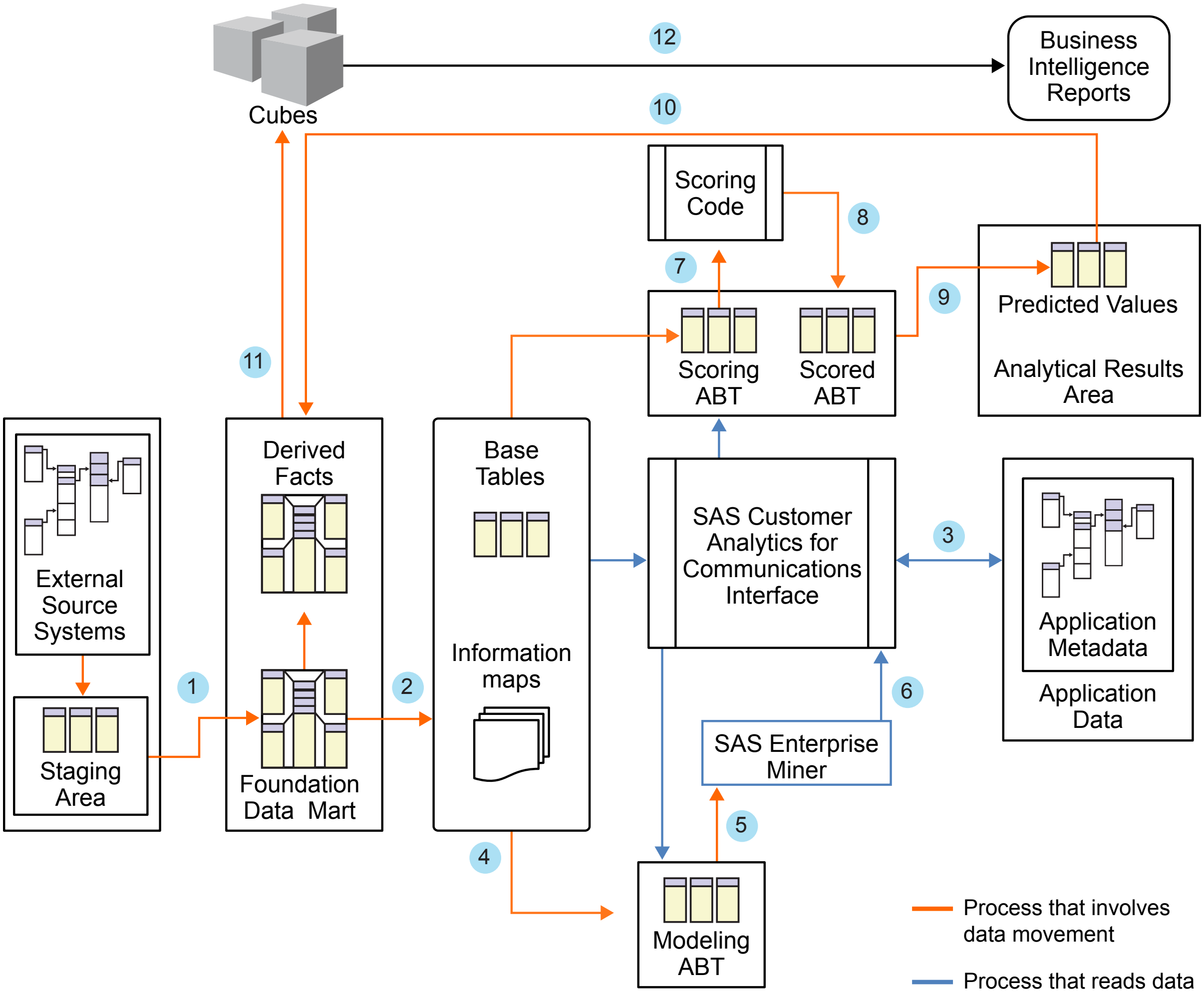
Table 1.4 *Client-Tier Components*

Component	Function
SAS BI Reporting Interface	Consists of the following client interfaces: • SAS OLAP Cube Studio • SAS Information Map Studio • SAS Web Report Studio • SAS Information Delivery Portal • SAS BI Dashboard • SAS BI Portlets For information and usage of these clients, see *SAS Intelligence Platform: Overview*, which is located at **http://support.sas.com/documentation/cdl/en/biov/63145/PDF/default/biov.pdf**.
SAS Enterprise Miner for Desktop and SAS Enterprise Guide	For information and usage of these clients, see *SAS Intelligence Platform: Overview*, which is located at **http://support.sas.com/documentation/cdl/en/biov/63145/PDF/default/biov.pdf**.
SAS Customer Analytics for Communications Web User Interface	Is the web-based user interface to support the end-to-end analytical workflow from building modeling ABT to scoring. For details, see *SAS Customer Analytics for Communications: User's Guide*.

Data Flow in SAS Customer Analytics for Communications

The following diagram illustrates the data marts that form the solution information model. The diagram also illustrates the data flow across various data marts and the internal working of SAS Customer Analytics for Communications.

Figure 1.2 *Data Flow in SAS Customer Analytics for Communications*



1 Denotes the extract, transform, and load (ETL) jobs that populate the Foundation data mart from the external source systems. These ETL jobs are not pre-packaged with the solution.

2 Denotes the ETL jobs that populate the base tables and information maps from the Foundation data mart. These ETL jobs are pre-packaged with the solution. The data in the base tables is used as the source for building modeling and scoring analytical base tables (ABTs). For details, see "Populating the Base Tables" on page 37.

3 SAS Customer Analytics for Communications stores and maintains the application metadata as the Application data. The metadata of the various objects, such as source tables, projects, ABTs, variables, and scoring templates, is maintained in various data sets in the Application data.

4 SAS Customer Analytics for Communications uses the base tables data and information maps data to build modeling ABTs. The modeling ABTs are registered with the SAS metadata.

5 SAS Enterprise Miner uses the registered modeling ABTs to create SAS Enterprise Miner models. These models are registered with the SAS metadata.

6 When you capture a model's information in SAS Customer Analytics for Communications, the solution reads the model metadata (such as significant variables, outcome variable, source code, and so on) and stores the information in the

Application data. When you create a scoring template, the solution reads the scoring code from the model metadata.

7 When you run a scoring job, it uses the data of base tables and information maps as source and creates a scoring ABT that contains the significant variables.

8 The scoring job applies the SAS Enterprise Miner scoring code on the scoring ABT to generate the scored ABT.

9 The analytical results (for example, the predicted values or segments) are written into the Analytical results area.

10 Denotes the ETL jobs that use the information stored in the Analytical results area to populate the fact tables and the dimension tables of the Foundation data mart.

11 When you build the OLAP cubes, the OLAP cubes are populated with data from the fact tables and the dimension tables the Foundation data mart.

12 The OLAP cubes are used as source for generating business intelligence reports. You can view these reports through SAS Web Report Studio. For details, see the *SAS Communications Analytics Architecture: User's Guide*.

Note: If the Foundation data mart is in a database other than SAS, such as Teradata, then all the processes that involve data movement as indicated in the above diagram are completed in the corresponding third-party database.

SAS Customer Analytics for Communications Solution Flow

The SAS Customer Analytics for Communications solution flow contains the following steps:

1. Populate the Foundation data mart with data from the external source system. To populate the Foundation mart with data from the external source system, you need to create ETL jobs. For details, see the *SAS Communications Analytics Architecture: Administrator's Guide*.

2. Populate the base tables with data from the Foundation data mart. You can use the prepackaged ETL jobs to do this. For details, see "Populating the Base Tables" on page 37.

3. Review the default configuration of the application and modify it, if required.

   SAS Customer Analytics for Communications has number of predefined libraries, source tables, time periods for behavioral variables, subset maps, and subject groups. These objects are preconfigured for use in the application. If required, you can modify these objects according to your requirements. You can also define and configure new objects. For details, see Chapter 4, "Configuring SAS Customer Analytics for Communications," on page 47.

4. Log on to SAS Customer Analytics for Communications as a business analyst and complete these tasks:

   a. Create a project for your business problem. For more information, see the *SAS Customer Analytics for Communications: User's Guide*.

   b. Define a subset criterion to define your target population—the population on which you want to perform your analysis. If you want to perform your analysis on the entire population, you can skip this step. For more information, see the *SAS Customer Analytics for Communications: User's Guide*.

c. Define and built a modeling analytical base table (ABT). Alternatively, you can define the ABT and deploy it. For more information, see the *SAS Customer Analytics for Communications: User's Guide*.

5. Log on to SAS Enterprise Miner, create an analytical model, and register the model with the SAS Metadata Server.

6. Log on to SAS Customer Analytics for Communications as a business analyst, and complete the following tasks. For more information, see the *SAS Customer Analytics for Communications: User's Guide*.

   a. Capture the details of the model that you created in SAS Enterprise Miner and registered with the SAS Metadata Server.

   b. Publish the model for scoring.

   Note: For a segmentation project, edit the names and descriptions of segments in the related Application data tables. For details, see "Editing Segment Details" on page 133.

   c. Create a scoring template for the published model.

7. Populate the Application data by running the scoring job. For details, see "The Scoring Job" on page 99.

8. Populate the Foundation data mart from the Analytical results area by using the prepackaged ETL jobs. For details, see "Writeback ETL Jobs" on page 40.

9. Run the OLAP cubes. For more information, see the *SAS Communications Analytics Architecture: Administrator's Guide*.

10. Log on to SAS Web Report Studio and view business intelligence reports. For more information, see the *SAS Communications Analytics Architecture: User's Guide*.

Chapter 2
Installation and Configuration

Pre-Installation Tasks

Verify System Requirements

Review the system requirements documentation to ensure that your system meets the appropriate requirements. For more information, see *System Requirements for SAS Customer Analytics for Communications*. You can access the document from the following location: **http://support.sas.com/resources/sysreq/**.

Complete the Pre-Installation Tasks for SAS Intelligence Platform

Before you begin to install SAS Intelligence Platform and SAS Customer Analytics for Communications, read the *SAS Intelligence Platform: Installation and Configuration Guide*, which is located at **http://support.sas.com/documentation/cdl/en/biig/62611/PDF/default/biig.pdf**. SAS Customer Analytics for Communications is designed to work with SAS Intelligence Platform. Therefore, reading the documentation will help you understand the pre-installation tasks and guide you through a typical installation of SAS Intelligence Platform.

Complete the Pre-Installation Tasks for SAS Communications Analytics Architecture

Before you install SAS Customer Analytics for Communications, make sure that you complete the pre-installation instructions that are detailed in *SAS Communications Analytics Architecture: Administrator's Guide*. For information about how to access SAS Communications Analytics Architecture documentation, see **http://support.sas.com/documentation/onlinedoc/securedoc/index_caa.html**.

Obtain a Deployment Plan

Before you can install SAS Customer Analytics for Communications, you must obtain a deployment plan. The deployment plan is a summary of the software that is installed and configured during your installation. A deployment plan file, named plan.xml, contains information about what software should be installed and configured on each machine in your environment. This plan serves as input to the SAS installation and configuration tools. SAS includes a standard deployment plan. You can use this standard plan or create your own plan. For more information, see “About Deployment Plans” in the *SAS Intelligence Platform: Installation and Configuration Guide*, which is located at **http://support.sas.com/documentation/cdl/en/biig/62611/PDF/default/biig.pdf**.

Note:

- When you create a plan file with the assumption that SAS Communications Analytics Architecture will be installed with SAS Customer Analytics for Communications, make sure that you select **SAS Communications Analytics Architecture** along with SAS Customer Analytics for Communications in the planning application.
- SAS/ACCESS for Teradata and MySQL are not readily available in the SAS Customer Analytics for Communications planning data. Therefore, you have to

select a product that uses SAS/ACCESS for Teradata and MySQL and include the appropriate SAS ACCESS product in your plan file data.

Create a SAS Software Depot

Download the software that is listed in your SAS Software Order with the SAS Download Manager. This creates a SAS Software Depot, which includes the SAS installation data (SID) file. The SID file is used by the SAS system to install and license SAS software. It is a control file that contains license information that is required to install SAS. After you have downloaded the SAS Software Depot, you can then use the SAS Deployment Wizard to install your software. For more information, see “Creating a SAS Software Depot” in the *SAS Intelligence Platform: Installation and Configuration Guide*, which is located at **http://support.sas.com/documentation/cdl/en/biig/62611/PDF/default/biig.pdf**.

Determine the Location of the SAS Environment URL

During deployment, you are prompted by the SAS Deployment Wizard to specify a URL location of the SAS environment file, named sas-environment.xml (for example, **http://<your HTTP server>/sas-environment.xml**). This file defines a set of SAS deployments at your site for client applications to use. The sas-environment.xml file does not need to physically exist at the URL location that you specify in the SAS Deployment Wizard before beginning the SAS installation.

However, knowing the intended value of this URL is important because every client installation is prompted for this value. If you do not specify the URL when SAS Customer Analytics for Communications is installed, then as a post-installation task, you must manually edit a file on every client machine to specify this URL. Therefore, it is beneficial to decide on a value for this URL during your planning process so that it can be provided to administrators who might be performing an installation.

For more information about the structure of this file, see “Configuring the SAS Environment File” in the *SAS Intelligence Platform: Middle-Tier Administration Guide*, which is located at **http://support.sas.com/documentation/cdl/en/bimtag/64207/PDF/default/bimtag.pdf**.

Setting Up the MySQL Database

Overview

SAS Customer Analytics for Communications requires a MySQL instance for the Application data. This instance should run successfully before you start the SAS Deployment Wizard.

Set Up the Environment

If you are installing a fresh MySQL Server, make sure that you have selected the **Include Bin Directory in Windows Path** option. If you do not select this option, you have to manually add the MySQL bin path as mentioned below:

For Windows environment, add the following path for the Windows PATH variable: **<MySQL Installation Dir>/MySQL Server 5.1/bin;**.

Note: Make sure that the Windows PATH variable contains the path to the directory where libmysql.ddl resides.

For UNIX environment perform the following steps:

1. Add the following lines in the .profile file:

   ```
   export PATH=$PATH:<MySQL_Install_Dir>/bin;
   ```

   ```
   export LIBPATH=$LIBPATH:<MySQL_Install_Dir>/lib
   ```

   ```
   LD_LIBRARY_PATH=<MySQL_Install_Dir>/lib:$LD_LIBRARY_PATH
   ```

   ```
   export LD_LIBRARY_PATH
   ```

   ```
   export MYSQL_LIBPATH=<MySQL_Install_Dir>/lib
   ```

   Note:

   - If you are installing MySQL on H6i/UNIX platform, add the following line in the .profile file:

     ```
     export SHLIB_PATH=<MySQL_Install_Dir>/lib
     ```

   - In order to confirm that the paths are updated in the session, you must log on to UNIX shell before restarting the SAS Deployment Wizard.

2. Make sure that you are installing 64-bit MySQL if you are installing MySQL on a 64-bit machine.

3. Add the path for **`basedir=`** and **`datadir= in`** in the **`<MySQL_Install_Dir>/support-files/mysql.server`** file as mentioned below:

   ```
   basedir=<MySQL_Install_Dir>
   ```

   ```
   datadir=<MySQL_Install_Dir>/data
   ```

   Note: If MySQL server is not installed on the same machine on which Customer Analytics for Communications Server is to be configured, then make sure that you install and configure the MySQL client software.

Create a MySQL Instance and User ID for Application Data

To create a MySQL database instance and user ID for the Application data:

1. Log on to the MySQL database with the default user.

2. Create a database for the Application data. For example, to create a database instance named cscapdm, you can enter the following command:

   ```
   CREATE DATABASE cscapdm
   ```

   The subsequent sections refer to this database instance as cscapdm.

3. Create a new user. For example, you can enter the following command:

   ```
   CREATE USER <User name>'@'<Machine name>' IDENTIFIED BY '<Password>'
   ```

4. Grant permission to this user on the database that you created in step 2. For example, if you have created a user named CACuser, you can enter the following command:

   ```
   GRANT ALL PRIVILEGES ON apdm.* TO 'CACuser'@'<machine Name>'
   ```

   The subsequent sections refer to this user as CACuser.

5. Perform the following steps if you are creating the database instance on a UNIX machine:

   a. Stop MySQL server.

   b. Open the my.cnf file that is available in the root MySQL installation folder and add the following line after the line **`myisam_sort_buffer_size`**:

      ```
      lower_case_table_names=1
      ```

c. Start the MySQL server.

6. Create a sample table in the cscapdm database by using the credentials that you created in step 3. This step ensures that the CACuser has the CREATE TABLE privileges. As a result, the CAC APDM DDL that will be run by the SAS Deployment Wizard will run without errors.

Prerequisite Steps for Teradata

Overview

If the Foundation data mart is on Teradata, make sure that you install and configure the Teradata client on a machine on which Customer Analytics for Communications Server Tier is to be installed and configured. Contact your Database Administrator to set up the client software. Also, make sure that the required databases are created on the Teradata Server

Create a Super User

Create a user on the Teradata server. This user will be a super user who will perform all the operations that are relevant for SAS Customer Analytics for Communications. In addition, this user would own all the databases that you will create.

Create Databases

Create appropriate databases on the Teradata server. A separate database should be created for each library of the Foundation data mart. The following table mentions the details of databases that you need to create. All these databases should reside under the super user. The child databases should be a part of the same parent database.

Table 2.1 *Solution-Specific Libraries and Databases*

Library Name and Library Path	Libref	Schema name
BASE	CFDBASE	CFDBASE
CACARM	ARM	ARM
Modeling ABT	CAC_MDL	CACMDL
Scoring ABT	CAC_SCR	CACSCR
Actual ABT	CAC_ACT	CACACT
Modeling ABT Scratch	CACMDSCR	CACMDSCR
Scoring ABT Scratch	CACSCSCR	CACSCSCR
Actual ABT Scratch	CACATSCR	CACATSCR

Grant Privileges to the Super User

By using the Teradata client, grant the following permissions to the super user. Contact your Database Administrator for assistance.

GRANT ALL ON <Database name> to <Super user name>

Replace <Database name> with the appropriate schema name. For details, see Table 2.1 on page 15.

Replace <Super user name> with the appropriate user name that you created earlier. For details, see "Create a Super User" on page 15.

For example, for the CFDBASE schema and the bppexec super user, enter the following command:

```
GRANT all on CFDBASE to bppexec;
```

Note: Make sure that you grant all permissions to the super user for each database that is listed in the above table (Table 2.1 on page 15).

In addition, if you are using SAS Scoring Accelerator for Teradata, grant the following privileges to the super user (bppexec):

```
grant EXECUTE PROCEDURE ON SAS_SYSFNLIB TO bppexec;

grant EXECUTE FUNCTION ON SAS_SYSFNLIB TO bppexec;

grant EXECUTE FUNCTION ON SYSLIB.MonitorVirtualConfig TO bppexec;
```

Note: For more details about the other prerequisite tasks for SAS Scoring Accelerator for Teradata, see Chapter 6, "Configuring SAS Analytics Accelerator for Teradata" of *Configuration Guide for SAS 9.3 Foundation for Microsoft Windows*, which is available at **http://support.sas.com/documentation/installcenter/en/ikfdtnwincg/64431/PDF/default/config.pdf**.

Export Environment Variables

Export environment variable for the Teradata client according to your platform.

Table 2.2 *Environment Variables*

Platform	Environment Variables

AIX	LIBPATH=TPT-API-LIBRARY-LOCATION
	NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION
	LC__FASTMSG=false
	Note: LC__FASTMSG contains two underscores.

HP-UX and HP-UX for the Itanium Processor Family	SHLIB_PATH=TPT-API-LIBRARY-LOCATION
	NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION

Linux for Intel Architecture, Linux for x64, and Solaris for x64	LD_LIBRARY_PATH=TPT-API-LIBRARY-LOCATION
	NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION

Specifying Library Names

Various Teradata libraries are created during SAS Deployment Wizard. You will be prompted to specify database name for each of these libraries. Default values are provided for these prompts as mentioned in the above table. However, you can change these values according to your planned database setup.

Note: It is not mandatory that these databases should be available during SAS Deployment Wizard installation. However, make sure that you create them before you execute the post-installation instructions. This is because during post-installation, you have to execute certain DDLs in order to create tables in these databases.

Providing 64-bit Libraries Path

You need to provide the 64-bit libraries path if you want to install SAS in 64-bit native mode. To confirm whether you are using 64-bit libraries for 64-bit SAS, execute the **`env`** command on UNIX platform or the **`set`** command on other platforms.

For example, the output of **`env`** execution on an AIX platform that lists variables specific to Teradata environment variables is as follows:

```
LIBPATH=/install/mysql/5.1.47/lib:/opt/teradata/client/13.10/

lib64:/opt/teradata/client/13.10/tdicu/lib64:/opt/teradata/client/ODBC_64/lib
```

Default File Locations

The SAS Deployment Wizard installs and configures your SAS software. The application installation files are installed in a default location referred to as <SAS Home>. For example, on a Windows machine, <SAS Home> is **`C:/Program Files/SASHome`**.

The configuration files are stored in a default location referred to as <SAS configuration directory>. For example, on a Windows machine, <SAS configuration directory> is **`C:/SAS/Config`**.

Note: You can deploy up to 10 configurations of the SAS products. The SAS Deployment Wizard specifies each configuration under a <Level> folder. For example, if you deploy a level 2 configuration, the default configuration directory is **`C:/SAS/Config/Lev2`**.

The following table lists the default locations of the installation and configuration files for SAS Customer Analytics for Communications.

Table 2.3 *Default Locations*

Location Name	Windows Path	UNIX Path
<SAS Home>	**`C:/Program Files/SASHome`**	**`../SASHome`**
<SAS configuration directory>	**`C:/SAS/Config`**	**`../SAS/Config`**

Installation Instructions

SAS Customer Analytics for Communications works with SAS Intelligence Platform, which is also installed and configured when you deploy SAS Customer Analytics for Communications. You use the SAS Deployment Wizard to install and configure your software. For more information, see the *SAS Deployment Wizard: User's Guide*, which is located at `http://support.sas.com/documentation/installcenter/en/ikdeploywizug/62130/PDF/default/user.pdf`.

This section explains installation instructions that are specific to SAS Customer Analytics for Communications. For complete installation and configuration instructions, see the *SAS Intelligence Platform: Installation and Configuration Guide*, which is located at `http://support.sas.com/documentation/cdl/en/biig/62611/PDF/default/biig.pdf`.

If you are installing SAS Customer Analytics for Communications along with SAS Communications Analytics Architecture, then make sure that you refer to the solution-specific installation and configuration instructions that are explained in the *SAS Communications Analytics Architecture: Administrator's Guide*.

To install SAS Customer Analytics for Communications and other relevant SAS products by using the SAS Deployment Wizard, complete these steps:

1. Log on to the machine on which you want to install SAS Customer Analytics for Communications.

   On a Windows machine, log on as a user who is a member of the Administrators group. On a UNIX machine, log on as a SAS user (for example, sas) that you defined when performing the SAS Intelligence Platform pre-installation tasks.

   Note: It is recommended that you do not log on as root to perform an installation on a UNIX machine.

2. Start the SAS Deployment Wizard from your SAS Software Depot. For example, on a Windows machine, double-click the setup.exe file, which is located in your SAS Software Depot folder.

3. Open the *SAS Intelligence Platform: Installation and Configuration Guide*, which is located at `http://support.sas.com/documentation/cdl/en/biig/62611/PDF/default/biig.pdf`.

   TIP On the Select Deployment Step and Products to Install page of the wizard, make sure that you select **SAS Customer Analytics for Communications Server** from the **Product** list.

4. (Optional) If you are installing SAS Customer Analytics for Communications along with SAS Communications Analytics Architecture, see the *SAS Communications Analytics Architecture: Administrator's Guide* for the solution-specific installation instructions.

5. Perform step-by-step installation and configuration as explained in the respective guides.

6. On the SAS Communications Analytics Architecture Server Configuration page, specify the database configuration information. For details, see *SAS Communications Analytics Architecture: Administrator's Guide*.

7. On the SAS Communications Common Server Configuration page, specify the following details for the APDM Database User Configuration, and then click **Next**.

Table 2.4 *APDM Database User Configuration Details*

Field Label	Value
APDM Database Administrative User ID	Enter the user ID of the user who has the administrative privileges.
Password	Enter the password for the **APDM Database Administrative User ID**.
Confirm Password	Enter the same password again. Click **Next** to specify the schema names.

8. On the SAS Communications Common Server Configuration page, specify the following details for the Base Tables Configuration, and then click **Next**.

Table 2.5 *Database Configuration Information Details*

Field Label	Value
Base Tables for Analytics Database Schema Name	Enter the schema name of the database that stores the base tables for analytics. The default value is CFDBASE.
Application Data Mart Database Schema Name	Enter the schema name of the database that stores the application data.

9. On the SAS Customer Analytics for Communications Server Configuration page, specify the following details, and then click **Next**:

Table 2.6 *Libraries*

Field Label	Value
ARM Database Schema Name	Enter the schema name of the ARM database. The default value is CACARM.
Modeling ABT Database Schema Name	Enter the schema name of the database that contains tables that are relevant for the modeling ABTs. The default value is CACMDL.
Scoring ABT Database Schema Name	Enter the schema name of the database that contains tables that are relevant for the scoring ABTs. The default value is CACSCR.
Actual ABT Database Schema Name	Enter the schema name of the database that contains tables that are relevant for the actual ABTs. The default value is CACACT.
Modeling ABT Scratch Database Schema Name	Enter the schema name of the database that contains the temporary tables that are relevant for the modeling ABTs. The default value is CACMDSCR.

Field Label	Value
Scoring ABT Scratch Database Schema Name	Enter the schema name of the database that contains the temporary tables that are relevant for the scoring ABTs. The default value is CACSCSCR.
Actual ABT Scratch Database Schema Name	Enter the schema name of the database that contains the temporary tables that are relevant for the actual ABTs. The default value is CACATSCR.

10. On the SAS Customer Analytics for Communications Database JDBC page, provide the location of the JDBC Driver for the MySQL database. Make sure that the MySQL connector JAR is the only JAR in the folder.
11. On the Deployment Summary page, review the list of products that you are about to install, and then click **Start**.
12. On the Deployment Complete page, the SAS Deployment Wizard indicates that the installation and configuration of your SAS software is complete. Each software component should have a check mark beside it in the list. Click **Next**.
13. On the Additional Resources page, review the manual configuration instructions in the instructions.html file. This file is saved to your SAS configuration directory during the deployment (for example, **`C:/SAS/Config/Lev1/Documents/instructions.html`**). You can also view the list of the links for additional resources about your deployment. You can print the list of the links for future reference.
14. Click **Finish** to close the SAS Deployment Wizard.

    Note: On a UNIX platform, certain files require setuid permissions. To set the setuid permissions, run the setuid.sh script, which is located in the **`/usr/local/SASHome/SASFoundation/9.3/utilities/bin`** directory.

Validation Instructions

Verify the Installation of SAS Customer Analytics for Communications

To verify whether SAS Customer Analytics for Communications has been installed properly, complete these steps:

1. Enter the application URL in the address field of your web browser, and then press ENTER. For example, you might enter **`http://server01.abc.com:9080/SASCustAnalyticsComm/`**.
2. Log on to SAS Customer Analytics for Communications as a user who has all the capabilities of the Business Analysis role. For details, see "Roles and Capabilities" on page 91.
3. You should see the following workspaces:
    - Projects

- Scoring

Note: You see the Administrative workspace only if you log on to the application as an administrator.

4. In the Projects workspace, check that you can create a project. After you create a project, you can open it and define its components such as a subset criterion, an analytical base table (ABT), and variables. For instructions on how to create a project and define its components, see *SAS Customer Analytics for Communications: User's Guide*.

   Note: Installing SAS Customer Analytics for Communications does not automatically populate the base tables. Base tables are populated when you run the ETL jobs. For details, see "Populating the Base Tables" on page 37. Therefore, unless you populate the base tables with the required data, you cannot perform the tasks that access the base tables. These tasks include the following:

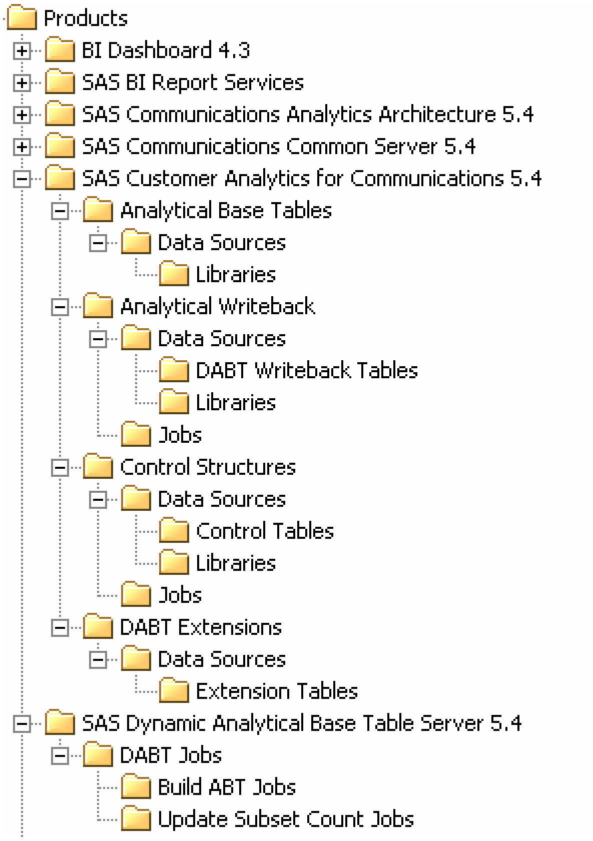
   - build a modeling ABT
   - view the number of members in the target population defined through a subset criterion (the Show Count operation)

Verify the Metadata Layout

To verify that all the required metadata components are created successfully, complete these steps:

1. Log on to SAS Management Console with a certain profile.
2. On the **Folders** tab, expand **Products** ⇨ **SAS Customer Analytics for Communications 5.4**.
3. Confirm that the following subfolders are created in each of these folders.

Figure 2.1 *Metadata Layout*



```
Products
  BI Dashboard 4.3
  SAS BI Report Services
  SAS Communications Analytics Architecture 5.4
  SAS Communications Common Server 5.4
  SAS Customer Analytics for Communications 5.4
    Analytical Base Tables
      Data Sources
        Libraries
    Analytical Writeback
      Data Sources
        DABT Writeback Tables
        Libraries
      Jobs
    Control Structures
      Data Sources
        Control Tables
        Libraries
      Jobs
    DABT Extensions
      Data Sources
        Extension Tables
  SAS Dynamic Analytical Base Table Server 5.4
    DABT Jobs
      Build ABT Jobs
      Update Subset Count Jobs
```

4. Close SAS Management Console.

Verify Roles and Capabilities

To verify that appropriate roles and capabilities are created, complete the following steps:

1. Log on to SAS Management Console with the profile of an administrator.
2. On the **Plug-ins** tab, expand **Environment Management** and select **User Manager**.
3. In the right pane, make sure that the following roles are available:
   - Cust Analytics Comm Mid: Administration
   - Cust Analytics Comm Mid: Business Analysis
   - Cust Analytics Comm Mid: Data Analysis
4. Right-click on any one of the roles and select **Properties**.
5. On the **Capabilities** tab, make sure that the role-specific capabilities are selected for the role. For details, see "Roles and Capabilities" on page 91.
6. Close SAS Management Console.

Post-Installation Tasks

Overview of Post-Installation Tasks

If the installation is successful, the SAS Deployment Wizard produces an HTML file named instructions.html. Follow the post-installation tasks in this file to complete the configuration of the server tier and the middle tier. The instructions.html file also describes when to perform the tasks that are outlined in this chapter.

Additional information about the instructions.html file and supplemental instructions are available in the SAS Intelligence Platform documentation.

Create Users for SAS Enterprise Miner

Overview

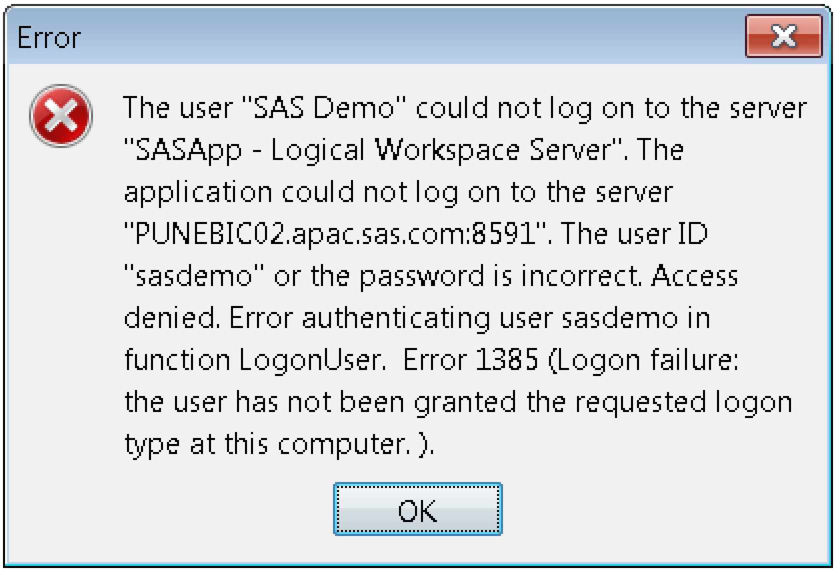
Analytical models for customer churn, customer segmentation, cross-sell and up-sell, and so on, are created using SAS Enterprise Miner. Therefore, you must create a user in the operating system for SAS Enterprise Miner. Also, you have to add this user ID in the SAS metadata by using SAS Management Console and grant privileges that are related to SAS Enterprise Miner.

Create Operating System User for SAS Enterprise Miner

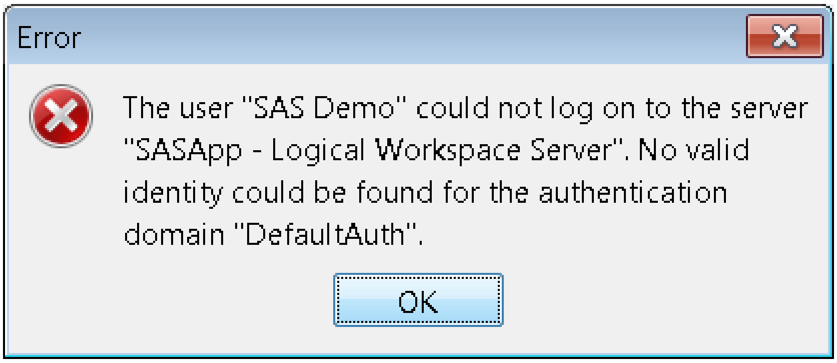
On a Windows machine, to create a user and add security policies, perform these steps:

1. Create a user in the operating system.
2. Go to **Control Panel** ⇨ **System and Security** ⇨ **Administrative Tools**.
3. Double-click **Local Security Policy**.
4. In the left pane, expand **Local Policies**, and then select **User Rights Assignment**.
5. Add the user ID that you have created to the following security policies:
    - act as part of the operating system
    - log on as a batch job

    Note: If you do not assign the security policies to the user, you might encounter the following errors when you work in SAS Enterprise Miner:



The user "SAS Demo" could not log on to the server "SASApp - Logical Workspace Server". The application could not log on to the server "PUNEBIC02.apac.sas.com:8591". The user ID "sasdemo" or the password is incorrect. Access denied. Error authenticating user sasdemo in function LogonUser. Error 1385 (Logon failure: the user has not been granted the requested logon type at this computer.).



The user "SAS Demo" could not log on to the server "SASApp - Logical Workspace Server". No valid identity could be found for the authentication domain "DefaultAuth".

Create User in SAS Management Console for SAS Enterprise Miner

Note: It is recommended that the user who creates projects in SAS Customer Analytics for Communications is the same as the SAS Enterprise Miner user who creates and registers the analytical models for these projects.

To create a user for SAS Enterprise Miner, complete these steps:

1. Open SAS Management Console with the default profile of sasadm user.
2. On the **Plug-ins** tab, expand **Environment Management**.
3. Right-click **User Manager**, and then select **New** ⇨ **User**. The New User Properties dialog box appears.
4. On the **General** tab, enter the appropriate name and description of the user, and then click **OK**.
5. Select the **Groups and Roles** tab.
   a. Select **Metadata Server: Unrestricted** from the **Available Groups and Roles** list and move it to the **Member of** list.
   b. Click **OK**.
6. Select the **Accounts** tab.
   a. Click **New**. The New Login Properties dialog box appears.
   b. Enter appropriate **User ID** and **Password**.

Note: Make sure that the user ID that you create is the same as the user ID of the operating system user that you have created. For details, see "Create Operating System User for SAS Enterprise Miner" on page 23.

c. Select **DefaultAuth** as the **Authentication Domain**.

d. Click **OK**.

7. Close SAS Management Console.

Check Log Files

Verify the log files that are created at the following locations:

Table 2.7 *Log Files*

Log Filename	**Path**
cscom_config_wrapper	**`<SAS configuration directory>/Lev1/AppData/SASCommunicationsCommonServer/5.4/logs`**
cac_dabt_config_wrapper	**`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm/5.4/logs`**

Verify Operating System Users and Grant Permissions

Verify that a user group named SAS Server Users exists at the operating system level. This user group is created as a part of the SAS installation. Make sure that this group includes the sassrv user (a SAS Server user) and the sasdemo user (a SAS user). Also, make sure that all users who will log on to the application are part of this user group.

Grant the SAS Server Users group Full Control permission (on Windows) or 775 permission (on UNIX) for the following folders and their subfolders:

- **`<SAS configuration directory>/Lev1/AppData/SASCommAnalyticsArch`**
- **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm`**
- **`<SAS configuration directory>/Lev1/AppData/SASDynamicAnalyBaseTblSvr`**
- **`<SAS configuration directory>/Lev1/AppData/SASCommunicationsCommonServer`**
- **`<SAS configuration directory>/Lev1/Applications/SASCommAnalyticsArch5.4`**
- **`<SAS configuration directory>/Lev1/Applications/SASCommunicationsCommonServer5.4`**
- **`<SAS configuration directory>/Lev1/Applications/SASCustAnalyticsCommServer5.4`**

- `<SAS configuration directory>/Lev1/Applications/SASDynamicAnalyBaseTblSvr5.4`

Define Users and Assign Roles

You need to define users who can log on to SAS Customer Analytics for Communications and perform tasks based on their assigned roles. SAS Customer Analytics for Communications is shipped with three predefined roles: Business Analysis, Data Analysis, and Administration. A predefined set of capabilities is available for each role.

You can define users and assign roles in SAS Management Console. For more information, see the *SAS 9.3 Management Console: Guide to Users and Permissions*. This document is available at `http://support.sas.com/documentation/cdl/en/mcsecug/63190/PDF/default/mcsecug.pdf`.

Add Users to the Appropriate User Group

After you create users and assign appropriate roles to them, add those users to the **Customer Analytics for Communications Users** group.

Note: Make sure that the **Customer Analytics for Communications Users** group has the WriteMetadata permission for the **Modeling ABT** library. This library is available at the following location on the **Folders** tab of SAS Management Console: **SAS Folders** ⇨ **SAS Customer Analytics for Communications 5.4** ⇨ **Analytical Base Tables** ⇨ **Data Sources** ⇨ **Libraries**.

Provide ReadMetadata and Read privilege on the following folders to **Customer Analytics for Communications Users** user group. In SAS Management Console, the metadata folders are available at the following locations:

- **Products** ⇨ **SAS Communications Analytics Architecture 5.4**
- **Products** ⇨ **SAS Communications Common Server 5.4**
- **Products** ⇨ **SAS Customer Analytics for Communications 5.4**
- **Products** ⇨ **SAS Dynamic Analytical Base Table Server 5.4**

Note: Make sure that the **SAS General Servers** user group is granted full permissions to the above-mentioned folders.

In addition, provide ReadMetadata and Read privilege on the below mentioned metadata folders to the **SAS General Servers** user group and the **Customer Analytics for Communications Users** user group. In SAS Management Console, the metadata folders are available at the following locations

- **System** ⇨ **Applications** ⇨ **SAS Communications Analytics Architecture**
- **System** ⇨ **Applications** ⇨ **SAS Communications Common Server Configuration**
- **System** ⇨ **Applications** ⇨ **SAS Customer Analytics for Communications**
- **System** ⇨ **Applications** ⇨ **SAS Dynamic Analytical Base Table Server**

Verify the Library Definitions in Metadata

Verify that the following library definitions available in the Metadata. Also, make sure that all libraries are assigned to your application server, which could be named as SASApp.

Table 2.8 Solution-Specific Libraries and Databases

Library Name and Library Path	Metadata Path in SAS Management Console	Libref	Schema name
BASE	**Products ⇨ SAS Communications Common Server 5.4 ⇨ Base Tables Mart ⇨ Data Sources ⇨ Libraries**	CFDBASE	CFDBASE
CACARM	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Writeback ⇨ Data Sources ⇨ Libraries**	ARM	ARM
Modeling ABT	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CAC_MDL	CACMDL
Scoring ABT	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CAC_SCR	CACSCR
Actual ABT	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CAC_ACT	CACACT
Modeling ABT Scratch	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CACMDSCR	CACMDSCR
Scoring ABT Scratch	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CACSCSCR	CACSCSCR

Library Name and Library Path	Metadata Path in SAS Management Console	Libref	Schema name
Actual ABT Scratch	**Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Base Tables ⇨ Data Sources ⇨ Libraries**	CACATSCR	CACATSCR

Verify Initialization Files

To verify the initialization files, run the following code in SAS Data Integration Studio.

```
%cacinit(set_sql_func_ex_append=Y);
```

Verify the log file. If any errors are logged, then it indicates that there could be a possible issue with installation or configuration properties. Verify all the properties.

Run the Auto-Increment Script

SAS Customer Analytics for Communications provides a script that enables you to initialize SK values of various entities such as subset maps, subjects of analysis, and so on.

To run the auto-incrment script, complete these steps:

1. Log on to SAS Data Integration Studio and connect to the appropriate profile.
2. In the code editor, open the cac_alter_auto_increment.sas file. This file is available at the following location:

   Windows path
   `<SAS Home>/SASFoundation/9.3/cacsrv/sasmisc/controlscripts`

   UNIX path
   `<SAS Home>/SASFoundation/9.3/misc/cacsrv/controlscripts`
3. Run the code.
4. Make sure that the log of the code editor does not contain any errors.

Execute Base and ARM DDLs

Perform these steps if your Foundation data mart is in Teradata.

Run the following DDL files in your Teradata client.

base_teradata.ddl
This file is available in the following location:

Windows path
`<SAS Home>/SASFoundation/9.3/cscomsrv/sasmisc/ddl/Teradata`

For example, on a Windows machine, this location can be: **`C:/Program Files/SASHome/SASFoundation/9.3/cscomsrv/sasmisc/ddl/Teradtata`**.

UNIX path

`<SAS Home>/SASFoundation/9.3/misc/cscomsrv/ddl/Teradata`

arm_teradata.ddl

This file is available in the following location:

Windows path

`<SAS Home>/SASFoundation/9.3/cacsrv/sasmisc/ddl/Teradtata`

For example, on a Windows machine, this location can be: **`C:/Program Files/SASHome/SASFoundation/9.3/cacsrv/sasmisc/ddl/Teradtata`**.

UNIX path

`<SAS Home>/SASFoundation/9.3/misc/cacsrv/ddl/Teradtata`

Publish SAS Formats in Teradata

If your Foundation data mart is in Teradata, you must publish SAS formats in Teradata. For details, see Publishing SAS Formats in Teradata in the *SAS 9.3 In-Database Products: User's Guide*. This guide is available at the following location: **`http://support.sas.com/documentation/cdl/en/indbug/65485/PDF/default/indbug.pdf`**.

Create User-Defined Functions in Teradata

In SAS Customer Analytics for Communications, date-related operations are performed using SAS date functions. However, when run on a Teradata database, these SAS functions import source data from the Teradata database into SAS, perform operations on the data, and then return the results to the Teradata database. This process deteriorates the overall performance because it involves data movement between SAS and Teradata.

To overcome this issue and to support Teradata processing of date-related operations, SAS Customer Analytics for Communications provides certain prepackaged SQL scripts. When you run these scripts in your Teradata database, these scripts create certain user-defined functions in your Teradata database. These functions are internally invoked by their equivalent SAS functions and perform the same operations in Teradata, which would, otherwise, be performed in SAS. This processing prevents data movement between Teradata and SAS, and thus optimizes the performance.

To create user-defined functions (UDFs) in Teradata, complete these steps:

1. Open your Teradata client (for example, BTEQ or SQL Query Assistant), and then connect to the desired Teradata database with at least Create Function and Execute Function privileges.

   Note: If you want to create global user-defined functions (functions are available to all databases in Teradata, and not restricted to one database), connect to the SYSLIB database.

2. Run the dabt_udf_teradata.sql file in your Teradata client. This SQL file is available in the following location:

   Windows path

   `<SAS Home>/SASFoundation/9.3/dabtsrv/sasmisc/ddl`

For example, on Windows, this location might be **`C:/Program Files/SASHome/SASFoundation/9.3/dabtsrv/sasmisc/ddl`**

UNIX path
`<SAS Home>/SASFoundation/9.3/misc/dabtsrv/ddl`

3. Grant the Execute Function privilege on the database that contains the user-defined functions to the Teradata database user.

If you create the functions in a database other than the SYSLIB database, you must also complete the following steps:

1. Start SAS Data Integration Studio, and then connect to the appropriate metadata server.

   Note: Make sure that you are connected to the correct application server context.

2. Select **Tools** ⇨ **Code Editor**.
3. Enter the following code in the Code Editor, and then click **Run**. If asked for a user name and password, use the sassrv user to run the code.

   Note: Before you run the code, replace <*database name*> with the name of the database in which you created the user-defined functions.

```
%cacinit(set_sql_func_ex_append=Y);

%let udf_database = <database name>;

data cscfunc.cscom_db_func_ext;

    set cscfunc.cscom_db_func_ext;

dbmsfuncname = "&udf_database.."||sasfuncname;

dbmsfuncnamelen = length("&udf_database.") + 1 + length(sasfuncname);

run;
```

Populate the Reference Tables

The value of the LEVEL_CD column of the LEVEL_MASTER table (in the CSCAPDM library) must be same as the value of the ANALYTICAL_MODEL_LEVEL_CD column of the ANALYTICAL_MODEL_LEVEL table (in the CFDREF library). Also, the value of the ANALYTICAL_MODEL_TYPE_CD column of the ANALYTICAL_MODEL_TYPE table (in the CFDREF library) must be same as the first two letters of the PURPOSE_CD column of the PURPOSE_MASTER table (in the CSCAPDM library).

To populate these tables with appropriate data based on these constraints, complete these steps:

1. Make sure that the ANALYTICAL_MODEL_LEVEL and ANALYTICAL_MODEL_TYPE tables do not contain any data.
2. Go to the **`<SAS configuration directory>/Lev1/<SAS Application Server context name>`** folder.
3. Depending on whether the operating system is Windows or UNIX, run the sas.bat or the sas.sh file respectively. For example, on the Windows machine, run the **`C:/SAS/Config93/Lev1/SASApp/sas.bat`** file.
4. View the pre-assigned libraries such as CFDREF, CFDDIM, and so on, in the SAS Explorer.

5. Open the cac_post_install_wrapper.sas macro. This macro is located in the following folder:

   Windows path
   `<SAS Home>/SASFoundation/9.3/cacsrv/sasmisc/controlscripts`.

   UNIX path
   `<SAS Home>/SASFoundation/9.3/misc/cacsrv/controlscripts`

6. Verify the arguments.

   ```
   %cac_post_install_wrapper(ref=CFDREF);
   ```

7. Enter appropriate values for the macro arguments. Ensure that the correct library reference for the reference library is provided as an argument to this macro.

8. Click **Save**.

9. Click **Run**.

10. To ensure successful execution of the INSERT scripts, view the log.

11. Close Base SAS.

Deploy the Master Loop Job

The MasterLoopDABTJob invokes the jobs that populate the ABT during the ABT building process.

To deploy the MasterLoopDABTJob job, complete these steps:

1. Log on to SAS Data Integration Studio and connect to an appropriate profile.

2. On the **Folders** tab, expand **Products** ⇨ **SAS Dynamic Analytical Base Table Server 5.4** ⇨ **Application Mart** ⇨ **Jobs**.

3. Right-click **MasterLoopDABTJob**, and then select **Scheduling** ⇨ **Deploy**.

4. In the Deploy a job for scheduling dialog box, from the **Batch Server** list, select **<Application server> - SAS DATA Step Batch Server**.

   Note: **<Application server>** represents the application server on which you installed SAS Customer Analytics for Communications. For example, if you installed SAS Customer Analytics for Communications on SASApp, select **SASApp – SAS DATA Step Batch Server**.

5. Click **New** to define the deployment directory. The New directory dialog box appears.

   a. In the **Name** field, enter a name for the deployment directory. For example, you can enter **`dabt_deployedjobs`**.

   b. In the **Directory** field, type **`<SAS configuration directory>/Lev1/Applications/SASDynamicAnalyBaseTblSvr5.4/deployed_jobs/dabt`**. Alternatively, you can click **Browse** to navigate to the deployment directory.

   c. Click **OK**.

6. Do not change the default values in the **Deployed Job Name** field and the **Location** field.

7. Click **OK**.

If the MasterLoopDABTJob job is deployed successfully, a file named MasterLoopDABTJob.sas is created in the deployment directory (**`<SAS configuration directory>/Lev1/Applications/SASDynamicAnalyBaseTblSvr5.4/deployed_jobs/dabt`**).

8. Edit the MasterLoopDABTJob.sas file and make the following changes:

   Note: You must edit the MasterLoopDABTJob.sas file and make these changes every time you deploy or redeploy the MasterLoopDABTJob job.

   a. Search for the line of code beginning with the following declaration: **`%let etls_controlName`**

      You will find two occurrences of this line of code.

   b. On the first occurrence, replace the value of the etls_controlName variable with the following: **`&outer_loop_remote_session_prefix.;`**

   c. On the second occurrence, replace the value of the etls_controlName variable with the following: **`&inner_loop_remote_session_prefix.;`**

9. Save the changes, and then close the MasterLoopDABTJob.sas file.

10. Close SAS Data Integration Studio.

Configure Parameters

SAS Customer Analytics for Communications has a set of parameters that are defined in the PARAMATER_MASTER table (in the CSCAPDM library). These parameters are essential to ensure proper functioning of the solution. During the installation, these parameters are assigned certain default values. Review these default values and modify them, as required in the PARAMETER_VALUE_DTL table (in the CSCAPDM library). For details, see “Parameters List” on page 111.

Unconfiguring SAS Customer Analytics for Communications

Prerequisite Tasks

Before you unconfigure SAS Customer Analytics for Communications, complete the following tasks:

1. Create a backup of the following folders:

   - **`<SAS configuration directory>/Lev1/AppData/SASCommunicationsCommonServer`**
   - **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm`**
   - **`<SAS configuration directory>/Lev1/AppData/SASDynamicAnalyBaseTblSvr`**

2. Create a backup of the application metadata if you have made any customizations that you want to save for later use. To do so, Start SAS Management Console, and

then open the appropriate connection profile to connect to the desired metadata server.

3. On the **Folders** tab, expand **SAS Folders** ⇨ **Products**.
4. Create a backup of the following folders:
   - **SAS Communications Common Server 5.4**
   - **SAS Customer Analytics for Communications 5.4**
   - **SAS Dynamic Analytical Base Table Server 5.4**
   - **User Folders**
5. Close SAS Management Console.
6. Stop the web application server (JBoss, WebSphere, or WebLogic, as applicable).

Remove SAS Customer Analytics for Communications

Use the SAS Deployment Manager to remove the following software components of SAS Customer Analytics for Communications:

- SAS Customer Analytics for Communications Mid-Tier
- SAS Customer Analytics for Communications Server
- SAS Communications Common Server

Post-Unconfiguration Tasks

After you have removed the software components of SAS Customer Analytics for Communications, delete the application metadata.

To delete the application metadata, complete the following tasks:

1. Start SAS Management Console, and then open the appropriate connection profile to connect to the desired metadata server.
2. On the **Folders** tab, expand **SAS Folders** ⇨ **Products**.
3. Delete the following folders:
   - **SAS Communications Common Server 5.4**
   - **SAS Customer Analytics for Communications 5.4**
   - **SAS Dynamic Analytical Base Table Server 5.4**
4. Close SAS Management Console.
5. Delete the following folders:
   - **`<SAS configuration directory>/Lev1/AppData/SASCommunicationsCommonServer`**
   - **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm`**
   - **`<SAS configuration directory>/Lev1/AppData/SASDynamicAnalyBaseTblSvr`**

Part 2

Data Management

Chapter 3

Chapter 3
Solution-Specific ETLs

Prerequisite Tasks

Before you proceed with performing the tasks that are mentioned in this chapter, make sure that the following tasks are complete:

1. Perform all the tasks that are detailed in the *SAS Communications Analytics Architecture: Administrator's Guide*. Especially, make sure that you have run all the extract, transform, and load (ETL) jobs. These are prerequisite jobs for running the ETL jobs that are required for SAS Customer Analytics for Communications.
2. Perform all the post-installation tasks that are detailed in this guide. For details, see "Post-Installation Tasks" on page 23.

Populating the Base Tables

In order to populate the base tables, you have to run the ETL jobs for analytics. These jobs extract data from the Foundation data mart and then load this data in the base tables.

The following table lists the jobs that you have to run for managing data that is required for the analytical components. For the recommended sequence in which you should run these jobs, see "Load Order Sequence for Bill-Monthly Jobs" on page 135 and "Load Order Sequence for Weekly Jobs" on page 136.

Table 3.1 ETL Jobs for Analytics — Prepaid

Job Name	Purpose	Primary Source Table	Target Table
cfd_pre_pd_subscrp_bucket_drvd_b_job	Loads various bucket-level customer variables that are precalculated for analytical processing such as calculating the duration between first recharge from activation. This job loads data for the completed week.	• PREPAID_NON_USAGE_CHARGE_F • PREPAID_USAGE_CHARGE_F • SUBSCRP_BUCKET_USAGE_SNPSHT_F	PRE_PD_SUBSCRP_BUCKET_DRVD_B
cfd_pre_pd_subscrp_usage_drvd_b_job	Loads various precalculated variables at subscription level. These variables are required for analytical processing such as calculating duration of last recharge from first usage. This job loads data for the completed week.	• PREPAID_NON_USAGE_CHARGE_F • PREPAID_USAGE_CHARGE_F	• PRE_PD_SUBSCRP_USAGE_DRVD_B • USAGE_SUMMARY_F • PAYMENT_F
cfd_pre_pd_cust_offer_snpsht_b_job	Loads the latest details of the all the offers that a prepaid customer owns.	• PRE_PD_CUST_TMP • OFFER_D	PRE_PD_CUST_OFFER_SNPSHT_B
cfd_pre_pd_cust_snpsht_b_job	Loads the details about prepaid customers. These details are related to current and previous offer bundle and subscription status (count of activation and deactivation).	• PRE_PD_CUST_TMP • OFFER_BUNDLE_D,SUBSCRP_D	PRE_PD_CUST_SNPSHT_B

Table 3.2 ETL Jobs for Analytics — Postpaid

Job Name	Purpose	Primary Source Table	Target Table
cfd_pst_pd_cust_acct_snpsht_b_job	Loads derived information about customers' accounts for postpaid customers. This job loads data for the completed billing cycle.	CUST_ACCT_BALANCE_SNPSHT_F	PST_PD_CUST_ACCT_SNPSHT_B
cfd_pst_pd_payment_drvd_b_job	Loads payment-related-derived information about postpaid customers. This job loads data for the completed billing cycle.	• BILL_X_PAYMENT_BRIDGE • BILL_F	PST_PD_PAYMENT_DRVD_B
cfd_pst_pd_cust_offer_snpsht_b_job	Loads the latest details of the all the offers that a postpaid customer owns.	• CUST_X_BILL_CYCLE_BRIDGE_TMP • OFFER_D	PST_PD_CUST_OFFER_SNPSHT_B

Job Name	Purpose	Primary Source Table	Target Table
cfd_pst_pd_cust_snpsht_b_job	Loads the details about postpaid customers. These details are related to current and previous offer bundle and subscription status (count of activation and deactivation).	• CUST_X_BILL_CYCLE_BRIDGE_TMP • OFFER_BUNDLE_D • SUBSCRP_D	PST_PD_CUST_SNPSHT_B

Table 3.3 ETL Jobs — Prepaid and Postpaid

Job Name	Purpose	Primary Source Table	Target Table
cfd_usage_weekly_summary_b_job	Aggregates usage data in the USAGE_WEEKLY_SUMMARY_B table at a weekly grain. This table is used as a data source for creating ABT variables.	• USAGE_SUMMARY_F • EVENT_FAILURE_SUMMARY_F	USAGE_WEEKLY_SUMMARY_B
cfd_usage_monthly_summary_b_job	Aggregates usage data in the USAGE_MONTHLY_SUMMARY_B table at a monthly grain. This table is used as a data source for creating ABT variables.	• USAGE_SUMMARY_F • EVENT_FAILURE_SUMMARY_F	USAGE_MONTHLY_SUMMARY_B

Configuration Job

You have to run the cfd_input_mart_load_dates_job configuration job in SAS Data Integration Studio after you run the ETLs that populate the base tables.

Table 3.4 Configuration Job Details

Job Name	Purpose	Primary Source Table	Target Table
cfd_input_mart_load_dates_job	Populates the latest load date from the SETUP_PARAM table into the INPUT_MART_LOAD_DATES table. The SETUP_PARAM table contains the latest date for which the data is loaded in the base tables. This date is populated into the INPUT_MART_LOAD_DATES table by using this job to synchronize the date in the base tables with the date to be chosen for the batch run of the scoring templates. The INPUT_MART_LOAD_DATES table is updated every time the input mart is loaded.	SETUP_PARAM	INPUT_MART_LOAD_DATE

Writeback ETL Jobs

The writeback ETL jobs write back the analytical scores to the Foundation data mart tables. You have to schedule these jobs for execution after you complete the scoring-related tasks. For details about the scoring tasks, see *SAS Customer Analytics for Communications: User's Guide* and Chapter 6, "Scoring," on page 99 of this guide. For the recommended sequence in which you should run these jobs, see "Load Order Sequence for Writeback Jobs" on page 136.

Table 3.5 Writeback Jobs

Job Name	Purpose	Primary Source Table	Target Table
cac_analytical_rule_master_dtl_job	Loads records that contain rules of MBA (market basket analysis) or sequence analysis models. A single record contains information about one rule. The target table contains one record for a combination of ANALYTICAL_MODEL_SK and RULE_SK.	SCORING_MODEL_RULE_MASTER	ANALYTICAL_RULE_MASTER_DTL

Job Name	Purpose	Primary Source Table	Target Table
cac_analytical_rule_dtl_job	Loads transformed records (that is, separate records for each entity in a specific rule) of rules of MBA or sequence analysis models. The target table contains one record for a combination of ANALYTICAL_MODEL_SK, RULE_SK, and RULE_UNIT_TYPE_CD per UNIT_ID.	SCORING_MODEL_RULE_DTLS	ANALYTICAL_RULE_DTL
cac_subscrp_rule_score_dtl_job	Loads scores of rules that are associated with models such as MBA and sequence analysis at subscription level. The SUBSCRP_RULE_SCORE_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK, SUBSCRP_MODEL_SCORE_DT, SUBSCRP_SK, and RULE_SK. The PREDICTIVE_MODEL_RUN_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and MODEL_SCORE_DT.	RULES_SCORING_RESULT_SUB	• SUBSCRP_RULE_SCORE_DTL • PREDICTIVE_MODEL_RUN_DTL
cac_cust_rule_score_dtl_job	Loads scores of rules that are associated with models such as MBA and sequence analysis at customer level. The CUST_RULE_SCORE_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK, CUST_MODEL_SCORE_DT, CUST_SK, and RULE_SK. The PREDICTIVE_MODEL_RUN_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and MODEL_SCORE_DT.	RULES_SCORING_RESULT_CUST	• CUST_RULE_SCORE_DTL • PREDICTIVE_MODEL_RUN_DTL

Job Name	Purpose	Primary Source Table	Target Table
cac_subscrp_model_score_dtl_job	Loads model scores at subscription level. SUBSCRP_MODEL_SCORE_DTL would contain one record per SUBSCRP_SK per ANALYTICAL_MODEL_SK per SUBSCRP_MODEL_SCORE_DT. PREDICTIVE_MODEL_RUN_DTL would contain one record per ANALYTICAL_MODEL_SK per MODEL_SCORE_DT.	PREDICTED_SCORING_RESULT_SUB	• SUBSCRP_MODEL_SCORE_DTL • PREDICTIVE_MODEL_RUN_DTL
cac_cust_model_score_dtl_job	Loads model scores at customer level. Also loads the aggregated scores (aggregated from subscription level records to customer level) to customer level tables. The CUST_MODEL_SCORE_DTL table contains one record for a combination of CUST_SK, ANALYTICAL_MODEL_SK, and CUST_MODEL_SCORE_DT. The PREDICTIVE_MODEL_RUN_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and MODEL_SCORE_DT.	PREDICTED_SCORING_RESULT_CUST	• CUST_MODEL_SCORE_DTL • PREDICTIVE_MODEL_RUN_DTL

Job Name	Purpose	Primary Source Table	Target Table
cac_prspct_cust_model_score_dtl_job	Loads model scores at prospective customer level. The PRSPCT_CUST_MODEL_SCORE_DTL table contains one record for a combination of PRSPCT_CUST_SK, ANALYTICAL_MODEL_SK, and PRSPCT_CUST_MODEL_SCORE_DT. The PREDICTIVE_MODEL_RUN_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and MODEL_SCORE_DT.	PREDICTED_SCORING_RESULT_PRS	• PRSPCT_CUST_MODEL_SCORE_DTL • PREDICTIVE_MODEL_RUN_DTL
cac_cust_analytical_sgmt_dtl_job	Loads analytical segment scores at customer level. The CUST_ANALYTICAL_SGMT_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK, CUST_MODEL_SCORE_DT, and CUST_SK. The SEGMENTATION_MODEL_RUN_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and SEGMENT_MODEL_SCORE_DT.	SEGMENT_SCORING_RESULT_CUST	• CUST_ANALYTICAL_SGMT_DTL • SEGMENTATION_MODEL_RUN_DTL

Job Name	Purpose	Primary Source Table	Target Table
cac_model_writeback_job	Loads all model details from the writeback tables to the Foundation data mart (FDM) tables. These details include model details such as model type code and model ID. This information also includes other attributes such as services or offers for cross-sell and up-sell and segment details for segmentation models. The ANALYTICAL_MODEL_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK, ENTITY_LEVEL_CD, and LEVEL_ENTITY_SK. The ANALYTICAL_SGMT_DTL table contains one record for a combination of ANALYTICAL_MODEL_SK and ANALYTICAL_SEGMENT_CD.	• SCORING_MODEL • SCORING_MODEL_RULE_MASTER • SCORING_MODEL_SEGMENT_MASTER • SERVICE_D • OFFER_D	• ANALYTICAL_MODEL_DTL • ANALYTICAL_SGMT_DTL • ANALYTICAL_MODEL_D

Part 3

Application Management

Chapter 4

Configuring SAS Customer Analytics for Communications

Configuring SAS Customer Analytics for Communications

Overview

SAS Customer Analytics for Communications is prepackaged with a number of libraries, data sources, subjects of analysis, time periods , subset maps, and subject groups. These objects are preconfigured for use in the application. However, if required, you can modify these objects according to your requirements. You can also define and configure new libraries, data sources, subjects of analysis, time periods, implicit subset criteria, implicit variables, subset maps, and subject groups.

Note:

- It is recommended that you make sure that no user is logged on to the application when you modify any of the predefined objects or define new objects. Also, the changes that you make are reflected in the application interface only on the next logon.
- Also, when you configure these objects, make sure that you restart the server-side web application. For details, see "Restarting the Server-Side Web Application" on page 89.

Configuring Libraries

SAS Customer Analytics for Communications provides a number of libraries that store the data sources, such as tables and information maps, that are prepackaged (and preconfigured for use). The LIBRARY_MASTER table (in the CSCAPDM library) stores the records of these libraries.

The following table shows the LIBRARY_MASTER table with sample records.

For descriptions of the columns in the LIBRARY_MASTER table, see *SAS Customer Analytics for Communications: Data Reference Guide.*

If you want to add new data sources for use in the application, you can add them to any of these preconfigured libraries. However, if required, you can also define a new library for your new data sources. To define a new library, you need to add a record for that library in the LIBRARY_MASTER table.

Note: Make sure that you enter a unique value for the LIBRARY_SK column. Also, make sure that this value is greater than 1000000.

Table 4.1 *LIBRARY_MASTER Table*

LIBRARY_SK	LIBRARY_ REFERENCE	LIBNAME_ STATEMENT	LIBRARY_ TYPE_CD	LIBRARY_ SHORT_NM	LIBRARY_ DESC
1	CSCAPDM		SRC_TBL	DABT APDM Library	DABT APDM Library
3	inpbase		SRC_TBL	BASE	BASE
5	FACT		SRC_TBL	FACT	FACT
6	DIM		SRC_TBL	DIM	DIM
13	CSC_IM		SRC_IM	Communications Common IM Library	Communications Common IM Library

Note: If you are not using the pre-assigned libraries, enter the correct library statement in the LIBNAME_STATEMENT column.

After you have added the record in the LIBRARY_MASTER table, the library name appears in the **Library** list in the Import Data Source window (in the Administrative workspace of the SAS Customer Analytics for Communications interface).

You can change the short name and description of a preconfigured library. You can also delete a library from the LIBRARY_MASTER table if the library is no longer used in the application—that is, there are no configured data sources that refer to that library.

Configuring Time Periods for Behavioral and Time-Based Variables

Before you can create behavioral and time-based variables in SAS Customer Analytics for Communications, you must define the time periods based on which the variables will be created. For behavioral variables, time periods determine the data aggregations that are possible when defining variables. For time-based variables, time periods determine the historical period that is to be considered when defining the variables. You can consider the entire historical period or a certain historical period. For example, the A_LST_PCE_INQTYP_B1M_INTRDT variable extracts historical data of only one month from the ABT build date and computes the latest interaction type code during that period. Similarly, the A_LST_PCE_INQTYP_HISTMT_INTRDT variable extracts all the historical data that is available in the source tables and computes the latest interaction type code. For more information about behavioral and time-based variables, see the *SAS Customer Analytics for Communications: User's Guide*.

SAS Customer Analytics for Communications provides a number of predefined time periods for behavioral and time-based variables. The SOURCE_TIME_PERIOD table (in the CSCAPDM library) stores the definitions of these time periods.

The following table shows the SOURCE_TIME_PERIOD table with sample time period definitions:

Table 4.2 SOURCE_TIME_PERIOD Table

TIME_ PERIOD_SK	TIME_ FREQUENCY_ SK	TIME_ PERIOD_ FROM	TIME_ PERIOD_TO	TIME_ PERIOD_CD	TIME_ PERIOD_ SHORT_NM	TIME_ PERIOD_ DESC
1	1	1	1	B1M	Base 1 Month	Base 1 Month
2	1	2	2	B2M	Base 2 Month	Base 2 Month
8	3	1	2	L2W	Last 2 Week	Last 2 Week
23	1	2	7	L27M	Last 2 to 7 Month	Last 2 to 7 Month
24	1	1	12	HISTMT	Over Entire History	Over Entire History
25	3	1	52	HISTWK	Over Entire History	Over Entire History

Note: You use the **Over Entire History** time periods for time-based variables. For this time period, the values of columns **TIME_PERIOD_FROM** and **TIME_PERIOD_TO** are configured as 1 and 12 for monthly time gain and 1 and 52 for weekly time grain. You can change these values, if you want to consider a larger historical period.

For descriptions of the columns in the SOURCE_TIME_PERIOD table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

You can use these predefined time periods to create behavioral variables and time-based variables. However, if required, you can also define new time periods. To define a new time period, you need to add a record for that time period in the SOURCE_TIME_PERIOD table.

Note: Make sure that you enter a unique value for the TIME_PERIOD_SK column. Also, make sure that this value is greater than 1000000.

You can change the short name and description of a preconfigured time period. You can also delete a time period from the SOURCE_TIME_PERIOD table if the time period is no longer used in the application—that is, there are no behavioral and time-based variables that are based on that time period.

Configuring Time Periods for Direct Variables

Before you can create direct variables in SAS Customer Analytics for Communications, you must define the time periods based on which the variables will be created. These time periods indicate the period for which you want to retrieve information for computing value of the variables. For more information about direct variables, see the *SAS Customer Analytics for Communications: User's Guide*.

SAS Customer Analytics for Communications provides a number of predefined time periods. The SOURCE_AS_OF_TIME_MASTER table (in the CSCAPDM library) stores the definitions of these time periods.

The following table shows the SOURCE_AS_OF_TIME_MASTER table with sample time period definitions:

Table 4.3 SOURCE_AS_OF_TIME_MASTER Table

AS_OF_TIME_SK	AS_OF_TIME_VALUE	TIME_FREQUENCY_SK	AS_OF_TIME_CD	AS_OF_TIME_SHORT_NM	AS_OF_TIME_DESC
1	0	1	CM	Latest	Latest
2	1	1	1MB	1 Month Back	1 Month Back
14	4	3	4WB	4 Week Back	4 Week Back
19	24	3	24WB	24 Week Back	24 Week Back

You can use these predefined time periods to create direct variables. For example, by using the **Latest** time period, you can derive direct variable values that are valid as of a particular ABT build date. By using time periods such as 1 Month Back (1MB) and 4 Week Back (4WB), you can derive direct variable values that are valid as of a particular time period from the ABT build date. If required, you can also define new time periods. To define a new time period, add a record for that time period in the SOURCE_AS_OF_TIME_MASTER table.

Note: Make sure that you enter a unique value for the AS_OF_TIME_SK column. Also, make sure that this value is greater than 1000000.

For descriptions of the columns in the SOURCE_AS_OF_TIME_MASTER table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

You can change the short name and description of a predefined time period. You can also delete a time period from the SOURCE_AS_OF_TIME_MASTER table if the time period is no longer used in the application—that is, there are no direct variables that are based on that time period.

Configuring Data Sources

In SAS Customer Analytics for Communications, data sources such as tables or information maps provide data for the subset criterion and analytical base table (ABT) variables that you define for a project.

SAS Customer Analytics for Communications provides a number of preconfigured data sources. When you log on to SAS Customer Analytics for Communications, the Administrative workspace shows a list of these preconfigured data sources. You can use these data sources to create subset maps and ABT variables in the application. However, if required, you can modify the default configuration according to your requirement.

Note: The prepackaged data sources and their columns are preconfigured for use. You do not need to modify the default configuration unless you modify the data sources (for example, add new columns or delete existing columns). However, dimensional attribute values (values of the columns with column type Dimensional Attribute) are specific to an implementation. Therefore, you must modify the prepackaged dimensional attribute values and create new ones according to your requirements.

You can also configure new data sources for use in the application. However, before that, you must import those data sources in the application. After importing a data source, you must configure it such that it can be used for creating subset maps or certain types of ABT variables. For example, you can configure a data source such that it can be used for creating only behavioral variables or only time-based variables.

A table that is configured for creating a particular type of variable appears in the **Data sources** list for that variable type in the New Variables window.

For information about how to configure data sources and perform other related tasks, see *SAS Customer Analytics for Communications: User's Guide*.

Configuring the Columns of a Data Source

If you add new columns in a data source that you have already imported into the application, you must import those columns before you can use them in the application. After you import the columns, you have to configure them according to your requirements. For details about how to import columns, configure them, and perform other related tasks, see *SAS Customer Analytics for Communications: User's Guide*.

Configure Dimensional Attribute Values for a Column

The values of columns, which are of Dimensional Attribute type can be implementation-specific and therefore cannot be preconfigured. Therefore, you have to configure all dimensional values according to your requirements. These values are required when you define a subset criterion or specify one or more dimensions when you define an ABT variable.

Note: For some columns of the dimensional attribute type, the values might be preconfigured. However, you must verify these values and change them according to your requirements. SAS Customer Analytics for Communications enables you to either import or define dimensional values. For details about how to configure the dimensional attributes of a source column, see *SAS Customer Analytics for Communications: User's Guide*.

Working with Information Maps

Predefined Information Maps

In order to optimize the performance of SAS Customer Analytics for Communications, most of the base tables are replaced with information maps. These information maps are used as data sources for defining ABT variables and subset criterion. As a result, information maps refer to the required data directly from the Foundation data mart tables and data duplication is reduced.

SAS Customer Analytics for Communications provides the following predefined information maps.

Table 4.4 *Predefined Information Maps*

Information Map Name	Description
CSCOM_BILL_USAGE_IM	Contains the bill usage data for postpaid customers.
CSCOM_CUST_BILL_IM	Contains bill-related information for postpaid customers.
CSCOM_CUST_BILL_NONUSAGE_IM	Contains non-usage related bill information for postpaid customers.

Information Map Name	Description
CSCOM_CUST_COMPLAINT_IM	Contains information about customer-level complaints that prepaid and postpaid customers have lodged.
CSCOM_CUST_INQUIRY_IM	Contains information about customer-level inquiries that prepaid and postpaid customers have made.
CSCOM_CUST_LOYALTY_IM	Contains customer-level loyalty points that prepaid and postpaid customers have earned.
CSCOM_CUST_OFFER_IM	Contains information about offer activation for TV customers.
CSCOM_DROPPED_USAGE_IM	Contains usage information about dropped calls for prepaid and postpaid customers.
CSCOM_EQUIP_ACTIVITY_IM	Contains information about status and the corresponding reason for status change for various pieces of equipment that are owned by prepaid and postpaid customers.
CSCOM_IMPLICIT_SUBSET_IMAP	Contains information that is required to define the implicit subset criterion for MBA and sequence analysis models.
CSCOM_IPTV_CHANNEL_USAGE_IM	Contains information about channel viewership for TV customers at monthly grain.
CSCOM_IPTV_USAGE_IM	Contains information about channel viewership for TV customers at daily grain.
CSCOM_PAYMENT_IM	Contains payment-related information for prepaid and postpaid customers.
CSCOM_PROGRAM_VIWERSHIP_IM	Contains information about program viewership for TV customers. For details, see "Setting Up Derived Columns of Program Viewership Information Map" on page 58.
CSCOM_SERVICE_ACTIVITY_IM	Contains information about various services and the corresponding status for these services that are owned by prepaid and postpaid customers.
CSCOM_SUBSCRP_BILL_NONUSAGE_IM	Contains non-usage bill information at subscription level for postpaid customers.
CSCOM_SUBSCRP_BUCKET_USAGE_IM	Contains the recharge closing balance and usage decrement value for all the prepaid customers. This is a daily snapshot table.
CSCOM_SUBSCRP_COMPLAINT_IM	Contains information about subscription-level complaints that prepaid and postpaid customers have lodged.
CSCOM_SUBSCRP_INQUIRY_IM	Contains information about subscription-level inquiries that prepaid and postpaid customers have made.
CSCOM_SUBSCRP_LOYALTY_IM	Contains subscription level loyalty points that prepaid and postpaid customers have earned.
CSCOM_SUBSCRP_TLCTITM_ORDR_IM	Contains information about telecast item activation for TV customers.

Information Map Name	Description
CSCOM_SUBSCRP_TLCTITM_SBST_IM	Contains information that is required to define subset criterion for MBA at telecast-item level and sequence analysis purposes.
CSCOM_USAGE_IM	Contains usage information of prepaid and postpaid customers.
CSCOM_USAGE_RECHARGE_IM	Contains recharge information of prepaid customers.
CSCOM_USAGE_WEEKLY_SUMMARY_IM	Contains usage summary information of prepaid customers. This information map contains weekly aggregated data.
CSCOM_USAGE_MONTHLY_SUMMARY_IM	Contains usage summary information of postpaid customers. This information map contains monthly aggregated data.

Note: The information maps, CSCOM_DROPPED_USAGE_IM,CSCOM_USAGE_IM, and CSCOM_IPTV_USAGE_IM are provided as prepackaged information maps with SAS Customer Analytics for Communications. However, their configuration is not predefined. In order to use these information maps, you must configure them as data sources through the SAS Customer Analytics for Communications interface. For details, see *SAS Customer Analytics for Communications: User's Guide*.

For details about the columns of the information maps and their data mapping with the Foundation data mart tables, see *SAS Customer Analytics for Communications: Data Reference Guide*.

In SAS Information Map Studio, you can view these information maps from the following location: **Products ⇨ SAS Communications Common Server 5.4 ⇨ Base Tables Mart ⇨ Data Sources ⇨ Infomaps**.

Predefined Dummy Subset Tables

Each of the predefined information maps contains one or both of the following predefined dummy subset tables depending on the levels at which the information map is configured:

- CSCOM_CUST_SUB
- CSCOM_SUBSCRP_SUB

For example, the information map is configured at customer level, then it contains the CSCOM_CUST_SUB table. However, if it is configured at both customer and subscriptions levels, then it contains both the tables.

These tables belong to the CSC_SUB library. In SAS Data Integration Studio, this library is available at the following location: **Products ⇨ SAS Communications Common Server 5.4 ⇨ Base Tables Mart ⇨ Data Sources ⇨ Libraries**. Similarly, these tables are available at the following location: **Products ⇨ SAS Communications Common Server 5.4 ⇨ Base Tables Mart ⇨ Data Sources ⇨ Subset Tables**.

When you define an information map, you must select one of these dummy subset tables as a data source depending on the level, such as customer or subscription, at which you want to populate your ABT variables. However, if you want to populate variables at any other level such as offer or service, then you must define the corresponding dummy subset table. Also, if you want to use an information map for defining subset criterion, then you need not select a dummy subset table.

Using an Information Map for Defining ABT Variables

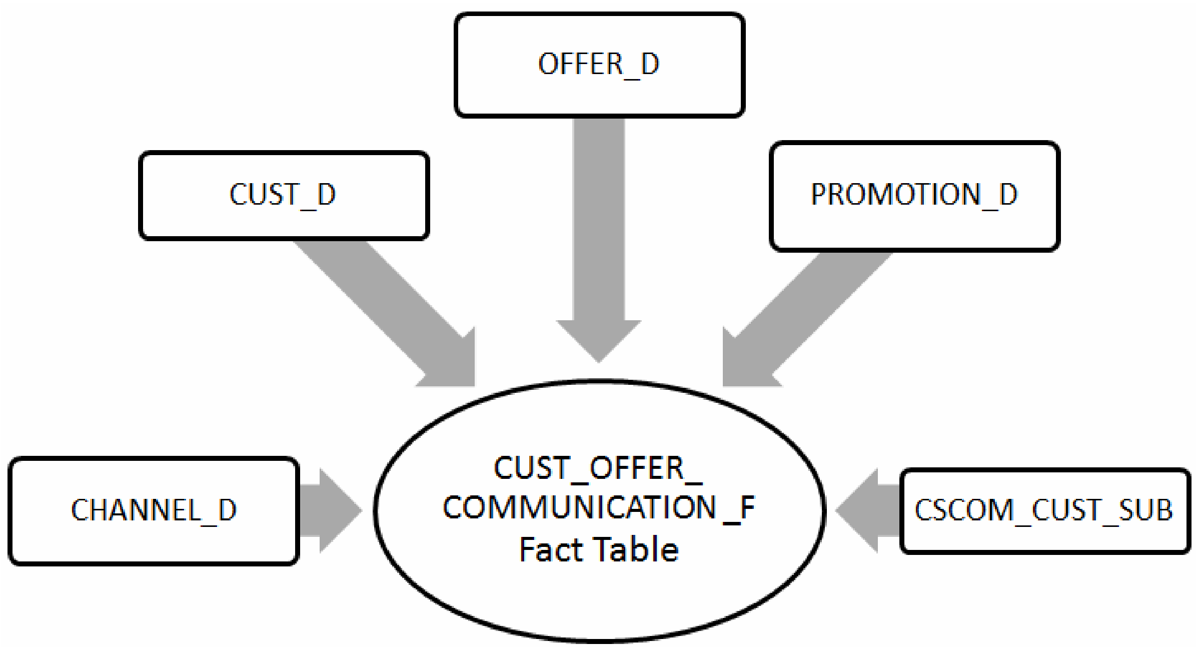
In addition to the predefined information maps, you might want to define new information maps. It is recommended that you refer to the following guidelines when you define a new information map.

Before you define an information map, perform the following prerequisite tasks:

1. Identify the ABT variables that you would want to define by using the information map.
2. Identify the columns that you would want to include in the information map.
3. Identify the source tables (fact and dimension tables of the Foundation data mart) for the information map.
4. Decide the level such as subscription or customer at which you want to populate the ABT variables.
5. Make sure that the dummy subset table is defined for the level at which you want to define your ABT variables.

Let us assume that you want to define ABT variables such as the number of times best offers were recommended to customers through a specific communication channel. In this case, you can define an information map, CSCOM_CUST_BEST_OFFERS_IM based on the CUST_OFFER_COMMUNICATION_F table of the Foundation data mart. Based on the above recommendations, for this information map, you can define the star schema as indicated in the diagram below:

Figure 4.1 *Sample Star Schema for Information Map*



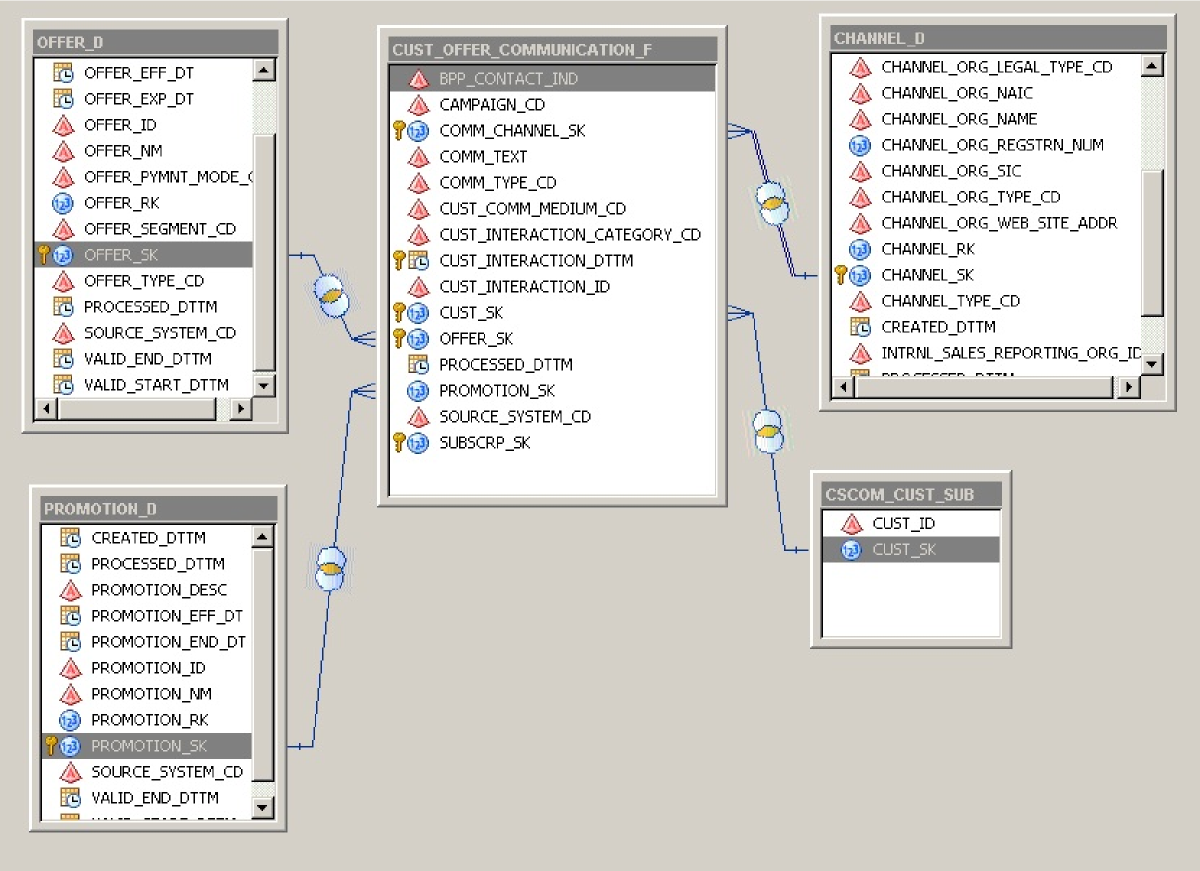
Also, you can decide that you want to populate customer-level data in your ABT variables. Therefore, you must select the CSCOM_CUST_SUB subset dummy table as one of the data sources of the information map.

To define the CSCOM_CUST_BEST_OFFERS_IM information map:

1. Log on to SAS Information Map Studio and connect to the profile of an administrator.
2. Define a new information map, CSCOM_CUST_BEST_OFFERS_IM in the following location: **Products** ⇨ **SAS Communications Common Server 5.4** ⇨ **Base Tables Mart** ⇨ **Data Sources** ⇨ **Infomaps**.
3. Add the following data sources from the corresponding libraries.
   - CSCOM_CUST_SUB
   - CUST_OFFER_COMMUNICATION_F
   - CHANNEL_D
   - OFFER_D
   - PROMOTION_D

   Note: If you want to add certain customer details other than CUST_ID, then you can also include the CUST_D table.
4. Define the relationships between the data sources as indicated in the diagram below.

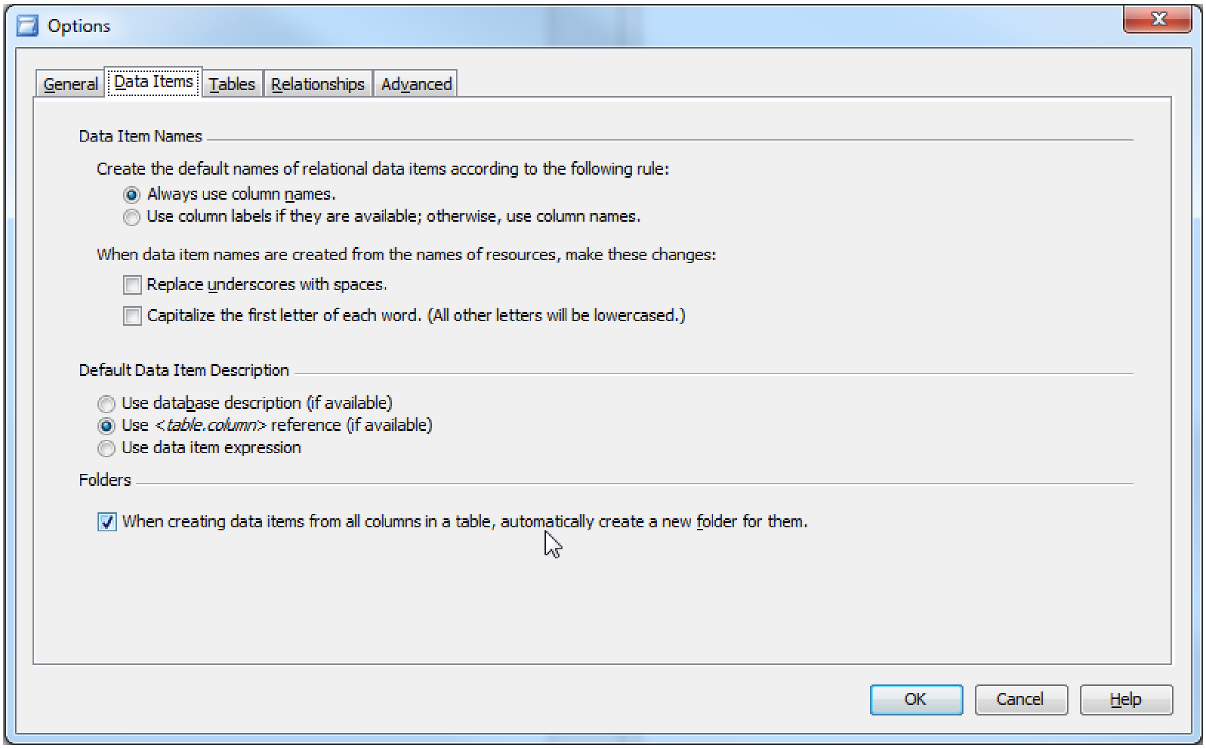
***Display 4.1** Relationships*



Note: Make sure that you define the relationship between the CUST_SK of CSCOM_SUB_CUST and CUST_OFFER_COMMUNICATION_F tables.

5. On the menu, select **Tools** ⇨ **Options**. On the **Data Items** tab, make sure that you set up the following options.

Display 4.2 *Options Window*



Data Item Names

Create the default names of relational data items according to the following rule:

● Always use column names.

○ Use column labels if they are available; otherwise, use column names.

When data item names are created from the names of resources, make these changes:

□ Replace underscores with spaces.

□ Capitalize the first letter of each word. (All other letters will be lowercased.)

Default Data Item Description

○ Use database description (if available)

● Use <*table.column*> reference (if available)

○ Use data item expression

Folders

☑ When creating data items from all columns in a table, automatically create a new folder for them.

6. On the **Design** tab, add the following columns from the respective tables:

   Note: Make sure that you first add the CUST_ID column from the dummy subset table.

   Table 4.5 *Information Map Columns*

Table Name	Column Name
CSCOM_CUST_SUB	CUST_ID
CHANNEL_D	CHANNEL_ID
CHANNEL_D	CHANNEL_TYPE_CD
OFFER_D	OFFER_TYPE_CD
CUST_OFFER_COMMUNICATION_F	COMM_TYPE_CD
CUST_OFFER_COMMUNICATION_F	CUST_INTERACTION_DTTM
CUST_OFFER_COMMUNICATION_F	CUST_INERACTION_ID
PROMOTION_D	PROMOTION_ID

You can also create a new data item by defining an expression. If your Foundation data mart is in Teradata, it is recommended that, in the expression, you use

SAS/ACCESS functions that support implicit pass-through. These functions support in-database processing.

For a list of SAS/ACCESS functions that support implicit pass-through, see "SAS/ACCESS Interface to Teradata" in *SAS/ACCESS 9.3 for Relational Databases: Reference*, which is located at `http://support.sas.com/documentation/cdl/en/acreldb/63144/PDF/default/acreldb.pdf`.

7. Click **Save** to save the information map in the following metadata path: **Products** ⇨ **SAS Communications Common Server 5.4** ⇨ **Base Tables Mart** ⇨ **Data Sources** ⇨ **Infomaps**.

Importing and Using the Information Map

When you define a new information map, you must import its metadata in the Application data and configure its columns. For details, see *SAS Customer Analytics for Communications: User's Guide*. After the configuration, you can use the information map to define ABT variables or subset criterion. When you define an ABT, depending on the subset criterion that you define, the dummy subset table, CSCOM_CUST_SUB is populated with IDs and SKs of customers who satisfy the subset criterion. As a result, when the ABT is built, it will automatically contain information pertaining to these customers only.

For example, based on the information map that you have defined above, you can define a behavioral variable. This variable can compute the number of times best offers were recommended to a customer through a certain communication channel such as phone, personal visit, or e-mail in a specified time period. For more information about how to define ABT variables, see *SAS Customer Analytics for Communications: User's Guide*.

Setting Up Derived Columns of Program Viewership Information Map

The cscom_program_viewership_im information map stores information about program viewership for TV customers. This information map contains three derived columns:

DAY_TYPE
indicates whether the program was watched on a weekday or a weekend.

TIME_OF_DAY
indicates at what time of day a particular program was viewed.

PRIME_NON_PRIME_SLAB
indicates whether a program was viewed during prime time or non-prime time.

The expressions that are defined for these columns are based on the following assumptions:

Variable Name	Definition
DAY_TYPE	Assuming that a week starts on Sunday, if weekday number for a day is 1 or 7, then the value of this variable is set as 'WEEKEND'. Otherwise, the value is set as 'WEEKDAY'.

Variable Name	Definition
TIME_OF_DAY	• If the hour when the program was watched is greater than or equal to 3 and less than 7, then the variable value is ‘EARLY MORNING’. • If the hour when the program was watched is greater than or equal to 7 and less than 12, then the variable value is ‘MORNING’. • If the hour when the program was watched is greater than or equal to 12 and less than 16, then the variable value is ‘AFTERNOON’. • If the hour when the program was watched is greater than or equal to 16 and less than 19, then the variable value is ‘EVENING’. • If the hour when the program was watched is greater than or equal to 19 and less than 22, then the variable value is ‘NIGHT’. • If none of the above conditions are satisfied, the variable value is ‘LATE NIGHT’.
PRIME_NON_PRIME_SLAB	If the hour when program was watched is greater than or equal to 7 and less than 9 or if it is greater than or equal to 19 and less than 21, then the variable value is ‘PRIME’. Otherwise, the variable value is ‘NON-PRIME’.

You can change the definitions of these columns according to you your requirement, and then update the configuration information of the information map that is stored in the Application data.

To change the expression of a column:

1. Log on to SAS Information Map Studio and connect to an appropriate profile.
2. Expand **Products** ⇨ **SAS Communications Common Server 5.4** ⇨ **Base Tables Mart** ⇨ **Data Sources** ⇨ **Infomaps**.
3. Open the cscom_program_viewership_im information map.
4. In the Information Map Contents window, right-click the column that you want to edit, and then select **Properties**. The Data Item Properties window appears.
5. Click **Edit** and modify the expression according to your requirements.
6. Click **Save** to save the changes that you have made to the expression.

   Note: In order to use the new expression of the derived column, you must update the configuration information of the information map. For details, see *SAS Customer Analytics for Communications: User’s Guide*.

Configuring Subjects of Analysis

SAS Customer Analytics for Communications provides you a set of predefined subjects of analysis. Typically, a subject of analysis can be mapped to a single column of a table in the Foundation data mart. For example, in a simple scenario, for the **Segmentation Postpaid** purpose, the subject of analysis can be **Customer**. In this case, the subject of analysis can be directly mapped to the CUST_ID column of the CUST_D table of the Foundation data mart. However, in complex scenarios, for certain purposes such as **Association Rules VOD and PPV**, the subject of analysis can be **Subscription Telecast Item Order Date**. This subject of analysis cannot be mapped to a single column of a table. Therefore, you have to manually configure this type of subjects of analysis in the Application data tables.

To add a new subject of analysis, you have to add record in the LEVEL_MASTER and the LEVEL_KEY_COLUMN_DTL tables. Make sure that you enter a unique value for the LEVEL_SK and KEY_COLUMN_SK columns of the respective tables. Also, make sure that the value that you enter for each column is greater than 1000000.

The following table briefly explains how you can define subjects of analysis for simple and complex scenarios.

Note: For details about the Application data tables, see *SAS Customer Analytics for Communications: Administrator's Guide*.

Table 4.6 *Configuring Subjects of Analysis*

Table Name of Application Data	Details to be Configured in the Table	Main Columns to be Configured	Sample Configuration Values for Simple Scenario	Sample Configuration Values for Complex Scenario
PURPOSE_ MASTER	Configure the details about purpose of your analysis.	PROJECT_SHORT_ NM	Segmentation Postpaid	Association rules VOD and PPV
LEVEL_MASTER	Configure details about the subject of analysis. Also, specify the details of the master table to which the subject of analysis can be mapped. If the subject of analysis cannot be mapped to a column of a single table, then specify the value of the MASTER_ SOURCE_TABLE_ SK as –1.	LEVEL_DESC	Customer	Subscription Telecast Item Order Date
		MASTER_ SOURCE_TABLE_ SK	8 (SK of CUST_D table)	–1

Table Name of Application Data	Details to be Configured in the Table	Main Columns to be Configured	Sample Configuration Values for Simple Scenario	Sample Configuration Values for Complex Scenario
PURPOSE_X_LEVEL	Configure the details about eligible combinations of a purpose and a subject of analysis.	LEVEL_SK	1 (SK of Customer level in LEVEL_MASTER)	14 (SK of Subscription Telecast Item Order Date level in LEVEL_MASTER)
		PURPOSE_SK	3 (SK of Segmentation Postpaid purpose in PURPOSE_MASTER	20 (SK of Association rules VOD and PPV in PURPOSE_MASTER)
SOURCE_TABLE_MASTER	Confirm that the details about the source table to which the subject of analysis belongs is configured. If the subject of analysis is mapped to more than one columns, then –1 indicates that there is no single master table for that subject of analysis. This table will be considered as the master table.	SOURCE_TABLE_SK	8 (SK of CUST_D table)	–1
SOURCE_COLUMN_MASTER	Confirm the details about one or more columns to which the subject of analysis is mapped.	SOURCE_COLUMN_SK	CUST_ID, CUST_SK	• SUBSCRP_ID • TELECAST_ITEM_ID • TLCST_ITM_ORDER_DT In this case, all these columns can belong to an information map that is defined based on TELECAST_ITEM_D, CALENDAR_D, and SUBSCRP_D tables.

Table Name of Application Data	Details to be Configured in the Table	Main Columns to be Configured	Sample Configuration Values for Simple Scenario	Sample Configuration Values for Complex Scenario
LEVEL_KEY_ COLUMN_DTL	Specify the details of the column to which the subject of analysis is mapped.	KEY_COLUMN_SK for which KEY_TYPE_CD= RETAIN_KEY	SK of CUST_ID column from SOURCE_ COLUMN_ MASTER where SOURCE_TABLE_ SK corresponds to SK of CUST_D Table	Insert one row each for the columns that forms the level. • SK of SUBSCRP_ID column from SOURCE_ COLUMN_ MASTER table where SOURCE_ TABLE_SK corresponds to the SK of table that stores association between subscription, telecast item, and telecast item order date. • SK of TELECAST_ ITEM_ID column from SOURCE_ COLUMN_ MASTER table where SOURCE_ TABLE_SK corresponds to the SK of table that stores association between subscription, telecast item, and telecast item order date. • SK of TLCST_ITM_ ORDER_DT column from SOURCE_ COLUMN_ MASTER table where SOURCE_ TABLE_SK corresponds to the SK of table that stores association between subscription, telecast item, and telecast item order date.

Table Name of Application Data	Details to be Configured in the Table	Main Columns to be Configured	Sample Configuration Values for Simple Scenario	Sample Configuration Values for Complex Scenario
		KEY_COLUMN_SK for which KEY_TYPE_CD= SURROGATE_KEY	SK of CUST_SK column from SOURCE_ COLUMN_ MASTER where SOURCE_TABLE_ SK corresponds to SK of CUST_D Table	As there is no master table, there is no need to configure the KEY_COLUMN_SK column for which KEY_TYPE_CD= SURROGATE_KEY.

Creating a New Subset Map

Overview

A subset map is a group of tables that contain interrelated data pertaining to a particular subject of analysis. For example, the probable subjects of analysis for a communications service provider can be customers and subscriptions. A subset map is designed in a specific way so that users can apply filters on any column of subset map tables. You apply filters on a subset map through a subset criterion. For more information about the subset criterion, see the *SAS Customer Analytics for Communications: User's Guide*.

SAS Customer Analytics for Communications provides a number of predefined subset maps. For details, see "Predefined Subset Maps" on page 127. You can also create new subset maps according to your business requirements. Internally, the filter conditions in a subset criterion might be on columns of one or more tables in a subset map. In such a case, the final SQL statement (filter condition) for the subset criterion needs a relationship among all the involved subset map tables. The relationship is in the form of join conditions among the tables.

To create a new subset map, complete the following tasks:

1. Configure all the tables that you plan to use in the subset map definition for subset criterion usage.

   In other words, for each table that you plan to use in the subset map definition, select **Subset Criterion** on the **Direct/Subset Criterion** tab (in the Configure Data Source Usage window). For details, see the *SAS Customer Analytics for Communications: User's Guide*.

2. Define the subset map. For details, see "Define the Subset Map" on page 64.

3. Define the relationships among the tables in the subset map. For details, see "Define Relationships among Subset Map Tables" on page 64.

4. Configure the availability of the subset map for various subjects of analysis. For details, see "Configure the Availability of the Subset Map" on page 65.

Define the Subset Map

To define the subset map, you need to add a record in the SUBSET_FROM_PATH_MASTER table (available in the CSCAPDM library). The following table shows the SUBSET_FROM_PATH_MASTER table with sample records:

Note: Make sure that you enter a unique value for the SUBSET_FROM_PATH_SK column. Also, make sure that this value is greater than 1000000.

Table 4.7 *SUBSET_FROM_PATH_MASTER Table*

SUBSET_FROM_PATH_SK	FROM_PATH_SHORT_NM	FROM_PATH_DESC
7	Subscriptions or Customers level Selection	Subscriptions or Customers level Selection
10	Customers with Active Status	Active Customers
11	Subscriptions with Active Status	Active Subscriptions

For descriptions of the columns in the SUBSET_FROM_PATH_MASTER table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Define Relationships among Subset Map Tables

Add one or more records in the SUBSET_TABLE_JOIN_CONDITION table to define relationships among the tables in the subset map. These relationships among the tables help in constructing the FROM clause of the final SQL statement.

Note: Make sure that you enter a unique value for the SUBSET_TABLE_JOIN_CONDITION_SK column. Also, make sure that this value is greater than 1000000.

The following table shows the SUBSET_TABLE_JOIN_CONDITION table with sample records:

Table 4.8 *SUBSET_TABLE_JOIN_CONDITION Table*

SUBSET_ TABLE_ JOIN_ CONDITION_ SK	SUBSET_ FROM_ PATH_SK	JOIN_ CONDITION_ SEQUENCE_ NUMBER	JOIN_ TYPE	LEFT_ TABLE_SK	LEFT_ COLUMN_ SK	RIGHT_ TABLE_SK	RIGHT_ COLUMN_ SK
42	7	1	INNER	9	103	15	666
43	7	2	INNER	15	665	60	673
10	10	1	NONE	8	99	.	.

For descriptions of the columns in the SUBSET_TABLE_JOIN_CONDITION table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Configure the Availability of the Subset Map

A subset map can be linked to one or more subjects of analysis. This linking specifies that the tables in a subset map can be used to define subset criteria in projects with the corresponding subjects of analysis. You need to define these links between the subset map and various subjects of analysis in the SUBSET_FROM_PATH_X_LEVEL table.

The following table shows the SUBSET_FROM_PATH_X_LEVEL table with sample records:

Table 4.9 SUBSET_FROM_PATH_X_LEVEL Table

SUBSET_FROM_ PATH_SK	LEVEL_SK	SELECT_ SOURCE_ COLUMN_SK	CORRSP_LEVEL_ KEY_COLUMN_ DTL_SK
7	4	102	.
7	1	103	.
10	1	99	.

For descriptions of the columns in the SUBSET_FROM_PATH_X_LEVEL table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Example: Creating a Subset Map

Overview

Configure a new subset map that enables the user to select the customers based on the following attributes:

- customer type
- customer tenure
- payment mode

In order to create this subset map, you need to use the following tables of the Foundation data mart: CUST_D, CUST_AGRMNT_D, and OFFER_BUNDLE_D. As a result, users can define the subset criterion based on the columns of these tables.

The subsequent topics explain how to configure this subset map.

Configure the SUBSET_FROM_PATH_MASTER Table

Assuming that 31 subset maps with SK in sequence are defined, add the following joining condition of the query in the SUBSET_FROM_PATH_MASTER table.

Table 4.10 Sample Record in SUBSET_FROM_PATH_MASTER Table

SUBSET_FROM_PATH_SK	FROM_PATH_SHORT_NM	FROM_PATH_DES
32	Customers with specific payment mode	Customers with either prepaid or postpaid payment mode

Configure the SUBSET_TABLE_JOIN_CONDITION Table

After you configure the values in the SUBSET_FROM_PATH_MASTER table, make sure that you enter correct values for joining condition in the SUBSET_TABLE_JOIN_CONDITION table. When a user defines a subset criterion based on a subset map, it is converted into a query. This query is configured in the SUBSET_TABLE_JOIN_CONDITION table.

To configure the SUBSET_TABLE_JOIN_CONDITION table, complete these steps:

1. Open the SOURCE_TABLE_MASTER table (available in the CSCAPDM library). Identify the SK of each of the CUST_D, CUST_AGRMNT_D, and OFFER_BUNDLE_D tables. For example, the SK for each table can be as mentioned in the following table.

Table 4.11 *Sample Records in SOURCE_TABLE_MASTER Table*

SOURCE_TABLE_NM	SOURCE_TABLE_SK
CUST_D	8
CUST_AGRMNT_D	15
OFFER_BUNDLE_D	60

2. Identify the joining keys in these tables.
   - CUST_D and CUST_AGRMNT_D tables are joined based on the CUST_ID column.
   - CUST_AGRMNT_D and OFFER_BUNDLE_D tables are joined based on the OFFER_ID and BASE_OFFER_ID columns in the respective tables.
3. Open the SOURCE_COLUMN_MASTER table (available in the CSCAPDM library). Identify the SK of the joining keys from the respective tables. For example, these SK values can be as shown in the table below.

Table 4.12 *Sample Records in the SOURCE_COLUMN_MASTER Table*

SOURCE_TABLE_NM	SOURCE_TABLE_SK	SOURCE_COLUMN_NM	SOURCE_COLUMN_SK
CUST_D	8	CUST_ID	99
CUST_AGRMNT_D	15	CUST_ID	666
CUST_AGRMNT_D	15	OFFER_ID	665
OFFER_BUNDLE_D	60	BASE_OFFER_ID	673

4. Open the SUBSET_TABLE_JOIN_CONDITION table (available in the CSCAPDM library). Assuming that the maximum SK value for the SUBSET_TABLE_JOIN_COONDITION_SK column is 43, add the following two records in the SUBSET_TABLE_JOIN_CONDITION table.

Table 4.13 *Sample Records in the SUBSET_TABLE_JOIN_CONDITION Table*

SUBSET_TABLE_JOIN_COONDITION SK	SUBSET_FROM_PATH_SK	JOIN_CONDITION_SEQUENCE_NUMBER	JOIN_TYPE	LEFT_TABLE_SK	LEFT_COLUMN_SK	RIGHT_TABLE_SK	RIGHT_COLUMN_SK
44	32	1	INNER	8	99	15	666
45	32	2	INNER	15	665	60	673

Configure the SUBSET_FROM_PATH_X_LEVEL Table

After you configure the subset map, you have to make it available in the SAS Customer Analytics for Communications interface. A subset map can be linked to one or more subject of analysis. The subset map that is created in this example can be made available only at customer level. Therefore, you have to identify the SK of the CUST_ID column.

To identify the SK of the customer level, complete these steps:

1. Open the LEVEL_MASTER table (available in the CSCAPDM library). Identify the LEVEL_SK, which corresponds to LEVEL_KEY_COLUMN_NM as CUST_ID. For example, this SK value can be 1.
2. Insert following record in the SUBSET_FROM_PATH_X_LEVEL table (available in the CSCAPDM library).

Table 4.14 *Sample Record in SUBSET_FROM_PATH_X_LEVEL Table*

SUBSET_FROM_PATH_SK	LEVEL_SK	SELECT_SOURCE_COLUMN_SK	CORRSP_LEVEL_KEY_COLUMN_DTL_SK
32	1	99	.

Subset Query

Let us assume that a user defines a subset criterion based on the subset map that you have created, to filter customer who have postpaid payment mode. In this case, the following query is created. The SELECT_SOURCE_COLUMN_SK column is selected as a part of the final query.

```
proc sql noprint;

create table tmp_lib.d_135 as select distinct CUST_D.CUST_ID


From DIM.CUST_D

(where =(&DABT_LOAD_USER_INPUT_DTTM between VALID_START_DTTM and VALID_END_DTTM))

INNER join

DIM.CUST_AGRMNT_D

(where =(&DABT_LOAD_USER_INPUT_DTTM between VALID_START_DTTM and VALID_END_DTTM))

on (CUST_D.CUST_ID = CUST_AGRMNT_D.CUST_ID)INNER join

DIM.OFFER_BUNDLE_D

(where =(&DABT_LOAD_USER_INPUT_DTTM between VALID_START_DTTM and VALID_END_DTTM))
```

```
on (CUST_AGRMNT_D.OFFER_ID = OFFER_BUNDLE_D.BASE_OFFER_ID)

where (OFFER_BUNDLE_D.BASE_OFFER_PYMNT_MODE_CD

IN %quote((&PST_PYMNT_MODE_VALS)));

quit;
```

This query can be split into three parts:

1. The first part (the selection part) is the configuration made in the SELECT_SOURCE_COLUMN_SK column of the SUBSET_FROM_PATH_X_LEVEL table.
2. The second part (the FROM condition) is the configuration made in the SUBSET_TABLE_JOIN_CONDITION table.
3. The last part indicates the filters that you define for the subset criterion that you defined for the subset map.

Configure the Subset Map at Subscription Level

If you want to define the same subset map at subscription level, one of the methods is to join the SUBSCRP_D table instead of CUST_D with the CUST_AGRMNT_D table. The joining column will be the same, that is, CUST_ID. The rest of the configuration will be the same for the SUBSET_TABLE_JOIN_CONDITION table. However, the LEFT_TABLE_SK and LEFT_COLUMN_SK columns will correspond to the SK (for example, 9) of the SUBSCRP_D table and the SK (for example, 103) of CUST_ID column of the SUBSCRP_D table. In this case, the records from the SUBSET_TABLE_JOIN_CONDITION table would look as shown below:

Table 4.15 Sample Records in SUBSET_TABLE_JOIN_CONDITION Table

SUBSET_ TABLE_ JOIN_ CONDITION_ SK	SUBSET_ FROM_ PATH_SK	JOIN_ CONDITION_ SEQUENCE_ NUMBER	JOIN_ TYPE	LEFT_ TABLE_SK	LEFT_ COLUMN_ SK	RIGHT_ TABLE_SK	RIGHT_ COLUMN_ SK
44	32	1	INNER	9	103	15	666
45	32	2	INNER	15	665	60	673

Edit the record in the SUBSET_FROM_PATH_X_LEVEL table. The value for the SELECT_SOURCE_COLUMN_SK column should represent the SK (for example, 103) of the CUST_ID column from the SUBSCRP_D table. Add one more row in the SUBSET_FROM_PATH_X_LEVEL table corresponding to subscription as a level. If 4 is the value of LEVEL_SK for SUBSCRP_ID level from the LEVEL_MASTER table and 102 is the SK of SUBSCRP_ID column from the SUBSCRP_D table, then the records in the SUBSET_FROM_PATH_X_LEVEL table would look as shown below:

Table 4.16 Sample Records in SUBSET_FROM_PATH_X_LEVEL Table

SUBSET_FROM_PATH_SK	LEVEL_SK	SELECT_SOURCE_COLUMN_SK
32	1	103
32	4	102

If you use the above subset map for the projects with Customer as the subject of analysis, then the record corresponding to LEVEL_SK equal to 1 will be selected in the final query. However, if you use the same subset map for the projects with Subscription as the subject of analysis, then the record corresponding to LEVEL_SK equal to 4 will be selected in the final query.

Note: You can join the input tables of a subset map through different joining conditions. You can also use LEFT and RIGHT joins depending on the type of data that you expect in the final ABT. If you want to use a single table in a subset map, the joining condition should be marked as NONE with entries only under LEFT_TABLE_SK and LEFT_COLUMN_SK columns. The RIGHT_TABLE_SK and RIGHT_COLUMN_SK columns should have Null values.

Creating a New Subset Map with an Information Map

Overview

SAS Customer Analytics for Communications enables you to define a subset map by using an information map as a data source. When you use an information map for defining a subset map, the final SQL statement (filter condition) of the subset criterion that is defined on the subset map contains the joins between the tables that constitute the information map.

To configure a new subset map by using an information map, complete the following tasks:

1. Identify the tables that you want to use in the information map. Also, identify the relationships between the tables. This information map will be used to define the subset map.

2. Define an information map by using the tables that you identified in step 1. For details, see "Using an Information Map for Defining ABT Variables" on page 55.

   In addition, consider the following recommendations when you define an information map that you will use for creating the subset map.

   - An information map can be used for defining ABT variables and a subset map. If you intend to use same information map for both the purposes, then you must use a dummy subset table. This dummy table should be selected depending on the subject of analysis at which you want to configure the information map.

   - If more than one table is used in the information map and if any of these tables contain valid from and valid to columns, then configure these columns (from each table) by using the Configure Data Source window of the SAS Customer Analytics for Communications interface. For details, see *SAS Customer Analytics for Communications: User's Guide*.

   - A subset map be defined based on a single information map (also called the source information map). This source information map cannot be nested into another information map. In addition, you cannot use another source table along with the source information map for defining the subset map. All the tables that are required for the subset specification should be a part of the source information map.

3. Configure the information map that you defined in step 2 so that it can be used for defining a subset criterion. For details, see *SAS Customer Analytics for Communications: User's Guide*.
4. Define the subset map. For details, see "Define the Subset Map" on page 76.
5. Configure the subset map. For details, see "Define Relationships among Subset Map Tables" on page 70.
6. Configure the availability of the subset map for various subjects of analysis. For details, see "Configure the Availability of the Subset Map" on page 71.

Define the Subset Map

To define the subset map, you need to add a record in the SUBSET_FROM_PATH_MASTER table (available in the CSCAPDM library).

Note: Make sure that you enter a unique value for the SUBSET_FROM_PATH_SK column. Also, make sure that this value is greater than 1000000.

The following table shows the SUBSET_FROM_PATH_MASTER table with sample records:

Table 4.17 *SUBSET_FROM_PATH_MASTER Table*

SUBSET_FROM_PATH_SK	FROM_PATH_SHORT_NM	FROM_PATH_DESC
37	Subset Map for Implicit Subset Queries	Subset Map for Implicit Subset Queries
38	TV Customers/ subscriptions and services/ offers TV Customers/ subscriptions and services/ offers	TV Customers/ subscriptions and services/ offers TV Customers/ subscriptions and services/ offers
39	TV Customers/ subscriptions and telecast items TV Customers/ subscriptions and telecast items	TV Customers/ subscriptions and telecast items TV Customers/ subscriptions and telecast items

Define Relationships among Subset Map Tables

Add one or more records in the SUBSET_TABLE_JOIN_CONDITION table to define relationships among the tables in the subset map. These relationships among the tables help in constructing the FROM clause of the final SQL statement.

Note: Make sure that you enter a unique value for the SUBSET_TABLE_JOIN_CONDITION_SK column. Also, make sure that this value is greater than 1000000.

Note: For information maps, you do not need to configure the joining condition between the source tables for the subset maps. The information map will be considered as a single data source to be configured. Therefore, for an information map, you need add a single record in the SUBSET_TABLE_JOIN_CONDITION table.

The following table shows the SUBSET_TABLE_JOIN_CONDITION table with a sample record when an information map is configured as a data source:

Table 4.18 SUBSET_TABLE_JOIN_CONDITION Table

SUBSET_ TABLE_ JOIN_ CONDITION_ SK	SUBSET_ FROM_ PATH_SK	JOIN_ CONDITION_ SEQUENCE_ NUMBER	JOIN_ TYPE	LEFT_ TABLE_SK	LEFT_ COLUMN_ SK	RIGHT_ TABLE_SK	RIGHT_ COLUMN_ SK
57	37	1	NONE	112	.	.	.

Configure the Availability of the Subset Map

A subset map can be linked to one or more subjects of analysis. This linking specifies that the tables in a subset map can be used to define subset criteria in projects with the corresponding subjects of analysis. You need to define these links between the subset map and various subjects of analysis in the SUBSET_FROM_PATH_X_LEVEL table.

The following table shows the SUBSET_FROM_PATH_X_LEVEL table with sample records:

Table 4.19 SUBSET_FROM_PATH_X_LEVEL Table

SUBSET_FROM_PATH_ SK	LEVEL_SK	SELECT_SOURCE_ COLUMN_SK	CORRSP_LEVEL_KEY_ COLUMN_DTL_SK
37	6	923	.
37	6	935	.
37	9	920	.
37	9	923	.
37	12	923	.
37	12	926	.
37	12	935	.
37	13	920	.
37	13	923	.
37	13	926	.

When you use an information map for defining ABT variables and a subset map, the column that identifies the subject of analysis (also called level key column) should be selected from the subset table. However, the column that is to be configured in the SUBSET_FROM_PATH_X_LEVEL.SELECT_SOURCE_COLUMN_SK should be considered from a non-subset table only. As the subset tables are used only during ABT building, they will not be used while applying filters through the subset criterion. Therefore, when you define the information map, you have to select the level key column from subset table and a non-subset table.

When you select the level key column from more than two data sources, it will result in the name of corresponding data item from information map to have different name. For example, CUST_ID is the level key column that is added in the information map twice. In this case, the first data item will have name as CUST_ID and second data item will have name as CUST_ID2. As a result, the name of column that is specified in the SUBSET_FROM_PATH_X_LEVEL.SELECT_SOURCE_COLUMN_SK column will not match with the level key column name in the master table. To avoid this, you will need to do an additional configuration. That is, you have to set the value of SUBSET_FROM_PATH_X_LEVEL.CORRSP_LEVEL_KEY_COLUMN_DTL_SK.

For example, in the above scenario, identify the value of LEVEL_KEY_COLUMN_DTL_SK from LEVEL_KEY_COLUMN_DTL table where LEVEL_SK = 1 (Customer level) and KEY_TYPE_CODE is RETAIN_KEY. Set this value under CORRSP_LEVEL_KEY_COLUMN_DTL_SK for your subset map, such that LEVEL_SK=1 and SELECT_SOURCE_COLUMN_SK corresponds to SK of CUST_ID2 from SOURCE_COLUMN_MASTER table of CSCAPDM library.

Example: Creating a Subset Map Using an Information Map

Overview of Sample Scenario

Create and configure a subset map that enables you to select customers based on the following attributes:

- Customer type
- Customer status
- Customer tenure

In addition, this information map can also be used for defining ABT variables as mentioned below:

- Total call duration in the past three months
- Total call count in Base 1 month
- Total Local SMS count in Base 2 month

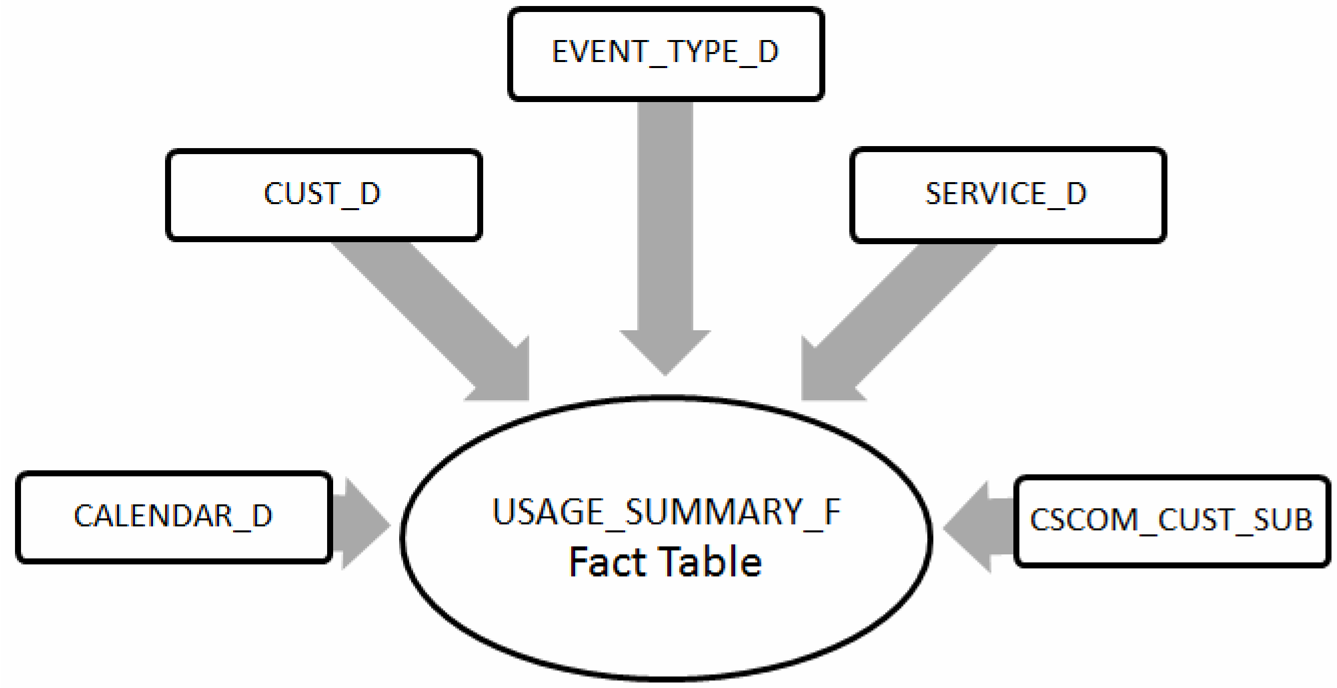
Identify the Source Tables of the Information Map

In order to create an information map for defining subset criteria and ABT variables as mentioned above, use following tables of the Foundation data mart:

- CUST_D
- USAGE_SUMMARY_F
- SERVICE_D
- CALENDAR_D

In order to build an ABT at customer level using this information map, you will also need to use the CSCOM_CUST_SUB dummy subset table.

Figure 4.2 *Source Tables of Information Map*

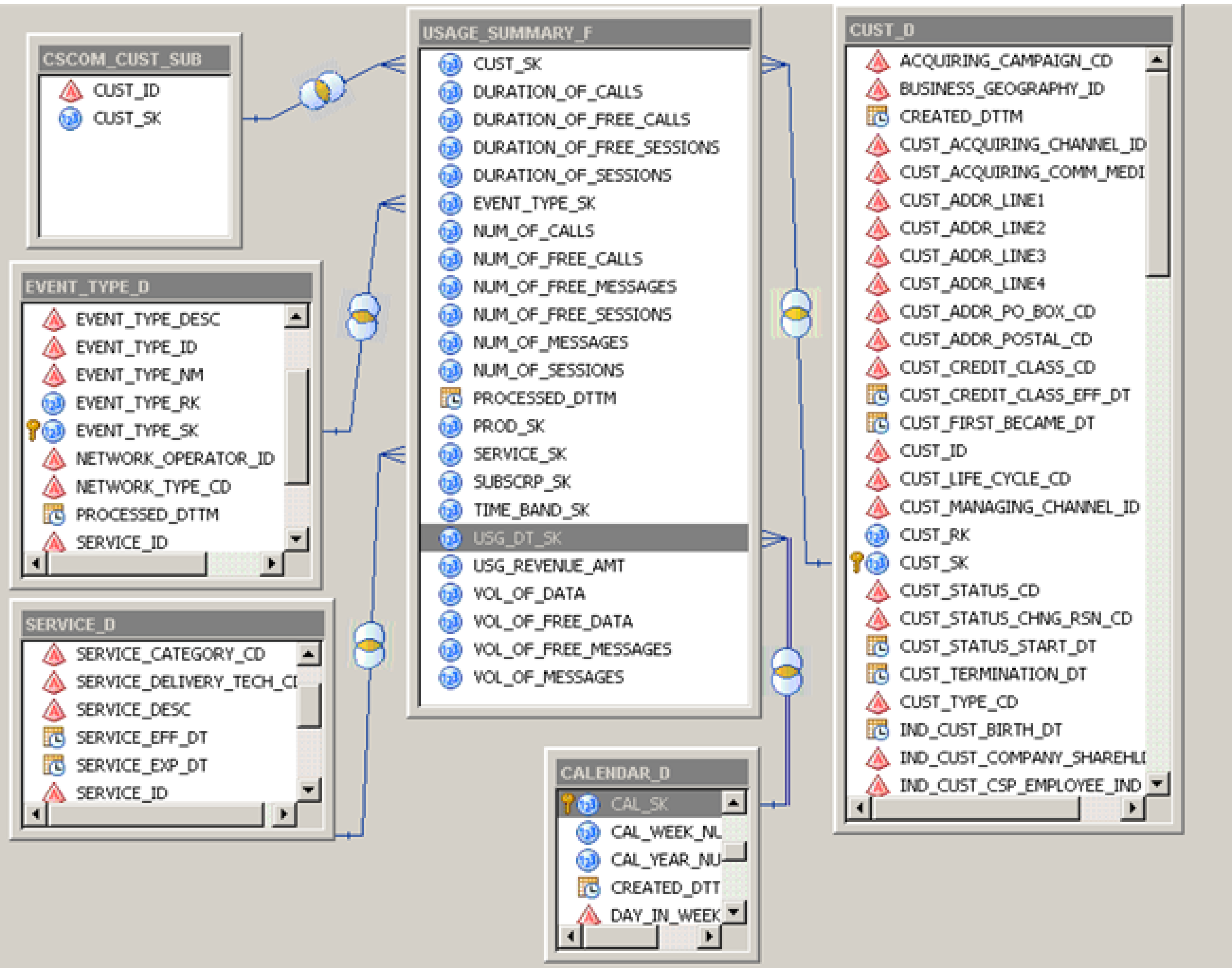


Define the Information Map

To define the information map:

1. Log on to SAS Information Map Studio and connect to the profile of an administrator.
2. Define a new information map, CSCOM_CUST_SUBSET_IM in the following location:**Products** ⇨ **SAS Communications Common Server 5.4** ⇨ **Base Tables Mart** ⇨ **Data Sources** ⇨ **Infomaps**.
3. Add the following data sources from the corresponding libraries:
   - CUST_D
   - USAGE_SUMMARY_F
   - EVENT_TYPE_D
   - SERVICE_D
   - CALENDAR_D
   - CSCOM_CUST_SUB
4. Define the relationships between the data sources.

Figure 4.3 Relationships



5. On the **Design** tab, add the following columns from the respective tables:

   Note: Make sure that you first add the CUST_ID column from the dummy subset table.

Table 4.20 Information Map Columns

Table Name	Column Name	Corresponding Data Item Name	Expression
CSCOM_CUST_SUB	CUST_ID	CUST_ID	
CUST_D	CUST_ID	CUST_ID2	
CUST_D	CUST_TYPE_CD	CUST_TYPE_CD	
CUST_D	CUST_STATUS_CD	CUST_STATUS_CD	
CUST_D	CUST_FIRST_BECAME_DT	CUST_FIRST_BECAME_DT	
CUST_D	VALID_START_DTTM	VALID_START_DTTM	

Table Name	Column Name	Corresponding Data Item Name	Expression
CUST_D	VALID_END_DTTM	VALID_END_DTTM	
USAGE_SUMMARY_F	NUM_OF_CALLS	NUM_OF_CALLS	CASE WHEN <<USAGE_SUMMARY_F.NUM_OF_CALLS>> IS NULL THEN 0 ELSE <<USAGE_SUMMARY_F.NUM_OF_CALLS>> END
USAGE_SUMMARY_F	NUM_OF_MESSAGES	NUM_OF_MESSAGES	CASE WHEN <<USAGE_SUMMARY_F.NUM_OF_MESSAGES>> IS NULL THEN 0 ELSE <<USAGE_SUMMARY_F.NUM_OF_MESSAGES>> END
USAGE_SUMMARY_F	DURATION_OF_CALLS	DURATION_OF_CALLS	CASE WHEN <<USAGE_SUMMARY_F.DURATION_OF_CALLS>> IS NULL THEN 0 ELSE <<USAGE_SUMMARY_F.DURATION_OF_CALLS>> END
EVENT_TYPE_D	EVENT_TYPE_ID	EVENT_TYPE_ID	
EVENT_TYPE_D	VALID_START_DTTM	VALID_START_DTTM2	
EVENT_TYPE_D	VALID_END_DTTM	VALID_END_DTTM2	
SERVICE_D	SERVICE_ID	SERVICE_ID	
SERVICE_D	VALID_START_DTTM	VALID_START_DTTM3	
SERVICE_D	VALID_END_DTTM	VALID_END_DTTM3	
CALENDAR_D	CAL_DT	USAGE_DT	

6. Click **Save**.

Configure the Information Map

Configure the information map by using the **Direct or Subset Criterion** tab that is available in the Configure Data Source Usage window of the SAS Customer Analytics for Communications interface, so that it can be used for defining subset criteria and ABT variables. On the **Direct or Subset Criterion** tab, provide the valid start and end date pairs. On the **Behavioral** tab, add USAGE_DT as **Date of Extract**. For details, see *SAS Customer Analytics for Communications: User's Guide*.

Define the Subset Map

Assume that 40 subset maps with SK in sequence are defined. Add the following joining condition of the query in the SUBSET_FROM_PATH_MASTER table.

Table 4.21 Joining Condition

SUBSET_FROM_PATH_SK	FROM_PATH_SHORT_NM	FROM_PATH_DESC
41	Customer-level selection and variable creation	Customer-level selection and variable creation

Configure the SUBSET_TABLE_JOIN_CONDITION Table

After you configure the values in the SUBSET_FROM_PATH_MASTER table, make sure that you enter correct values for joining condition in the SUBSET_TABLE_JOIN_CONDITION table. When a user defines a subset criterion based on a subset map, it is converted into a query. This query is configured in the SUBSET_TABLE_JOIN_CONDITION table.

To configure the SUBSET_TABLE_JOIN_CONDITION table, complete these steps:

1. Open the SOURCE_TABLE_MASTER table (available in the CSCAPDM library). Identify the SK of the CSCOM_CUST_SUBSET_IM table. For example, the SK for the table can be as mentioned in the following table.

Table 4.22 SOURCE_TABLE_MASTER Table

SOURCE_TABLE_NM	SOURCE_TABLE_SK
CSCOM_CUST_SUBSET_IM	123

2. Open the SOURCE_COLUMN_MASTER table (available in the CSCAPDM library). Identify the SK of the CUST_ID2 column from table where SOURCE_TABLE_SK=123.

Table 4.23 SOURCE_COLUMN_MASTER Table

SOURCE_TABLE_NM	SOURCE_TABLE_SK	SOURCE_COLUMN_NM	SOURCE_COLUMN_SK
CSCOM_CUST_SUBSET_IM	123	CUST_ID2	1178

3. Open the SUBSET_TABLE_JOIN_CONDITION table (available in the CSCAPDM library). Assuming that the maximum SK value for the SUBSET_TABLE_JOIN_COONDITION_SK column is 66, add the following record in the SUBSET_TABLE_JOIN_CONDITION table.

Table 4.24 SUBSET_TABLE_JOIN_CONDITION Table

SUBSET_TABLE_JOIN_CONDITION_SK	SUBSET_FROM_PATH_SK	JOIN_CONDITION_SEQUENCE_NUMBER	JOIN_TYPE	LEFT_TABLE_SK	LEFT_COLUMN_SK	RIGHT_TABLE_SK	RIGHT_COLUMN_SK
67	41	1	NONE	1178			

Configure the SUBSET_FROM_PATH_X_LEVEL Table

After you configure the subset map, you have to make it available in the SAS Customer Analytics for Communications interface. A subset map can be linked to one or more subjects of analysis. The subset map that is created in this example can be made available only at customer level. Therefore, you have to identify the SK of the CUST_ID column.

To identify the SK of the customer level, complete these steps:

1. Open the LEVEL_MASTER table (available in the CSCAPDM library). Identify the LEVEL_SK, which corresponds to LEVEL_KEY_COLUMN_NM as CUST_ID. For example, this SK value can be 1.
2. The name corresponding to SELECT_SOURCE_COLUMN_SK (CUST_ID2) of the SUBSET_FROM_PATH_X_LEVEL table is not same as the LEVEL_KEY_COLUMN_NM (CUST_ID) of the LEVEL_MASTER table. Therefore, open the LEVEL_KEY_COLUMN_DTL table. Identify the value of LEVEL_KEY_COLUMN_DTL_SK where LEVEL_SK=1 and KEY_TYPE_CD=RETAIN_KEY. This value needs to be added for the CORRSP_LEVEL_KEY_COLUMN_DTL_SK column of the SUBSET_FROM_PATH_X_LEVEL table.

Table 4.25 LEVEL_KEY_COLUMN_DTL Table

LEVEL_KEY_COLUMN_DTL_SK	KEY_TYPE_CD	KEY_COLUMN_SK	LEVEL_SK
1	RETAIN_KEY	99	1

3. Insert the record in the SUBSET_FROM_PATH_X_LEVEL table (available in the CSCAPDM library).

Table 4.26 SUBSET_FROM_PATH_X_LEVEL Table

SUBSET_FROM_PATH_SK	LEVEL_SK	SELECT_SOURCE_COLUMN_SK	CORRSP_LEVEL_KEY_COLUMN_DTL_SK
41	1	1178	1

Subset Query

Assume that a user defines a subset criterion based on the subset map that you have created. This subset criterion filters customers whose type code is individual, status is “CS_1” (suspended), and tenure is of at least 3 months. In this case, the following query is created. The SELECT_SOURCE_COLUMN_SK column is selected as a part of the final query.

```
Proc sql;

Create table cacmdscr.M_10000015_L1_2 as SELECT DISTINCT table0.CUST_ID AS CUST_ID

FROM CFDDIM.EVENT_TYPE_D table1

Inner join

CFDFACT.USAGE_SUMMARY_F table3 on table1.EVENT_TYPE_SK = table3.EVENT_TYPE_SK

Inner join

CFDDIM.CUST_D table0 on table0.CUST_SK = table3.CUST_SK

Inner join

CFDDIM.SERVICE_D table2 on table2.SERVICE_SK = table3.SERVICE_SK

WHERE ((('31DEC10:23:59:59'DT >= CUST_D.VALID_START_DTTM and

'31DEC10:23:59:59'DT <= CUST_D.VALID_END_DTTM)

AND ('31DEC10:23:59:59'DT >= EVENT_TYPE_D.VALID_START_DTTM and

'31DEC10:23:59:59'DT <= EVENT_TYPE_D.VALID_END_DTTM)

AND ('31DEC10:23:59:59'DT >= SERVICE_D.VALID_START_DTTM and

'31DEC10:23:59:59'DT <= SERVICE_D.VALID_END_DTTM))

AND (((CUST_D.CUST_FIRST_BECAME_DT <=

(dabt_func_intnx_dttm_mnth_same('DTMONTH', '31DEC10:23:59:59'DT,-3, 'SAME')))

AND (CUST_D.CUST_STATUS_CD IN ('CS_1')) AND (CUST_D.CUST_TYPE_CD IN ('INDIVIDUAL')))));

quit;
```

This query can be split into three parts:

- The first part (the selection part) is the configuration made in the SELECT_SOURCE_COLUMN_SK column of the SUBSET_FROM_PATH_X_LEVEL table.
- The second part (the FROM condition) is the configuration made in the SUBSET_TABLE_JOIN_CONDITION table.
- The last part indicates the filters that you define for the subset criterion that you defined for the subset map.

Creating a Subject Group

Overview

A subject group is a predefined group of members that share a common set of attributes. For example, all customers of individual type can be grouped together. A subject group, together with the subset criterion that you define, determines the records in the final ABT data set. When an ABT is built, only those records that satisfy the following criteria are used to populate the ABT:

- records that belong to the specified subject group

- records that satisfy the filter conditions specified in the subset criterion

For more information, see the *SAS Customer Analytics for Communications: User's Guide*.

Subject groups are associated with a particular purpose and subject of analysis. The subject group that you can select for a project depends on the purpose and the subject of analysis that you select for the project. For projects with subject of analysis as customer and subscription , SAS Customer Analytics for Communications provides the Individual Customers or Subscriptions as the predefined subject group.

You can also create your own subject groups. To create a new subject group, complete these steps:

1. Create a subset criterion that defines the subjects expected by the subject group that you plan to create. For details, see "Create a Subset Criterion" on page 79.
2. Define a subject group based on the subset criterion that you created in step 1. For details, see "Define a Subject Group" on page 80.
3. Configure the availability of the subject group for certain combinations of purposes and subjects of analyses. For details, see "Configure the Availability of the Subject Group" on page 80.

Create a Subset Criterion

A subject group defines a group of members that share a common set of attributes. These common attributes are internally specified in the form of a subset criterion. Therefore, to create a subject group, you first need to create a subset criterion. This involves certain configurations in the SUBSET_QUERY_MASTER, SUBSET_FILTER_NODE, SUBSET_FILTER_NODE_EXPRESSION, and FILTER_NODE_EXPRSSN_X_VALUE tables. These tables are available in the CSCAPDM library. For descriptions of these tables and their columns, see *SAS Customer Analytics for Communications: Data Reference Guide*.

You can either manually edit these tables and enter the configuration details for the new subset criterion, or you can use the SAS Customer Analytics for Communications interface to create a subset criterion. When you create a subset criterion through the interface, these tables are automatically updated with the configuration details of the new subset criterion.

To create a new subset criterion (through the interface), complete these steps:

1. Log on to SAS Customer Analytics for Communications with an appropriate profile.
2. Create a project with the subject of analysis for which you want to configure the subject group. For instructions on how to create a project, see the *SAS Customer Analytics for Communications: User's Guide*.
3. Open the project, and then create a subset criterion with the required subset map and filters such that the subset criterion defines the members (with a common set of attributes) as required by the subject group. For instructions on how to create a subset criterion, see the *SAS Customer Analytics for Communications: User's Guide*.
4. Note the project ID that is assigned to the project, and then log off from SAS Customer Analytics for Communications.
5. Open the PROJECT_MASTER table (available in the CSCAPDM library). This table contains a record for each project that is created in the application. Locate the record corresponding to your project. You can use the project ID to locate the record.

6. For this record, note the value in the SUBSET_QUERY_SK column. The value in this column uniquely identifies the new subset criterion definition in the SUBSET_QUERY_MASTER table.
7. Set the value of the SUBSET_QUERY_SK column to NULL. This breaks the link between the subset criterion and the corresponding project. That means, when you delete the project, the associated subset criterion is not deleted. Save the changes, and then close the PROJECT_MASTER table.
8. Log on to SAS Customer Analytics for Communications with an appropriate profile.
9. Delete the project that you created in step 2.

Define a Subject Group

To define a subject group based on the subset criterion that you created in “Create a Subset Criterion,” you need to insert a record in the SUBJECT_GROUP_MASTER table (available in the CSCAPDM library). While doing that, in the SUBSET_QUERY_SK column, insert the value that you noted in step 6 in “Create a Subset Criterion.”

Note: Make sure that you enter a unique value for the SUBJECT_GROUP_SK column. Also, make sure that this value is greater than 1000000.

The following table shows the SUBJECT_GROUP_MASTER table with a sample record:

Table 4.27 *SUBJECT_GROUP_MASTER Table*

SUBJECT_ GROUP_SK	SUBJECT_ GROUP_CD	SUBJECT_ GROUP_ SHORT_NM	SUBJECT_ GROUP_DESC	SUBSET_ QUERY_SK
3	INDCST	Individual Customers or Subscriptions	Individual Customers or Subscriptions	57

For descriptions of the columns in the SUBJECT_GROUP_MASTER table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Configure the Availability of the Subject Group

After you create a subject group, you must configure it such that it is available for all the desired combinations of purposes and subjects of analyses in a project. To do this, you need to enter appropriate records in the SUBJECT_GROUP_SPCFCN_DTL table (available in the CSCAPDM library). Typically, you should configure a subject group such that it is available in projects with the subject of analysis same as that of the project that you created in step 2 in “Create a Subset Criterion.”

The following table shows the SUBJECT_GROUP_SPCFCN_DTL table with sample records:

Table 4.28 SUBJECT_GROUP_SPCFCN_DTL Table

PURPOSE_SK	LEVEL_SK	SUBJECT_GROUP_SK
3	1	3
4	1	3
5	4	3
9	1	3
9	4	3
10	1	3
10	4	3

Configuring Outcome-Based Filtering

Outcome-based filtering enables the user to select correct observations in the modeling and scoring ABTs. If for a given subject, the event under consideration is already satisfied within the performance window itself, then such subjects are removed from the corresponding modeling or scoring ABT. For more information about outcome-based filtering, see the *SAS Customer Analytics for Communications: User's Guide*.

The values in the following columns of the PURPOSE_LEVEL_CONFIG_DTL table (available in the CSCAPDM library) determine whether outcome-based filtering is enabled for modeling and scoring ABTs (for a given combination of purpose and subject of analysis):

APPLY_OC_PST_IMPL_SBST_MDL_FLG
indicates whether outcome-based filtering is enabled for modeling ABTs.

APPLY_OC_PST_IMPL_SBST_SCR_FLG
indicates whether outcome-based filtering is enabled for scoring ABTs.

By default, for all purposes except for customer retention, cross-sell, and up-sell, the values of these columns are set to N. This is because, outcome-based filtering is applicable only to customer retention, cross-sell, and up-sell projects. To enable outcome-based filtering, set the value of these columns to Y.

If outcome-based filtering is enabled, the value for the outcome variable is temporarily calculated with reference to the ABT build date. The final ABT data set contains only those records for which the value of the outcome variable as of the ABT build date matches the value of the OC_BSD_PST_IMPL_SBST_VALUE parameter. For details about this parameter, see "Parameters List" on page 111.

Configuring Build-Date Cap

Build-date cap is applied while calculating the value of an outcome variable in the modeling ABT. The variables that are involved in the calculation of the outcome variable might refer to a period that falls before the modeling ABT build date. In this case, only the data pertaining to the period after the ABT build date is considered for the calculation. For more information about the build-date cap, see the *SAS Customer Analytics for Communications: User's Guide*. The APPLY_BLD_DT_CAP_OUTCM_VAR_FLG column of the PURPOSE_MASTER table (available in the CSCAPDM library) determines whether the build-date cap is enabled. By default, the value of the APPLY_BLD_DT_CAP_OUTCM_VAR_FLG column is set to Y only for customer retention, cross-sell, and up-sell purposes. This is because, the build-date cap is not applicable to customer segmentation, customer lifetime value, market basket analysis, and customer acquisition projects. To disable the build-date cap, set the value of the APPLY_BLD_DT_CAP_OUTCM_VAR_FLG column to N.

Defining Implicit Subset Criterion

Overview

A subset criterion enables you to define the target population for your project. Certain business rules need to be applied to the base population to derive the correct target population. These business rules are mandatory and need to be applied irrespective of whether a subset criterion is defined for a project. These business rules can be applied in the form of an implicit subset criterion. For example, while creating a scoring ABT for a customer retention project, it is recommended to consider customers who have the active status. To enforce this business rule, an implicit subset criterion needs to be defined. An implicit subset criterion depends on the purpose and the subject of analysis. Here is the list of a few predefined implicit subset criteria thatSAS Customer Analytics for Communications provides:

- Postpaid Customers or Subscriptions
- Prepaid Customers or Subscriptions
- Fresh Prospects
- Contacted Prospects

You can also define your own implicit subset criterion. Creating an implicit criterion involves the following tasks:

1. Create an implicit subset criterion that defines the business rule that you want to apply to select the right population. For details, see "Create an Implicit Subset Criterion" on page 82.
2. Configure the availability of the implicit subset criterion for certain combinations of purposes and subjects of analyses. For details, see "Configure the Availability of the Implicit Subset Criterion" on page 83.

Create an Implicit Subset Criterion

An implicit subset criterion determines a group of members that satisfy certain business rules or attributes. These common attributes are internally specified in the form of a subset criterion. Therefore, to create an implicit subset criterion, you first need to create a subset criterion. This task involves certain configurations in the SUBSET_QUERY_MASTER, SUBSET_FILTER_NODE, SUBSET_FILTER_NODE_EXPRESSION, and

FILTER_NODE_EXPRSSN_X_VALUE tables. These tables are available in the CSCAPDM library. For descriptions of these tables and their columns, see *SAS Customer Analytics for Communications: Data Reference Guide*.

You can either manually edit these tables and enter the configuration details for the new subset criterion, or you can use the SAS Customer Analytics for Communications interface to create a subset criterion. When you create a subset criterion through the interface, these tables are automatically updated with the configuration details of the new subset criterion. For details, see "Create a Subset Criterion" on page 79.

Note: When you manually enter a new subset criterion, make sure that you enter a unique value for the SUBSET_QUERY_SK column. Also, make sure that this value is greater than 1000000.

Configure the Availability of the Implicit Subset Criterion

After you have created an implicit subset criterion, you must configure it such that it is available for all the desired combinations of purposes and subjects of analyses in a project. To do this, you need to enter appropriate records in the IMPLICIT_SUBSET_SPCFCTN_DTL table (available in the CSCAPDM library). Typically, you should configure an implicit subset criterion such that it is available in projects with the subject of analysis same as that of the project for which you created the implicit subset criterion as explained above.

Note: The implicit business rules that are to be applied to a project can be different for modeling and scoring ABTs. Therefore, you must configure the implicit subset criterion for the modeling and scoring ABTs accordingly. To do so, in the IMPLICIT_SUBSET_SPCFCTN_DTL table, specify MDL or SCR as the value for the APPLICABLE_FOR_PROCESS_TYPE_CD column for modeling and scoring ABT respectively.

The following table shows the IMPLICIT_SUBSET_SPCFCTN_DTL table with sample records:

Table 4.29 *IMPLICIT_SUBSET_SPCFCTN_DTL Table*

PURPOSE_SK	LEVEL_SK	APPLICABLE_FOR_PROCESS_TYPE_CD	SUBSET_QUERY_SK
3	1	MDL	8
3	1	SCR	60
4	1	MDL	7
4	1	SCR	58
5	4	MDL	8
5	4	SCR	61

Creating an Implicit Variable

Overview

Similar to an implicit subset criterion, there can be some business rules, which require some mandatory variables to be a part of the modeling and or scoring ABT. These mandatory variables are called implicit variables. After you configure these variables, they are automatically added in the modeling and or scoring ABT. You will not able to edit or delete such variables from the SAS Customer Analytics for Communications interface.

For example, while creating an ABT for customer retention, the subscription's total revenue in past 6 weeks or last 6 month's revenue needs to be present in a scoring ABT. Therefore, these variables need to be configured as implicit variables. The implicit variables that you configure, are added by default in the modeling ABT. You can decide whether you want to add them in the scoring ABT. An implicit variable can be of any standard type of variable, which can be created by using the SAS Customer Analytics for Communications interface.

SAS Customer Analytics for Communications provides the following predefined implicit variables:

- Customer's First Activation Date
- Customer's Termination Date
- Customer's tenure until ABT building or reference date in months
- Customer's tenure until ABT building or reference date in weeks
- Total revenue generated by a subscription over past 6 months
- Total revenue generated by a subscription over past 6 weeks
- Total TV usage for past 6 months

You can also create your own implicit variable. Creating a new implicit variable involves the following tasks:

1. Create an ABT variable that satisfies a given business rule. For details, see "Create an Implicit ABT Variable" on page 84.
2. Configure the availability of the implicit variable for certain combinations of purposes, subjects of analysis and the time grain. For details, see "Configure the Availability of the Implicit Variable" on page 85.

Create an Implicit ABT Variable

An implicit variable is the variable, which is mandatory for the modeling and or scoring ABT for given purpose, subject of analysis, and time grain. Therefore, to create an implicit variable, you first need to create an ABT variable. This involves certain configurations in the VARIABLE_MASTER, MODELING_ABT_MASTER, MODELING_ABT_X_VARIABLE tables and any of the BEHAVIORAL_VARIABLE, RECENT_VARIABLE, SUPPLEMENTARY_VARIABLE, DERIVED_VARIABLE, and DERIVED_VAR_X_EXPRESSION_VAR tables based on the type of implicit variable that you want to create. These tables are available in the CSCAPDM library.

For descriptions of these tables and their columns, see *SAS Customer Analytics for Communications: Data Reference Guide*.

You can either manually edit these tables, enter the configuration details for the new ABT variable or you can use the SAS Customer Analytics for Communications interface to create an ABT variable. When you create a variable through the interface, these tables are automatically updated with the configuration details of the new variable.

Note: When you manually enter a new implicit variable, make sure that you enter a unique value for the VARIABLE_SK column. Also, make sure that this value is greater than 1000000.

To create a new ABT variable (through the interface), complete these steps:

1. Log on to SAS Customer Analytics for Communications with an appropriate profile.
2. Create a project with the subject of analysis for which you want to configure the implicit variable. For instructions about how to create a project, see the *SAS Customer Analytics for Communications: User's Guide*.
3. Open the project, and then create an ABT. Save the ABT details. Create a variable that you want to configure as an implicit variable. For instructions about how to create an ABT and its variables, see the *SAS Customer Analytics for Communications: User's Guide*.
4. Note the project ID that is assigned to the project, and then log off from SAS Customer Analytics for Communications.
5. Open the PROJECT_MASTER table (available in the CSCAPDM library). This table contains a record for each project that is created in the application. Locate the record corresponding to your project. You can use the project ID to locate the record.
6. Open the MODELING_ABT_MASTER table. Set the value of the PROJECT_SK column to NULL. This breaks the link between the modeling ABT and the corresponding project. That means, when you delete the project, the associated modeling ABT is not deleted. Save the changes, and then close the PROJECT_MASTER and MODELING_ABT_MASTER table.
7. Log on to SAS Customer Analytics for Communications with an appropriate profile.
8. Delete the project that you created in step 2.

Configure the Availability of the Implicit Variable

After you have created an implicit variable, you must configure it such that it is available for all the desired combinations of purposes and subjects of analysis and time grains in a project. To do this, you need to enter appropriate records in the IMPLICIT_VAR_SPECIFICATION_DTL table (available in the CSCAPDM library). Typically, you should configure an implicit variable such that it is available in projects with the subject of analysis same as that of the project that you created as explained above. Whenever you configure an implicit variable, it is automatically added in the modeling ABT. You can decide whether you want to include it in the scoring ABT. You should also configure whether the implicit variable is an outcome variable.

The following table shows the IMPLICIT_VAR_SPECIFICATION_DTL table with sample records:

Table 4.30 *IMPLICIT_VAR_SPCFCTN_DTL Table*

PURPOSE_ SK	LEVEL_SK	AVAILBLE_ FOR_ABT_ TIME_ GRAIN_SK	VARIABLE_ SK	MDL_ABT_ OUTCOME_ VARIABLE_ FLG	APPLICABLE_ FOR_SCR_ PROCESS_ FLG	APPLICABLE_ FOR_ACT_ PROCESS_ FLG
5	4	3	10000004	N	Y	N
6	4	1	10000003	N	Y	N
9	4	3	10000004	N	Y	N
10	4	1	100000003	N	Y	N
11	4	3	10000004	N	Y	N

Filtering Statistically Significant Rules

The analytical model that is built in SAS Enterprise Miner for association rules analysis generates the output in the form of a rules data set. This data set contains the association rules that are derived from the association rules analysis. All these rules are written into the MODEL_RULE_MASTER table (in the CSCAPDM library). These rules are filtered and marked as statistically significant or statistically insignificant based on certain filtering logic. For details, see *SAS Customer Analytics for Communications: User's Guide*.

If you want to change the default logic to filter rules, you have to modify the cac_rules_filter.sas macro.

Here is the list of macros that are relevant to rules filtering.

cac_rules_filter.sas

> contains the code for filtering rules from the rules data set. This macro is called by the dabt_new_capture_mdl_wrapper.sas macro by using the MCR_DABT_RULES_FILTER parameter. This parameter is a column of the DABT_CONFIG_PARAM_DTLS table (in the CSCAPDM table). The value of this parameter is cac_rules_filter, which is the name of the macro itself. If you rename the cac_rules_filter.sas macro, you also have to change the value of this parameter accordingly.
>
> The cac_rules_filter.sas macro is located at the following location: **`<SAS Home>/SASFoundation/9.3/dabtsrv/cmacros`**.

dabt_new_capture_mdl_wrapper.sas

> calls the cac_rules_filter.sas macro and inserts the rules that this macro filters into the MODEL_RULE_MASTER table (in the CSCAPDM library). The dabt_new_capture_mdl_wrapper.sas macro is located at the following location: **`<SAS Home>/SASFoundation/9.3/cacsrv/ucmacros`**.

Deploying the Modeling ABT

SAS Customer Analytics for Communications provides you the feature of deploying a modeling ABT. After you define an ABT, define its variables, and mark the outcome variable, if applicable, you can deploy the ABT. For details about how to deploy an ABT, see *SAS Customer Analytics for Communications: User's Guide*. When you deploy an ABT, the ABT-building job is exported, so that you can schedule a batch run to build the ABT. In addition, the mdl_abt_deploy_job folder is created in the following location:**`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm/5.4/project/<Project ID>`**. The information that is required for building an ABT is exported in this folder. This information includes the following items:

query files
: These files are exported if the ABT variables are defined by using an information map as a data source.

data sets
: These data sets contain the following information:

- the level at which ABT is to be built
- the source libraries of the data sources
- subset criterion that is defined after ABT creation
- information map and its data items that are to be used
- source tables, if any are used
- subset maps used in the subset criterion that is defined for the ABT
- library type of the source tables that are used to define the subset criterion
- types of variables that are defined
- outcome variable that is defined for the ABT
- source variable that is used to define the outcome variable

You can schedule the ABT-building job by using a suitable environment and use the exported information for building the ABT.

Creating a Post-Action Macro

Overview

SAS Customer Analytics for Communications provides a framework to perform tasks starting from project creation to scoring template execution. There can be a situation in which certain additional processing is required for a project, ABT, model, published model, or a scoring template. To do so, SAS Customer Analytics for Communications provides the flexibility to execute certain additional processes along with the predefined process to arrive at the desired result. These processes can be run as post-action macros. You can run these macros at certain instances, termed as actions. After the specific action is complete, you can run the macros that you have configured.

For projects with customer as a subject of analysis and Customer Lifetime (CLTV) value as purpose, SAS Customer Analytics for Communications provides following predefined post-action macros:

cac_convert_lftm_val
After the scoring template is executed successfully for CLTV, this macro converts the customer's expected tenure from months or weeks to days depending on the base offer payment mode of the customer.

cac_modify_timeids
After the CLTV ABT is built, this macro converts the customer's activation and deactivation dates to date format from datetime format.

You can also create your own post-action macro. Creating a new post-action macro involves the following tasks:

1. Create the post-action macro, which accesses the keys of the action after which the macro will be executed. For details, see "Create a Post-Action Macro" on page 88.
2. Configure the availability of the post-action macro for certain combinations of action, purpose, and the macro name. For details, see "Configure the Availability of the Post-Action Macro" on page 89.

Create a Post-Action Macro

These macros are executed after a specific action from the SAS Customer Analytics for Communications interface is complete. SAS Customer Analytics for Communications provides a predefined set of actions after which a post-action macro can be executed. These are non-editable actions.

The following table lists the predefined actions that can be performed through the SAS Customer Analytics for Communications interface. You must not change the details in this table or add new actions to it.

Note: You can define a post-action macro for any of the actions that are listed below.

Table 4.31 *ACTION_MASTER Table*

ACTION_SK	ACTION_CD	ACTION_SHORT_NM
1	DEL_PRJ	Delete Project
2	NEW_PRJ	New Project
3	NEW_ABT	New ABT
4	DEL_ABT	Delete ABT
5	NEW_MDL	New Model Capture
6	EDT_MDL	Edit Model
7	DEL_MDL	Delete Model
8	PBL_MDL	Publish Model for Scoring
9	NEW_SCR_TMPLT	New Scoring Template

ACTION_SK	ACTION_CD	ACTION_SHORT_NM
10	DEL_SCR_TMPLT	Delete Scoring Template
11	BLD_MDL_ABT	Build Modeling ABT
12	SCR_JOB	Scoring Job Execution
13	ACT_JOB	Actual Value Job Execution

Write the SAS code in the form of a SAS macro to execute the extra processes that you run. Save the SAS code at the following location:

Windows path

```
<SAS Home>/SASFoundation/9.3/cacsrv/ucmacros
```

UNIX path

```
<SAS Home>/SASFoundation/9.3/ucmacros/cacsrv
```

Configure the Availability of the Post-Action Macro

After you create the SAS macro, you must configure it and make it available for all desired combinations of purpose and the action after which it is to be called. To do so, you need to enter appropriate records in the POST_ACTION_MACRO_CONFIG_DTL table (available in the CSCAPDM library).

The following table shows the POST_ACTION_MACRO_CONFIG_DTL table with sample records:

Table 4.32 *POST_ACTION_MACRO_CONFIG_DTL Table*

ACTION_SK	PURPOSE_SK	POST_ACTION_MACRO_NM
12	15	cac_convert_lftm_val
12	16	cac_convert_lftm_val
11	15	cac_modify_timeids
11	16	cac_modify_timeids

Restarting the Server-Side Web Application

When you configure objects such as subject groups or time periods, which cannot be configured from the SAS Customer Analytics for Communications interface, certain tables (in the CSCAPDM library) of the Application data are updated. In order to make these changes effective, you have to restart the server-side web application. For example, you define new time periods in the SOURCE_TIME_PERIOD table. For details, see “Configuring Time Periods for Behavioral and Time-Based Variables” on page 49 and “Configuring Time Periods for Direct Variables” on page 50. In order that these time periods are available when you define a behavioral variable by using the SAS Customer

Analytics for Communications interface, you have to restart the server-side web application.

Chapter 5

Performing Middle-Tier Administrative Tasks

Middle-Tier Administration Overview

The middle-tier component synchronizes all components of SAS Customer Analytics for Communications and enables them to function together as an integrated system. It interacts with the web-based user interface of SAS Customer Analytics for Communications and the SAS library. Also, it connects to the SAS run-time environment in order to run analytical processes.

You have to perform certain administrative activities in order to set up and configure the middle-tier component on various servers such as the metadata server, the application server, and the SAS library. Also, you have to maintain the error logs that are generated by SAS Customer Analytics for Communications.

Roles and Capabilities

Different users of SAS Customer Analytics for Communications might have access to different functionality depending on their assigned roles. Each role is mapped to a set of predefined capabilities. A capability, also known as an application action, defines the operations that a user can perform. One or more roles can be assigned to a user who can access SAS Customer Analytics for Communications. If multiple roles are assigned to a user, then the least restrictive capability of each role is granted to the user.

The following table lists the capabilities that are defined in SAS Customer Analytics for Communications.

Note: The project-related capabilities enable a user to view or work on only those projects and their components, such as subset criteria, analytical base tables (ABTs), and models, that the user has created. A user cannot view or work on the projects that other users have created.

TIP The Create or the Delete capability on an object does not enable a user to create or delete the object unless the user also has the View capability on that object.

Table 5.1 *Capabilities in SAS Customer Analytics for Communications*

Workspace Name	Capability Name	Description
Projects	View Project	Enables a user to view projects.
	Create Project	Enables a user to create and edit projects.
	Delete Project	Enables a user to delete projects.
	View Subset Criterion	Enables a user to view subset criteria.
	Create Subset Criterion	Enables a user to create, save, edit, and delete subset criteria and filter nodes.
	View Modeling ABT	Enables a user to view modeling ABTs.
	Create Modeling ABT	Enables a user to create, edit, and delete modeling ABTs and variables. This capability also enables the user to import variables and share ABTs with other users.
	Deploy Modeling ABT	Enables a user to deploy a modeling ABT, so that it can be built any time later.
	Build Modeling ABT	Enables a user to build modeling ABTs.
	Register Modeling ABT	Enables a user to register modeling ABTs with the SAS Metadata Server.
	View Model	Enables a user to view analytical models.
	Capture Model	Enables a user to capture and edit a model’s information.
	Delete Model	Enables a user to delete a model’s captured information.
	Publish Model	Enables a user to publish models for scoring.
Scoring	View Scoring Template	Enables a user to view scoring templates.
	Create Scoring Template	Enables a user to create and edit scoring templates.
	Delete Scoring Template	Enables a user to delete scoring templates.
Administrative	Manage Configuration of the Application Mart	Enables a user to import and configure source tables, columns, dimensional attribute values, and time periods for behavioral variables for use in SAS Customer Analytics for Communications.

SAS Customer Analytics for Communications is shipped with three predefined roles. Each role is pre-assigned a set of capabilities. The following table shows the default mapping of roles to capabilities.

Note: Using SAS Management Console, you can modify the roles and specify the capabilities according to your requirements. You can also define new roles. For more information about defining users and granting roles and capabilities, see *SAS 9.3 Management Console: Guide to Users and Permissions*. This document is available at `http://support.sas.com/documentation/cdl/en/mcsecug/63190/PDF/default/mcsecug.pdf`.

Table 5.2 *Mapping of Roles to Capabilities*

	Roles		
Capability	**Cust Analytics Communication: Business Analysis**	**Cust Analytics Communication: Data Analysis**	**Cust Analytics Communication: Administration**
View Project	Y	Y	Y
Create Project	Y	Y	N
Delete Project	Y	Y	N
View Subset Criterion	Y	Y	Y
Create Subset Criterion	Y	Y	N
View Modeling ABT	Y	Y	Y
Create Modeling ABT	Y	Y	N
Deploy Modeling ABT	Y	Y	N
Build Modeling ABT	Y	Y	N
Register Modeling ABT	Y	Y	N
View Model	Y	N	Y
Capture Model	Y	N	N
Delete Model	Y	N	Y
Publish Model	Y	N	N
View Scoring Template	Y	N	Y
Create Scoring Template	Y	N	N
Delete Scoring Template	Y	N	N
Manage Configuration of Application Data Mart Tables	N	N	Y

Change the Owner of a Project

When a user creates a project, the user is the default owner of the project. Only the owner of a project can view and work on the project. Other users cannot view or work on the project. However, according to the business requirements, you can assign the ownership of the project to another user.

Note: Only one user at a time can be the owner of a project.

To change the owner of a project:

1. Log on to SAS Data Integration Studio, and then connect to the desired metadata server.
2. On the menu, select **Tools** ⇨ **Code Editor**.
3. Enter the following code in the Code Editor, and then click **Run**. If asked for a user name and password, use an appropriate user (for example, sassrv) to run the code.

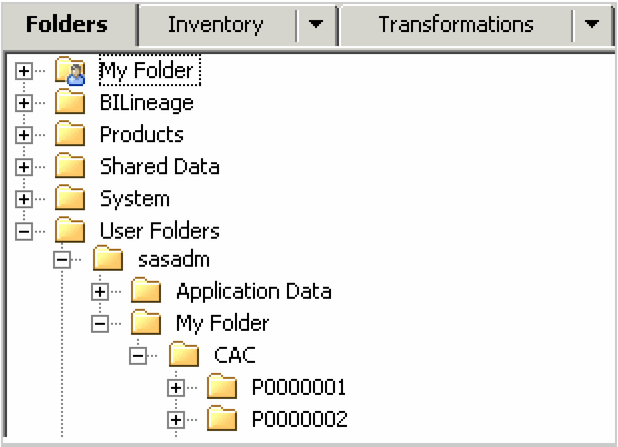
   In the code, replace <*Project ID*> with the ID of the project. Replace <*User ID*> with the user ID of the new owner.

   ```
   %cacinit(set_sql_func_ex_append=Y);
   %dabt_change_project_ownership
   (m_project_id = <Project ID>, m_change_owner_to = <User ID>);
   ```

4. Move the project folder from the CAC folder of the previous owner to the CAC folder of the new owner. For instructions on how to move a folder to another folder in SAS Data Integration Studio, see the SAS Data Integration Studio Help.

   When a user creates the first project (in the Projects workspace), a folder structure for that user is automatically created in the SAS metadata. For each subsequent project that the user creates, a folder is automatically created under the CAC folder. The name of the folder is the same as the project ID. For example, for sasdemo, the folder structure looks similar to the following in SAS Data Integration Studio:

   Display 5.1 *Folder Structure*



   Note: If the new owner of the project has not yet created a project, you must first manually create the required folder structure (up to the CAC folder) in SAS Data Integration Studio. Then, move the project folder from the CAC folder of the previous owner to the CAC folder of the new owner.

5. Close SAS Data Integration Studio.

Working with Software Component Properties

Overview of Software Component Properties

SAS Customer Analytics for Communications has properties defined for the following software components:

- SAS Dynamic Analytical Base Table Server
- SAS Communications Common Server
- SAS Customer Analytics for Communications Server
- SAS Customer Analytics for Communications Middle Tier

It is recommended that you do not modify the default values of these properties. However, if you modify the values, you must redeploy SAS Customer Analytics for Communications on your web application server.

Note: The subsequent topics mention only the software component properties that are visible in SAS Management Console.

SAS Dynamic Analytical Base Table Server

To view or modify the application properties of the SAS Dynamic Analytical Base Table Server component:

1. Open SAS Management Console, and then connect to the appropriate metadata server.
2. On the **Plug-ins** tab, expand**SAS Management Console** ⇨ **Application Management** ⇨ **Configuration Manager**.
3. Right-click **Dynamic Analy Base Tbl Svr 5.4**, and then select **Properties**.
4. In the Properties dialog box, select the **Advanced** tab. On this tab, you can see a list of the properties and their values. To change a value, click the value in the **Property Value** column change it according to your requirement. Click **OK**.

   The following table lists the properties of the SAS Dynamic Analytical Base Table Server component:

Table 5.3 *Properties of SAS Dynamic Analytical Base Table Server*

Property	Default Value	Description
dabt.stp.path	SBIP://METASERVER/System/SAS Dynamic Analytical Base Table Server/ Dynamic Analy Base Tbl Svr 5.4/dabt_stored_procs/	Location in which the stored procedures that are required by SAS Dynamic Analytical Base Table are registered.
dabtsrvc.config.dir	C:\SAS\Config\Lev1	Location in which Dynamic ABT server is configured.
dabtsrvc.version	5.4	Version of the Dynamic ABT server.

SAS Customer Analytics for Communications Server

To view or modify the properties of the SAS Customer Analytics for Communications Server software component:

Note: You must not modify the default values of the following properties. Doing so might result in unexpected behavior or might even cause the application to fail. However, if you accidentally change these values, you can replace the changed values with the default values.

1. Open SAS Management Console, and then connect to the appropriate metadata server.
2. In SAS Management Console, on the **Plug-ins** tab, expand the following sequence of nodes: **SAS Management Console ⇨ Application Management ⇨ Configuration Manager**.
3. Right-click **Cust Analytics Comm Server 5.4**, and then select **Properties**.
4. In the Properties dialog box, select the **Advanced** tab. On this tab, you can see a list of the properties and their values. To change a value, in the **Property Value** column, click the value and enter the desired value. Click **OK**.

   The following table lists the properties of the Cust Analytics Comm Svr 5.4 software component:

Table 5.4 Properties of SAS Customer Analytics for Communications Server

Property	Default Value	Description
cacsrvc.dabt.app_init_stmt	%cacinit(set_sql_func_ex_append=Y);	The INIT statement that needs to be executed before every process in SAS Customer Analytics for Communications is run.
cacsrvc.dabt.usepooled	TRUE	Indicates whether the Pooled Workspace Server is available. You can change value to FALSE if Pooled workspace server is not available or if it is desired to use the Logical Workspace Server instead.

SAS Customer Analytics for Communications Middle Tier

To view or modify the properties of the SAS Customer Analytics for Communications Middle Tier component:

1. In SAS Management Console, on the **Plug-ins** tab, expand the following sequence of nodes: **SAS Management Console ⇨ Application Management ⇨ Configuration Manager ⇨ SAS Application Infrastructure**.
2. Right-click **Cust Analytics Communications Mid 5.4**, and then select **Properties**.
3. In the Properties dialog box, click the **Advanced** tab. On this tab, you can see a list of the properties and their values. To change a value, in the Property Value column, click the value and enter the desired value. Click **OK**.

The following table lists the properties of the Cust Analytics Communications Mid 5.4 software component:

Table 5.5 *Properties of SAS Customer Analytics for Communications Middle Tier*

Property	Default Value	Description
cacmid.flex.LOCALE	en_US	The locale that is used if no locale is defined in the request or the locale that is defined in the request is not supported.
cacmid.flex.THEMES_DEFAULT	Corporate	Specifies the default Flex theme.
cacmid.flex.TIMEOUT ENABLED	TRUE	Indicates whether session timeout is enabled.
cacmid.flex.TIMEOUT PINGWINDOW_INTERVAL	10	Intervals (in milliseconds) at which the application checks whether the application server is active.
cacmid.flex.TIMEOUT WARNING_DURATION	300	Duration (in seconds) for which the warning dialog box is displayed before the session times out.

Configuring the Logging Folder

The logs of the middle-tier component are maintained in the **`<SAS configuration directory>/Lev1/Web/Logs`** folder. For example, on a Windows computer, this location can be **`C:/SAS/Config/Lev1/Web/Logs`**.

The logs of the middle-tier component use log4j and therefore support the following features:

- Increase or decrease the level of detail that is generated in the logs.
- Define the location where the logs can be generated.
- Select the preferred format for the log files.

The log configuration file is maintained in the **`<SAS configuration directory>/Lev1/Web/Common/LogConfig`** folder. You can change the logging configuration for the middle-tier component. However, before you do so, see the log4j documentation. For more information, see **`http://logging.apache.org/log4j/`**.

Chapter 6
Scoring

The Scoring Process

Overview

Scoring is the process of applying an analytical model to new data in order to compute outputs. In SAS Customer Analytics for Communications, scoring is performed with the help of a scoring template. A scoring template enables you to associate the scoring process with a model that has been published for scoring. For each scoring template, SAS Customer Analytics for Communications automatically creates a scoring job in the **`<`*`SAS configuration directory`*`>/AppData/SASCustAnalyticsComm/5.4/scoring/<`*`Scoring template ID`*`>/scr_act_run_exported_code`** folder. (This is the scoring path.)

The scoring_run_job_exec_<scoring template ID>.sas scoring job is created. The <scoring template ID> represents the scoring template identification number, which is a unique number that is automatically assigned to each scoring template when it is created. For information about how to create a scoring template, see the *SAS Customer Analytics for Communications: User's Guide*. You can use the scoring job to generate the predicted outcome values or scores for the current list of subject of analysis.

SAS Customer Analytics for Communications does not provide the framework to run or schedule this job. As an administrator, you must manually schedule this job (through an external scheduler). Typically, the scoring job is scheduled to run at the end of a month after the Foundation data mart is successfully loaded for that month.

Note: This job uses certain pre-assigned libraries defined in the SAS metadata server (such as the CSCAPDM library). Therefore, ensure that these pre-assigned libraries are available to the session in which this job is run.

The Scoring Job

The scoring job, scoring_run_job_exec_<scoring_template_ID>.sas, contains code similar to the following:

```
%cacinit(set_sql_func_ex_append=Y);

%dabt_build_scr_abt_wrapper (m_scoring_template_id=5, m_scoring_as_of_dt=,

m_perform_scoring_flg=Y, m_populate_arm_flg=Y);
```

In the above code:

m_scoring_template_id
is the unique ID that is assigned to each scoring template when it is created.

m_scoring_as_of_dt
specifies the scoring date. By default, this parameter does not have a value assigned to it. As a result, when the scoring job is run, it identifies the scoring date by using some internal logic. For details, see the steps that are mentioned below. These steps indicate the tasks that are performed when you run the scoring job.

If you want to perform scoring for a specific date, you can edit the scoring_run_job_exec_<scoring_template_ID>.sas file and specify the date by using the m_scoring_as_of_dt parameter. You must specify the date in the DDMONYYYY format. For example, if you want to perform scoring for 31 August 2010, the modified code should look similar to the following:

```
%cacinit(set_sql_func_ex_append=Y);

%dabt_build_scr_abt_wrapper(m_scoring_template_id=5, m_scoring_as_of_dt=31AUG2010,

m_perform_scoring_flg=Y, m_populate_arm_flg= Y);
```

m_perform_scoring_flg
indicates whether the SAS Enterprise Miner scoring code is to be applied on the scoring ABT. The possible values of the parameter are Y and N. Y is the default value. If the value is Y, then the scoring job will also apply the SAS Enterprise Miner scoring code to the scoring ABT and generate the corresponding Scored ABT in the Scoring ABT library. If the value of the flag is set to N, then the scoring job will stop its execution after preparing the Scoring ABT.

m_populate_arm_flg
indicates whether the scoring results are to be written to the Analytical results area. The possible values of the parameter are Y and N. Y is the default value. If you are applying the scoring code on your scoring ABT externally (that is, the value of the m_perform_scoring_flg parameter is set to N), the scoring results are not written to the Analytical results area, regardless of the value of the m_populate_arm_flg parameter. In this case, you can use populate_scoring_arm_job_<scoring_template_ID>.sas to write the scoring results to the Analytical results area. This job is available at the same location where the scoring job is located. Here is the structure of this job:

```
%cacinit(set_sql_func_ex_append=Y);

%dabt_populate_score_result(m_scoring_template_id=12, m_input_table_nm =,

m_scoring_bld_dttm =, m_divert_log_to_new_file = Y);
```

m_input_table_nm
name of the scored ABT prefixed with libref (for example, arm.scrd_abt_1_30APR2011). By default, no name is specified.

m_scoring_bld_dttm
scoring date in the DDMONYYYY:hh:mm:ss format (for example, 10JAN2009:23:59:59). By default, no date is specified.

m_divert_log_to_new_file

indicates whether the log is to be diverted to a new file. The default value is Y. The name of the log file that is created is populate_scoring_arm_job_<scoring_template_id>.log. If the value of this parameter is set to N, the log is written to the SAS log. The log file is created in the following location: **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm/5.4/scoring/<scoring template id>/scoring_run/log`**.

If you specify a value for the m_input_table_nm parameter, you must also specify a value for the m_scoring_bld_dttm parameter and vice versa. If you do not specify values for these parameters, the scoring job identifies all the successful scoring runs whose scoring results are yet to be written to the Analytical results area. That is, for a given scoring template, the scoring job checks the SCORING_CONTROL_DETAIL table for records where the SCORING_STATUS_SK column has a value of 1 and the LOAD_ARM_STATUS_SK column has a value other than 1.

When the scoring_run_job_exec_<scoring_template_id>.sas is run, it performs the following tasks:

1. Identifies the scoring date—the date for which the scoring is to be performed.

   For the first scoring run

   The scoring job checks the INPUT_MART_LOAD_DATES table (available in the Control library) for the Foundation data mart load date—the date up to which the Foundation data mart contains data. This date is the scoring date for the first scoring run.

   Note: The INPUT_MART_LOAD_DATES table is updated every time the Foundation data mart is loaded. This table has two columns: LOAD_DTTM and LOAD_STATUS. The latest date in the LOAD_DTTM column, with a value of Y in the LOAD_STATUS column, is treated as the scoring date for the first scoring run. For example, 31MAR2009:23:59:59 indicates that the Foundation data mart contains data up to 31Mar11:23:59:59. If the Foundation data mart has not yet been populated (there is no record in the INPUT_MART_LOAD_DATES table), or if there is no record with a value of Y in the LOAD_STATUS column, the scoring job stops its execution.

   Note: Also, make sure that you enter the time as 23:59:59 only. Otherwise, the scoring process will consider the previous month's or week's end date as the scoring date. For example, the value of the LOAD_DTTM column is 31Mar11:22:47:35. In this case, for an ABT of monthly grain, the scoring date will be considered as 28Feb11 instead of 31Mar11.

   For the subsequent scoring runs

   a. For a given scoring template, the scoring job checks the INPUT_MART_LOAD_DATES table for the Foundation data mart load date. Let us call it Date 1.

   b. For a given scoring template, the scoring job checks the SCORING_CONTROL_DETAIL table (available in the CSCAPDM library) for its corresponding last scoring date—the date as of which this scoring job was last run successfully. Let us call it Date 2. Then, the job compares Date 2 with Date 1. If Date 2 is earlier than Date 1 (Date 2 < Date 1), Date 1 is identified as the scoring date.

      If scoring has already been performed for the latest data in the Foundation data mart (Date 2 matches Date 1), the scoring job stops its execution.

2. Builds the scoring analytical base table (ABT) with reference to the scoring date. The scoring ABT is created with the significant variables.

   The scoring ABT is created with the name that was specified in the **Table name** field in the New Scoring Template window.

3. Applies the SAS Enterprise Miner scoring code on the scoring ABT. The SAS Enterprise Miner scoring code is available at the following location: **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm/5.4/scoring/<Scoring tempalte ID>/scr_act_run_exported_code/scoring_eminer_code_<Scoring template ID>.sas`**.

   The scored ABT is created with the following name: scrd_abt_<Scoring template ID>_<Scoring date>..This ABT contains the scoring output (for example, predicted values or customer segments).

4. Writes the scoring output to corresponding tables in the analytical result tables of the Analytical results area.

   Table 6.1 *Analytical Results Area Tables*

Table Name	Description
PREDICTED_SCORING_RESULT_CUST	Stores the values that are predicted by the analytical models for which the subject of analysis is Customer.
PREDICTED_SCORING_RESULT_SUB	Stores the values that are predicted by the analytical models for which the subject of analysis is Subscription.
PREDICTED_SCORING_RESULT_PRS	Stores the values that are predicted by the customer acquisition models for which the subject of analysis is Prospect Customer.
SEGMENT_SCORING_RESULT_CUST	Stores the segments that are created by segmentation models for which the subject of analysis is Customer.
RULES_SCORING_RESULT_CUST	Stores the association rules that are created by association models for which the subject of analysis is Customer.
RULES_SCORING_RESULT_SUB	Stores the association rules that are created by association models for which the subject of analysis is Subscription.

   Note: The records in these tables are later used to populate the Foundation data mart.

5. Inserts a record in the SCORING_CONTROL_DETAIL table. This table contains a record for each scoring run. The record contains information such as the scoring date, the status of the scoring run, and so on. The following table shows the SCORING_CONTROL_DETAIL table with sample records:

Table 6.2 SCORING_CONTROL_DETAIL Table

SCORING CONTROL DETAIL_ SK	SCORING TEMPLAT SK	SCORING AS_OF_ DTTM	BUILD_ ABT_ STATUS_ SK	SCORING STATUS_ SK	LOAD_ ARM_ STATUS_ SK	SCORED_ ABT_ NM	SCORING PROCESS DTTM	OUTCOME PERIOD_ END_ DTTM	ACTUAL_ RESULT_ RUN_ ACTIVE_ FLG
1	1	30APR2011:23:59:59	1	1	1	scrd_ abt_1_ 30APR201	10MAY2011:05:00:00	31JUL2011:23:59:59	N
2	1	31MAY2011:23:59:59	1	1	1	scrd_ abt_1_ 31MAY20	10JUN2011:05:00:00	31AUG2011:23:59:59	N
3	1	30JUN2011:23:59:59	1	1	1	scrd_ abt_1_ 30JUN201	10JUL2011:05:00:00	30SEP2011:23:59:59	Y
4	1	31JUL2011:23:59:59	1	1	1	scrd_ abt_1_ 31JUL201	10AUG2011:05:00:00	31OCT2011:23:59:59	Y

For descriptions of the columns in the SCORING_CONTROL_DETAIL table, see *SAS Customer Analytics for Communications: Data Reference Guide*.

After the execution of the scoring job, check the SCORING_CONTROL_DETAIL table (available in CSCAPDM) to view the status of the executed jobs. If the status_sk (BUILD_ABT_STATUS_SK, SCORING_STATUS_SK, and LOAD_ARM_STATUS_SK) of the corresponding jobs refer to Success, then the job is run successfully. For status values, see the STATUS_MASTER table of the CSCAPDM library.

Note: If the execution of the scoring job fails, check the log files that are generated during the scoring process. For details, see "Troubleshooting the Scoring Process" on page 142.

Scores Aggregation

Overview

SAS Customer Analytics for Communications supports analytical models at various levels such as customer, subscription, prospect customer, and so on. The scores are generated at the corresponding level.

Depending on the requirement, scores can be aggregated from one level to another. For example, customer churn is observed at subscription level. Therefore, the churn scores are always generated at subscription level. However, business also needs to generate the churn scores at customer level. In order to do so, the scores at subscription level are aggregated at customer level. The scores are aggregated by using the weighted average method. The usage revenue in the past 6 months for postpaid subscriptions and in the past 6 weeks for prepaid subscriptions is used for assigning scores at customer level. For example, let us assume that the subscriptions generate revenue worth 15 USD, 85 USD, and 25 USD in the past six months respectively. In this case, the churn scores can be derived by using the weighted average method as explained below:

CUST_MODEL_SCORE_NUM = SUM(WT_CHURN_SCORE) / SUM (REVENUE_AMOUNT)

Weighted churn score = Subscription probability * REVENUE_AMOUNT

Therefore, for the above example the churn score is calculated as follows:

CUST_MODEL_SCORE_NUM = (15*0.35+0.8*85+0.45*25) /125 = 0.676 ~ 0.68

This flags the customer as a churner.

The scores are aggregated mainly for customer churn, cross-sell, and up-sell models, which are built at subscription level.

The following high-level tasks that are performed when the scores are aggregated for a given model.

1. When a modeling ABT is defined for customer churn, cross-sell, or up–sell purposes, the S_SUM_PSU_URVAT_L6M or S_SUM_PRU_URVAT_L6W variables are added in the corresponding ABT depending on whether the ABT is to be built for postpaid or prepaid payment mode. These variables are configured as implicit variables for the above-mentioned purposes and also applicable for the scoring ABT. Therefore, when a scoring ABT is generated for any one of these purposes, these variables are also added in the scoring ABT. You have to configure these variables as the aggregation variables in the PARAMETER_VALUE_DTL table of the CSCAPDM library. Also, the value of these variables is used as the weighing value while aggregating the scores. If you want to add any other variable as a weighing variable, then you have to configure that variable as an implicit variable. For details, see “Creating an Implicit Variable” on page 84.

2. After the modeling ABT is built, a customer churn, cross-sell, or up-sell model is built on the modeling ABT. When the customer churn, cross-sell, or up-sell model’s information is captured through the SAS Customer Analytics for Communications interface, the respective variable (S_SUM_PSU_URVAT_L6M or S_SUM_PRU_URVAT_L6W) is marked as PREDICTED_OUTCOME_VALUE_1 in the MODEL_SCORING_OUTPUT_COLUMN table of the CSCAPDM library. This mapping is performed through the CAC_UPDATE_SCR_RSLT_COLUMN SAS macro, which resides at the following location:

   Windows path
   ```
   <SAS Home>/SASFoundation/9.3/cacsrv/ucmacros
   ```

   UNIX path
   ```
   <SAS Home>/SASFoundation/9.3/ucmacros/cacsrv
   ```

   This macro contains the logic of mapping the corresponding weighing column with the scoring output column. If you want to change this mapping and point it to any other column, then modify this macro accordingly. If you want to change the name of the macro, then configure the new name as a value of the MCR_DABT_UPDATE_SCR_RSLT_COLUMN column of the PARAMETER_VALUE_DTL table of the CSCAPDM library.

3. After the model’s information is captured and the model is published for scoring, the scoring template is created on the scoring model. The scoring template is executed and the scores at subscription level are written into the PREDICTED_SCORING_RESULT_SUB table of the ARM library.

4. The scores are also written to the writeback tables of the Foundation data mart by using the SAS Data Integration Studio jobs. In SAS Data Integration Studio, these jobs are available in the following path: **Products ⇨ SAS Customer Analytics for Communications 5.4 ⇨ Analytical Writeback ⇨ Jobs**. The cac_subscrp_model_score_dtl_job job aggregates the scores at customer level as

mentioned above if the scores aggregation flags, namely CFDN_AGGR_CHURN_SCRS_FLG or CFDN_AGGR_XS_SCRS_FLG are set to Y. The customer-level aggregated scores are first written to PREDICTED_SCORING_RESULT_CUST table of the ARM library, and then to the CUST_MODEL_SCORE_DTL table of the Foundation data mart by using the cac_cust_model_score_dtl_job job. The subscription-level scores are also written back to the SUBSCRP_MODEL_SCORE_DTL table of the Foundation data mart.

Note: For TV customers, the revenue that is generated does not vary much. Therefore, the subscriptions viewership usage over past 6 months (D_SUM_IPU_IPUSG_L6M) is used as the weighing variable. Otherwise, the steps involved in scores aggregation for churn, cross-sell, and up-sell models of TV customers are the same as mentioned above.

Deleting a Scoring Model and the Associated Data

Overview

If you no longer need a scoring model and the associated data (scoring results), you can delete them. You can use the dabt_delete_scoring_model macro to do this. When you run this macro, it deletes the specified scoring model and the associated data from the Analytical results area. However, if you want to retain the scoring model and delete only the associated data, you must manually delete the data from the Analytical results area.

Delete a Scoring Model and the Associated Data

1. Log on to SAS Customer Analytics for Communications.
2. In the Scoring workspace, select **Published Models**.
3. Note the ID of the model that you want to delete. Make sure that a scoring template is not defined for this model. If a scoring template is defined, then delete it. For details, see *SAS Customer Analytics for Communications: User's Guide*.
4. In SAS Data Integration Studio, enter the following code in the Code Editor, and then click **Run**.

   In the code, replace <scoring_model_id> with the ID of the model that you noted in step 3.

   ```
   %cacinit(set_sql_func_ex_append=Y);

   %dabt_delete_scoring_model(m_scoring_model_id = <scoring_model_id>);
   ```

   For example, if the model ID is 1000005, the code looks similar to the following:

   ```
   %cacinit(set_sql_func_ex_append=Y);

   %dabt_delete_scoring_model(m_scoring_model_id = 1000005);
   ```

   The log for this code is generated in the **`<SAS configuration directory>/Lev1/AppData/SASCustAnalyticsComm/5.4/admin/<`*`User ID`*`>/log`** folder. The name of the log file is dabt_delete_scoring_model_<*scoring_model_id*>.log

   This code deletes the scoring model and all the data associated with that model. In addition, it deletes data from the Analytical results area. Therefore, make sure that you load that data in the Application data or take a backup of that data before running this code.

Deleting Scoring Results

If you want to delete only the data (scoring results) associated with a scoring model and retain the model for later use, you must manually delete the data from the Analytical results area (certain tables in the ARM library). According to your requirements, you can delete all the data associated with a scoring model or delete only the data pertaining to specific time periods.

Based on the scoring as of date (SCORING_AS_OF_DTTM) for a scoring model (SCORING_MODEL_SK), you need to delete records from the appropriate Application data . The following table lists the tables that store scoring values for scoring models for different projects. For more information about these tables, see *SAS Customer Analytics for Communications: Data Reference Guide*.

Table 6.3 *Analytical Results Area Tables*

Purpose	Subject of Analysis	Scoring Results Table Name
Association Rules Postpaid	Customer Offer Agreement Date	Rules_scoring_result_cust
Association Rules Postpaid	Customer Service Activation Date	Rules_scoring_result_cust
Association Rules Postpaid	Customer Offer	Rules_scoring_result_cust
Association Rules Postpaid	Customer Service	Rules_scoring_result_cust
Association Rules Postpaid	Subscription Service Activation Date	Rules_scoring_result_sub
Association Rules Postpaid	Subscription Service	Rules_scoring_result_sub
Association Rules Prepaid	Customer Offer Agreement Date	Rules_scoring_result_cust
Association Rules Prepaid	Customer Service Activation Date	Rules_scoring_result_cust
Association Rules Prepaid	Customer Offer	Rules_scoring_result_cust
Association Rules Prepaid	Customer Service	Rules_scoring_result_cust
Association Rules Prepaid	Subscription Service Activation Date	Rules_scoring_result_sub
Association Rules Prepaid	Subscription Service	Rules_scoring_result_sub
Association rules TV offers & services	Customer Service Activation Date	Rules_scoring_result_sub
Association rules TV offers & services	Customer Offer	Rules_scoring_result_cust
Association rules TV offers & services	Customer Offer Start date	Rules_scoring_result_cust
Association rules TV offers & services	Customer Service	Rules_scoring_result_cust
Association rules TV offers & services	Subscription Service Activation Date	Rules_scoring_result_sub
Association rules TV offers & services	Subscription Service	Rules_scoring_result_sub
Association rules VoD & PpV	Subscription Telecast Item	Rules_scoring_result_sub
Association rules VoD & PpV	Subscription Telecast Item Order Date	Rules_scoring_result_sub
Churn Postpaid	Subscription	Predicted_scoring_result_sub

Purpose	Subject of Analysis	Scoring Results Table Name
Churn Prepaid	Subscription	Predicted_scoring_result_sub
Churn TV	Subscription	Predicted_scoring_result_sub
Cross-sell Postpaid	Customer	Predicted_scoring_result_cust
Cross-sell Postpaid	Subscription	Predicted_scoring_result_sub
Cross-sell Prepaid	Customer	Predicted_scoring_result_cust
Cross-sell Prepaid	Subscription	Predicted_scoring_result_sub
Cross-sell TV	Customer	Predicted_scoring_result_cust
Cross-sell TV	Subscription	Predicted_scoring_result_sub
Customer Acquisition	Prospect Customer	Predicted_scoring_result_prs
Customer Lifetime Postpaid	Customer	Predicted_scoring_result_cust
Customer Lifetime Prepaid	Customer	Predicted_scoring_result_cust
Customer Lifetime TV	Customer	Predicted_scoring_result_cust
Segmentation Postpaid	Customer	Segment_scoring_result_cust
Segmentation Prepaid	Customer	Segment_scoring_result_cust
Segmentation TV	Customer	Segment_scoring_result_cust
Up-Sell Postpaid	Customer	Predicted_scoring_result_cust
Up-Sell Postpaid	Subscription	Predicted_scoring_result_sub
Up-Sell Prepaid	Customer	Predicted_scoring_result_cust
Up-Sell Prepaid	Subscription	Predicted_scoring_result_sub
Up-Sell TV	Customer	Predicted_scoring_result_cust
Up-Sell TV	Subscription	Predicted_scoring_result_sub

Part 4

Appendixes

Appendix 1
Parameter Configuration

As a post-installation task, you have to configure certain parameters that are defined in the PARAMETER_MASTER table (available in the CSCAPDM library). The PARAMETER_VALUE_DTL table stores the default values for these parameters. Depending on whether you can edit the parameter, you can change the default values of these parameter.

The following table lists the solution-specific parameters that are defined in the PARAMETER_MASTER table and the default value that is set for each of these parameters in the PARAMETER_VALUE_DTL table. You can change the default value if the parameter can be edited. A parameter that can be edited is marked with a **Y** in the **Can Be Edited** column.

Table A1.1 *Solution-Specific Parameters*

Parameter Name	Description	Default Value	Can Be Edited
CUST_ACT_STATUS_ VALS	This column is used as a macro variable. It stores the values of customer status codes that indicate an active status of the customer.	"CS_1"	Y
DATETIME_COLMN_ TYP_CD	This column is used as a macro variable. It defines the column data type code that indicates the datetime format for a date variable. The value of this parameter can be changed if the configuration of a timestamp date variable is changed in the COLUMN_DATA_TYPE_ MASTER table of the CSCAPDM library.	"DTT"	Y
INDV_CUST_TYPE_CD_ VALS	This column is used as a macro variable. It stores the code that is assigned to the Individual type of customer.	"INDIVIDUAL"	Y

Parameter Name	Description	Default Value	Can Be Edited
MBA_TRANSPOSE_ FLAG_VALUE	This parameter stores the value of transpose flag that is available in the output table. This table is generated by the association node in SAS Enterprise Miner that is used for the association analysis purpose.	Y	N
NO_CAMPAIGN_VALS	This column is a macro variable. It indicates whether the prospective customer was not contacted through any campaign.	"NAVL"	Y
NON_SETTOP_OFFER_ SGMNT_CD_VALS	This macro variable drives the values of offer segment codes that identify an offer segment for non-TV customers. All the offer segments that correspond to a particular line of business and that are not specific to TV offers should be set as value of this parameter.	"4_OFFER_SE"	Y
PR_CLT_PURPOSE_CD	This parameter stores the code that is assigned to the Prepaid CLTV purpose.	CLR	Y
PR_CROSSSELL_ PURPOSE_AGG_VAR_ NM	For the Prepaid Cross-Sell purpose, this parameter stores the name of the variable that is used to aggregate the scores from subscription level to customer level.	S_SUM_PRU_URVAT_ L6W	Y
PR_CROSS_SELL_ PURPOSE_CD	This parameter stores the code that is assigned for the Prepaid Cross-Sell purpose.	XSR	Y
PR_MBA_PURPOSE_CD	This parameter stores the code that is assigned to the Prepaid Market Basket Analysis code.	MBR	Y
PR_RETENTION_ PURPOSE_AGG_VAR_ NM	For the Prepaid Customer Retention purpose, this parameter stores the name of variable that is used to aggregate scores from subscription level customer level.	S_SUM_PRU_URVAT_ L6W	Y

Parameter Name	Description	Default Value	Can Be Edited
PR_RETENTION_PURPOSE_CD	This parameter stores the code that is assigned for the prepaid customer retention purpose.	CRR	Y
PR_SEGMENT_PURPOSE_CD	This parameter stores the code that is assigned for the prepaid segmentation code.	CSR	Y
PR_UPSELL_PURPOSE_AGG_VAR_NM	For the Prepaid Up-Sell purpose, this parameter defines the variable name that is used to aggregate the scores from subscription level to customer level.	S_SUM_PRU_URVAT_L6W	Y
PR_UP_SELL_PURPOSE_CD	This parameter stores the code that is assigned for the Postpaid Up-Sell code.	USR	Y
PRE_PYMNT_MODE_VALS	This column is a macro variable. It stores the value of the code that identifies the prepaid method of payment mode.	"PREPAID"	Y
PST_CAQ_PURPOSE_CD	This parameter stores the code that is assigned for the Customer Acquisition purpose.	CAQ	Y
PST_CLT_PURPOSE_CD	This parameter stores the code that is assigned for the Postpaid Customer Lifetime (CLTV) purpose.	CLP	Y
PST_CROSSSELL_PURPOSE_AGG_VAR_NM	For the Postpaid Cross-Sell purpose, this parameter defines the variable name that is used to aggregate the scores from subscription level scores to customer level.	S_SUM_PSU_URVAT_L6M	Y
PST_CROSS_SELL_PURPOSE_CD	This parameter stores the code that is assigned for the Postpaid Cross-Sell purpose.	XSP	Y
PST_MBA_PURPOSE_CD	This parameter stores the code that is assigned for the postpaid market basket analysis (MBA) purpose.	MBP	Y

Parameter Name	Description	Default Value	Can Be Edited
PST_PYMNT_MODE_VALS	This parameter determines the code that identifies a postpaid method of payment mode.	"POSTPAID"	Y
PST_RETENTION_PURPOSE_AGG_VAR_NM	For the Postpaid Customer Retention purpose, this parameter defines the variable name that is used to aggregate subscription-level scores to customer-level scores.	S_SUM_PSU_URVAT_L6M	Y
PST_RETENTION_PURPOSE_CD	This parameter stores the name of the purpose code that is assigned for the Postpaid Customer Retention purpose.	CRP	Y
PST_SEGMENT_PURPOSE_CD	This parameter stores the name of the purpose code that is defined for the Postpaid Segmentation purpose.	CSP	Y
PST_UPSELL_PURPOSE_AGG_VAR_NM	For the postpaid up-sell purpose, this parameter defines the variable name that is used to aggregate the scores from subscription level to customer level.	S_SUM_PSU_URVAT_L6M	Y
PST_UP_SELL_PURPOSE_CD	This parameter stores the name of the purpose code that is defined for the Prepaid Up-Sell purpose.	USP	Y
SEQ_TRANSPOSE_FLAG_VALUE	This parameter stores the value of transpose flag that is available in the output table. This table is generated by the association node in SAS Enterprise Miner that is used for the sequence analysis purpose.	1	N
SETTOP_CROSS_SELL_AVG_VAR_NM	This parameter is applicable for the cross-sell TV or up-sell TV purpose. It defines the variable name that is used for aggregating the subscription-level scores to customer-level scores.	D_SUM_IPU_IPUSG_L6M	Y
SETTOP_CROSS_SELL_PURPOSE_CD	This parameter contains the name of the cross-sell TV purpose code	XSS	Y

Parameter Name	Description	Default Value	Can Be Edited
SETTOP_LIFETIME_PURPOSE_CD	This parameter contains the name of the Customer Lifetime TV purpose code.	CLS	Y
SETTOP_OFFER_SEGMENT_CD_VALS	This macro variable drives the values of offer segment codes that identify an offer segment for TV.	"SETTOP"	Y
SETTOP_OFRSRV_MBA_PURPOSE_CD	This parameter contains the name of the offer-level or service-level code for the Association rules TV offers and services purpose.	ARS	Y
SETTOP_RETENTION_AVG_VAR_NM	This parameter is applicable for the Churn TV purpose. It defines the variable name that is used for aggregating the subscription-level scores to customer-level scores.	D_SUM_IPU_IPUSG_L6M	Y
SETTOP_RETENTION_PURPOSE_CD	This parameter contains the name of the Churn TV purpose code.	CRS	Y
SETTOP_SEGMENT_PURPOSE_CD	This parameter contains the name of the Segmentation TV purpose code.	CSS	Y
SETTOP_TLITM_MBA_PURPOSE_CD	This parameter contains the name of the TV telecast item-level association rule purpose code.	ARV	Y
SETTOP_UP_SELL_AVG_VAR_NM	This parameter is applicable for TV Up-sell purpose. It defines the variable name that is used for aggregating the subscription-level scores to customer-level scores.	D_SUM_IPU_IPUSG_L6M	Y
SETTOP_UP_SELL_PURPOSE_CD	This parameter contains the name of the Up-sell TV purpose code.	USS	Y
SETTOP_PPV_IND_VALS	This parameter contains the value for the Pay per view (PPV) indicator.	1	Y
SETTOP_VOD_IND_VALS	This parameter contains the value for the Video on demand (VOD) indicator.	1	Y

Parameter Name	Description	Default Value	Can Be Edited
SUBSCRP_ACT_STATUS_ VALS	This column is a macro variable. It determines the value of subscription status code that indicates an active status of the subscription.	"SS_1"	Y

The following table lists the common parameters that are defined in the PARAMETER_MASTER table and the default value that is set for each of these parameters in the PARAMETER_VALUE_DTL table. You can change the default value for SAS Customer Analytics for Communications if the parameter can be edited. A parameter that can be edited is marked with a **Y** in the **Can Be Edited** column.

Table A1.2 *Common Parameters*

Parameter Name	Description	Default Value	Can Be Edited
APPLY_NULL_CONDTN_ FOR_CPX_VAR	This column is used as a macro variable. It determines the value that a complex variable returns if any of the variables of the expression that is defined for a complex variable has a NULL value. If you set the value of this parameter to Y, then the complex derived variable returns a NULL value if any of the variables of the expression has a NULL value.	N	Y
CAB_APPL_DEBUG_FLG	This parameter sets the value of the flag that debugs the SAS code. If you do not want to debug the code, then set the value to N. This flag is mostly used to debug a macro that is called as an action in the application. These macros are mostly related to some processing other than the batch processing. When you set the value to N, the temporary table is deleted at the end of process. This value also indicates that separate logs are not created in the log files. However, these logs are diverted to the log file of JBoss or WebLogic. It is recommended that you set the value of this parameter to N.	Y	Y

Parameter Name	Description	Default Value	Can Be Edited
CASE_CONDN_CHR_DATA_TYPE	This parameter identifies whether the column of the table that is being imported is of Character data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(kupcase(TYPE)='CHAR')	Y
CASE_CONDN_DATE_COLUMN_TYPE	This parameter identifies whether the column of the table that is being imported is of Date or Datetime data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(&CASE_CONDN_DATE_DATA_TYPE or &CASE_CONDN_DTTM_DATA_TYPE)	Y
CASE_CONDN_DATE_DATA_TYPE	This parameter identifies whether the column of the table that is being imported is of Date data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(1 = 0)	Y
CASE_CONDN_DIM_COLUMN_TYPE	This parameter identifies whether the column of the table that is being imported is of Dimensional Attribute data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(kupcase(TYPE)='CHAR' and SUBSTR(NAME ,length(NAME)-2,3) not in ('_RK', '_SK', '_ID'))	Y
CASE_CONDN_DTTM_DATA_TYPE	This parameter identifies whether the column of the table that is being imported is of datetime data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(kupcase(TYPE) ='NUM' and ((SUBSTR(NAME ,length(NAME)-2,3)='_DT' OR SUBSTR(NAME ,length(NAME)-4,5)='_DTTM') OR (SUBSTR(FORMAT,1,4) = 'DATE' OR SUBSTR(FORMAT,1,8) = 'DATETIME') OR SUBSTR(FORMAT,1,6) IN ('NLDATE','NLDATM')))	Y

Parameter Name	Description	Default Value	Can Be Edited
CASE_CONDN_KEY_ COLUMN_TYPE	This parameter identifies whether the column of the table that is being imported is of Key (retained, surrogate, or foreign) data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	((kupcase(TYPE)='NUM' or kupcase(TYPE)='CHAR') and SUBSTR(NAME ,length(NAME)-2,3) in ('_RK', '_SK', '_ID'))	Y
CASE_CONDN_MSR_ COLUMN_TYPE	This parameter identifies whether the column of the table that is being imported is of measure data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(kupcase(TYPE)='NUM' and (SUBSTR(NAME ,length(NAME)-2,3) not in ('_RK', '_SK', '_ID', '_DT') or SUBSTR(NAME ,length(NAME)-4,5) not in ('_DTTM')))	Y
CASE_CONDN_NUM_ DATA_TYPE	This parameter identifies whether the column of the table that is being imported is of Numeric data type. It contains the condition for validating the data type. This condition is based on the column of dictionary.columns table.	(kupcase(TYPE)='NUM')	Y
DABT_ACT_CALC_ OUTCM_END_ONLY_ FLG	This parameter indicates the default value of whether to perform 'Calculate Actual result calculation process only at the end of Outcome Period. It has possible values of Y or N. If you specify Y, then Actual result is calculated and populated only at end of outcome period for a given scoring run. This value is the default value in the deployed job for actual calculation.	Y	Y
DABT_ACTUAL_CALC_ ABT_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y
DABT_ACTUAL_CALC_ SCR_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y

Parameter Name	Description	Default Value	Can Be Edited
DABT_AGG_WHILE_EXTR_FOR_BEH_FLG	This parameter indicates whether to compute aggregations that are required for behavioral variable at the time of extraction from source data or first extract the required data from source table and then aggregate the data.	Y	Y
DABT_APPLY_TRYLOCK_ACT_ARM_FLG	This parameter decides whether to apply try-lock while populating the Actual Values ARM. It has possible values of Y or N. It should have value of Y if ARM tables are SAS data sets.	N	Y
DABT_APPLY_TRYLOCK_SCR_ARM_FLG	This parameter decides whether to apply try-lock while populating the Scoring ARM. It has possible values of Y or N. Its value should be Y if ARM tables are SAS data sets.	N	Y
DABT_ARM_TIMEOUT_IN_SECONDS	This parameter signifies the total number of seconds for which try lock will be tried before showing the time-out error. This is applicable while populating ARM if DABT_APPLY_TRYLOCK_SCR_ARM_FLG or DABT_APPLY_TRYLOCK_ACT_ARM_FLG are set to Y.	600	Y
DABT_ARM_WAIT_TIME_BETN_TRYLOCK	This parameter signifies the amount of time in seconds that the macro should wait in between each attempt to lock the ARM table. This is applicable while populating ARM if DABT_APPLY_TRYLOCK_SCR_ARM_FLG or DABT_APPLY_TRYLOCK_ACT_ARM_FLG are set to Y.	1	Y

Parameter Name	Description	Default Value	Can Be Edited
DABT_BACKTESTING_ ABT_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y
DABT_BACKTESTING_ SCR_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y
DABT_DEBUG_FLG	This parameter sets the value of debug flag for batch processes such as Build Modeling ABT, Back Testing, Scoring, or Actual Run. If you do not want to debug any of the batch processes, then set this parameter to N. This value indicates that the temporary folders and data sets are deleted at the end of the processing. However, logs are still maintained.	Y	Y
DABT_INDB_OPT_PRE_ BLD_ABT_JOB	This parameter determines whether the ABT building process will be executed in database other than SAS, such as Terdata. The value of this parameter is set before each ABT is built.	DBIDIRECTEXEC sql_ip_trace=Note	Y
DABT_INDB_OPT_PRE_ SORT	This parameter is set at the start of every sort process if it is supposed to be executed in the database itself.	SORTPGM=BEST SQLGENERATION=DBM S	Y
DABT_INDB_ PROCESSING_FLG	This flag indicates whether the ABT building process is run in the database other than SAS. If the process is run in a database such as Teradata, then the value of the parameter is set to Y.	N	Y
DABT_LOG_CLEAR	This parameter indicates whether a new log file will be created at the beginning of each batch process.	new	N

Parameter Name	Description	Default Value	Can Be Edited
DABT_MAX_ALLOWED_VAR_CLMN_LNGHTH	This parameter specifies the maximum length that is permissible for the column name of an ABT variable. If ABTs are created in Teradata, then the value of this parameter is set to 26. However, if ABTs are created in SAS, then the value of this parameter is set to 28. As a result, 4 extra characters are available if the SAS Enterprise Miner code adds additional characters to the column name of the ABT variable.	26 or 28	Y
DABT_MODELING_ABT_LIBREF	This parameter stores the libref of library, which stores information about the modeling ABT.	cac_mdl	Y
DABT_MODELING_SCR_LIBREF	This parameter stores the libref of library, which stores the information that is generated when the modeling ABT is built. This parameter is used when the source data is in the RDBMS and ABT building processing needs to be run in the database.	cacmdscr	Y
DABT_POOLING_ABT_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y
DABT_POOLING_SCR_LIBREF	This parameter is not applicable for SAS Customer Analytics for Communications.		Y
DABT_POPULATE_ACTUAL_ARM_FLG	This parameter decides whether the deployed actual value calculation job will perform the step of populating the ARM with the scoring result. The possible values are Y or N. If it is set as N, it indicates that the deployed job will perform only the step of preparing the scoring ABT and applying scoring code on it. However, it will not populate the ARM.	Y	Y

Parameter Name	Description	Default Value	Can Be Edited
DABT_POST_INIT_ OPTION_STMT	The SAS statement that is written as value of this parameter is executed at start of execution any macro during application. If you need to debug the macro processing, then you can include the option mprint mlogic symbolgen as value of this parameter.		Y
DABT_RETAIN_APPL_ SCRATCH_FLG	This flag is related to any macro that is called when some action takes place in the application. These macros are related to processing other than batch processing. In a normal scenario, when you do not encounter any problem (that is, you are not debugging), set the value of this parameter to N. The default value is N. This value indicates that any temporary scratch data set is deleted at the end of that process.	Y	Y
DABT_SCORING_ABT_ LIBREF	The libref of library, which stores the information about the Scoring ABT.	cac_scr	Y
DABT_SCORING_SCR_ LIBREF	The libref of library, which stores the scratch information that is generated while building the Scoring ABT. This parameter is used when the source data is in RDBMS and the ABT building processing needs to be run in the database.	cacscscr	Y
export_code_folder_nm	This parameter stores the name of the folder under scoring workspace that contains the files exported for scoring and actual job execution.	scr_act_run_exported_code	N
IM_SUBSET_TABLE_ LIBREF	This parameter stores the name of the libref of the library, which contains dummy subset tables for all subjects of analysis.	CSC_SUB	N

Parameter Name	Description	Default Value	Can Be Edited
libref_dabt_input_mart_control	This parameter contains the name of library that contains the input mart control table in defined format as required by the application. This table defines the latest date until which input mart is successfully loaded. The latest input mart load date is used while scoring and actual run execution to decide the latest date for which scoring or actual jobs can be run.	control	Y
LIST_OF_MODEL_TBL_TO_DEL	List of tables from which the model-related records should be deleted while deleting the information of the captured model. This macro variable (global variable) contains the name of the model-related dataset from which the records related to the current model_sk (of model for deletion) would be deleted.	MODEL_RULE_MASTER	N
log_clear	This macro variable means that a new log file will be created at start of each new DABT application SAS macro processing.	new	N
MCR_DABT_CREATE_EVENT_DATA	This parameter stores the name of macro that is called for event-driven ABT calculations. This macro creates a table that contains the EVENT_KEEP_KEY column and event date for each key. You need not set up this parameter if event-driven calculations are not applicable for your solution.	cac_create_event_data	N
MCR_DABT_RULES_FILTER	This is the name of the solution-specific macro. It is called only in case of Market Basket Analysis type of models. Hence, parameter needs to be set only in that case.	cac_rules_filter	N

Parameter Name	Description	Default Value	Can Be Edited
MCR_DABT_UPDATE_ ACT_RSLT_COLUMN	This is the name of solution-specific macro that updates the CORRSPNDNG_ACT_ RSLT_COLUMN_NM in the scr_model_x_act_outcome_ var .	cac_update_act_rslt_ column	N
MCR_DABT_UPDATE_ SCR_RSLT_COLUMN	This is the name of solution-specific macro that updates the CORRSPNDNG_SCR_ RSLT_COLUMN_NM in the model_scoring_output_ column table.	cac_update_scr_rslt_ column	N
OC_BSD_PST_IMPL_ SBST_VALUE	By default, Modeling or Scoring ABT is applied with Outcome Based Filter (that is, subjects that are already meeting an event under consideration are dropped from the Modeling or Scoring ABT). For example, in Churn ABT, records of customers who have already churned as of the ABT Build Date itself are dropped from the final modeling ABT. This global macro variable OC_BSD_PST_IMPL_ SBST_VALUE specifies the value of temporary outcome variable calculated as of the ABT build date. Only the records with this value will be stored in the final Modeling or Scoring ABT if outcome-based post ABT filter is applicable.	0	Y
SCR_USE_LOCK_SEQ	This parameter decides whether session-specific prefix should be generated by using sequence generator.	N	N

Parameter Name	Description	Default Value	Can Be Edited
table_dabt_input_mart_control	This parameter contains the name of table in which the library that is defined by libref_dabt_input_mart_control resides. This control table should be defined in a format as required by the application. It should have column with name LOAD_DTTM and LOAD_STATUS (Values: Y or N). This table defines the latest date until which Foundation data mart is successfully loaded. The latest Foundation data mart load date is used while scoring and actual run execution to decide the latest date for which scoring or actual jobs can be run.	INPUT_MART_LOAD_DATES	Y
WEEK_END_DAY_NM	In the Build ABT window of the SAS Customer Analytics for Communications interface, for a weekly grain, you can select the last day of a week as the build ABT date. This parameter defines the day, which is considered as the last day of week. The day that you define as the last day of week is the only weekday that a user can select while choosing an ABT build date. Depending on the day that you define, only the corresponding date of each week is available for selecting the ABT build date. For example, you define the last day of week as Saturday. In this case, for the month of July 2012, the following dates are available for selecting the ABT build date: July 7, July 14, July 21, and July 28.	SATURDAY	Y

Appendix 2
Predefined Subset Maps

The following table lists the predefined subset maps that SAS Customer Analytics for Communications provides.

Table A2.1 *Predefined Subset Maps*

Subset Map	Purpose	Source Table
All Prospects	Enables you to select only prospective customers in your target population. You must use this subset map when you want to build an analytical model for customer acquisition.	PROSPECT_CUST_D
Customer Offer Selection Postpaid	Enables you to create the modeling ABT for association rules analysis. You should select this subset map to analyze behavior of postpaid customers with reference to agreement dates of offers. In addition, after you choose this subset map, you can define filter conditions based on various offer-level and customer-level attributes. The target population contains customers who satisfy the subset criteria. Therefore, the ABT would contain records of postpaid customers for the offers that they signed up for at different points of time.	• CUST_D • PST_PD_CUST_OFFER_SNPSHT_B • OFFER_D
Customer Offer Selection Prepaid	Enables you to create the modeling ABT for association rules analysis. You should select this subset map to analyze behavior of prepaid customers with reference to agreement dates of offers. In addition, after you choose this subset map, you can define filter conditions based on various offer-level and customer-level attributes. The target population contains the customers who satisfy the subset criteria. Therefore, the ABT would contain records of prepaid customers for the offers that they signed up for at different points of time.	• CUST_D • PRE_PD_CUST_OFFER_SNPSHT_B • OFFER_D

Subset Map	Purpose	Source Table
Customer Service Selection	Enables you to create the modeling ABT for association rules analysis for postpaid or prepaid customers. You should select this subset map to analyze behavior of postpaid or prepaid customers with reference to the first activation of services. In addition, after you choose this subset map, you can define filter conditions that are based on various service-level and customer-level attributes. The target population contains the subscriptions that satisfy the subset criteria. Therefore, the ABT would contain records of postpaid or prepaid customers for the services that are activated by these customers at different points of time.	• CUST_D • SUBSCRP_SERVICE_ ACTIVITY_F • SERVICE_D
Subscription Service Selection	Enables you to create the modeling ABT for association rules analysis for postpaid or prepaid subscriptions. You should select this subset map to analyze behavior of postpaid or prepaid subscriptions with reference to the first activation of services. In addition, after you choose this subset map, you can define filter conditions that are based on various service-level and subscription-level attributes. The target population contains the subscriptions that satisfy the subset criteria. Therefore, the ABT would contain records of postpaid or prepaid subscriptions for the services that are activated by these customers at different points of time.	• SUBSCRP_D • SUBSCRP_SERVICE_ ACTIVITY_F • SERVICE_D
Rule-Based Customer Selection	Enables you to select customer population based on output of association rules analysis, which is rules. You should select this subset map to create a cross-sell or an up-sell model that is based on the association rules analysis models that you have created. In addition, after you choose this subset map, you can create filters based on various attributes of rules such as rule text, rule unit type (left hand or right hand of rule), and the ID values of services or offers, which are part of the rule.	• ANALYTICAL_RULE_DTL • CUST_RULE_SCORE_DTL • ANALYTICAL_MODEL_DTL

Subset Map	Purpose	Source Table
Rule-Based Subscription Selection	Enables you to select subscription population based on output of association rules analysis, which is rules. You should select this subset map to create a cross-sell or an up-sell model that is based on the association rules analysis models that you have created. In addition, after you choose this subset map, you can create filters based on various attributes of rules such as rule text, rule unit type (left hand or right hand of rule), and the ID values of services or offers, which are part of the rule.	• ANALYTICAL_RULE_DTL • SUBSCRP_RULE_SCORE_DTL • ANALYTICAL_MODEL_DTL
TV Customers or Subscriptions and Services or Offers	Enables you to create the modeling ABT for any of the TV-related purposes. You should select this subset map to analyze behavior of TV customers or TV subscriptions. For churn, segmentation, cross-sell, and up-sell purposes, this subset map extracts data at customer or subscription level. If you want to build ABT for association rules analysis for customers or subscriptions for service-related transactions, this subset map extracts data with reference to activation date of services. However, if you want to build ABT for association rules analysis for customers or subscriptions for offer-related transactions, this subset map extracts data with reference to start date of offers. In addition, when you choose this subset map, you can define filter conditions that are based on various service-level, offer-level, subscription-level, and customer-level attributes. The target population contains TV subscriptions or TV customers that satisfy the subset criteria.	• OFFER_D • SUBSCRP_X_OFFER_BRIDGE • SUBSCRP_D • CUST_D • SUBSCRP_SERVICE_ACTIVITY_F • SERVICE_D

Subset Map	Purpose	Source Table
TV Customers or Subscriptions and Telecast Items	Enables you to create the modeling ABT for association rules analysis for various telecast items . You should select this subset map to analyze behavior of TV subscriptions with reference to the order date for telecast items. In addition, after you choose this subset map, you can define filter conditions based on various channel-level and subscription-level attributes. If you want to analyze telecast items of specific types such as pay per view (PPV) or view on demand (VOD), then you should manually enter these filter conditions while creating the subsets by using the SAS Customer Analytics for Communications interface. The target population contains the subscriptions that satisfy the subset criteria. Therefore, the ABT would contain records of TV subscriptions for the telecast items that are purchased by these subscriptions at different points of time.	CSCOM_SUBSCRP_TLCSTITM_SBST_IM
Subscriptions or Customers Level Selection	Enables you to select customers or subscriptions with specific attributes. For example, you might want the target population to contain customers who have only mobile subscriptions. To do so, when you define a subset criterion, select this subset map. When you define the filter condition, select the **Subscription Type** variable, = (equal to) as the operator, and **Postpaid Mobile** as the value of the variable. *Note:* You have to configure the dimensional attribute values such as Prepaid Mobile and Postpaid Mobile for the **Subscription Type** column of the Subscription Dimension table. For details, see *SAS Customer Analytics for Communications: User's Guide*.	• SUBSCRP_D • CUST_AGRMNT_D • OFFER_BUNDLE_D
Customers with Active Status	Selects customers whose status is active. *Note:* This subset map is used internally. Therefore, make sure that you do not select this subset map when you define a subset criterion for a project that you have created.	CUST_D

Subset Map	Purpose	Source Table
Subscriptions with Active Status	Selects subscriptions whose status is active. *Note:* This subset map is used internally. Therefore, make sure that you do not select this subset map when you define a subset criterion for a project that you have created.	SUBSCRP_D
Customers or Subscription for BG	Selects customers or subscriptions that belong to a specific business group (BG). You can use this subset map if you have installed SAS Offer Optimization for Communications along with SAS Customer Analytics for Communications.	• SUBSCRP_D • CUST_X_BUSINESS_GROUP_BRIDGE

Appendix 3
Editing Segment Details

When you capture a segmentation model (on the Models page in the Projects workspace), you specify the number of segments in the Capture Model Information window. When you publish the model for scoring, the model creates the specified number of segments and inserts a record in the SCORING_MODEL_SEGMENT_MASTER table (available in the CSCAPDM library) for each segment. Each segment is automatically assigned a default name, short name, and description. The automatically assigned short names and descriptions are in the format of Segment_1, Segment_2, Segment_3, and so on.

The short names and descriptions of the segments appear in the segmentation-related business intelligence reports. Therefore, you might want to change these automatically assigned values and assign the segments more meaningful short names and descriptions. However, you must do this after you publish the segmentation model for scoring and before you populate the Analytical results area with these segments.

The scores are written back into the fact tables and the dimension tables of the Foundation data mart from the Analytical results area. This is when the records in the SCORING_MODEL_SEGMENT_MASTER table are used to populate the ANALYTICAL_SGMT_DTL Application data tables. These records are used to generate segmentation-related business intelligence reports. Typically, a business analyst provides these meaningful short names and descriptions based on the characteristics of each segment. As an administrator, you must manually edit the SCORING_MODEL_SEGMENT_MASTER table and update the values in the SEGMENT_SHORT_NAME and SEGMENT_DESC columns for the desired segments.

Note: When you edit the SCORING_MODEL_SEGMENT_MASTER table, ensure that no other users are performing any tasks that insert or extract data from the SCORING_MODEL_SEGMENT_MASTER table. These tasks include publishing a segmentation model for scoring and populating the Application data.

The following table shows the SCORING_MODEL_SEGMENT_MASTER table with modified short names and descriptions sample records:

Table A3.1 *SCORING_MODEL_SEGMENT_MASTER Table*

SCORING_ MODEL_ SEGMENT_SK	SCORING_ MODEL_SK	SEGMENT_NM	SEGMENT_ SHORT_NM	SEGMENT_ DESC
100000001	10000004	1	All-around callers	Good, evenly spread representation of all call types and lengths of calls

SCORING_ MODEL_ SEGMENT_SK	SCORING_ MODEL_SK	SEGMENT_NM	SEGMENT_ SHORT_NM	SEGMENT_ DESC
100000002	10000004	2	National callers	Small number of calls almost entirely national calls
100000003	10000004	3	Heavy International users	Lots of calls with good proportion of international calls
100000004	10000004	4	Low users	Only few, very short calls

Appendix 4
Load Order Sequence

Scheduling ETL Jobs

The tables that are listed below provide information about the ETL flows and the dependencies that exist between various jobs of a group. This information will help you schedule the ETL jobs. However, consider the following instructions before you schedule ETL jobs.

- Consider the order in which the jobs of a particular job group are to be run. The Group Order column represents the parent group, and the Level Within Group column indicates the sublevels within each group.
- Execute the jobs in one group only after all jobs of previous groups are executed. For example, the jobs in group 2 should be started only when all the jobs in group 1 are executed.
- Within a group, jobs should be run based on the value in the Level within Group column. For example, there can be two jobs within a group that have level number 1 and level number 2. In this case, Job 2 should be run only after Job 1 is complete.
- Within a group, jobs that have the same level number can be run in parallel.
- The bill-monthly and weekly jobs can be run in parallel.

Load Order Sequence for Bill-Monthly Jobs

Table A4.1 *Load Order for Bill-Monthly Jobs*

Group Order	Level within Group	Job Name
1	1	cfd_pst_pd_cust_acct_snpsht_b_job
1	1	cfd_pst_pd_payment_drvd_b_job
1	1	cfd_pst_pd_cust_offer_snpsht_b_job
1	1	cfd_pst_pd_cust_snpsht_b_job
1	1	cfd_usage_monthly_summary_b_job

Load Order Sequence for Weekly Jobs

Table A4.2 *Load Order for Weekly Jobs*

Group Order	Level within Group	Job Name
1	1	cfd_pre_pd_subscrp_bucket_drvd_b_job
1	1	cfd_pre_pd_subscrp_usage_drvd_b_job
1	1	cfd_pre_pd_cust_offer_snpsht_b_job
1	1	cfd_pre_pd_cust_snpsht_b_job
1	1	cfd_usage_weekly_summary_b_job

Load Order Sequence for Writeback Jobs

Table A4.3 *Load Order for Writeback Jobs for SAS Database*

Group Order	Level within Group	Job Name
1	1	cac_model_writeback_job
1	2	cac_cust_analytical_sgmt_dtl_job
1	3	cac_analytical_rule_master_dtl_job
1	4	cac_analytical_rule_dtl_job
1	5	cac_subscrp_rule_score_dtl_job
1	6	cac_cust_rule_score_dtl_job
1	7	cac_prspct_cust_model_score_dtl_job
1	8	cac_subscrp_model_score_dtl_job
1	9	cac_cust_model_score_dtl_job

Table A4.4 *Load Order for Writeback Jobs for Databases Other than SAS*

Group Order	Level within Group	Job Name
1	1	cac_model_writeback_job
1	2	cac_cust_analytical_sgmt_dtl_job
2	1	cac_analytical_rule_master_dtl_job

Group Order	Level within Group	Job Name
2	2	cac_analytical_rule_dtl_job
3	1	cac_subscrp_rule_score_dtl_job
3	1	cac_cust_rule_score_dtl_job
3	1	cac_prspct_cust_model_score_dtl_job
3	1	cac_subscrp_model_score_dtl_job
4	1	cac_cust_model_score_dtl_job

Appendix 5
Troubleshooting

Troubleshooting General Errors and Warnings

If you encounter any execution errors while working in the SAS Customer Analytics for Communications interface, you need to get more details about these errors in order to debug them. These details are stored in the relevant log files and SAS data sets.

This section describes the parameters that control the generation of log files and temporary data sets during the various actions that are performed in the application. These parameter values are configured in the PARAMETER_VALUE_DTL table of the Application data.

CAB_APPL_DEBUG_FLG
: This parameter controls whether you want to generate log files for the application code. Possible values are Y or N. By default, the value is set to N. If you want to turn the debugging feature on, set the value to Y. When set to Y, the log files are generated for the macros that are called during the application usage. These macros are related to processing other than DABT batch processing. When set to N, the log files are not created separately in .log files; they are diverted to JBoss or WebLogic logs. Current logic requires that this parameter should be set to N for proper exception handling.

DABT_RETAIN_APPL_SCRATCH_FLG
: This parameter controls whether you want to retain the temporary data sets that are generated when macros are called during the application usage. These macros are related to processing other than DABT batch processing. If you are not getting any errors, it is recommended that you let the value of this parameter set to N (default). This means that temporary data sets are deleted at the end of the process.

DABT_DEBUG_FLG
: This parameter sets the value of debug flag for batch processes such as Build Modeling ABT, Back Testing, Scoring, or Actual Run. If you do not want to debug any of the batch processes, then set this parameter to N. This value indicates that the temporary folders and data sets are deleted at the end of the processing. However, logs are still maintained.

Troubleshooting a Modeling ABT Building Process

This section describes the locations of the log files and the temporary data sets that are generated during a modeling ABT building process. It also gives details about the various log files. If a modeling ABT building process fails, you can use the information in these log files to troubleshoot the issue. Search these files for text containing ERROR, WARNING, or Syntax to get to the preliminary analysis of the problem.

If the Foundation data mart is in SAS (the source data is in SAS), during the ABT building process, the temporary data sets are created in the following folder:

```
<Project path>/<Project ID>/build_modeling_abt/<ABT table
name>/scratch/abt_scratch/<ABT build date>
```

If the Foundation data mart is in Teradata, the temporary data sets are created in the schema that is specified in the DABT_MODELING_SCR_LIBREF parameter. This value of this parameter is configured in the PARAMETER_VALUE_DTL table of the Application data.

The log files are created in the following folder:

```
<Project path>/<Project ID>/build_modeling_abt/<ABT table
name>/log
```

<Project path> represents the repository for the Projects workspace. For example, on a Windows computer, this location can be **`C:/SAS/Config/Lev1/AppData/SASCustAnalyticsCommServer/5.4/project`**. In this location, there is a folder for each project, with the corresponding project ID as the folder name. This folder contains the temporary data sets and the logs for all the activities that are performed in that project.

The project path is defined in the cacsrvc.dabt.projectpath property of the Cust Analytics Comm Server 5.4 software component. For details, see “SAS Customer Analytics for Communications Server” on page 96.

<Project ID> is a unique number (PROJECT_SK) that is automatically assigned to a project when it is created.

<ABT table name> is the **Table Name** that you specify for an ABT when you create the ABT.

For example, for a project with project ID 1000008 and ABT table name ACC_RTN_ABT, the location of the log files (on a Windows computer) can be:

```
C:/SAS/Config/Lev1/AppData/SASCustAnalyticsCommServer/5.4/
project/
```

```
1000008/build_modeling_abt/ACC_RTN_ABT/log
```

The following table lists the various log files and the details that the log files contain:

Table A5.1 *Log Files: Modeling ABT Building Process*

Log File	Description
export_mdl_abt_wrapper_<ABT_SK>.log	ABT_SK is a unique number that is automatically assigned to an ABT when it is created. This log file contains the logs for the initial and the final steps in the modeling ABT building process. These steps include reading the ABT metadata from the application mart, transforming it into the required format, and then storing it in the staging area.
build_mdl_abt_wrapper_<ABT_SK>.log	Contains the logs that are generated when the SQL statements that are returned by an information map are stored in files, in the required format. It also contains the logs that are generated when the application data is updated to reflect the status of the ABT building process.
masterloop_<ABT_SK>.log	Contains the logs that are generated during the process when the master loop job invokes the other associated jobs for execution.

For a standard ABT, the **`<Project path>/<Project ID>/build_modeling_abt/<ABT table name>/log`** folder also contain a folder corresponding to each ABT build date. For example, if you built the ABT as of **31 August 2010**, there is a folder named 31082010. A folder corresponding to an ABT build date contains a log file for each job that is invoked and run during the ABT building process as of that date.

For a stacked ABT, the **`<Project path>/<Project ID>/build_modeling_abt/<ABT table name>/log`** folder contains a folder for each stacked date specified during an ABT building process. A folder corresponding to a stacked date contains a log file for each job that is invoked and run during the ABT building process as of that date.

The name of the log file is determined by the JOB_ID column of the PARAMETER_LIST_DABT table. This table contains the list of various jobs that are run during the modeling ABT building process. A log file is created for each job that is available in the PARAMETER_LIST_DABT table. These jobs are run internally while building an ABT and are invoked in parallel by SAS Data Integration Studio jobs.

If the value of the DABT_DEBUG_FLG is set to Y, the PARAMETER_LIST_DABT table is created in the **`<Project path>/<Project ID>/build_modeling_abt/<ABT table name>/stg`** folder. If the value of the DABT_DEBUG_FLG is set to N, the PARAMETER_LIST_DABT table is created in the **`<Project path>/<Project ID>/build_modeling_abt/<ABT table name>/log`** folder with the name LOG_COPY_PARAMETER_LIST_DABT. The original PARAMETER_LIST_DABT table is then deleted from the **`<Project path>/<Project ID>/build_modeling_abt/<ABT table name>/stg`** folder.

This table also contains the job start time, end time, and total time (in seconds). The RETURN_CD column of the PARAMETER_LIST_DABT table indicates the status of a job. A value of 0 indicates that the job ran successfully. A value of 1 to 4 indicates a warning. A value greater than 4 indicates an error.

Troubleshooting the Scoring Process

This section describes the locations of the log files and the temporary data sets that are generated during the scoring processes (that is, during the execution of scoring jobs). It also gives details about the various log files. If the execution of a scoring job fails, you can use the information in these log files to troubleshoot the problem. Search these files for text containing ERROR, WARNING, or Syntax to get to the preliminary analysis of the problem.

If the Foundation data mart is in SAS, during the scoring process, the temporary data sets are created in the following folder:

```
<Scoring path>/<Scoring template
ID>/scoring_run/scratch/abt_scratch/<ABT build date>
```

If the Foundation data mart is in Teradata, the temporary data sets are created in the schema that is specified in the DABT_SCORING_SCR_LIBREF parameter. This value of this parameter is configured in the PARAMETER_VALUE_DTL table of the Application data.

The log files that are generated while creating a scoring ABT are stored in the following folder:

```
<Scoring path>/<Scoring template ID>/scoring_run/log
```

<Scoring path> represents the repository for the Scoring workspace. For example, on a Windows computer, this location can be **C:/SAS/Config/Lev1/AppData/SASCustAnalyticsCommServer/5.4/scoring**. In this location, there is a folder for each scoring template, with the corresponding scoring template ID as the folder name. This folder contains the temporary data sets and the logs for all the scoring activities corresponding to that scoring template.

The scoring path is defined in the cacsrvc.dabt.scoringpath property of the Cust Analytics Comm Server 5.4 software component. For details, see "SAS Customer Analytics for Communications Server" on page 96.

<Scoring template ID> is a unique number (SCORING_TEMPLATE_SK) that is automatically assigned to a scoring template when it is created.

For example, for a scoring template with scoring template ID 10000005, the location of the log files (on a Windows computer) can be:

C:/SAS/Config/Lev1/AppData

/SASCustAnalyticsCommServer/5.4/scoring/

10000005/scoring_run/log.

The following table lists the various log files and the details that the log files contain:

Table A5.2 *Log Files: Scoring Process*

Log File	Description
scoring_run_job_wrapper_<SCORING_TEMPLATE_ID>.log	Contains the logs for the initial and the final steps in the scoring ABT building process. These steps include the following: • Read the input mart load date and the last scoring date (the date as of which scoring was last performed). • Import the required metadata from the scoring and actual calculation jobs. • Initiate the ABT building process. • Apply scoring code on the scoring ABT. • Write the scoring results to the Analytical results area.
masterloop_<SCORING_TEMPLATE_ID>.log	Contains the logs that are generated during the process when the master loop job invokes the other associated jobs for execution.
populate_scoring_arm_job_<SCORING_TEMPLATE_ID>.log	Contains the logs that are generated when the scores are written back into the Analytical results area. *Note:* This log file is generated only when the scores are written back into the Analytical results area through the dabt_populate_score_result job, and the value of the m_divert_log_to_new_file parameter is set to Y.

The **`<Scoring path>/<Scoring template ID>/scoring_run/log`** folder also contains a folder corresponding to each scoring date. For example, for a scoring date of **30 September 2010**, there is a folder named 30092010. This folder contains a log file for each job that is invoked and run during the scoring ABT building process.

The name of the log file is determined by the JOB_ID column of the PARAMETER_LIST_DABT table. This table contains the list of various jobs that are run during the scoring ABT building process. A log file is created for each job that is available in the PARAMETER_LIST_DABT table. These jobs are run internally while building an ABT and are invoked in parallel by SAS Data Integration Studio jobs.

If the value of the DABT_DEBUG_FLG is set to Y, the PARAMETER_LIST_DABT table is created in the **`<Scoring path>/<Scoring template ID>/scoring_run/stg`** folder. If the value of the DABT_DEBUG_FLG is set to N, the PARAMETER_LIST_DABT table is copied into the **`<Scoring path>/<Scoring template ID>/scoring_run/log`** folder with the name LOG_COPY_PARAMETER_LIST_DABT. The original PARAMETER_LIST_DABT table is then deleted from the **`<Scoring path>/<Scoring template ID>/scoring_run/stg`** folder.

This table also contains the job start time, end time, and total time (in seconds). The RETURN_CD column of the PARAMETER_LIST_DABT table indicates the status of a job. A value of 0 indicates that the job ran successfully. A value of 1 to 4 indicates a warning. A value greater than 4 indicates an error.

Glossary

ABT variable

See analytical base table variable

analytical base table

a highly denormalized data structure that is designed to build an analytical model or to generate scores based on an analytical model. Short form: ABT.

analytical base table variable

a column in an analytical base table that is used to build a statistical model to predict defaults. Short form: ABT variable.

analytical model

a statistical model that is designed to perform a specific task or to predict the probability of a specific event.

churn

the process of losing active customers and their related revenue. Churn can be classified as either voluntary or involuntary, depending on the reason for discontinuing the subscription or service.

churn score

a process that uses analytical data and process models to predict the likelihood of customer churn. The churn models are developed based on data from account, client, household, subscription, and equipment information. The

churner

a subscriber that involuntarily or voluntarily disconnects a subscription.

cube

See OLAP cube

data mart

a subset of the data in a data warehouse. A data mart is optimized for a specific set of users who need a particular set of queries and reports.

dimension

a data element that categorizes values in a data set into non-overlapping categories that can be used to group, filter, and label the data in meaningful ways. Hierarchies within a dimension typically represent different groupings of information that pertains to a single concept. For example, a Time dimension might consist of two hierarchies: (1) Year, Month, and Date, and (2) Year, Week, and Day.

ETL
See extract, transform, load

ETL job
a set of instructions that is used to specify ETL processes that are needed to create output.

extract, transform, load
a data warehousing process in which data is extracted from outside sources, transformed according to operational and quality needs, and loaded into a target database.

fact table
the central table in a star schema or snowflake schema. The fact table contains the individual facts that are being stored in the database as well as the keys that connect each fact to the appropriate value in each dimension.

OLAP
See online analytical processing

OLAP cube
a logical set of data that is organized and structured in a hierarchical, multidimensional arrangement to enable quick analysis of data. A cube includes measures, and it can have numerous dimensions and levels of data.

online analytical processing
a software technology that enables users to dynamically analyze data that is stored in multidimensional database tables (cubes).

Index