

# **SAS<sup>®</sup> Cost and Profitability Management 8.1: Data Administration Guide**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2014. *SAS® Cost and Profitability Management 8.1: Data Administration Guide*. Cary, NC: SAS Institute Inc.

**SAS® Cost and Profitability Management 8.1: Data Administration Guide**

Copyright © 2014, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

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

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

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

**U.S. Government License Rights; Restricted Rights:** The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

April 2017

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

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

P6:abmdref

---

# Contents

## PART 1 Managing Permissions 1

<b>Chapter 1 • User Capabilities and Groups</b>	<b>3</b>
Overview	3
Creating Roles	4
Creating Groups	12
Creating Users	17
Sample Allocation of Capabilities	21
Table of Capabilities and Model Access	26
<b>Chapter 2 • How To Manage Permissions</b>	<b>35</b>
Create a Role	35
Create a Group	35
Create Users	35
Change the Owner of an Item	35
Grant Read or Read/Write Access to an Item	36

## PART 2 Surveys 37

<b>Chapter 3 • Introduction to Surveys</b>	<b>39</b>
Types of Surveys	39
Logging On	40
Log On with IWA	41
The Survey Process in a Nutshell	41
<b>Chapter 4 • Exporting Data to Use with Surveys</b>	<b>47</b>
Exporting Survey Data	47
<b>Chapter 5 • Surveys: User Interface</b>	<b>53</b>
Home	53
Survey Manager	53
User Manager	54
<b>Chapter 6 • Creating Surveys</b>	<b>57</b>
Create the Survey	57
Create a Group Survey	62
Create Follow-up Surveys	63
Create an Aggregated Survey	64
<b>Chapter 7 • Taking a Survey</b>	<b>69</b>
Take a Driver Survey	69
Take a Non-Driver Survey	73
Submit a Survey	75
<b>Chapter 8 • Manage Users</b>	<b>77</b>
Overview	77
Reassign a Survey to Another User	77

Add Attributes to a User .....	78
<b>Chapter 9 • Administer Surveys .....</b>	<b>81</b>
Survey Administrator .....	81
Mark a Survey as Incomplete .....	82
Audit Surveys .....	83
View Approved Assignments .....	84
Send an e-mail .....	85
Enable E-mail Notification .....	86
Delete a Survey .....	88
Delete Exported Model Survey Data .....	88
Reassign a Survey .....	89
<b>Chapter 10 • Preferences .....</b>	<b>91</b>
Overview .....	91
Global Preferences .....	92
General Preferences .....	92
Application Preferences .....	93
Model Preferences .....	94
<b>Chapter 11 • Importing Survey Data back into the Model .....</b>	<b>101</b>
Importing Survey Data .....	101
 PART 3   Staging Tables for Import and Export   105	
<b>Chapter 12 • About Staging Tables .....</b>	<b>107</b>
Introduction .....	107
Staging Tables for Importing a New Model .....	108
Periodic Data versus Structural Data .....	108
Staging Table Dependencies .....	109
Naming Conventions .....	109
Reference Conventions .....	114
Data Types .....	115
<b>Chapter 13 • Staging-Table Schemas .....</b>	<b>117</b>
Table Headings .....	118
Account table .....	119
Assignment table .....	122
AssignmentNonUnique table .....	123
CurrencyRate table .....	124
Dimension table .....	127
DimensionAttributeAssociation table .....	128
DimensionLevel table .....	129
DimensionMember table .....	129
DimensionOrder table .....	130
DimMemberDimAttrAssociation .....	130
DimMemberValueAttrAssociation .....	131
Driver table .....	131
EnteredCostElement table .....	132
Model table .....	133
Module table .....	137
Period table .....	137
PeriodLevel table .....	138
RollupAccount table .....	138



Scenario table .....	140
ScenarioLevel table .....	140
ValueAttribute table .....	141
ValueAttributeAssociation table .....	141
ValueAttributePeriodicDef table .....	142

## PART 4 Export Registered Tables 143

### **Chapter 14 • Working with Registered Tables 145**

Overview .....	145
Model-Structure Tables .....	146
Model-Independent Views .....	147
Fact-Table Related Views .....	148
How To Export and Register Tables .....	150

### **Chapter 15 • Registered Table Schemas 153**

Column Descriptions for All Tables .....	154
PUBLICMODEL .....	154
PUBLICMODELSTATUS .....	155
PUBLICPERIOD .....	155
PUBLICSCENARIO .....	156
modelRef_ACCOUNTMAP .....	156
modelRef_ASSIGNMENTMAP .....	159
modelRef_PV_ACCOUNT .....	159
modelRef_PV_ASSIGNMENT .....	162
modelRef_PV_ASSIGNMENTTEXT .....	164
modelRef_PV_ATTRIBUTE .....	165
modelRef_PV_ATTRIBUTEVALUE .....	166
modelRef_PV_DIMENSION .....	167
modelRef_PV_DRIVER .....	167
modelRef_PV_ENTEREDCE .....	168
modelRef_PD_dimShortRef<suffix> .....	169
modelRef_PD_COSTELEMENT<suffix> .....	169
modelRef_PD_DRIVER<suffix> .....	170
modelRef_PD_MODULE<suffix> .....	171
modelRef_PD_PERIOD<suffix> .....	172
modelRef_PD_SCENARIO<suffix> .....	172
modelRef_PD_YESNO<suffix> .....	173
modelRef_PF_<suffix> .....	174

## PART 5 Backup and Restore 177

### **Chapter 16 • Backup and Restore Utility 179**

Overview .....	179
Backup .....	180
Restore .....	181

## PART 6 Using the API 185

<b>Chapter 17 • Programming the API</b>	<b>187</b>
Overview	187
Progress reports	188
Log files	189
Writing a program in Java	190
<b>Chapter 18 • API Functions</b>	<b>191</b>
AcquireModelLock	191
Calculate	192
CancelOperation	192
Connect	193
CopyModelData	194
CopyPasteModel	194
Disconnect	194
ExportCubeConfigurations	195
ExportModelData	195
ExportRegisterTables	196
GenerateCube	196
GetOperationLog	196
GetOperationProgress	197
ImportCubeConfigurations	197
ImportModelData	197
OnOperationComplete	198
PublishPeriodsScenarios	199
ReleaseModelLock	199
TerminateWriteLock	199
<b>Chapter 19 • XML Passed to the API</b>	<b>201</b>
Overview	203
XML to Calculate a Model	203
XML to Copy and Paste a Model	206
XML to Copy Period/Scenario Data	209
XML to Export Cube Configurations	210
XML to Export a Model	211
XML to Export and Register Tables	221
XML to Generate a Cube	224
XML to Import Cube Configurations	227
XML to Import a Model	229
XML to Publish Period/Scenario Associations	242
<b>Chapter 20 • Easy API</b>	<b>245</b>
Using Easy API	245
Example of Using Easy API	251
 PART 7 SAS Visual Analytics	 265
<b>Chapter 21 • Introduction</b>	<b>267</b>
SAS Visual Analytics	267
SAS LASR Analytic Server	267
The Process in a Nutshell	268
<b>Chapter 22 • Working with SAS LASR Analytic Server</b>	<b>273</b>
Give Users Capabilities for SAS Visual Analytics	273
Create a New SAS LASR Analytic Server in Metadata	275

Use an Existing SAS LASR Analytic Server . . . . .	282
Create a New LASR-Enabled Metadata Library . . . . .	287
Use an Existing LASR-Enabled Metadata Library . . . . .	294
Determine the Port Number of a LASR Server . . . . .	295
Authorize Users for the LASR-Enabled Metadata Library . . . . .	296
Authorize Users to Manage LASR . . . . .	299
Start the SAS LASR Analytic Server Using SAS Visual Analytics . . . . .	302
Explore the Data . . . . .	305
<b>Chapter 23 • Sample Reports for SAS Visual Analytics . . . . .</b>	<b>307</b>
Overview . . . . .	308
Step 1 – Import the Models . . . . .	311
Step 2 – Import the Cube Configurations . . . . .	316
Step 3 – Start the SAS LASR Analytic Server . . . . .	319
Step 4 – Export and Register Tables . . . . .	322
Step 5 – Calculate the Models and Generate Cubes . . . . .	327
Step 6 – Import the Sample Reports . . . . .	335
Step 7 – View the Sample Reports . . . . .	342
View the ITBM_Dashboard Report . . . . .	344
View the ITBM_Fact_View Report . . . . .	357
View the ITBM_Account_View Report . . . . .	363
 PART 8 SAS Profitability Management 367	
<b>Chapter 24 • Working with SAS Profitability Management . . . . .</b>	<b>369</b>
Specify the SAS Profitability Management Input Library . . . . .	369
Use Account Data with SAS Profitability Management . . . . .	371
Mark Accounts as Behaviors . . . . .	373
Publish Behaviors to SAS Profitability Management . . . . .	374
Create a SAS Profitability Management Input Library . . . . .	378
Map to the Behavior Table . . . . .	382



## **Part 1**

---

# Managing Permissions

### *Chapter 1*

**User Capabilities and Groups** ..... 3

### *Chapter 2*

**How To Manage Permissions** ..... 35



## Chapter 1

# User Capabilities and Groups

---

<b>Overview</b> .....	<b>3</b>
<b>Creating Roles</b> .....	<b>4</b>
Overview .....	4
Create a Role .....	7
<b>Creating Groups</b> .....	<b>12</b>
Overview .....	12
Create a Group .....	14
Creating Groups versus Creating Roles .....	16
<b>Creating Users</b> .....	<b>17</b>
<b>Sample Allocation of Capabilities</b> .....	<b>21</b>
Groups Created During Installation .....	21
Roles .....	23
Modelers and Business Users .....	23
SAS OLAP Cube Creators .....	24
<b>Table of Capabilities and Model Access</b> .....	<b>26</b>
About the Table of Capabilities and Model Access .....	26
How to Read this Table .....	30

---

## Overview

As an administrator, you create users, roles, and groups. In general, the sequence you will follow is the following:

1. Create roles See [“Creating Roles” on page 4](#).  
A role is a set of capabilities. When you make a group be a member of a role, users who are members of that group inherit the capabilities of the role.
2. Create groups See [“Creating Groups” on page 12](#).  
A group is a set of users who share the same capabilities. It is convenient to create groups before creating users because if groups exist, then when you create a user you can assign the user to one or more groups and thereby determine the user's capabilities.
3. Create users and assign them to groups. See [“Creating Users” on page 17](#).

*Note:* To access SAS Cost and Profitability Management, a user must be a member of the Cost and Profitability Management Users group, either directly or indirectly (for example, by being a member of a group that is a member of the Cost and Profitability Management Users group).

What features of the SAS Cost and Profitability Management client are available to a user is determined by the combination of the following two factors:

- the capabilities that the user inherits from the groups to which the user belongs
- the permissions that are granted to the groups to which the user belongs

For a table that lists all the features of the SAS Cost and Profitability Management client and specifies what features are available to a user based on those two factors, see [“Table of Capabilities and Model Access” on page 26](#).

For examples of allocating capabilities using roles and groups, see [“Sample Allocation of Capabilities” on page 21](#).

---

## Creating Roles

### Overview

A role is a set of capabilities. There are three categories of capabilities associated with SAS Cost and Profitability Management:

**capabilities related to SAS Cost and Profitability Management Surveys**

- Create Surveys
- Take Surveys

**capabilities related to SAS Cost and Profitability Management What-If Analysis**

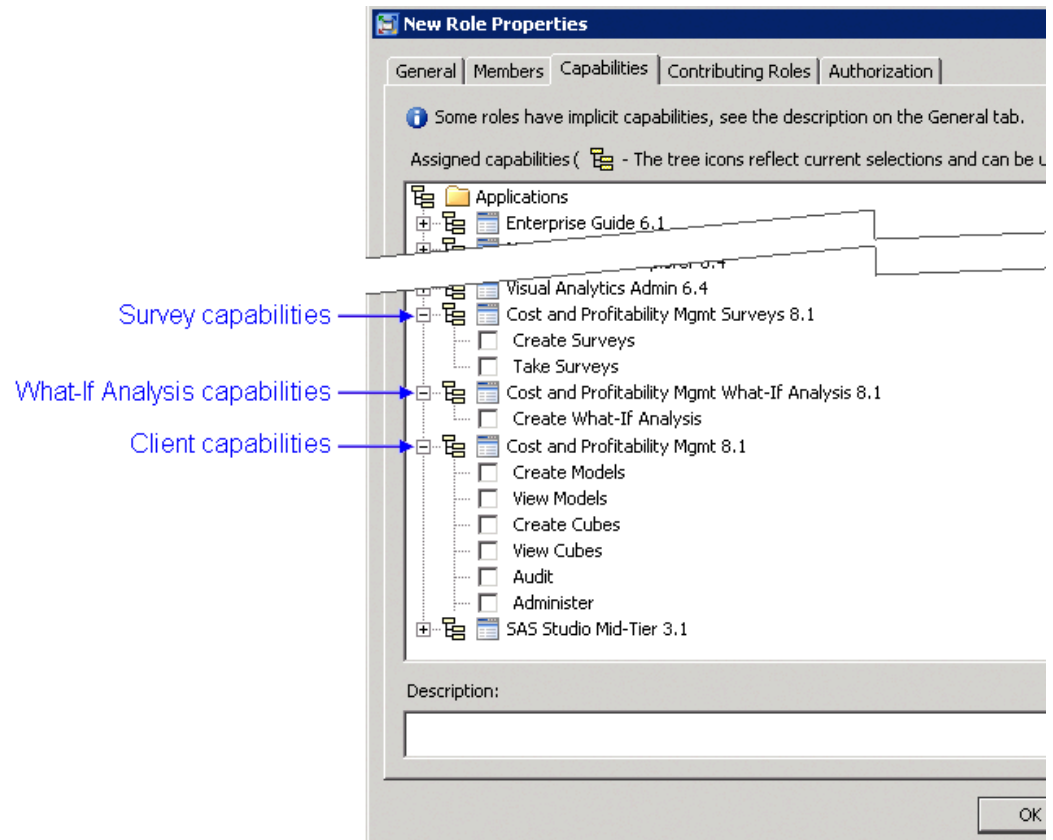
- Create What-If Analysis

**capabilities related to SAS Cost and Profitability Management client**

- Create Models
- View Models
- Create Cubes
- View Cubes
- Audit
- Administer

The following picture shows SAS Management Console displaying the SAS Cost and Profitability Management capabilities that can be assigned to a role.





The following table shows the roles that are automatically created for you on installation of SAS Cost and Profitability Management. The table also shows the capability that each role possesses.

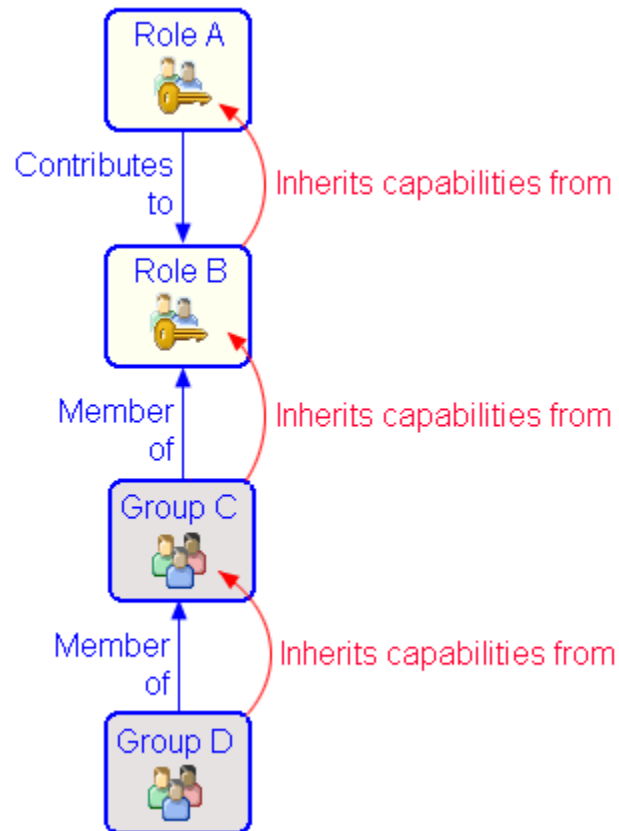
Role	Capabilities
Cost and Profitability Management: Administration	Administer
Cost and Profitability Management: Create Model	Create Models
Cost and Profitability Management: View Models	View Models
Cost and Profitability Management: Create Surveys	Create Surveys See <a href="#">Chapter 6, “Creating Surveys,”</a> on page 57.
Cost and Profitability Management: Take Surveys	Take Surveys See <a href="#">Chapter 7, “Taking a Survey,”</a> on page 69.
Cost and Profitability Management: Create What-If Analysis	Create What-If Analysis See “Create a What-If Analysis” in Chapter 2 of <i>SAS Cost and Profitability Management: What-If Analysis</i> .

As an administrator you create roles with one or more capabilities. Then when you create a group, you make the group be a member of a role so that the group inherits capabilities from the role.

For examples of allocating capabilities using roles and groups, see “[Table of Capabilities and Model Access](#)” on page 26.

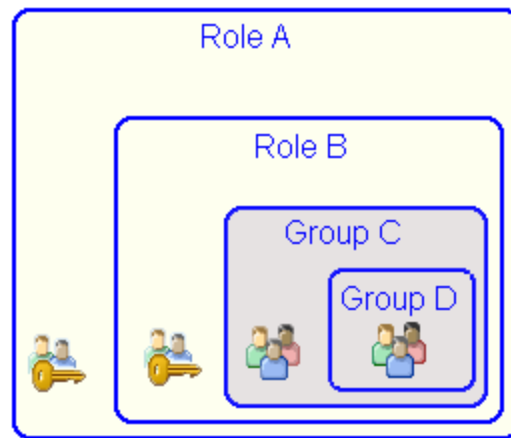
The following picture summarizes the inheritance of capabilities from roles and groups:

- Role B inherits from role A because role A is a contributing role to role B.
- Group C inherits from role B because group C is a member of role B.
- Group D inherits from group C because group D is a member of group C.



*Note:* The relationship inherits from is transitive. That is if C inherits from B, and B inherits from A, then C inherits from A.

Another way to look at the inheritance of capabilities is as one of inclusion as shown in the following picture which (assumes the Contributes to and Member of relationships in the picture above):

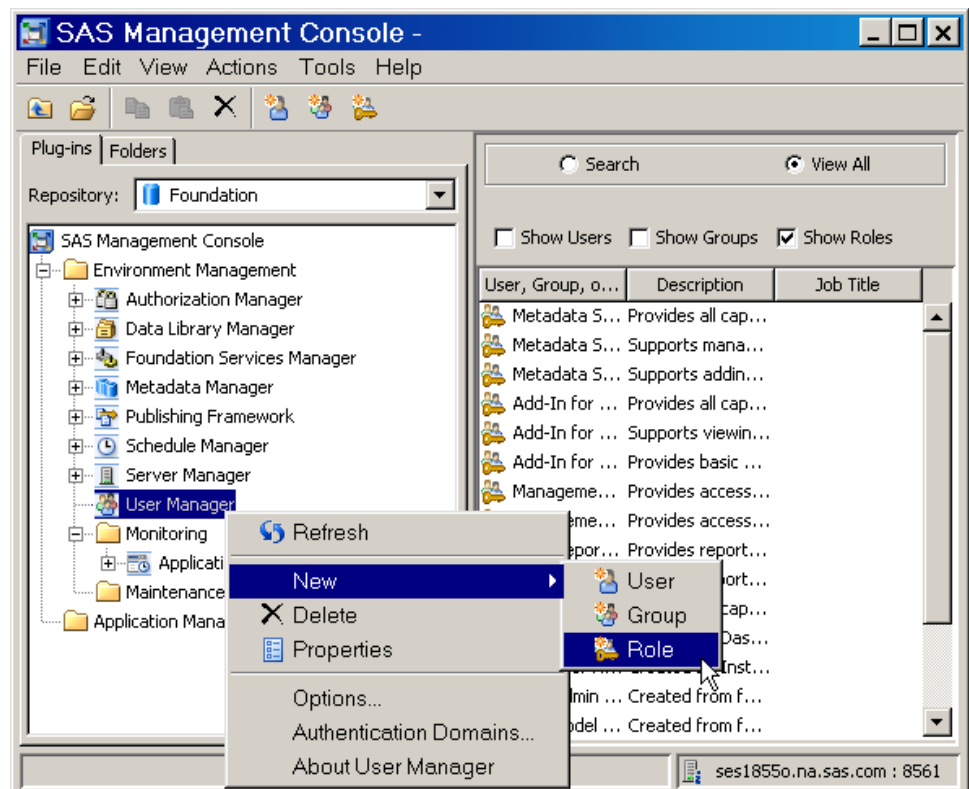


In this picture:

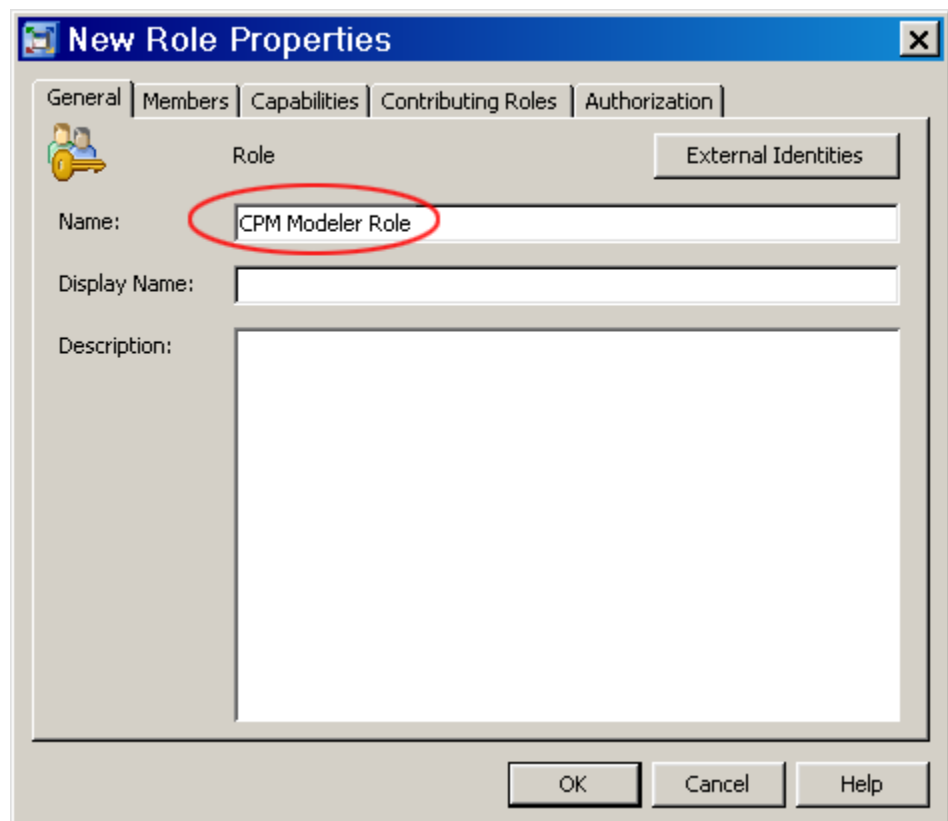
- Users in Role A have only the capabilities of Role A.
- Users in Role B have the capabilities of Role A and Role B.
- Users in Group C have the capabilities of Role A and Role B (and they could have whatever capabilities not pictured that Group C inherits from other roles).
- Users in Group D have the capabilities of Role A and Role B and Group C (and they could have whatever capabilities not pictured that Group D inherits from other roles or groups).

### Create a Role

1. Open SAS Management Console, connecting to your metadata server.
2. Select **User Manager**, and then select **New** ⇒ **Role**.

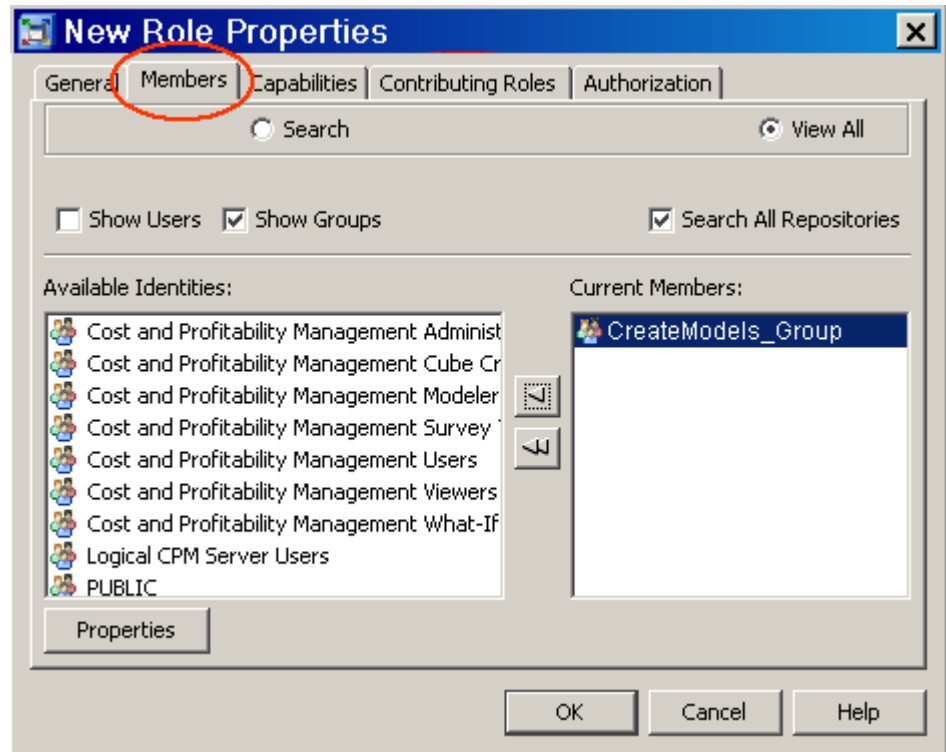


3. Name the role (for example CPM Modeler Role).

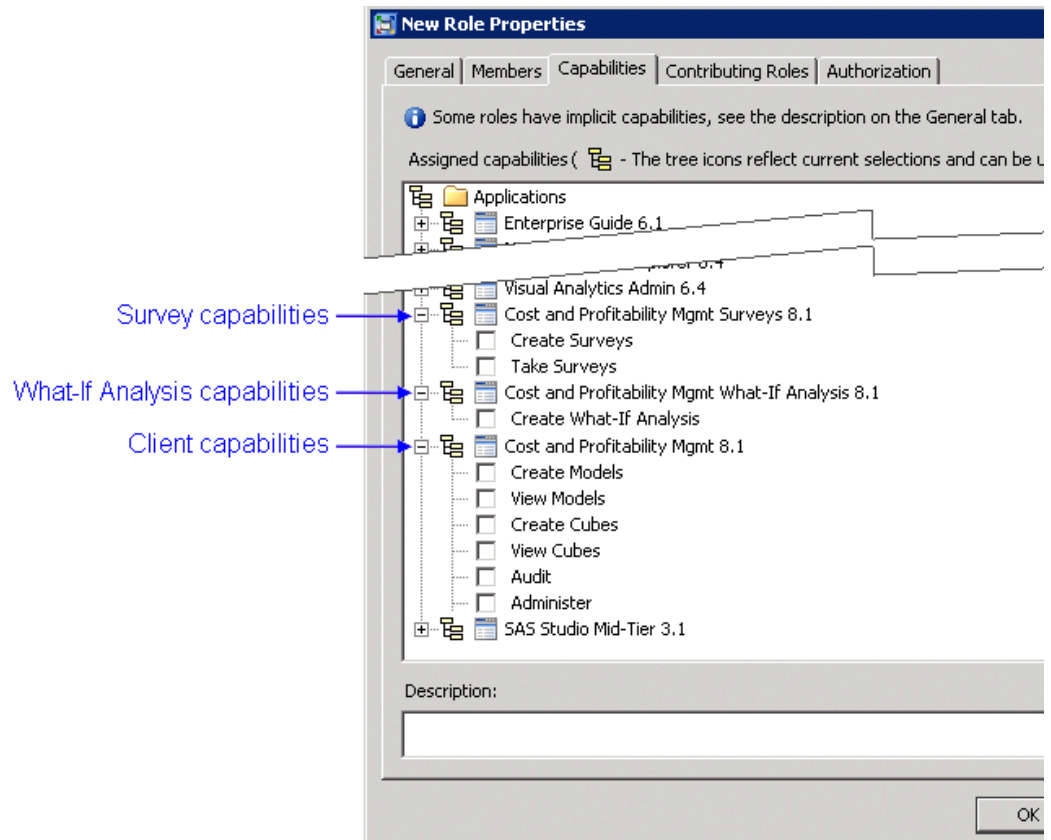


4. Click the **Members** tab and add, as members of this role, the groups that you want to inherit the capabilities of the role.

For example, the following picture shows CreateModels\_Group added as a member of the new role. This causes the CreateModels\_Group to inherit the capabilities of the role being created.





5. Click the **Capabilities** tab and select the capabilities that you want to assign to this role—and indirectly to any group that is a member of this role.

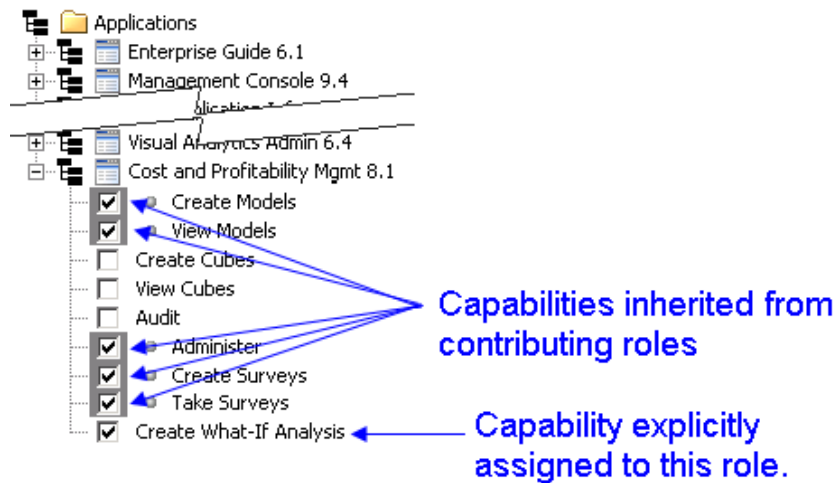


The **Create Models** capability gives full access to a model including model creation and deletion, cube creation and viewing, etc. Plus, it gives users some other abilities not related to a particular model such as creating column layouts and setting up exchange rates. The Create Models capability provides the abilities of a Modeler in previous releases of SAS Activity-Based Management.

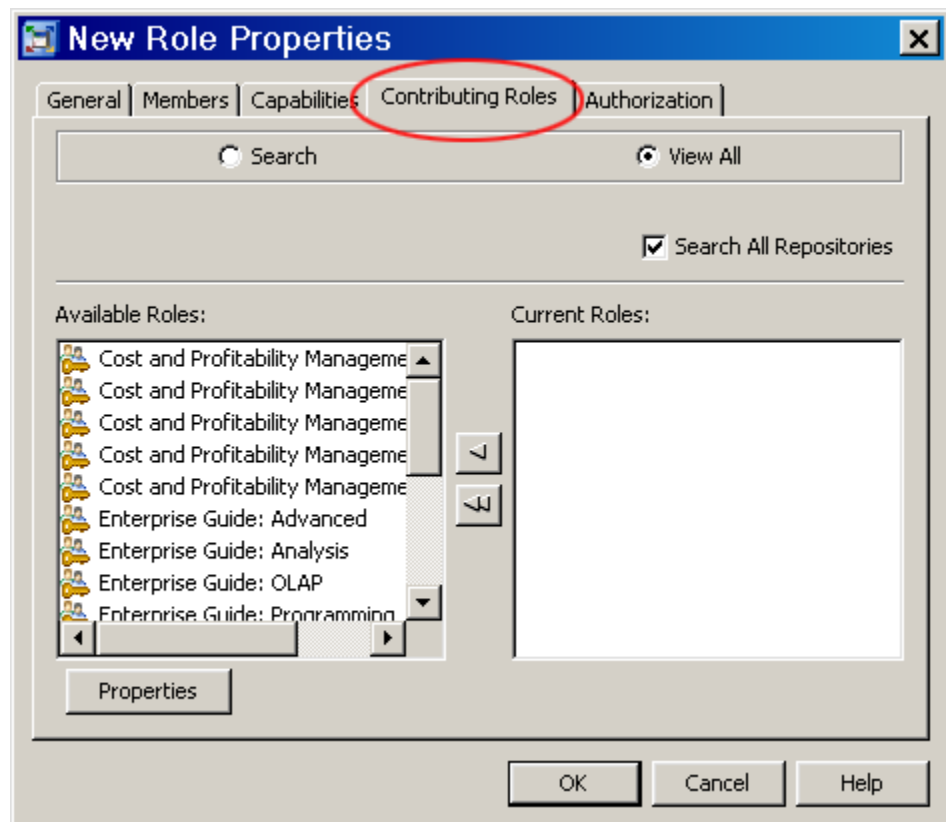
The **View Models** capability provides the abilities of a Business User in previous releases of SAS Activity-Based Management.

For a table that lists all the features of the SAS Cost and Profitability Management client and specifies what features are available to a user based on those two factors, see “[Table of Capabilities and Model Access](#)” on page 26.

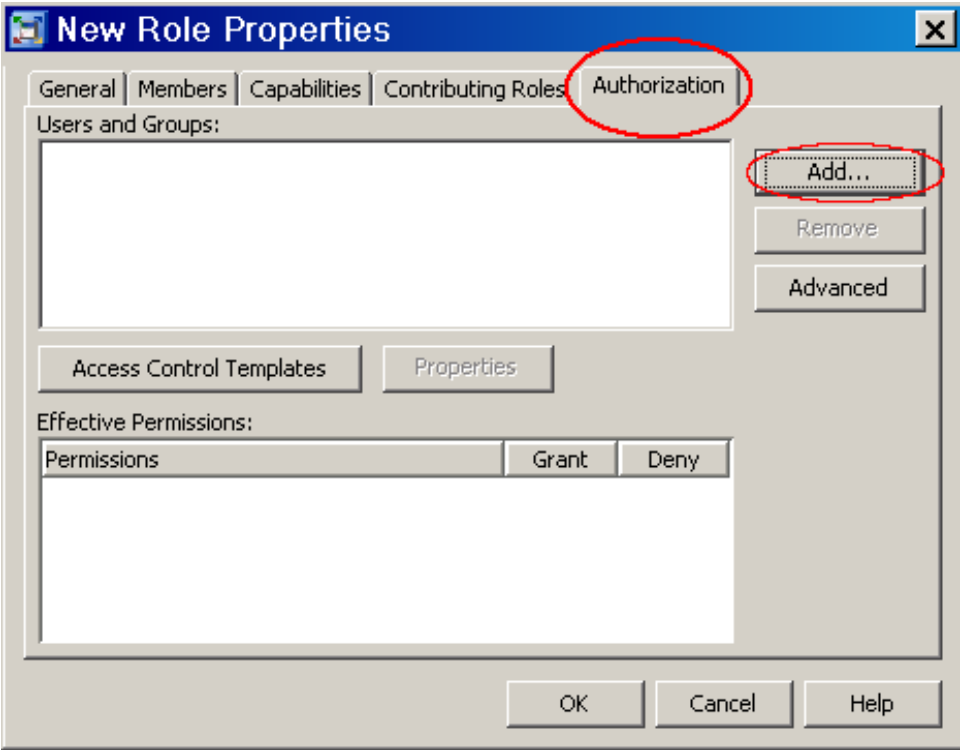
*Note:* In the **Capabilities** tab of SAS Management Console, the following icon  indicates a capability that has been explicitly assigned to a role, whereas the following icon  indicates a capability that has been inherited from a contributing role. For example, in the following picture you can see that five capabilities have been inherited from one or more contributing roles, whereas only one capability (Create What-If Analysis) has been explicitly assigned to the role.



6. Click the **Contributing Roles** tab and add any roles that you want to contribute capabilities to the role being created. This step is optional.



7. Click the **Authorization** tab and add any groups or roles that you want to have authorization to access the role being created. This step is optional.



8. Click **OK** to finish creating the role.

# Creating Groups

## Overview

A group is a set of users who share the same capabilities. It is convenient to create groups before creating users because if groups exist, then when you create a user you can immediately assign the user to one or more groups and thereby determine the user's capabilities. The following table shows the groups that are automatically created on installation of SAS Cost and Profitability Management. The table also shows the capabilities that each group inherits by being a member of a group or a role.

Group	is a member of:	inherits these capabilities:
Cost and Profitability Management Users	none	none (a user must be, directly or indirectly, a member of this group to access SAS Cost and Profitability Management)
Cost and Profitability Management Administrators	<ul style="list-style-type: none"><li>Cost and Profitability Management Users (group)</li><li>Cost and Profitability Management: Administration (role)</li></ul>	Administer



Group	is a member of:	inherits these capabilities:
Cost and Profitability Management Modelers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Models (role)</li> <li>Cost and Profitability Management: Create What-If Analysis (role)</li> <li>Cost and Profitability Management: Create Surveys (role)</li> </ul>	Create Models Create What-If Analysis Create Surveys Take Surveys
Cost and Profitability Management Viewers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: View Models (role)</li> </ul>	View Models
Cost and Profitability Management Cube Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> </ul>	none
Cost and Profitability Management What-If Analysis Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Models (role)</li> <li>Cost and Profitability Management: Create What-If Analysis (role)</li> </ul>	Create What-If Analysis
Cost and Profitability Management Survey Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Survey (role)</li> </ul>	Create Surveys Take Surveys
Cost and Profitability Management Survey Takers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Take Survey (role)</li> </ul>	Take Surveys
Cost and Profitability Management Server Users	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> </ul>	Access the IOM Server (for internal use)

*Note:* In order to access SAS Cost and Profitability Management, a user must be a member of either:

- the group Cost and Profitability Management Users
- a group that is a member of the group Cost and Profitability Management Users. Notice that all the groups that are automatically created on installation (listed in the table above) are members of the Cost and Profitability Management Users group.

*Note:* In the case of each group listed above, the capability possessed by the group is inherited from a role, and not from another group.

*Note:* The Cost and Profitability Management Cube Creators group does not inherit any capability. For information on giving capabilities to this group to work with SAS

OLAP cubes, see [“SAS OLAP Cube Creators” on page 24](#). The Create Cubes capability is not necessary for creating Microsoft Analysis Services cubes.

Creating a group is the second part of a two-step process:

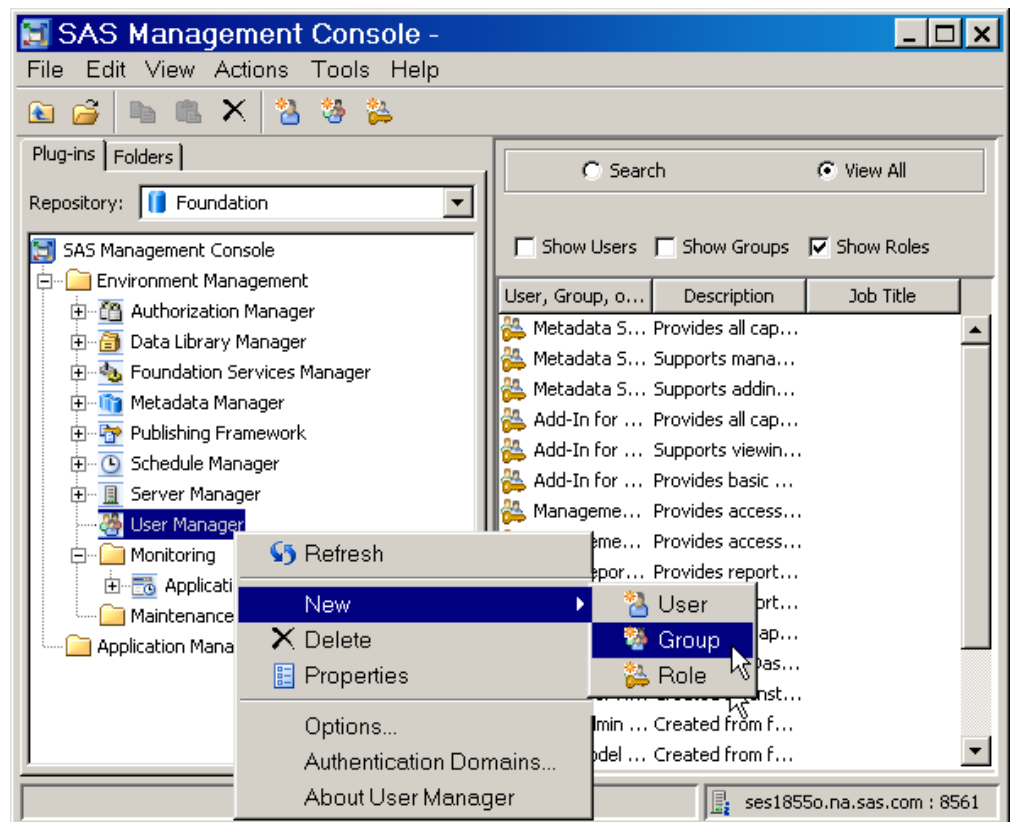
1. Create roles with the capabilities that you want to assign to the group.
2. Create a group and make it a member of the roles that you created. This gives the group the capabilities of the roles.

For examples of allocating capabilities using roles and groups, see [“Table of Capabilities and Model Access” on page 26](#).

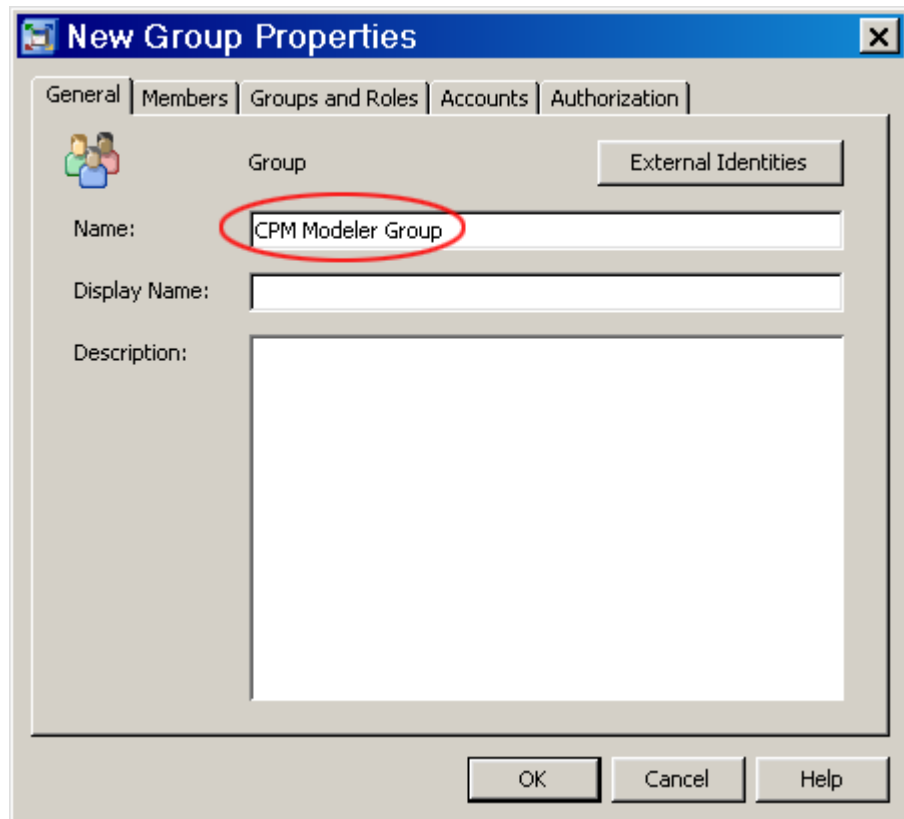
## Create a Group

To create a group, do the following:

1. Open SAS Management Console, connecting to your metadata server.
2. Select User Manager, and then select **New** ⇒ **Group**.



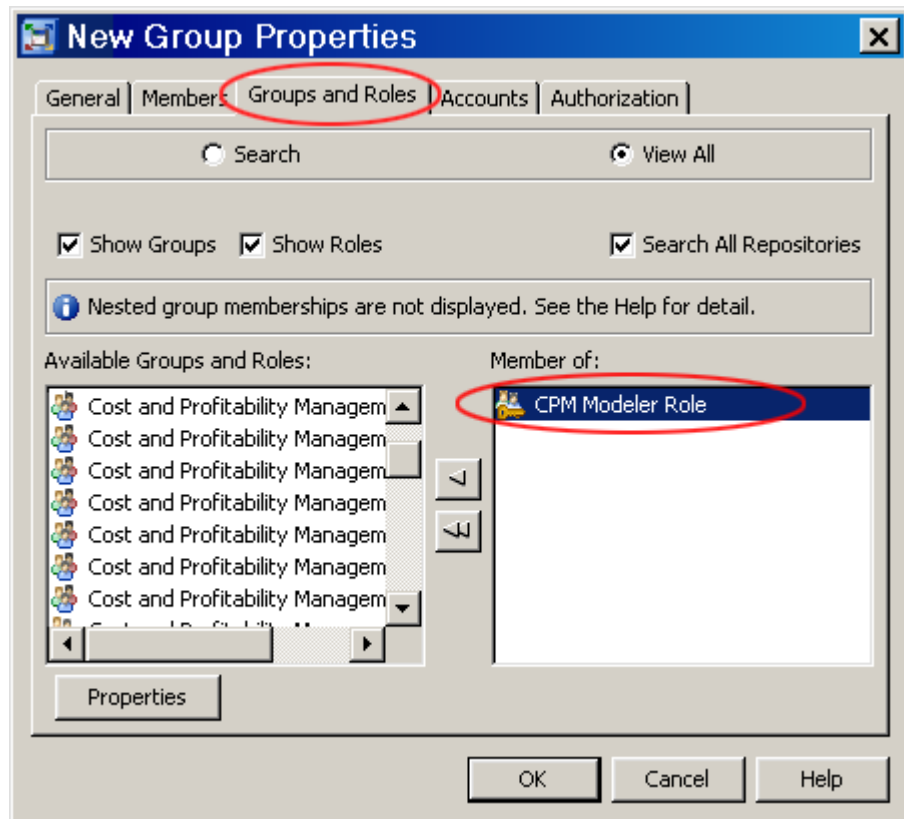
3. Name the group (for example CPM Modeler Group).



4. Click the **Groups and Roles** tab and add the roles whose capabilities you want the group to have.

We say that the group is a member of the role. Such roles contribute their capabilities to the group. Alternatively, we can say that the group inherits its capabilities from the roles.

*Note:* To access SAS Cost and Profitability Management, a user must be a member of the Cost and Profitability Management Users group, either directly or indirectly (for example, by being a member of a group that is a member of the Cost and Profitability Management Users group).

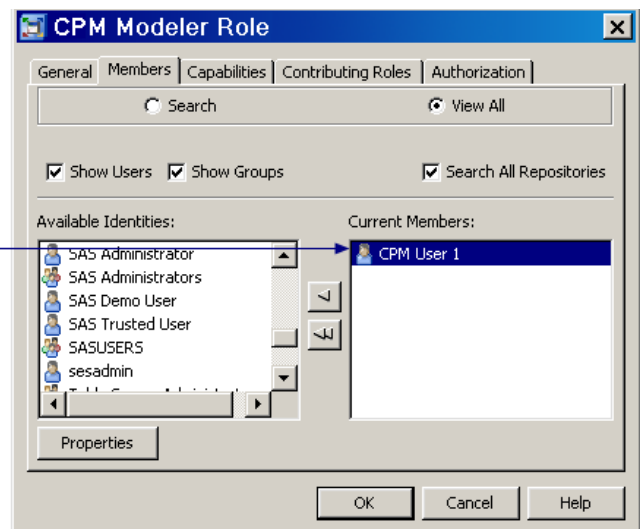


5. Click **OK** to finish creating the group.

### Creating Groups versus Creating Roles

You might be aware of the fact that an administrator can give capabilities to a user by making the user a member of a role. The following picture shows CPM User 1 as a member of the CPM Modeler Role. By being a member of this role, CPM User 1 acquires the capabilities of the role—in this case Create Model.

An administrator can make a user a member of a role as well as a member of a group



So, you might wonder why an administrator should bother to create a group, CPM Modeler Group, that inherits from the CPM Modeler Role. Why not simply make users a

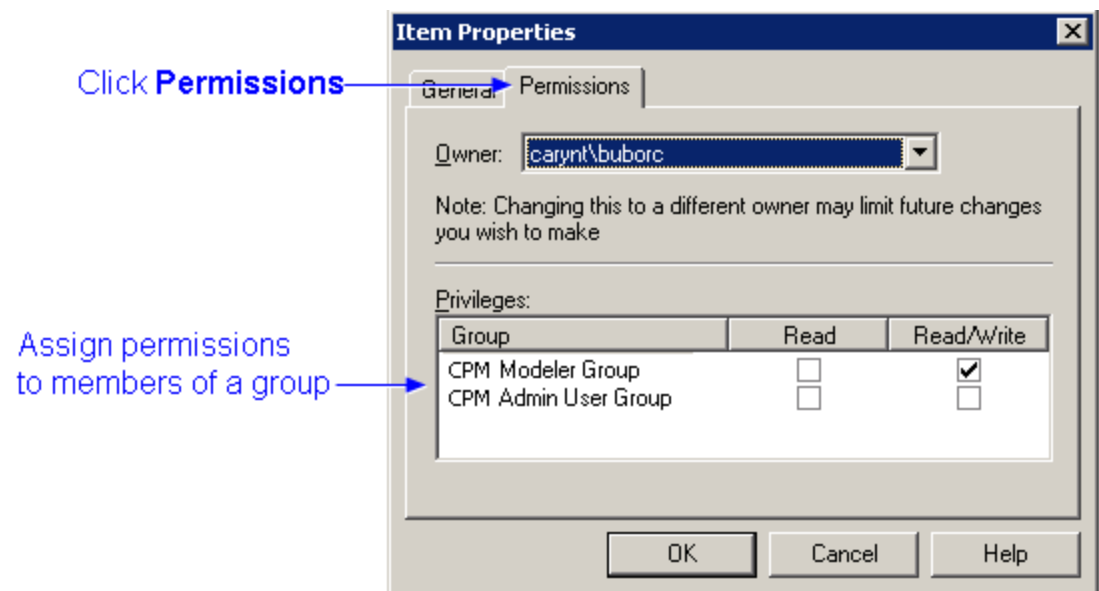
member of the CPM Modeler Role or some other role? The answer is that in order to access SAS Cost and Profitability Management, a user must be either:

- a member of the group Cost and Profitability Management Users
- a member of a group that is a member of the group Cost and Profitability Management Users.

Access to individual workspace items (models, cubes, and so forth) is assigned to groups and not to roles. So, if a user is not a member of a group, then that user cannot access Cost and Profitability Management objects (such as models) even if the user is a member of a role.

Additionally, the owner of a workspace item (or an administrator) can assign read or read/write access to that item. To assign access to a workspace item, the owner or administrator does the following:

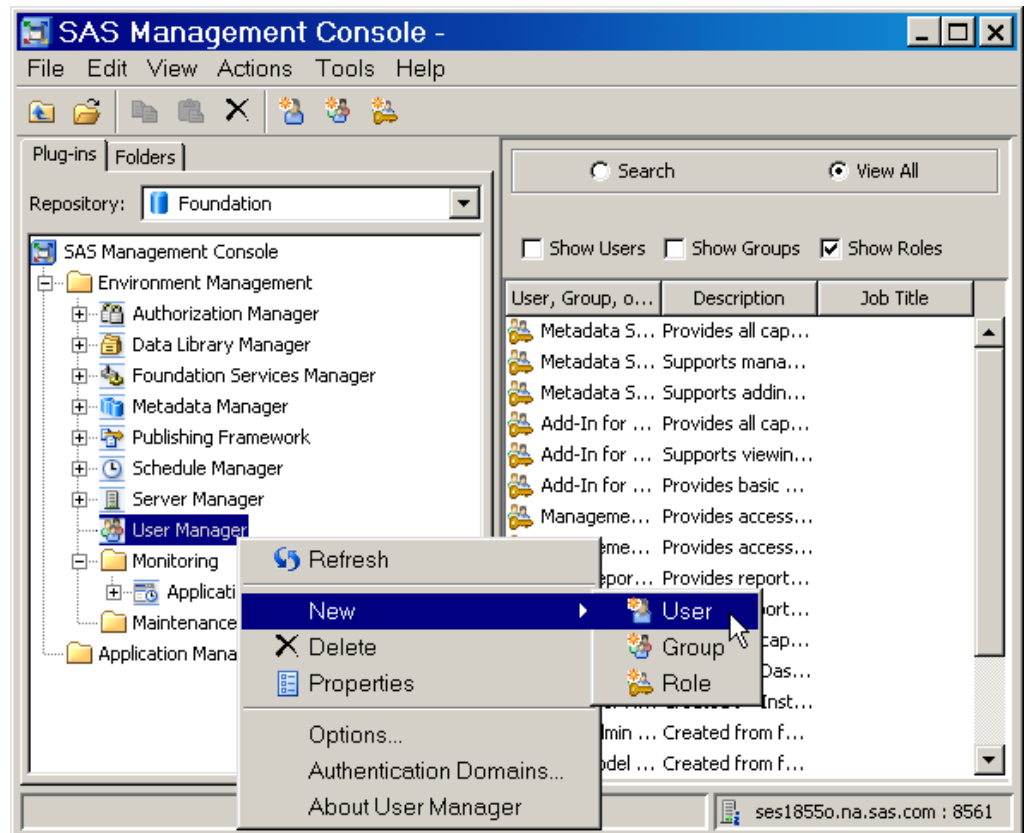
1. Select the workspace item.
2. Right-click and select **Item Properties** (or select **Edit** ⇒ **Item Properties**).
3. Click the **Permissions** tab.
4. Assign Read or Read/Write access to the desired groups. (Only groups are listed—not roles.)



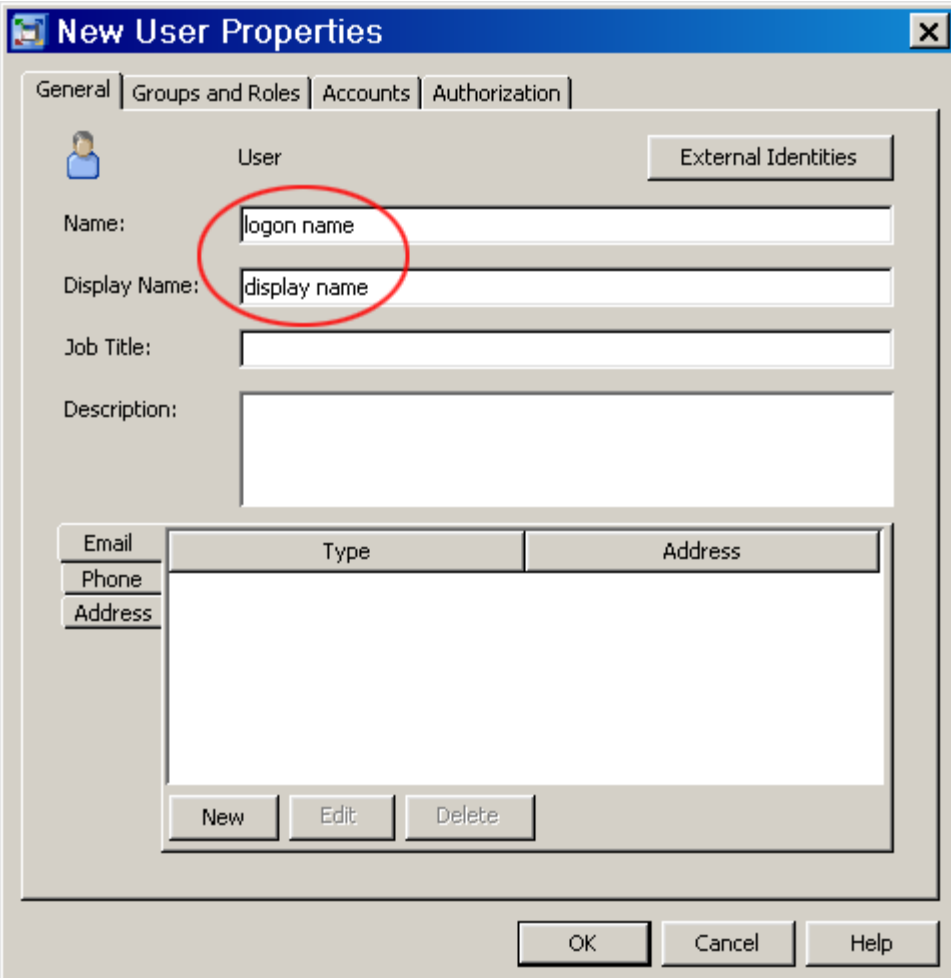
## Creating Users

To create a user, do the following:

1. Open SAS Management Console, connecting to your metadata server.
2. Select **User Manager**, and then select **New** ⇒ **User**.



3. Enter the user's Name (which is the user's CPM logon name) and the user Display Name.

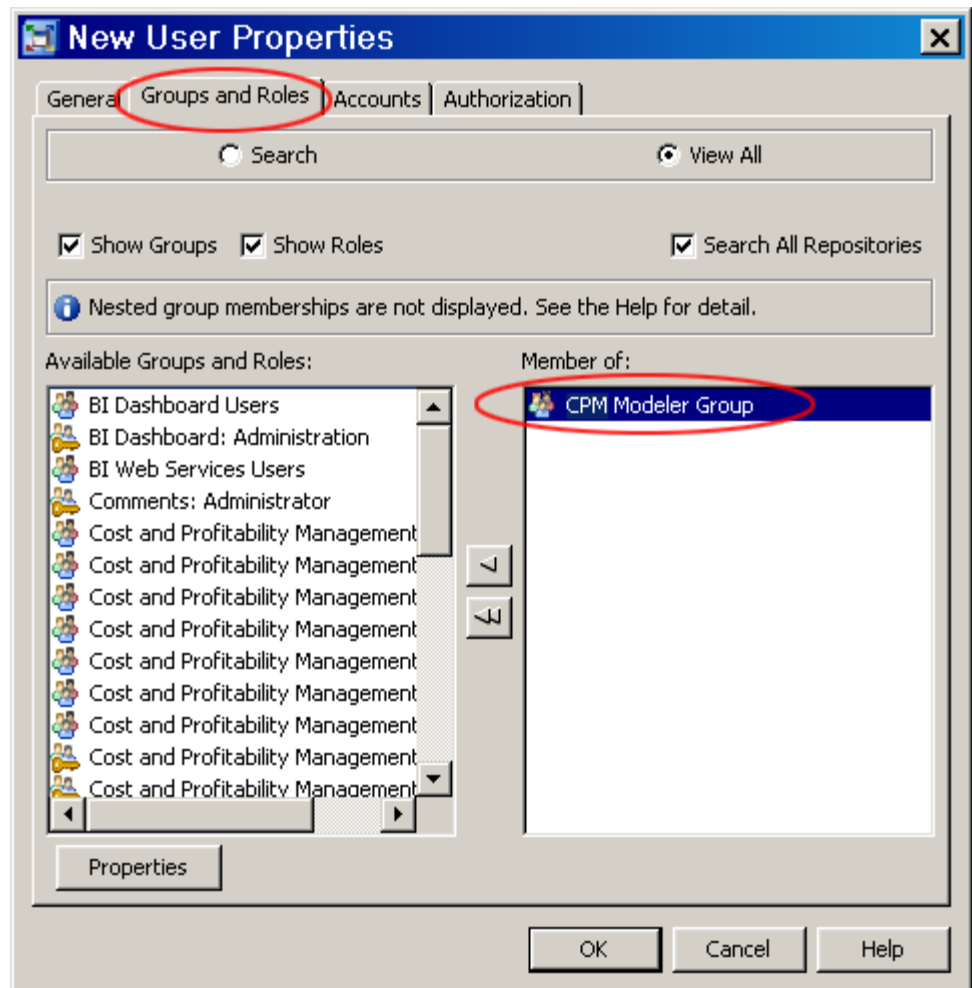


The image shows a Windows-style dialog box titled "New User Properties". It has four tabs: "General", "Groups and Roles", "Accounts", and "Authorization". The "General" tab is selected. Inside the "General" tab, there is a "User" section with a user icon and an "External Identities" button. Below this are four text input fields: "Name:" (containing "logon name"), "Display Name:" (containing "display name"), "Job Title:", and "Description:". The "Name:" and "Display Name:" fields are circled in red. At the bottom of the "General" tab is a table with columns "Email", "Phone", and "Address". The table is currently empty. Below the table are three buttons: "New", "Edit", and "Delete". At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

Email	Type	Address

4. Click the **Groups and Roles** tab and add the group or groups of which the user is a member.

This gives the user all the capabilities of the group. If the user is a member of multiple groups, then the user has the union of the capabilities of the groups of which the user is a member.

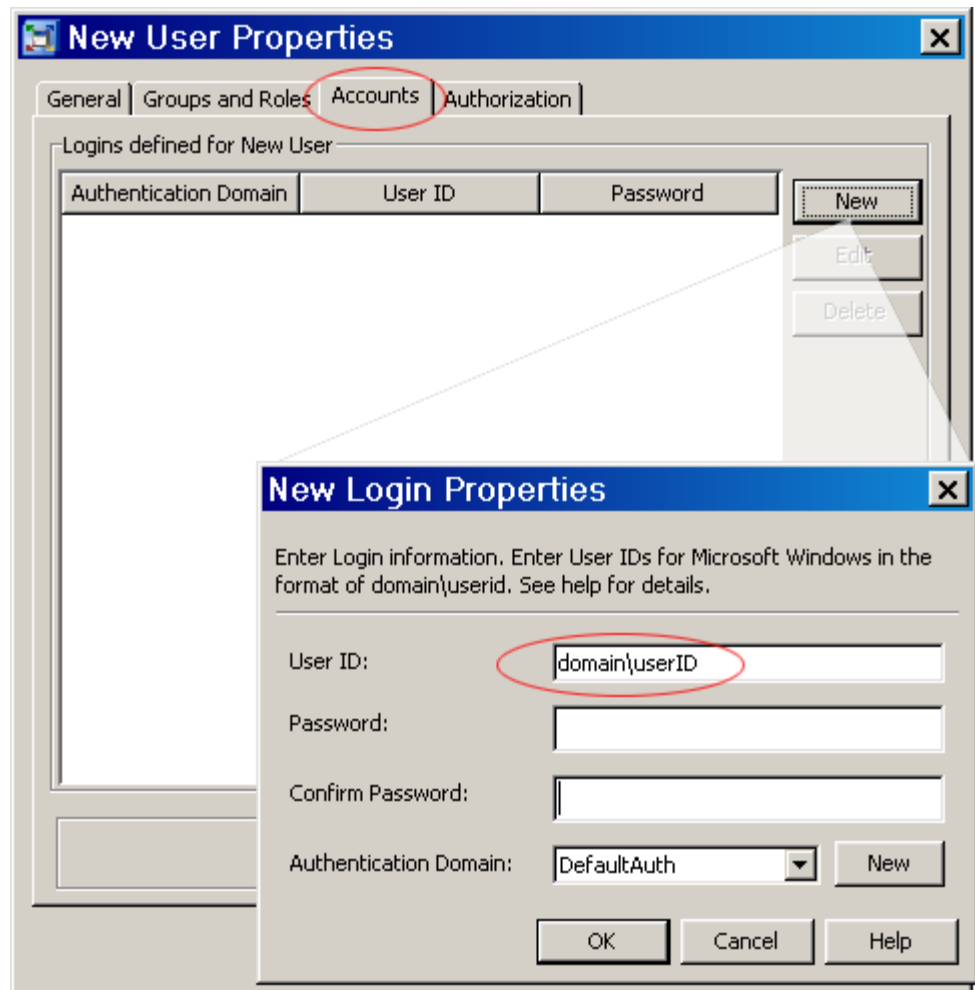


*Note:* To access SAS Cost and Profitability Management, a user must be a member of the Cost and Profitability Management Users group, either directly or indirectly (for example, by being a member of a group that is a member of the Cost and Profitability Management Users group).

5. Click the **Accounts** tab and click **New** to add a new account. Enter the user's domain and user ID.

You should leave the password blank. When the user logs onto SAS Cost and Profitability Management, the password that the user enters will be verified against the user's password on the system.





6. Click **OK** twice to finish creating the user.

*Note:* If the user is defined only on the local machine rather than in a domain, make sure to add the user to Local Users and Groups using Microsoft Management Console (with the Administrative Tool “Computer Management”).

---

## Sample Allocation of Capabilities

### ***Groups Created During Installation***

During installation of SAS Cost and Profitability Management, the following groups are created automatically:

Group	is a member of:	inherits these capabilities:
Cost and Profitability Management Users	none	none (a user must be, directly or indirectly, a member of this group to access SAS Cost and Profitability Management)
Cost and Profitability Management Administrators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Administration (role)</li> </ul>	Administer
Cost and Profitability Management Modelers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Models (role)</li> <li>Cost and Profitability Management: Create What-If Analysis (role)</li> <li>Cost and Profitability Management: Create Surveys (role)</li> </ul>	Create Models Create What-If Analysis Create Surveys Take Surveys
Cost and Profitability Management Viewers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: View Models (role)</li> </ul>	View Models
Cost and Profitability Management Cube Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> </ul>	none
Cost and Profitability Management What-If Analysis Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Models (role)</li> <li>Cost and Profitability Management: Create What-If Analysis (role)</li> </ul>	Create What-If Analysis
Cost and Profitability Management Survey Creators	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Create Survey (role)</li> </ul>	Create Surveys Take Surveys
Cost and Profitability Management Survey Takers	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> <li>Cost and Profitability Management: Take Survey (role)</li> </ul>	Take Surveys
Cost and Profitability Management Server Users	<ul style="list-style-type: none"> <li>Cost and Profitability Management Users (group)</li> </ul>	Access the IOM Server (for internal use)

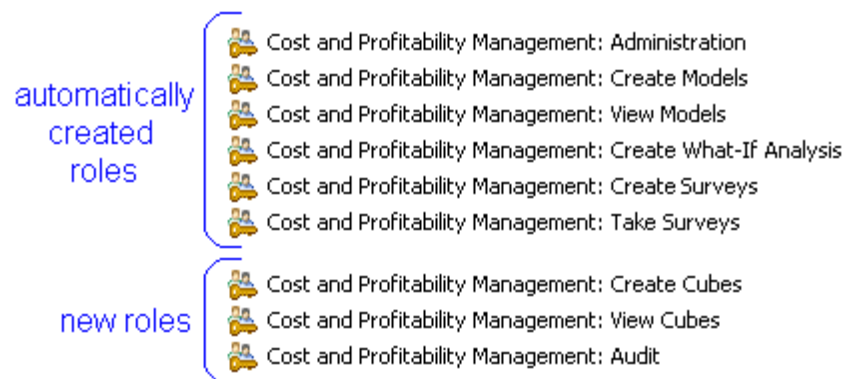
See “Creating Groups” on page 12.

## Roles

During installation of SAS Cost and Profitability Management, the following roles are created automatically.

Role	Capabilities
Cost and Profitability Management: Administration	Administer
Cost and Profitability Management: Create Model	Create Models
Cost and Profitability Management: View Models	View Models
Cost and Profitability Management: Create Surveys	Create Surveys See <a href="#">Chapter 6, “Creating Surveys,”</a> on page 57.
Cost and Profitability Management: Take Surveys	Take Surveys See <a href="#">Chapter 7, “Taking a Survey,”</a> on page 69.
Cost and Profitability Management: Create What-If Analysis	Create What-If Analysis See “Create a What-If Analysis” in Chapter 2 of <i>SAS Cost and Profitability Management: What-If Analysis</i> .

As an administrator, you can also create your own roles. One option is to create a separate role for each capability for which there is not a role created automatically. This gives you maximum flexibility in assigning capabilities. Alternatively, you can create roles each of which has multiple capabilities. This gives you less flexibility but makes it easier to assign multiple capabilities to a group.



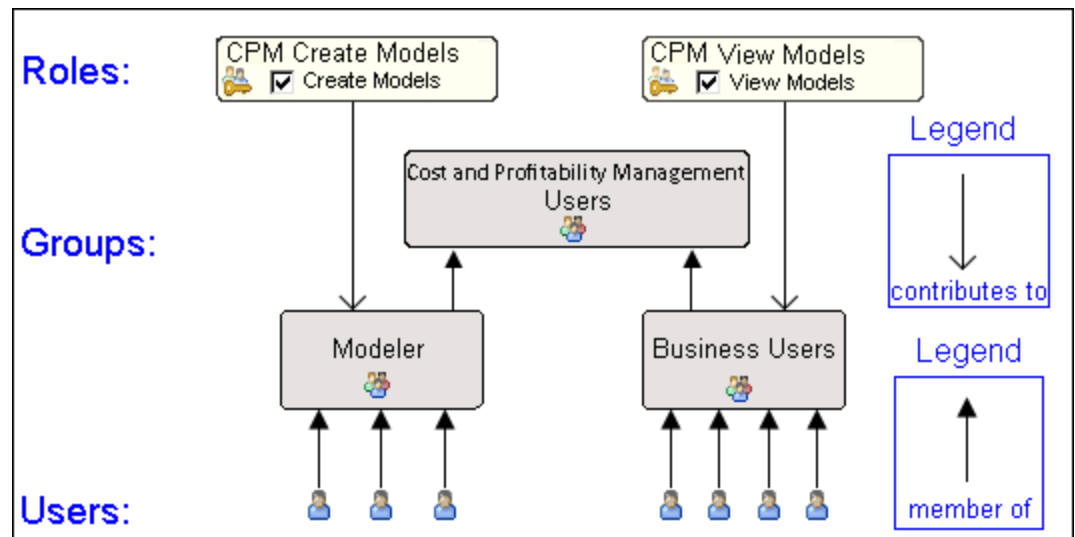
See “Creating Roles” on page 4.

## Modelers and Business Users

The following picture shows an example of creating two groups:

- Modeler
- Business Users

*Note:* The arrangement that is pictured is only a suggestion. It is not required.



The Modeler group inherits the Create Models capability, which gives users in the group full access to a model. With full access, users can create and delete models, create cubes, view cubes, and also perform other tasks such as creating column layouts and setting up exchange rates. The Create Models capability provides the abilities of a Modeler in previous releases of SAS Activity-Based Management.

The Business Users group inherits the View Models capability. The View Models capability provides the abilities of a Business User in previous releases of SAS Activity-Based Management.

For full details of what access to features the capabilities provide, see “[Table of Capabilities and Model Access](#)” on page 26.

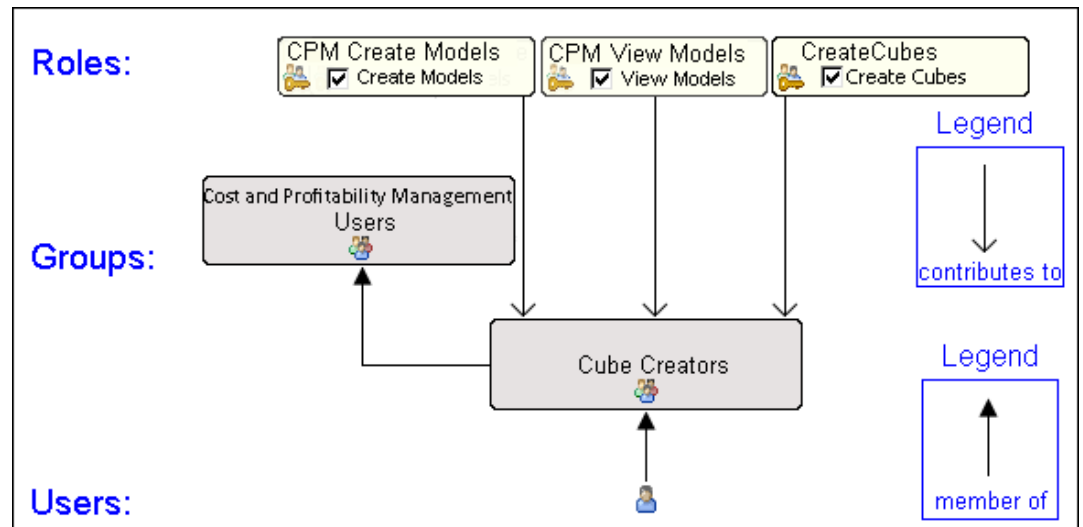
*Note:* The Modeler group and the Business Users group are both members of the Cost and Profitability Management Users group. This enables every user in the Modeler group and the Business Users group to log onto SAS Cost and Profitability Management.

## SAS OLAP Cube Creators

A user who wishes to create SAS OLAP cubes must be a member of the Cost and Profitability Management Cube Creators group. Membership in this group is not required for creating Microsoft Analysis Services cubes.

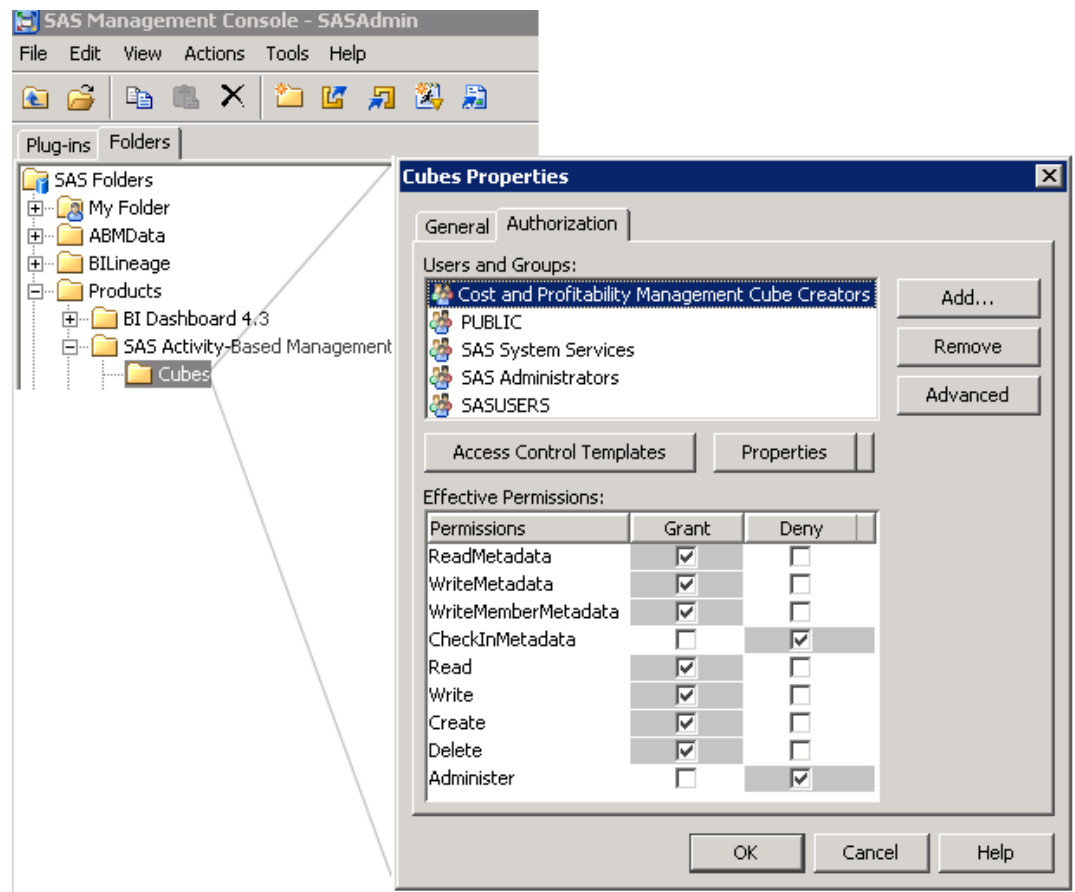
The Cost and Profitability Management Cube Creators group should have the following capabilities:

- Create Models
- View Models
- Create Cubes



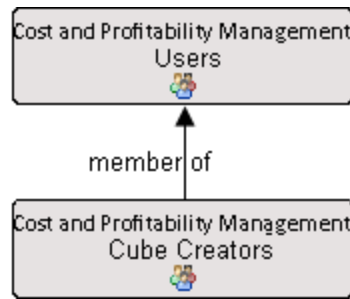
Alternatively, members of the Cost and Profitability Management Cube Creators group can inherit those capabilities by belonging to other groups with those capabilities.

*Note:* Members of the Cost and Profitability Management Cube Creators group inherit the authorizations shown in the following picture to access the Cubes folder for the SAS Cost and Profitability Management product. These authorizations are required for creating SAS OLAP cubes.



*Note:* The Cost and Profitability Management Cube Creators group is a member of the Cost and Profitability Management Users group. This enables every member of the

Cost and Profitability Management Cube Creators group to access SAS Cost and Profitability Management.



## Table of Capabilities and Model Access

### About the Table of Capabilities and Model Access

The following table shows how the combination of a capability and having Read or Read/Write access to a model determines a user's access to features of SAS Cost and Profitability Management.

#### Capability

This table lists only the capabilities that are specific to the SAS Cost and Profitability Management client. These capabilities are:

- Create Models
- View Models
- Create Cubes
- View Cubes
- Audit
- Administer

This table does not include capabilities related to associated applications.

- Create Surveys
- Take Surveys
- Create What-If Analysis

#### Read or Read/Write access to a model

The owner of a model (who, by default, is the creator of the model) can grant Read or Read/Write access to that model to the members of a group. See [“Grant Read or Read/Write Access to an Item”](#) on page 36.

See [“How to Read this Table”](#) on page 30.

Product Feature or Area	Create Models		View Models	Create Cubes	View Cubes	Audit	Administ er
	Model R	Model RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW
Acquire Write Lock (on model)		✓					
Open Model (with Read lock)	✓	✓	✓				
New Model Wizard (*)	✓	✓					
Import Model Data	✓	✓					
Export Model Data	✓	✓	✓				
Export and Register Tables	✓	✓	✓				
Cube Configuration Wizard	✓	✓	✓	✓			
Cube Configuration Import/Export	✓	✓	✓	✓			
New OLAP View Wizard	✓	✓	✓	✓	✓		
OLAP View Import/Export	✓	✓	✓	✓	✓		
Column layouts Create/Save/Import/Export	✓	✓	✓	✓			
Publish Model Period/Scenario		✓					
Publish Behaviors		✓					
Calculate		✓					
Generate Fact Tables		✓		✓			
Generate Cubes		✓		✓			
Audit Log	R	R	R			R Purge	
Periods & Scenarios (*)	R W D	R W D	R W	R			

Product Feature or Area	Create Models		View Models	Create Cubes	View Cubes	Audit	Administer
	Model R	Model RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW
Exchange Rates (*)	R W	R W	R	R			
Dimensions	R	R W D	R	R			
Model Properties Dialog	R	R W	R	R			
Period/Scenario Associations	R	R W D	R	R			
Copy Period/Scenario Data		✓					
Manage Attributes	R	R W D	R	R			
Drivers	R	R W D	R	R			
Edit model using Model Views	R	R W D	R	R			
Diagnostics	✓	✓	✓	✓			✓
Automation API		✓					
Workspace Items: models, cube configurations, cubes, column layouts, OLAP views, etc.							
Create	✓	✓	✓	✓			
Create Folders	✓	✓	✓	✓			
Open (Run, Invoke, View)	OwnerX	OwnerX	OwnerX	OwnerX1	OwnerX2		
Edit	OwnerRW	OwnerRW	OwnerRW	OwnerRW1			
Delete/Rename/Cut (Move)	Owner	Owner	Owner	Owner			✓
Item Properties	Owner	Owner	Owner	Owner			✓
My Shortcuts	✓	✓	✓	✓			
Operation Summaries(*)	R E D	R E D	R E	R E			
Manage Tasks(*)	R D	R D	R	R D			R D
User Options(*)	R W	R W	R W	R W	R W	R W	R W



**Legend**

\* = This facility is not model specific and cannot be controlled on a model-by-model basis

**R** = Read

**W** = Write

**D** = Delete

**P** = Publish

**E** = Export

**Purge** = Purge

*Note:* A user with Audit capability can purge an audit log without having either Read or Read/Write access to a model.

**Owner** = The Owner of the Workspace item has the ability

**OwnerX** = The Owner or any user in an authorized Read or Read/Write Group for the item has the ability.

- **OwnerX1** = The Owner or any user in an authorized Read or Read/Write Group for the item has the ability for Cube configurations and Saved Cube Views only. It excludes all other workspace items.
- **OwnerX2** = The Owner or any user in an authorized Read or Read/Write Group for the item has the ability for Saved Cube Views only. It excludes all other workspace items.

**OwnerRW** = The Owner or any user in an authorized Read/Write Group for the item has the ability.

- **OwnerRW1** = The Owner or any user in an authorized Read/Write Group for the item has the ability for Cube Configurations and Saved Cube Views only. It excludes all other workspace items.

**Create** = This refers to the ability to create folders and links to other workspace items. Creation of workspace items is not done in the workspace. Creation of workspace items is performed via operations on models and model data. For the ability to create workspace items, see the section at the top of the table that enumerates the wizards and dialogs that are used to create workspace items.

**Publish Model** = Model Viewers with read only access to a model cannot see data in a model unless it is in a published period/scenario.

**Workspace items** = Ability to perform operations on workspace items means that the user has permission to the workspace item and access to the model data (if any) needed for it. For example, to launch a Cube View, the user must have privileges to the Cube View and to the model it refers to.

**Edit** = This means to edit workspace items (other than models). For example, to edit an existing column layout, etc. and change its contents without changing its name.

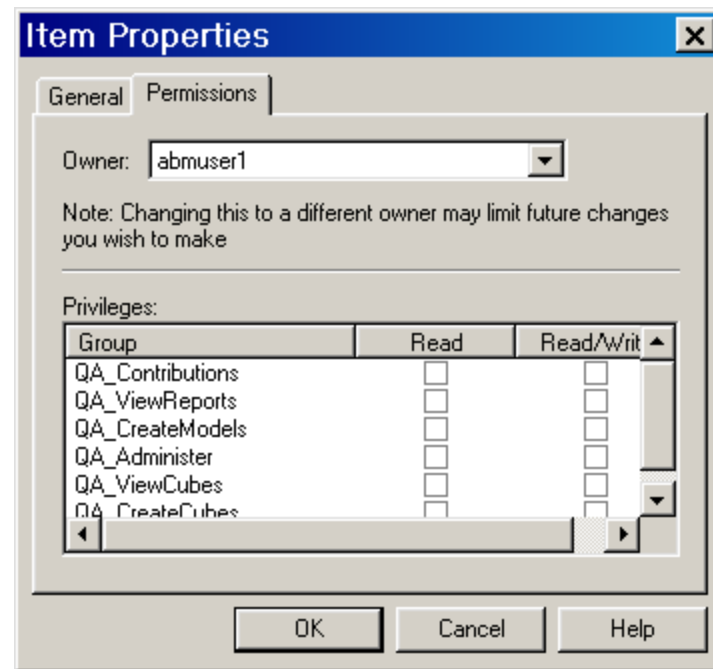
**Import Model Data** = The permissions on a model are applied if a user overwrites an existing model on import. The user is not allowed to overwrite the model unless the user has Write permission on the model.

**User Options** = Items that affect how the tool behaves and appears to the user. Example: background colors, number of displayable significant digits. These do not affect the Model data stored on the server in any way.

**Model Views** = Includes the Module views, Attributes view, Dimensions view, and Drivers view. Editing a model includes all editing tasks within those model views.

**My Shortcuts** = This includes all operations that can be performed under the My Shortcuts section of the Workspace Manager.

**Item Properties** = Open the Item Properties dialog box to change the owner of the item, edit its description, or assign permission to the item to groups.



## How to Read this Table

### 1. Capabilities

The column headings (Create Models, View Models, Create Cubes, View Cubes, Audit, and Administer) refer to capabilities that are granted to groups, and roles via SAS Management Console.

Product Feature or Area	Create Models		View Models	Create Cubes	View Cubes	Audit	Administer
	Model R	Model RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW	Model R or RW

SAS Management Console → **Cost and Profitability Management: Properties**

## 2. Model R and Model RW Access

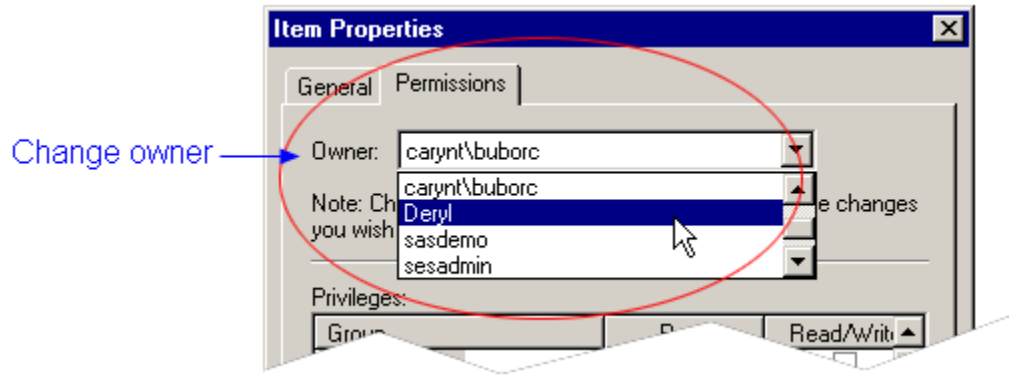
The column headings Model R and Model RW refer to the access that a user has to a particular model.

Product Feature or Area	Create Models		View Models
	Model R	Model RW	Model R or RW

A user can acquire the access in either of the following ways:

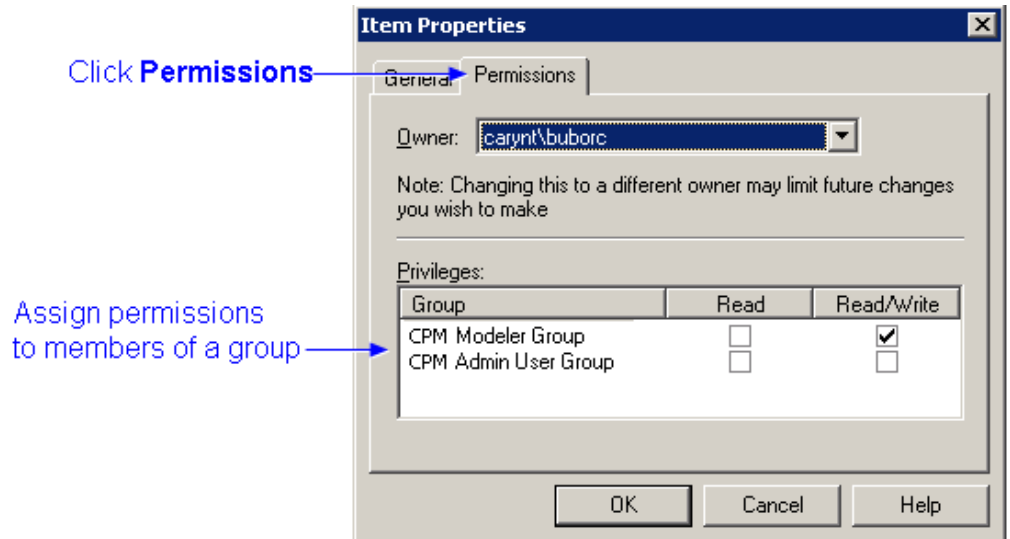
- The owner of a model automatically has Read/Write (RW) access to the model.

When a model is created, the creator becomes the model owner. However, the owner of a model (or an administrator) can transfer ownership of the model to another user by selecting the model in the Models Workspace and selecting **Edit** ⇒ **Item Properties** and then selecting a new owner from the drop-down list of users.



*Note:* The drop-down list of potential owners includes those users who have Create capability for the item. For example, you can transfer ownership of a model only to those users who have Create Models capability.

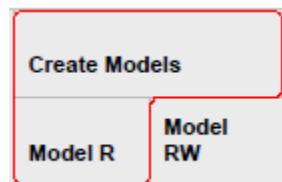
- The owner of a model (or administrator) can grant Read or Read/Write model access to members of a group by selecting the model in the Models Workspace and selecting **Edit** ⇒ **Item Properties**. Then, by checking either Read or Read/Write, the owner grants that model access to all members of the group.



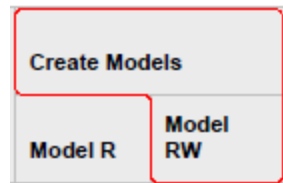
### 3. Understanding the Columns

The columns in the table refer to a set of users with a combination of a capability and a Read or Read/Write access to a model.

For example column 1 refers to the set of user with Create Models capability and Read access to a model. Because the owner of a model automatically has Read/Write access the model, users with only Read access are those users who are members of a group that has been granted Read access to the model by the model's owner.



Similarly, column 2 refers to the set of users with Create Models capability and Read/Write access to a model. Users with Read/Write access are the model owner and users who are members of a group that has been granted Read/Write access to the model by the model's owner.



*Note:* By itself, the Create Models capability primarily provides the ability to run the New Model Wizard and a few other facilities such as running diagnostics. It is mostly in conjunction with model permissions that the Create Models capability fulfills its potential.

#### 4. Understanding the Rows

Each row in the table refers to an action or facility of SAS Cost and Profitability Management. For example row 1 refers to the New Model Wizard. Row 2 refers to the action of importing model data.

#### 5. Understanding the Cells

The cells in the table specify either of the following (or both):

- Who specifically has the ability referred to in the row.
- What aspects of the ability does someone have

#### Who specifically has the ability referred to in the row

The following table lists the cell indicators that specify who specifically has the ability referred to in the row, and what those cell indicators mean.

cell indicator	what it means
✓	A user with both the capability and the Read or Read/Write access in the column heading has the ability.
Owner	<p>A user has the ability if the user:</p> <ul style="list-style-type: none"> <li>• has both the capability and the Read or Read/Write access in the column heading, and</li> <li>• is the model owner.</li> </ul>
OwnerX	<p>A user has the ability if the user:</p> <ul style="list-style-type: none"> <li>• has both the capability and the Read or Read/Write access in the column heading, and</li> <li>• is the model owner, or</li> <li>• is a member of an authorized Read or Read/Write group for the item.</li> </ul>

cell indicator	what it means
OwnerRW	<p>A user has the ability if the user:</p> <ul style="list-style-type: none"> <li>• has both the capability and the Read or Read/Write access in the column heading, and</li> <li>• is the model owner, or</li> <li>• is a member of an authorized Read/Write group for the item.</li> </ul>

### What Aspects of the Ability Does Someone Have

The following table lists the cell indicators that specify what aspect of an ability someone has, and what those cell indicators mean.

cell indicator	what it means
R	A user with both the capability and the Read or Read/Write access in the column heading has Read access to the item.
W	A user with both the capability and the Read or Read/Write access in the column heading has Write access to the item.
D	A user with both the capability and the Read or Read/Write access in the column heading has Delete access to the item.
P	A user with both the capability and the Read or Read/Write access in the column heading has Publish access to the item.
Purge	A user with both the capability and the Read or Read/Write access in the column heading has Purge access to the item.

## Chapter 2

# How To Manage Permissions

---

Create a Role .....	35
Create a Group .....	35
Create Users .....	35
Change the Owner of an Item .....	35
Grant Read or Read/Write Access to an Item .....	36

---

## Create a Role

See “Create a Role” on page 7.

## Create a Group

See “Create a Group” on page 14.

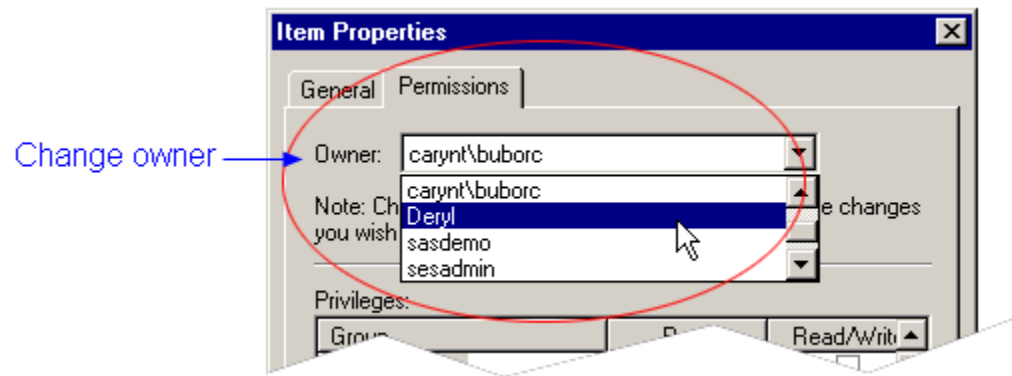
## Create Users

See “Creating Users” on page 17.

## Change the Owner of an Item

When a model or other workspace item is created, the creator becomes the item owner. However, the owner (or an administrator) can transfer ownership of the item to another user.

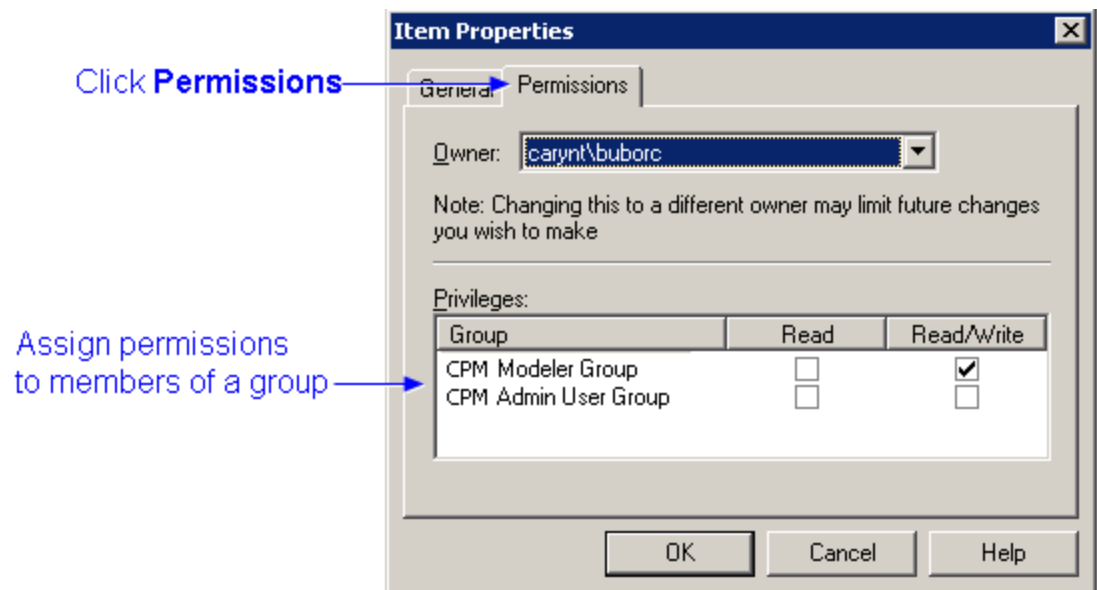
1. Select the item Workspace Manager and select **Edit** ⇒ **Item Properties**.
2. Select a new owner from the drop-down list of users.



## Grant Read or Read/Write Access to an Item

The owner of a model or other workspace item (or an administrator) can grant Read or Read/Write access to the item to members of a group. To grant access to the item:

1. Select the workspace item.
2. Right-click and select **Item Properties** (or select **Edit** ⇒ **Item Properties**).
3. Click the **Permissions** tab.
4. Grant Read or Read/Write access to the desired groups. (Only groups are listed—not roles.)





## Part 2

---

# Surveys

<i>Chapter 3</i>	
<b>Introduction to Surveys</b> .....	<i>39</i>
<i>Chapter 4</i>	
<b>Exporting Data to Use with Surveys</b> .....	<i>47</i>
<i>Chapter 5</i>	
<b>Surveys: User Interface</b> .....	<i>53</i>
<i>Chapter 6</i>	
<b>Creating Surveys</b> .....	<i>57</i>
<i>Chapter 7</i>	
<b>Taking a Survey</b> .....	<i>69</i>
<i>Chapter 8</i>	
<b>Manage Users</b> .....	<i>77</i>
<i>Chapter 9</i>	
<b>Administer Surveys</b> .....	<i>81</i>
<i>Chapter 10</i>	
<b>Preferences</b> .....	<i>91</i>
<i>Chapter 11</i>	
<b>Importing Survey Data back into the Model</b> .....	<i>101</i>



## Chapter 3

# Introduction to Surveys

---

<b>Types of Surveys</b> .....	<b>39</b>
<b>Logging On</b> .....	<b>40</b>
<b>Log On with IWA</b> .....	<b>41</b>
<b>The Survey Process in a Nutshell</b> .....	<b>41</b>
Overview .....	41
Export a model .....	42
Create a survey .....	42
Assign survey items to survey takers .....	42
Take the survey .....	42
Administer the survey .....	44
Reimport the model .....	45

---

## Types of Surveys

One of the most difficult tasks in maintaining a model is keeping its data accurate and up-to-date. Now you can create Web surveys to solicit data from the people who are directly responsible for the activities and accounts in your model. Data from the surveys is written directly to staging tables that have been exported from the model.

The following table shows the types of surveys that you can create for each module and the fields that a survey taker can update for each type of survey.

*Note:* Each field name is qualified by the staging table that the field is in.

Type of survey	Fields that can be updated
Driver	Assignment.DriverQuantityFixed See “ <a href="#">Assignment table</a> ” on page 122.
Entered Cost Element	EnteredCostElement.EnteredUnitCost EnteredCostElement.EnteredCost See “ <a href="#">EnteredCostElement table</a> ” on page 132.
Output Quantities	Account.OutputQuantityUE See “ <a href="#">Account table</a> ” on page 119.

Type of survey	Fields that can be updated
Attribute	ValueAttributeAssociation.NumericValue ValueAttributeAssociation.StringValue See “ValueAttributeAssociation table” on page 141.

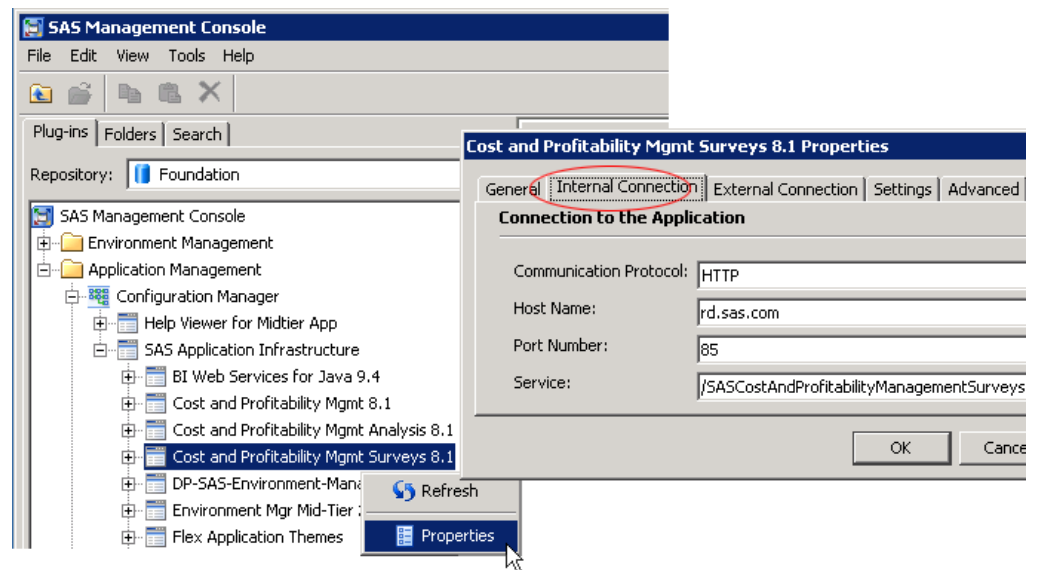
## Logging On

Log on to surveys by typing the survey URL into the command line of a browser. To determine the URL, you can do the following:

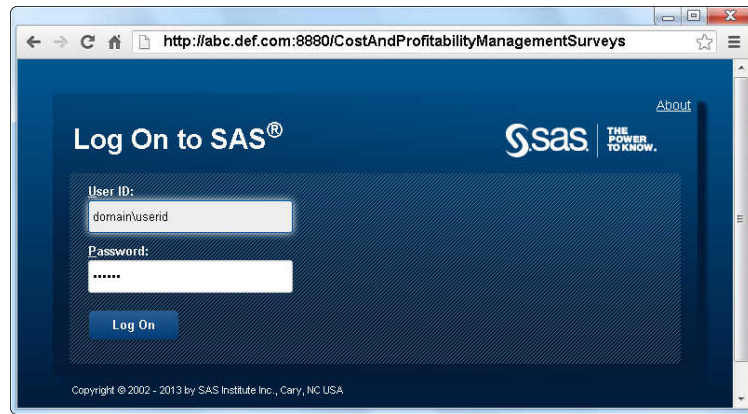
1. Log on to SAS Management Console as an administrator, and access the SAS Cost and Profitability Management middle-tier server.
2. Click the **Plug-Ins** tab.
3. Expand **Application Management**.
4. Expand **Configuration Manager**.
5. Expand **SAS Application Infrastructure**.
6. Right-click **Cost and Profitability Mgmt Surveys 8.1** and select **Properties**.
7. Click the **Connection** tab on the Properties window.

The connection information appears. In the following picture, you can see that the URL for invoking surveys is:

`http://xyz.sas.com:85/SASCostAndProfitabilityManagementSurveys`



8. Type the survey URL into the command line of a browser.



9. Log on with a user ID and password. The user must exist in the SAS Metadata Server with the following capabilities:

- To create a survey, the Create Surveys capability
- To take a survey, the Take Surveys capability

See Chapter 1, “User Capabilities and Groups,” on page 3.

---

## Log On with IWA

If your mid-tier is configured for IWA (Integrated Windows Authentication) then you can use IWA from your browser to log on if you have configured your browser to use IWA.

For information see “Support for Integrated Windows Authentication” in *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide, Third Edition*: <http://support.sas.com/documentation/cdl/en/bimtag/68217/HTML/default/viewer.htm#p1871e69gmwdr0n1o182krs1c10p.htm>.

---

## The Survey Process in a Nutshell

### Overview

The steps for using surveys are the following:

- “Export a model” on page 42
- “Create a survey” on page 42
- “Assign survey items to survey takers” on page 42
- “Take the survey” on page 42
- “Administer the survey” on page 44
- “Reimport the model” on page 45

**Export a model**

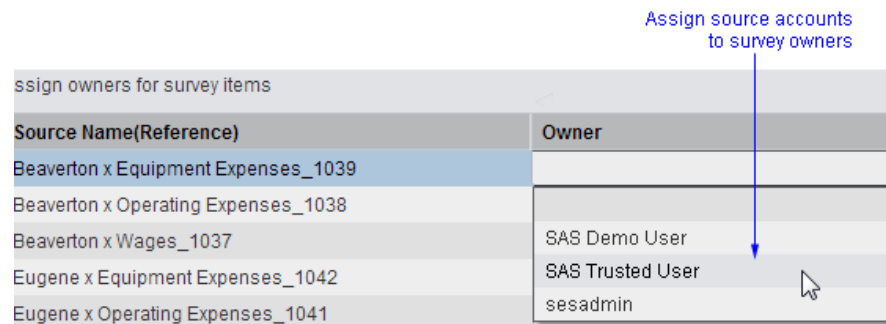
See [Chapter 4, “Exporting Data to Use with Surveys,”](#) on page 47.

**Create a survey**

See [Chapter 6, “Creating Surveys,”](#) on page 57.

**Assign survey items to survey takers**

The following picture shows how to assign a driver survey.



*Note:* The users that are available for assignment and that are shown in the drop-down list are members of the Survey Takers group that inherit the Take Surveys capability. See [“Creating Groups”](#) on page 12.

When you finish creating the survey, survey takers are notified by e-mail.

**Take the survey**

1. Answer the survey questions.

The survey taker logs on and begins taking the assigned survey.



The following picture shows a driver survey.

File Help Home Survey Manager User Manager

Manage Surveys  
By Definitions  
By Users  
Assigned Surveys

Double-click a source account →

Survey Items: pe tutorial

Select Period/Scenario: 2008 Q1/Actual Open Search

Driver Survey: Double click/open an item to fill survey measures. ?

Source Name(Re...	Percentage Complete	Status	New As...
Beaverton x Equipme...	0%	Incomplete	
Beaverton x Operating...	0%	Incomplete	

Fill Quantities For: Beaverton x Equipment Expenses\_10

Save As +

Period/Scenario(s): 2008 Q1/Actual Values are in terms of Percentage

Measure Name	Percentage
Beaverton x Air Distribution_2020	25.0000
Beaverton x Expedite Package Ship...	3.0000

Fill in values for the destination accounts →

2. Add additional assignments.

While taking a driver survey, a survey taker can add assignments that are not currently in the model. See [“Add New Assignments” on page 72](#).

Fill Quantities For: Beaverton x Equipment Expenses\_1039

Click **Add assignments**

Period/Scenario(s): 2008 Q1 in terms of Percentage

Measure Name	Percentage	Surveyed
<input checked="" type="checkbox"/> Beaverton x Air Di		
<input type="checkbox"/> Beaverton x Exp		

(to the destination account)

Select one or more additional destination accounts

Add Assignments

Select the desired item for surveying

Account Name	Account Reference	Type
Beaverton x Equipment Expenses_1039		
Beaverton x Operating Expenses_1038		
Beaverton x Wages_1037		
Eugene x Operating Expenses_1041		

3. Save the survey items as surveyed.

When each survey item has been satisfactorily responded to, the survey taker saves it as surveyed.

*Note:* As soon as a survey taker saves an item as surveyed, it is written to a staging table in the database. If the survey taker subsequently saves the item as not surveyed, the value that was previously saved to the database is not rolled back.

Select **Save As > Surveyed** or **Save As > Not surveyed**

for the selected account

Fill Quantities For: Beaverton x Equipment Expenses\_1039

Save As

Surveyed

Not surveyed

08 Q1/Actual Values are in terms of Percentage

Measure Name	Percentage	Surveyed
Beaverton x Air Distribution_2020	25.0000	No
Beaverton x Expedite Package Ship...	3.0000	No
Beaverton x Inspect 2019	25.0000	No

The Survey Manager displays the **Percentage Complete**—the percentage of survey items that have been saved as surveyed.

SAS Demo User - Survey Items:pe tutorial

Select Period/Scenario: 2008 Q1/Actual | Mark Item As | Audit

Driver Survey: Double click/open an item to fill survey measures. ?

Source Name(Ref...	Percentage Comp...	Status	New Assignments
Beaverton x Equip...	28%	Incomplete	No
Beaverton x Operat...	0%	Incomplete	No

#### 4. Submit the survey.

When all survey items have been saved as surveyed, the survey taker can submit the survey.

After all accounts have been saved as surveyed  
click **Submit the survey**

Fill Quantities For: Beaverton x Equipment Expenses\_1039

Save As | Search

Period/Scenario(s): 2008 Q1/Actual | Values are in terms of Percentage

Measure Name	Percentage	Surveyed
Beaverton x Air Distribution_2020	25.0000	Yes
Beaverton x Expedite Package Shipment...	3.0000	Yes
Beaverton x Inspect_2019	25.0000	Yes

**Submit**

Are you sure you want to submit the survey?

☐ Notify administrator on survey submit.

Click **Yes**

Yes No

Once the survey taker submits the survey, its status changes to **Completed**, and the survey taker can no longer change any responses.

Survey Items:pe tutorial

Select Period/Scenario: 2008 Q1/Actual | Open | Search

Driver Survey: Double click/open an item to fill survey measures. ?

Source Name(Refer...	Percentage Complete	Status	New Assignments
Beaverton x Equipment E...	100%	Completed	
Beaverton x Operating Ex...	0%	Incomplete	

The survey creator receives a notification e-mail.

### Administer the survey

The administrator of a survey is the survey creator.

Among other administration tasks, a survey creator can do the following:

#### 1. Reopen a completed survey.

On receiving notification that a survey has been submitted, a survey creator has the option of reopening the survey by marking it as **Incomplete**. If a survey creator reopens the survey, then:

- the survey status changes to **Incomplete**.
- the survey taker receives a notification.



- the survey taker retakes the survey and submits it again after saving all items as surveyed.

See [“Mark a Survey as Incomplete” on page 82](#).

*Note:* Staging tables in the database are not rolled back if a survey creator reopens a survey.

## 2. Audit new assignments.

If the survey taker has added any new assignments, then a survey creator must audit the assignments to either approve or disapprove.

- If the survey creator approves, then the new assignment values, as entered by the survey taker, are written to the staging table in the database.
- If the survey creator disapproves, then the new assignment is removed from the Assignment staging table in the database—it is as though the assignment was never made.

See [“Audit Surveys” on page 83](#).

## ***Reimport the model***

To reimport survey data:

1. Select **File** ⇒ **Import**.
2. Select **Surveys**.

For general information on importing, see Chapter 74, “Import Model Data from a Database,” in *SAS Cost and Profitability Management: User’s Guide*.

3. Select an existing model to update with survey data. You cannot create a new model from survey data.

*Note:* When importing into an existing model, make sure that the survey data is for the correct model. If the data is from a different model, the import can corrupt the existing model.

**Import Data - Model** Step 2 of 6

**Model**

Choose the model you want to import the data into.

Enter new model name or Select the existing model you want to import into:

☐ New model:

\* Model name:

\* Model Reference:

☒ Existing model:

[Select Model]

Select survey model to update an existing model:

Select one of the option to import new data when importing model structures and data:

Options

☒ Update all data in the model, then import new data

☐ Periodic import

☐ Remove all data in the model, then import new data

< Back Next > Finish Cancel Help

**See Also**

Chapter 11, “Importing Survey Data back into the Model,” on page 101

## Chapter 4

# Exporting Data to Use with Surveys

Exporting Survey Data .....	47
-----------------------------	----

## Exporting Survey Data

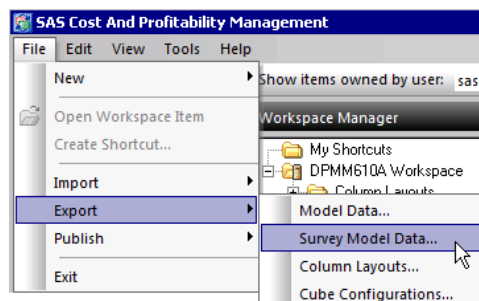
To create a survey, you do not have to export everything out of your model. Here are the steps to ensure you export only the required fields.

*Note:* Unless specified otherwise, accept all the default Export Wizard selections.

To export survey data, do the following:

*Note:* The Surveys application should not be open when you export survey data because the Surveys application caches model data when it opens and will not be aware of newly exported survey data.

1. Select **File** ⇒ **Export** ⇒ **Survey Model Data**.



The Survey Export – Select Model dialog opens.

2. Specify your choices on the Survey Export – Select Model dialog.

**Survey Export - Select Model** Step 1 of 7

Select a model and one or more period/scenario associations to export.

Select a model to export:

[Select Model]

☒ New Survey Model

Enter survey model name:

☐ Update Existing Survey Model

Select period/scenario associations you want to export from:

Select All

Clear All

< Back Next > Finish Cancel Help

a. Select the model to be exported.

b. Select one of the following:

- **New Survey Model**, and then enter a new survey model name.

You will use this name to access the model while working with the survey.

*Note:* You can export survey data multiple times for the same model—for example, once for each period in the model. In this case, you would use a different survey model name for each export.

- **Update Existing Survey Model**, and then select an existing survey model name.

This option is for updating the following two tables related to attributes: ValueAttribute table and ValueAttributeAssociation table. The two tables must have been exported previously.

c. Select the Period/Scenario associations to be exported.

d. Click **Next**.

3. Select the tables to export.

*Note:* You cannot unselect the required tables.

**Survey Export - Select Tables** Step 2 of 7

Select the tables to export. Tables which are required are marked with asterisk[\*].

Source Table	Target Table
<input checked="" type="checkbox"/> *Module	WMXXXX_Module
<input checked="" type="checkbox"/> *Account	WMXXXX_Account
<input checked="" type="checkbox"/> *Assignment	WMXXXX_Assignment
<input checked="" type="checkbox"/> *EnteredCostElement	WMXXXX_EnteredCostElement
<input checked="" type="checkbox"/> *Driver	WMXXXX_Driver
<input checked="" type="checkbox"/> ValueAttribute	WMXXXX_ValueAttribute
<input checked="" type="checkbox"/> ValueAttributeAssociation	WMXXXX_ValueAttribAssn

Select All

Clear All

Note: You can not rename the target tables names. XXXX in the table name will replace by id value.

< Back Next > Finish Cancel Help

If you plan to survey Numeric Attributes, then you also must check the following two tables:

#### ValueAttribute table

See “ValueAttribute table” on page 141.

#### ValueAttributeAssociation table

See “ValueAttributeAssociation table” on page 141.

4. Select common dimension columns (that is, select the columns that will always be exported for each table).

*Note:* You cannot unselect required columns.

**Survey Export - Select Common Dimension Columns** Step 3 of 7

Select the dimension columns to export for each table which includes dimension information. Select the most common dimension columns you wish to export. Columns which are required are marked with asterisk[\*].

Column Name
<input checked="" type="checkbox"/> *DimRef
<input type="checkbox"/> DimName
<input checked="" type="checkbox"/> *DimMemberRef
<input type="checkbox"/> DimMemberName
<input type="checkbox"/> DimLevel
<input type="checkbox"/> DimLevelName

< Back Next > Finish Cancel Help

Columns that you select are selected by default on the next Export Wizard page. You can, however, change your selection on the next page. That is, you can deselect a field that you had selected, or select a field that you had deselected.

**Export - Select Common Dimension Columns** Step 4 of 8

Select the dimension columns to export for each table which includes dimension information. Select the most common dimension columns you wish to export.

Column Name
<input checked="" type="checkbox"/> DimRef
<input checked="" type="checkbox"/> DimName
<input checked="" type="checkbox"/> DimMemberRef
<input checked="" type="checkbox"/> DimMemberName
<input type="checkbox"/> *DimLevel
<input type="checkbox"/> DimLevelName

Select these...

**Export - Select Dimension Columns by Table** Step 5 of 8

For each table which contains module and dimension specific information, choose the dimension columns to include in the exported tables. Fields which are required for importing are marked with an asterisk [\*].

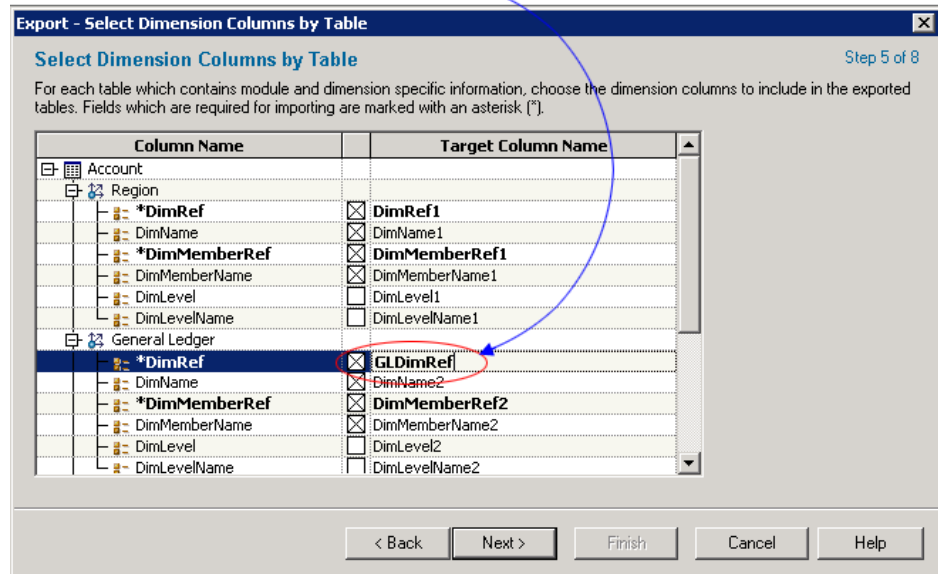
Column Name	Target Column Name
Account	
Region	
*DimRef	<input checked="" type="checkbox"/> DimRef1
DimName	<input checked="" type="checkbox"/> DimName1
*DimMemberRef	<input checked="" type="checkbox"/> DimMemberRef1
DimMemberName	<input checked="" type="checkbox"/> DimMemberName1
DimLevel	<input type="checkbox"/> DimLevel1
DimLevelName	<input type="checkbox"/> DimLevelName1
General Ledger	

and they are selected by default on the next Wizard page

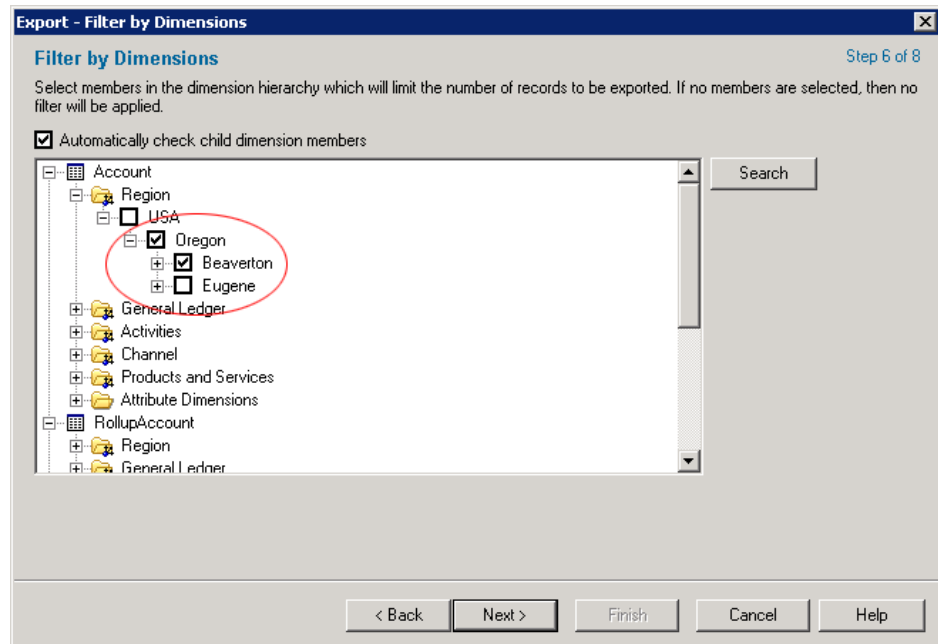
5. Select dimension columns to export for each table.

*Note:* You can overwrite the name of the target column.

You can overwrite the target name



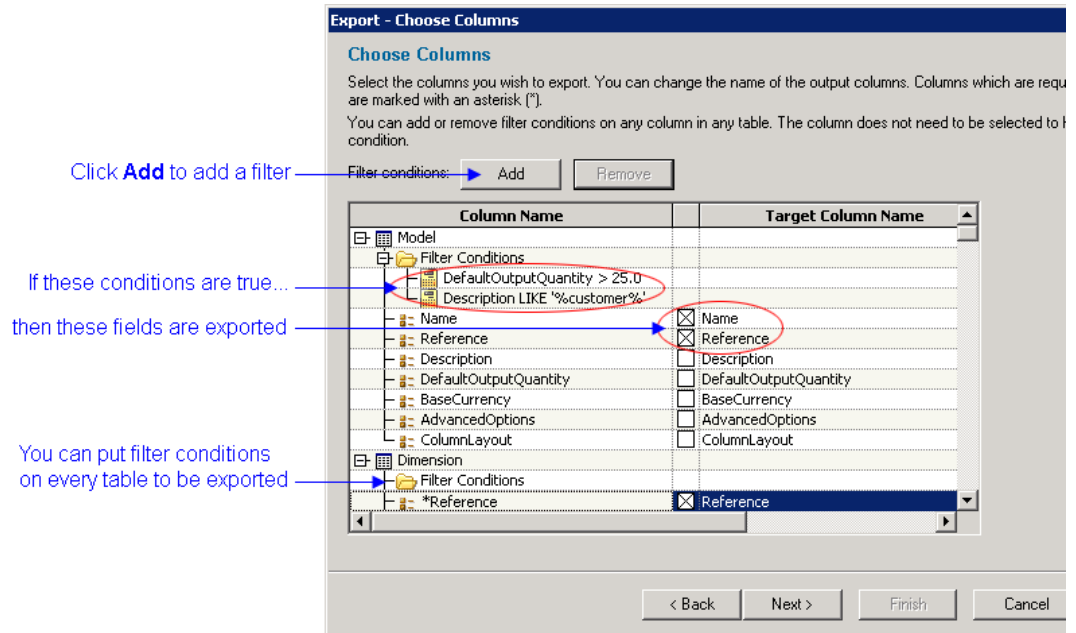
6. Filter by dimensions (that is, select those dimension members that will be exported for each table).



7. Choose the columns to export for each table.

Click **Add** to add filters to further select the columns to export. Selected columns are exported only if they pass the filter. In other words, for a column to be exported, it must both

- be selected
- pass whatever filters exist for the table



In addition to all default selections, make sure the following columns are checked:

#### Account table

DriverName  
Name  
OutputQuantityUE  
PeriodicNote (optional - only if you have Account Notes)

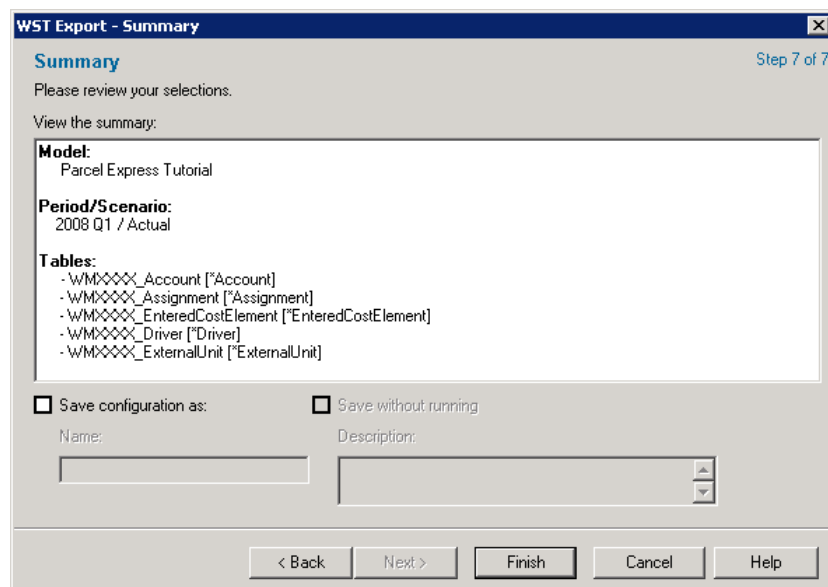
#### Assignment table

Source Accounts.DriverName  
DriverQuantityFixed

#### EnteredCostElement table

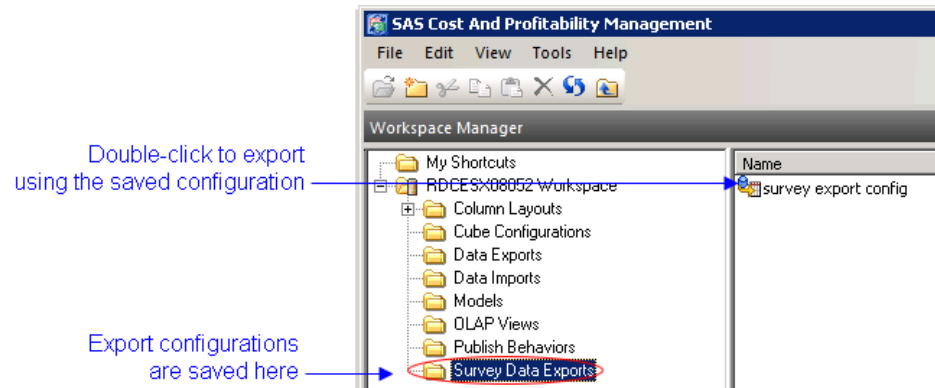
EnteredCost  
EnteredUnitCost

8. Verify the summary and click **Finish**.



Select **Save configuration as** to save your selections. The selections are saved in the **Survey Data Exports** folder.

Double-click a saved configuration to begin exporting using the saved options. You can modify the options while using the Export Wizard.





## Chapter 5

# Surveys: User Interface

---

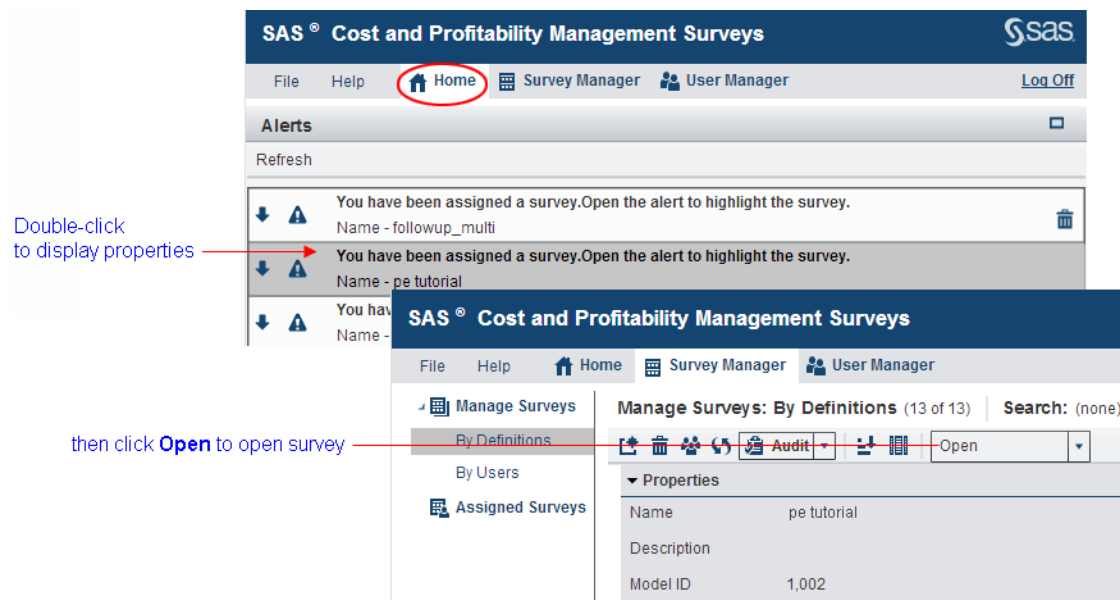
<b>Home</b> .....	<b>53</b>
<b>Survey Manager</b> .....	<b>53</b>
<b>User Manager</b> .....	<b>54</b>

---

## Home

The Home page shows you all the alerts that you have received.

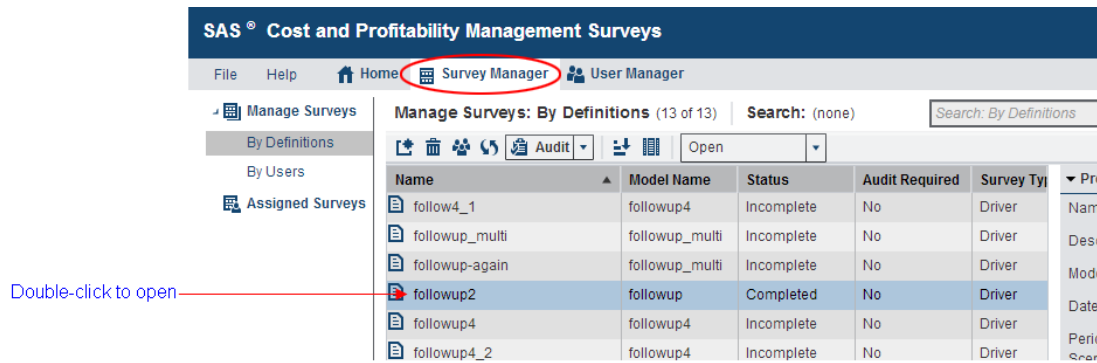
Double-click an alert to open the corresponding item.



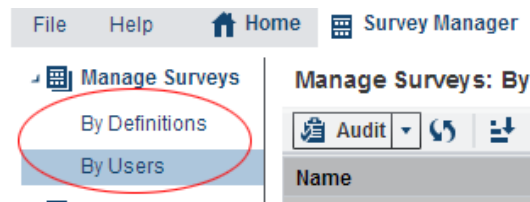
## Survey Manager

The Survey Manager lists surveys. Double-click a survey to open it. Use the Survey Manager to

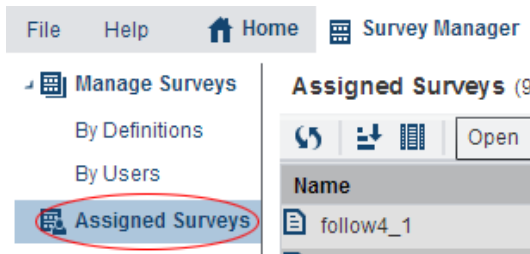
- Create a survey. See “Create the Survey” on page 57.
- Take a survey. See “Take a Driver Survey” on page 69.
- Audit surveys. See “Audit Surveys” on page 83.
- View approved assignments. See “View Approved Assignments” on page 84.
- Delete an assignment. See “Delete a Survey” on page 88.
- Send an e-mail. See “Send an e-mail” on page 85.



You can choose to list surveys by definition or by user.



Click **Assigned Surveys** to see the surveys assigned to you.



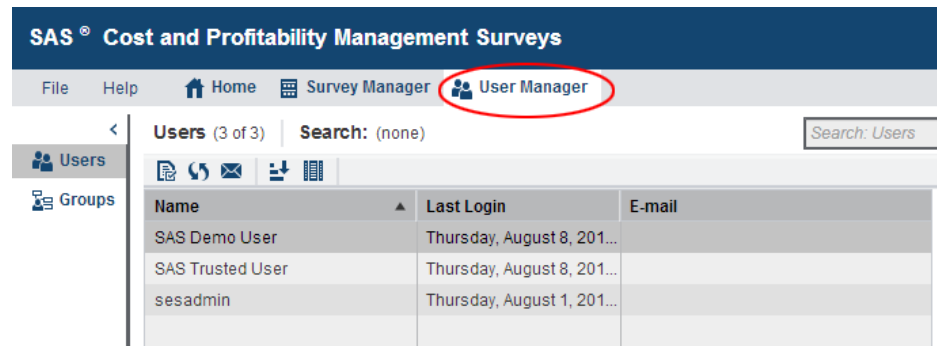
## User Manager

The User Manager is available only to users who have Create Model capability. There is no Administer Surveys capability. For a discussion of roles and capabilities, see “Creating Roles” on page 4.

Use the User Manager to

- Reassign survey items to users. See “Reassign a Survey to Another User” on page 77.

- Manage user attributes. See “Add Attributes to a User” on page 78.
- Send e-mails to individual users. See “Send an e-mail” on page 85.



The screenshot displays the SAS User Manager interface. At the top, a dark blue header bar contains the text "SAS® Cost and Profitability Management Surveys". Below this, a light gray navigation bar includes links for "File", "Help", "Home", "Survey Manager", and "User Manager". The "User Manager" link is circled in red. On the left side, a sidebar shows "Users" and "Groups" options. The main content area is titled "Users (3 of 3)" and includes a search bar with the text "Search: (none)". Below the search bar, there is a table with three columns: "Name", "Last Login", and "E-mail". The table lists three users: "SAS Demo User", "SAS Trusted User", and "sesadmin".

Name	Last Login	E-mail
SAS Demo User	Thursday, August 8, 201...	
SAS Trusted User	Thursday, August 8, 201...	
sesadmin	Thursday, August 1, 201...	



## Chapter 6

# Creating Surveys

---

<b>Create the Survey</b> .....	<b>57</b>
General Procedure .....	57
Driver Survey .....	59
Entered Cost Element Survey .....	60
Attribute Survey .....	60
Output Quantities Survey .....	61
<b>Create a Group Survey</b> .....	<b>62</b>
<b>Create Follow-up Surveys</b> .....	<b>63</b>
<b>Create an Aggregated Survey</b> .....	<b>64</b>

---

## Create the Survey

### General Procedure

To create a survey, a user must have Create Model capability. There is no Create Survey capability. For a discussion of roles and capabilities, see [“Creating Roles” on page 4](#).

To create a survey:

1. Log on using a browser.

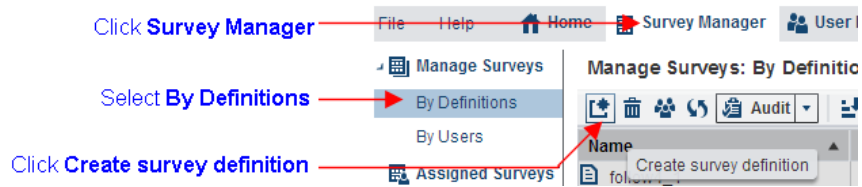
The logon URI uses the form: `http://machine_name:port/SASCostAndProfitabilityManagementSurveys`.

The machine name depends on where the SAS Cost and Profitability Management middle-tier server is installed, for example:

`http://abcdef.xyz.com:8880/SASCostAndProfitabilityManagementSurveys`

See [“Logging On” on page 40](#).

2. Click **Survey Manager**.
3. Select **By Definitions**.
4. Select **Create a survey definition**.



5. Name the survey and (optionally) provide a description, and then click **Next**.

The 'New Definition' form is shown, indicating 'Step 1 of 3'. The title is 'Definition details'. The instruction is 'Select name and description of definition'. There are two input fields: 'Name' with a red asterisk indicating it is required, and 'Description'. Both fields are currently empty.

6. Select the model to survey.
7. Select the period/scenario associations to survey.
8. Select the module to survey.

The 'Filters' form is shown, indicating 'Step 2 of 3'. The instruction is 'Select necessary details to proceed'. There are three selection sections:
 

- Select Model:** A dropdown menu with 'PE' selected. A blue arrow points from the text 'Select a model' to this dropdown.
- Select Period/Scenario:** A list box containing '2008 Q1/Actual' and '2008 Q1/Plan'. A blue arrow points from the text 'Select periods' to this list box.
- Select module:** A dropdown menu with 'Resource' selected. A blue arrow points from the text 'Select module' to this dropdown. The dropdown menu is open, showing options: 'External Unit', 'Resource' (highlighted), 'Activity', and 'CostObject'.

9. The next steps depend on the type of survey that you want to define.

The steps for each type of survey are described below:

- “Driver Survey” on page 59
- “Entered Cost Element Survey” on page 60
- “Attribute Survey” on page 60
- “Output Quantities Survey” on page 61

Driver Survey

With a Driver survey, a survey taker can update the following field in the Assignment table (see “Assignment table” on page 122):

Field
DriverQuantityFixed

To create a Driver survey, first follow the general procedure for creating any survey. See “General Procedure” on page 57.

To finish creating a Driver survey, do the following:

1. Select **Driver** for the **Type of Survey**.
2. For **Top-Level Dimension Reference**, do one of the following:

Specify **Any Dimension**

All accounts in the module are eligible for assignment to survey owners

Select a specific dimension

Only those accounts in the module under the specified dimension are eligible for assignment to survey owners.

*Note:* A restricted list of dimension members is displayed in the drop-down list of dimension members available for filtering. The number of dimensions displayed is also affected by the application preference **Max Top-Level Dimensions**. See “Application Preferences” on page 93.

3. For **Driver Name**, do one of the following:

Specify **Any Driver**

All accounts in the module with a driver are presented for assignment to survey owners

Select a specific driver

Only those accounts in the module using the selected driver are presented for assignment to survey owners.

4. Click **Next**.

5. Assign source accounts for the driver to survey owners.

*Note:* If you assign a survey to yourself, then you must click Refresh for the survey to show up in the list of surveys assigned to users.

### Entered Cost Element Survey

With an Entered Cost Element survey, a survey taker can update the following fields in the EnteredCostElement table (see “[EnteredCostElement table](#)” on page 132):

#### Field

EnteredCost

EnteredUnitCost

To create an Entered Cost Element survey, first follow the general procedure for creating any survey. See “[General Procedure](#)” on page 57.

To finish creating an Entered Cost Element survey, do the following:

1. Select **Entered Cost and Unit Cost** for the **Type of Survey**..

The screenshot shows a web-based configuration form. At the top, there is a section labeled 'Type of Survey:' with a dropdown menu currently displaying 'Entered Cost and Unit cost'. Below this is a section labeled 'Define scope of query' which contains a 'Top-level Dimension Reference:' dropdown menu set to 'Any Dimension'. A help icon (question mark in a circle) is visible to the right of the dropdown.

2. For **Top-Level Dimension Reference**, do one of the following:

#### Specify **Any Dimension**

All Resource accounts are eligible for assignment to survey owners.

#### Select a specific dimension

Only those Resource accounts under the specified dimension are eligible for assignment to survey owners.

*Note:* A restricted list of dimension members is displayed in the drop-down list of dimension members available for filtering. The number of dimensions displayed is also affected by the application preference **Max Top-Level Dimensions**. See “[Application Preferences](#)” on page 93.

3. Click **Next**.
4. Assign accounts to survey owners.

### Attribute Survey

With an Attribute survey, a survey taker can update the following fields in the ValueAttributeAssociation table (see “[ValueAttributeAssociation table](#)” on page 141):

#### Field

NumericValue



**Field**

StringValue

To create an Attribute survey, first follow the general procedure for creating any survey. See [“General Procedure” on page 57](#).

To finish creating an Attribute survey, do the following:

1. Select **Attribute** for the **Type of Survey**.

Type of Survey:  
Attribute

Attribute Type:  
Numeric

Attribute Name:  
Cost per Inspection

2. Specify the following:

**Attribute Type**

If you select **Numeric**, then only numeric attributes are listed for selection.

If you select **Text**, then only text attributes are listed for selection.

**Attribute Name**

Select an attribute for surveying. All accounts in the selected module that have that attribute attached are included in the survey.

See [“ValueAttributeAssociation table” on page 141](#).

3. Click **Next**.
4. Assign accounts to survey owners.

## Output Quantities Survey

With an Output Quantities Survey, a survey taker can update the following field in the Account table (see [“Account table” on page 119](#)):

**Field**

OutputQuantityUE

To create an Output Quantities Survey, first follow the general procedure for creating any survey. See [“General Procedure” on page 57](#).

To finish creating an Output Quantity Survey, do the following:

1. Select **Output Quantities**.

Type of Survey:  
 Output Quantities ▼

Define scope of query  
 Account Name Contains:

2. For **Account Name Contains** (under **Define scope of query**), you can limit the accounts to survey by specifying a character string that each account's name must contain.

The character string can contain blanks, and case does not matter. The string can occur at the beginning, in the middle, or at the end of the account name.

3. Click **Next**.
4. Assign accounts to survey owners.

## Create a Group Survey

In addition to assigning a survey to an individual survey taker, you can assign it to a group. Then, when all the members of the group have completed the survey, their individual responses are automatically added together and entered into the database.

You can assign only driver surveys (for any module) to a group. And, you can assign any type of driver survey to a group except for Percentage drivers. This means that if you select **Any Driver** for the driver name, and if any of the drivers to be assigned is a Percentage driver, then you can not assign the survey to a group.

Driver Name:

As an example, suppose there is a group of survey takers (named Group of Surveyors) with two members: Ashwin and Chirayu. For information about groups, see [“Creating Groups”](#) on page 12.

Name	Identity Name
Group of Surveyors	Group of Surveyors
Ashwin	Ashwin
Chirayu	Chirayu

Suppose also that you create a Resource Driver survey for a particular driver and assign the Operating Expenses account to the group.

Source Name(Reference)	Owner
Beaverton x Equipment Expenses_1039	SAS Trusted User
Beaverton x Operating Expenses_1038	SAS Trusted User
Beaverton x Wages_1037	Group of Surveyors

Then, both Ashwin and Chirayu will receive an e-mail informing them that they have been assigned a survey.

Suppose Ashwin takes the survey, and for each destination account assigns a quantity of 1.

Measure Name	FTEs
<input checked="" type="checkbox"/> Beaverton x Air Distribution_2020	1.0000
<input checked="" type="checkbox"/> Beaverton x Expedite Package Shipments_2016	1.0000
<input checked="" type="checkbox"/> Beaverton x Inspect_2019	1.0000

Suppose also that when Chirayu takes the survey, he assigns a quantity of 2 to each of the same destination accounts.

Measure Name	FTEs
<input checked="" type="checkbox"/> Beaverton x Air Distribution_2020	2.0000
<input checked="" type="checkbox"/> Beaverton x Expedite Package Shipments_2016	2.0000
<input checked="" type="checkbox"/> Beaverton x Inspect_2019	2.0000

The Driver Quantity Fixed that is written to the Assignment staging table is 3—the sum of the quantities entered by both Ashwin and Chirayu.

## Create Follow-up Surveys

If you select the model preference, **Follow-up driver surveys**, then whenever you create a Driver survey and assign it to a survey taker for a source account, then if there are assignments from the destination account to other accounts the following occurs:

- a survey is automatically created from the destination account to the other accounts
- the survey is assigned to the same survey taker
- the system looks at the other (new destination) accounts to see if there are assignments from those accounts to still other accounts. And, if so, it creates a survey

to those other (new destination) accounts. And so on from those other (new destination) accounts.

In short, follow-up surveys are created for as long as there is an uninterrupted chain of assignments.

For more information, see [“Follow-up Driver Surveys” on page 98](#).

---

## Create an Aggregated Survey

In an *aggregated survey* the Driver Quantity Fixed (DQF) that is written to a staging table is calculated (aggregated) according to a formula. The formula allows you to fix a weight to the relative contributions to the DQF of the participants who determine the DQF.

*Note:* Do not confuse aggregated surveys with calculated drivers. Unlike a driver formula, the formula in an aggregated survey is not attached to the driver whose DQF is being surveyed. The formula exists only when the survey is being taken.

You can create an aggregated survey for a standard driver because its Driver Driven Cost for each assignment is determined by the ratio of the DQF for each assignment relative to the total DQF for all assignments—not by the absolute value of each DQF. The formula for an aggregated survey calculates the relative contribution to the DQF—not the absolute value of the DQF.

An aggregated survey allows one survey taker to answer for multiple people and to assign a weight to the contribution of each type of person.

Aggregated surveys are best explained through an example. The following text describes one such survey in detail.

*Note:* In an aggregated survey, values must total 100% if you selected the application preference **Enforce hundred percent for percentage values**. See [“Application Preferences” on page 93](#).

1. Add attributes to a user. For information, see [“Add Attributes to a User” on page 78](#).

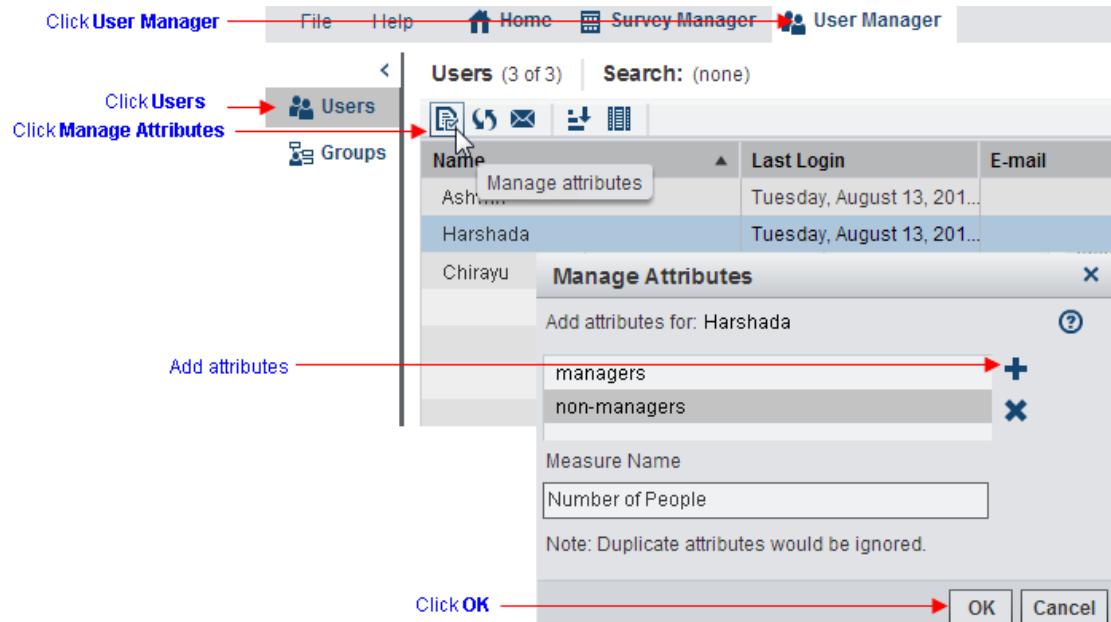
*Note:* The attributes that you add to a user for surveys have nothing to do with the attributes that you add to accounts in SAS Cost and Profitability Management. The attributes that you add to a user for surveys are not written to any staging table.

- a. Click **User Manager**.
- b. Click **Users**.
- c. Click **Manage Attributes**.
- d. Add attributes.

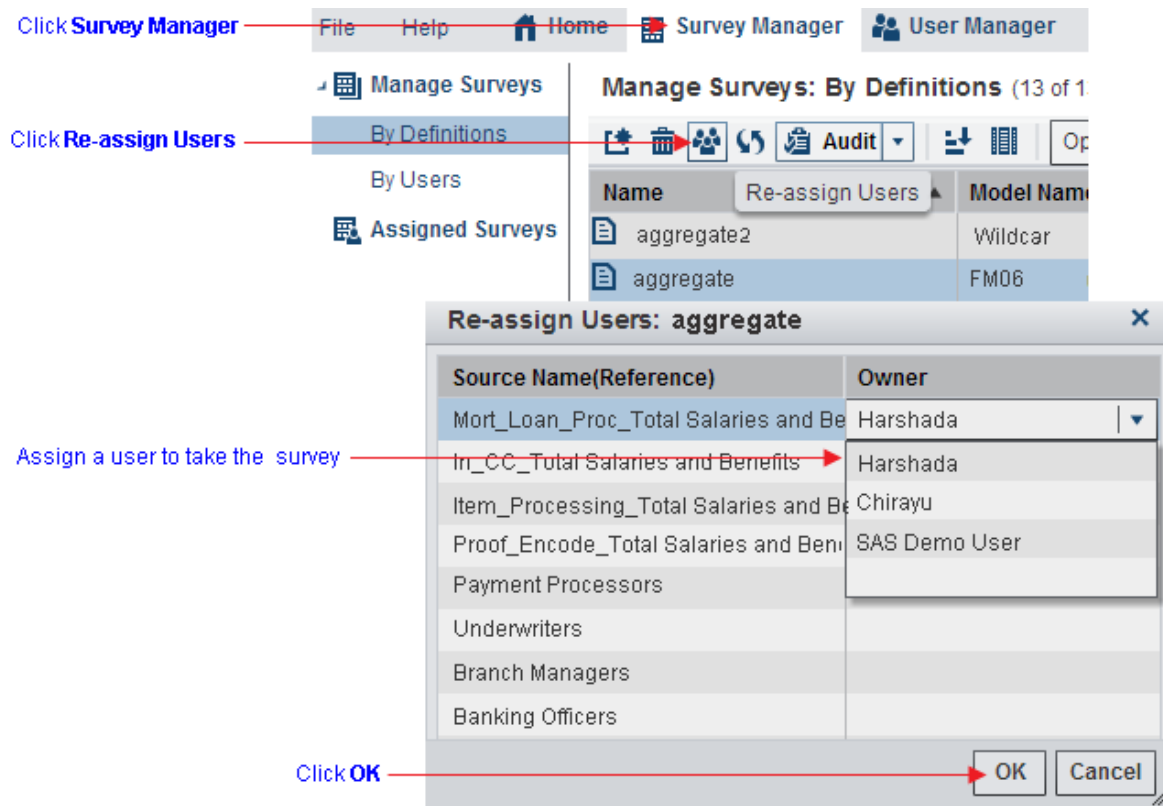
*Note:* The number of attributes that can be added is limited by the application preference **Maximum Attributes**. See [“Application Preferences” on page 93](#).

- e. Click **OK**.

The following picture shows how to add two attributes, *managers* and *non-managers*, to the user Harshada.



2. Assign a survey to the user with the attributes.



3. The user takes the survey.

Double-click the survey →

Double-click the source account →

Specify data for destination accounts

Period/Scenario(s): 2029_Q1/Actual		Percentage values are based on: Number of Hours	
Destination Name (reference)	managers	non-managers	DQF(Weighted)
<input type="checkbox"/> Research Payments(PMT-003)	25	30	23.68
<input type="checkbox"/> Perform Payment Balancing(PMT..)	35	40	23.74
<input type="checkbox"/> Enter Payment Data(PMT-001)	40	30	31.58
Parameter	managers	non-managers	
Total	100%	100%	
Number of People	4	75	
Relative Weighting	3	30	

This picture shows the user, Harshada, entering the following values for the percent of DQF accounted for by managers and non-managers, respectively:

Account Name (Reference)	Percent of DQF accounted for by managers	Percent of DQF accounted for by non-managers
Enter Payment Data(PMT-001)	25	30
Perform Payment Balancing(PMT-002)	35	40
Research Payments(PMT-003)	40	30
<b>Total percent:</b>	100	100

There are two things to notice about these numbers:

- They are entered as percentages regardless of the type of driver. However, the resulting DQF that is calculated by the system represents the unit of measure for the driver being surveyed. In the case of this example, the unit of measure is Number of Hours. It is this number that is stored in the Assignment staging table.

Number of Hours
DQF(Weighted)
23.68
23.74
31.58

- The total percentage equals 100 in the case of both managers and non-managers. Whether the total must equal 100% or not is determined by the application preference, **Enforce hundred percent for percentage values**. See “[Application Preferences](#)” on page 93.

The previous picture also shows a user entering the following values for the number of managers vs. non-managers and the relative weighting to be allocated to managers vs. non-managers.

Parameter	Managers	Non-managers
Number of people	4	75
Relative weighting	3	30

To explain the formula that is used to derive the resulting DQF, we can label the cells in the previous two tables as follows:

Account Name (Reference)	Percent of DQF accounted for by managers	Percent of DQF accounted for by non-managers
Enter Payment Data(PMT-001)	A: 25	U: 30
Perform Payment Balancing(PMT-002)	B: 35	V: 40
Research Payments(PMT-003)	C: 40	W: 30
<b>Total percent:</b>	D: 100	X: 100
Number of people	E: 4	Y: 75
Relative weighting	F: 3	Z: 30

The following picture shows the labelling

Period/Scenario(s):2029_Q1/Actual		Percentage values are based on :Number of Hours		
<input type="checkbox"/>	Destination Name (reference) ▾	managers	non-managers	DQF(Weighted)
<input type="checkbox"/>	Research Payments(PMT-003)	A: 25	U: 30	23.68
<input type="checkbox"/>	Perform Payment Balancing(PMT-002)	B: 35	V: 40	23.74
<input type="checkbox"/>	Enter Payment Data(PMT-001)	C: 40	W: 30	31.58
Parameter		managers	non-managers	
Total		D: 100%	X: 100%	
Number of People		E: 4	Y: 75	
Relative Weighting		F: 3	Z: 30	

The formula used to derive the resulting DQF for the first account (Enter Payment Data) is the following:

$$\begin{aligned} & ((A/D) * E * F) + ((U/X) * Y * Z) \\ & ((25/100) * 4 * 3) + ((30/100) * 75 * 30) = 3 + 675 = 678 \end{aligned}$$

divided by:

$$\begin{aligned} & (E * F) + (Y * Z) \\ & (4 * 3) + (75 * 30) = 12 + 2250 = 2262 \end{aligned}$$

times:

$$\begin{aligned} & ((A/D) * E) + ((U/X) * Y) \\ & + ((B/D) * E) + ((V/X) * Y) \\ & + ((C/D) * E) + ((W/X) * Y) \\ & ((25/100) * 4) + ((30/100) * 75) = 1 + 22.5 = 23.5 \\ & + ((35/100) * 4) + ((40/100) * 75) = 1.4 + 30 = 31.4 \\ & + ((45/100) * 4) + ((30/100) * 75) = 1.6 + 22.5 = 24.1 \\ & 23.5 + 31.4 + 24.1 = 79 \end{aligned}$$

or:

$$(678/2262) * 79 = 23.679045$$

The result is rounded to the number of decimal places as specified in the application preference, **Maximum Decimal Places**. See [“Application Preferences” on page 93](#). In the case of this example, it is rounded to 23.68.

The resulting DQF is calculated in similar fashion for the other two accounts. The resulting DQF for the three assignments are:

- **23.68**
- **23.74**
- **31.58**

And the total Driver Quantity Calculated is **23.68+23.74+31.58=79**.

So, the Driver Rates for the three assignments are:

- **23.68/79=0.2997**
- **23.74/79=0.3005**
- **31.58/79=0.3997**



## Chapter 7

# Taking a Survey

---

<b>Take a Driver Survey</b> .....	<b>69</b>
General Procedure .....	69
Add New Assignments .....	72
<b>Take a Non-Driver Survey</b> .....	<b>73</b>
<b>Submit a Survey</b> .....	<b>75</b>

---

## Take a Driver Survey

### General Procedure

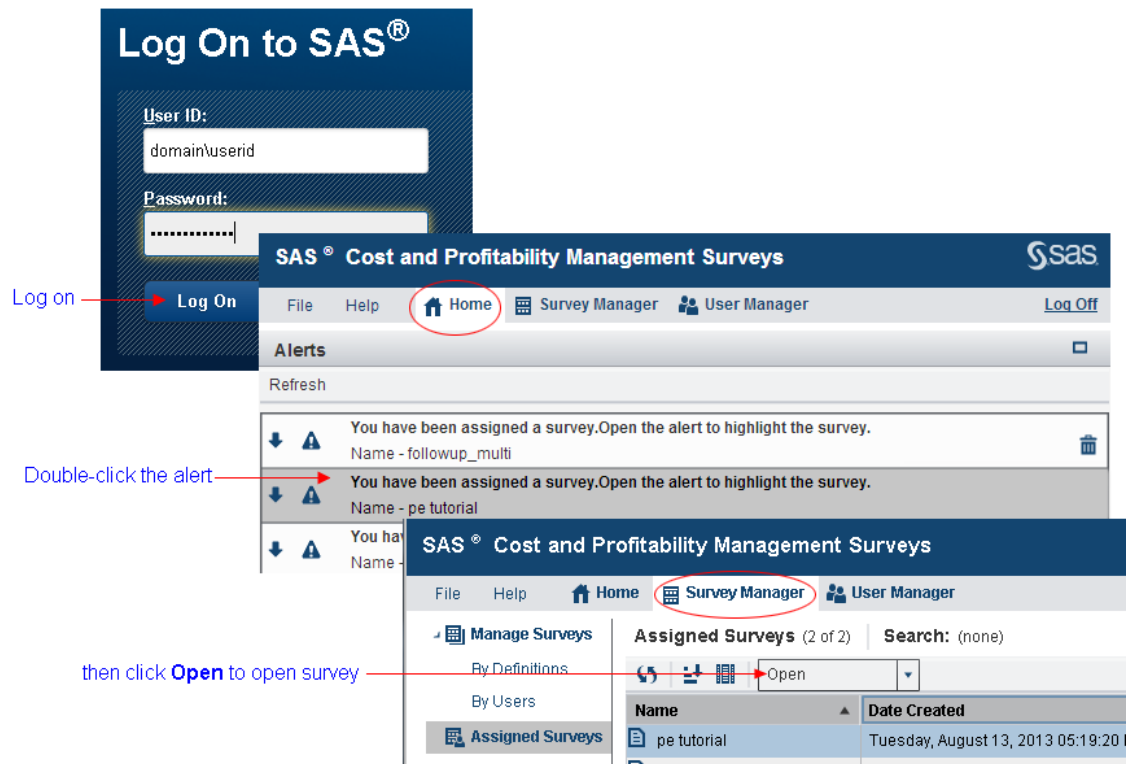
For information on creating a driver survey, see [“Driver Survey” on page 59](#).

To take any survey, a user must have the Take Surveys capability. For a discussion of roles and capabilities, see [“Creating Roles” on page 4](#).

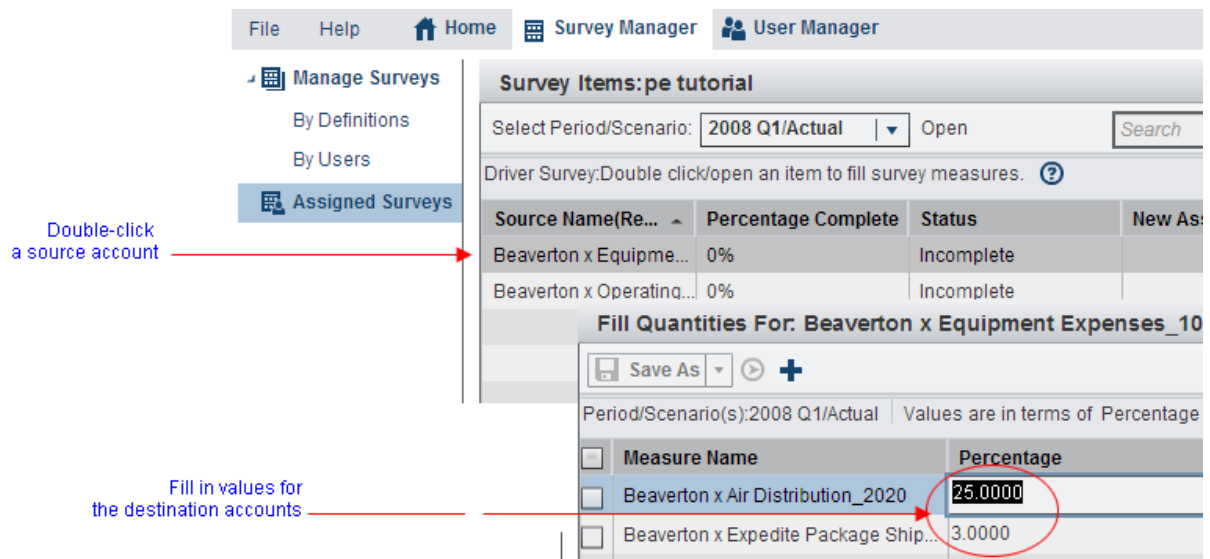
When the survey creator assigns a survey, the survey taker receives a notification email. The email contains a link to the survey. After clicking on the link, the survey taker:

1. Logs on to SAS Cost and Profitability Management Surveys.
2. In the Home page, double-clicks the survey notification to open the Survey Manager. The survey to take is highlighted.
3. Double-clicks the highlighted survey to open and take it.

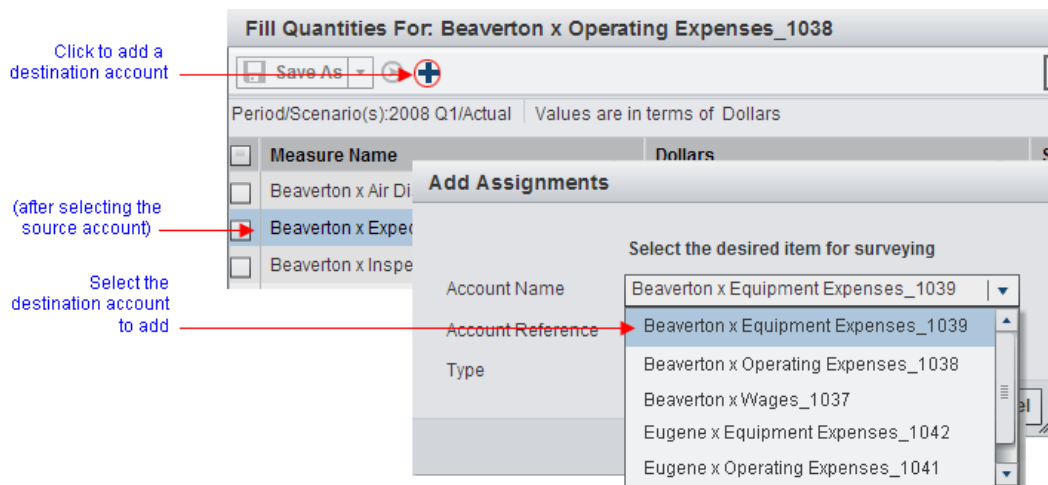
*Note:* You can open a survey directly from the Survey Manager.



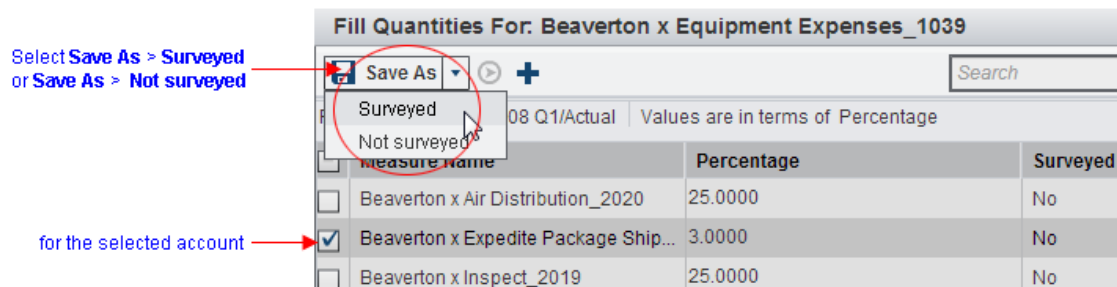
4. Double-click a source account.
5. Fill in values for the destination account.



6. If adding new destination accounts is allowed and you want to add a new destination account, then click the plus sign.
- See “Allow a Survey Taker to Add New Destination Accounts” on page 96.
7. To add a new destination account, select the desired account, and then click **OK**.



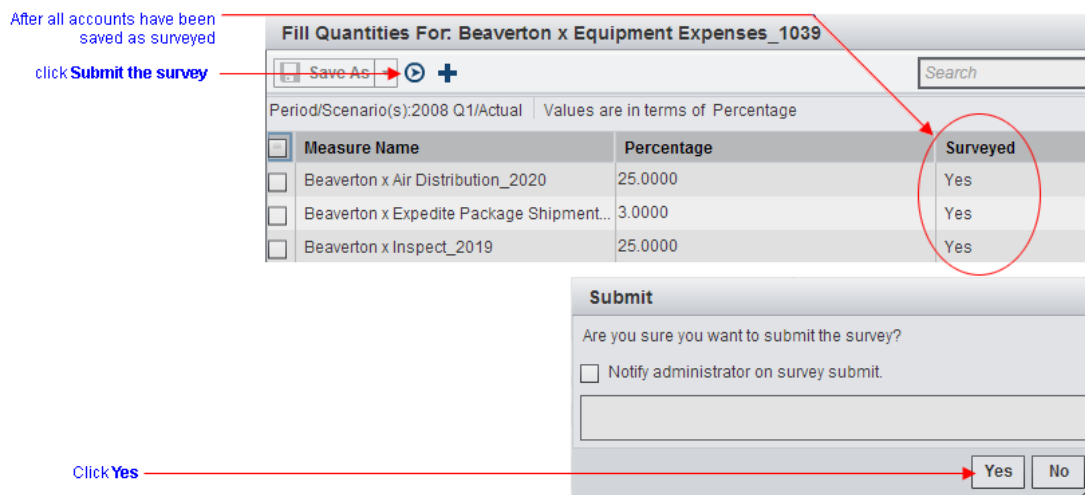
8. For each account, select **Save As** ⇒ **Surveyed** or **Save As** ⇒ **Not surveyed**.



The Survey Manager displays the Percentage Complete—the percentage of survey items that have been saved as surveyed.

*Note:* When a user selects **Save As** ⇒ **Surveyed**, then values are written to staging tables in the database. When a user selects **Save As** ⇒ **Not surveyed**, then nothing happens to staging tables—in particular, the database is not rolled back.

Once all items have been saved as surveyed, the survey taker can submit the survey.



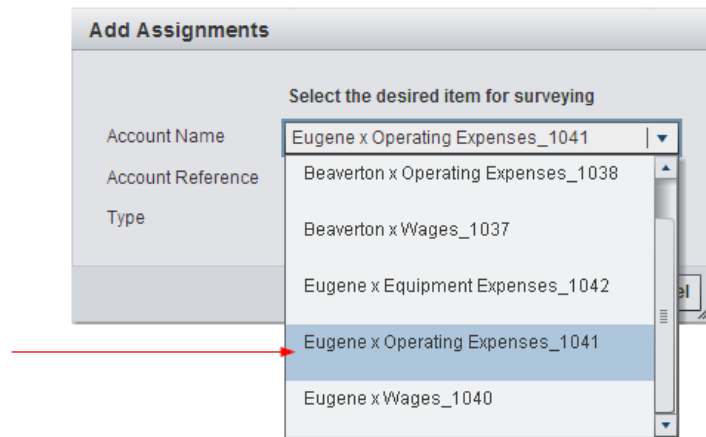
See “Submit a Survey” on page 75.



3. Select a new destination account from the drop-down list of accounts, and then click **OK**.

*Note:* The items in the drop-down list are controlled by the **Allow smart filtering of accounts** Model preference. If this option checked and there are no accounts underneath the top-level account, then the drop-down list will not display any accounts. In that case, you can turn off **Allow smart filtering of accounts** to see additional accounts. See “[Model Preferences](#)” on page 94.

3. Select a new destination account



4. The new destination account is added to the list of destinations..

4. The destination account is added to the list

The screenshot shows the 'Fill Quantities For: Beaverton x Operating Expenses\_1038' window. It includes a 'Save As' button, a search bar, and a table of measures. The table has columns for 'Measure Name', 'Dollars', and 'Surveyed'. The 'Eugene x Operating Expenses\_1041' row is circled in red.

Measure Name	Dollars	Surveyed
Beaverton x Air Distribution_2020	83,000.0000	No
Beaverton x Expedite Package Shipments...	8,000.0000	No
Beaverton x Inspect_2019	54,000.0000	No
Eugene x Operating Expenses_1041	0.0000	No

## Take a Non-Driver Survey

The following is a list of non-driver surveys:

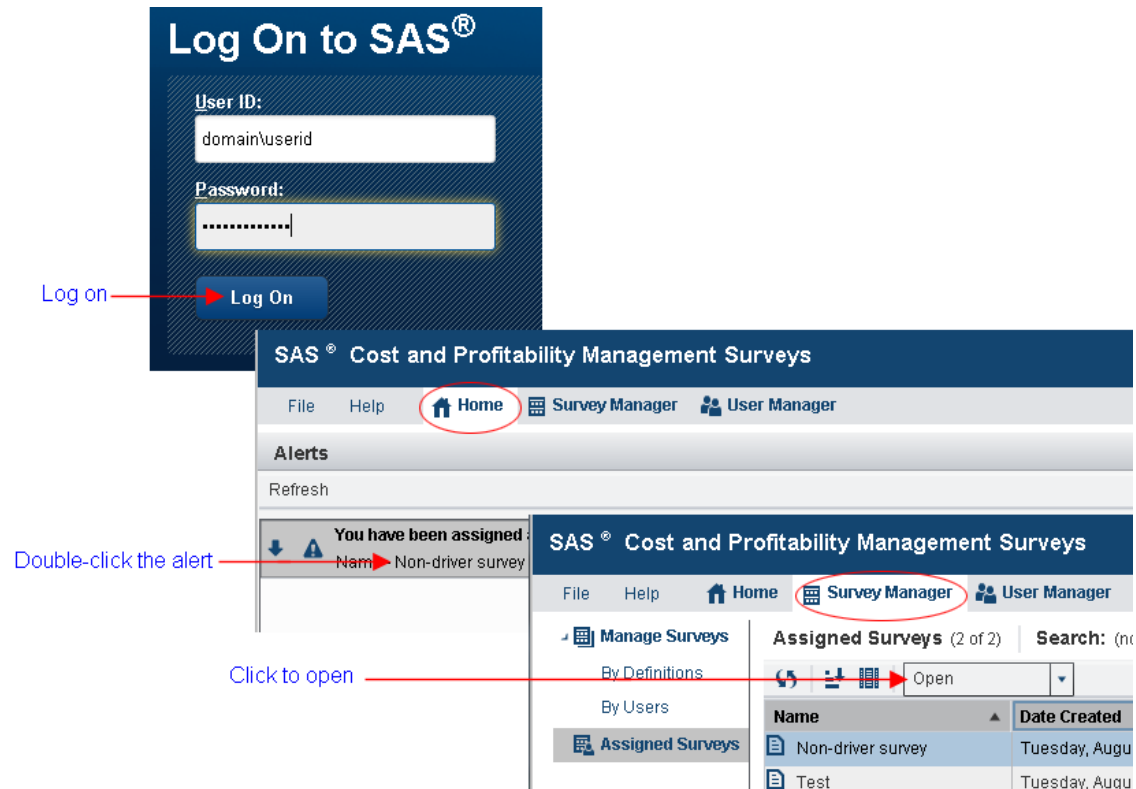
- “Entered Cost Element Survey” on page 60
- “Attribute Survey” on page 60
- “Output Quantities Survey” on page 61

To take a survey, a user must have the Take Surveys capability. For a discussion of roles and capabilities, see “[Creating Roles](#)” on page 4.

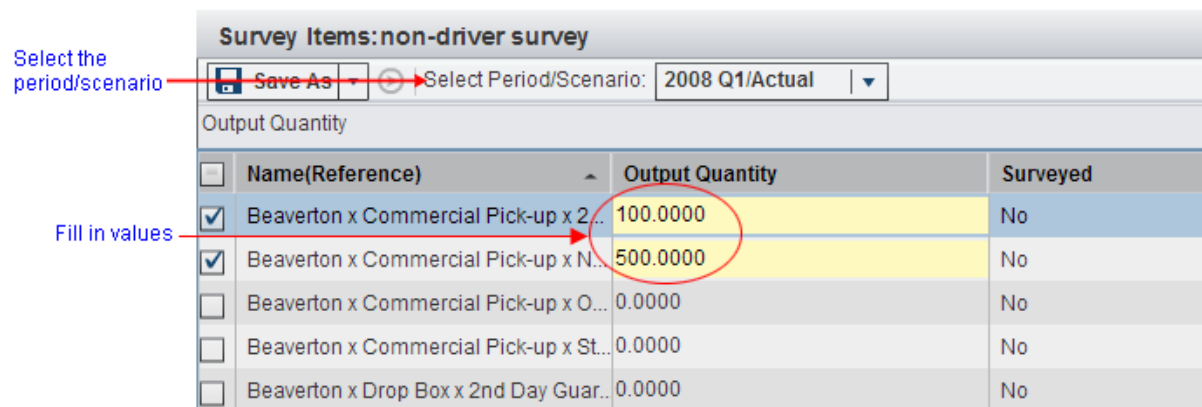
When a survey creator assigns a survey, the survey taker receives a notification e-mail. The e-mail contains a link to the survey. After clicking on the link, the survey taker

1. Logs on to SAS Cost and Profitability Management Surveys.
2. In the Home page, double-clicks the survey notification to open the Survey Manager. The survey to take is highlighted.
3. Double-clicks the highlighted survey to open and take it.

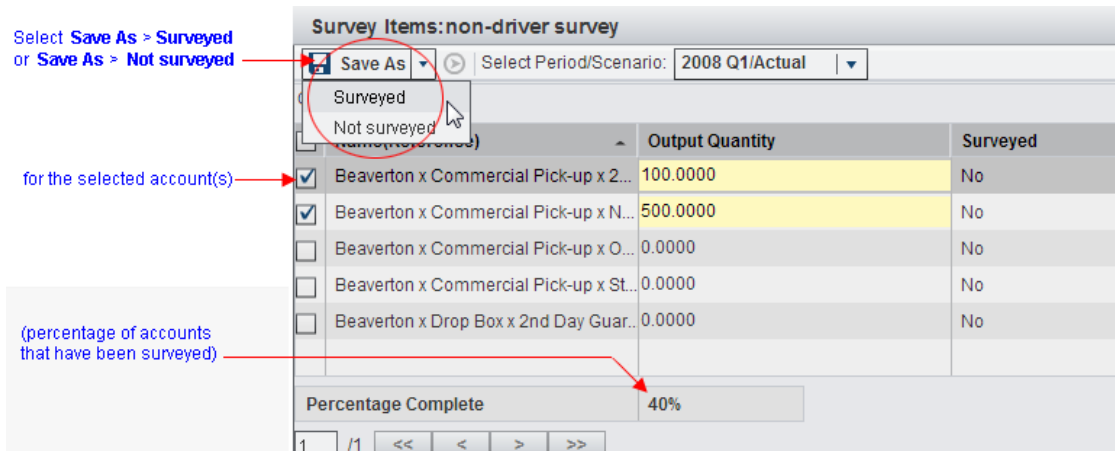
*Note:* You can open a survey directly from the Survey Manager.



4. Select the period/scenario for the survey.
5. Fill in values for the survey.



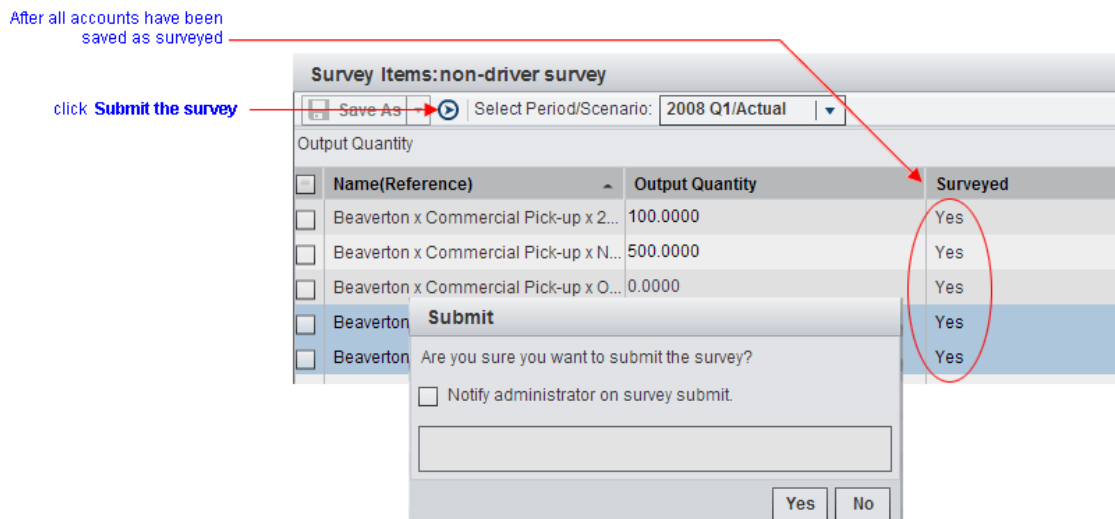
6. For each account, select **Save As** ⇒ **Surveyed** or **Save As** ⇒ **Not surveyed**.



The Survey Manager displays the Percentage Complete—the percentage of survey items that have been saved as surveyed.

*Note:* When a user selects **Save As** ⇒ **Surveyed**, then values are written to staging tables in the database. When a user selects **Save As** ⇒ **Not surveyed**, then nothing happens to staging tables—in particular, the database is not rolled back.

Once all items have been saved as surveyed, the survey taker can submit the survey.



See “Submit a Survey” on page 75.

## Submit a Survey

The following description applies to any survey. After a survey taker has saved all accounts as surveyed, the survey taker can submit the survey. To submit a survey:

1. Click **Submit the survey**.
  - a. Select **Notify administrator on survey submit** to send an alert to the survey creator.
  - b. Type text into the text field if you want it to go into an e-mail to the administrator (the survey creator).

- c. Click **Yes** to submit the survey.

After all accounts have been saved as surveyed

click **Submit the survey**

Click **Yes**

**Fill Quantities For: Beaverton x Equipment Expenses\_1039**

Save As +

Period/Scenario(s): 2008 Q1/Actual Values are in terms of Percentage

Measure Name	Percentage	Surveyed
Beaverton x Air Distribution_2020	25.0000	Yes
Beaverton x Expedite Package Shipment...	3.0000	Yes
Beaverton x Inspect_2019	25.0000	Yes

**Submit**

Are you sure you want to submit the survey?

☐ Notify administrator on survey submit.

Yes No

2. The survey status changes to **Completed**.

**Survey Items: pe tutorial**

Select Period/Scenario: 2008 Q1/Actual Open Search

Driver Survey: Double click/open an item to fill survey measures. ?

Source Name(Refer...	Percentage Complete	Status	New Assignments
Beaverton x Equipment E...	100%	Completed	
Beaverton x Operating Ex...	0%	Incomplete	

*Note:* Once a survey has been completed, the survey taker can no longer change any responses unless the survey creator changes the survey back to Incomplete. See “Mark a Survey as Incomplete” on page 82.



## Chapter 8

# Manage Users

---

Overview .....	77
Reassign a Survey to Another User .....	77
Add Attributes to a User .....	78

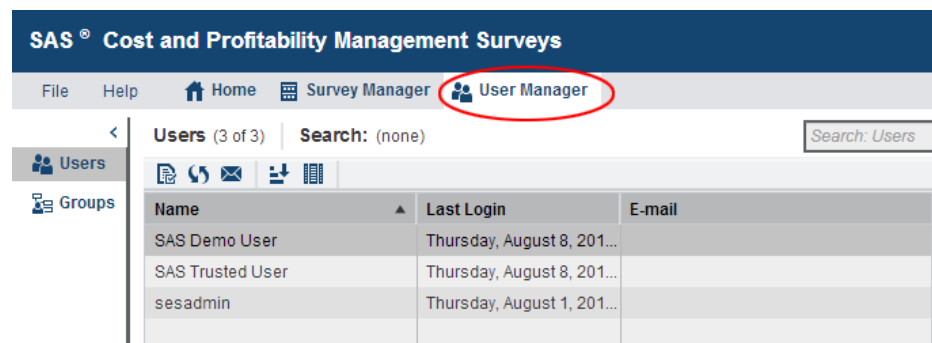
---

## Overview

The User Manager is available only to users who have Create Model capability. There is no Administer Surveys capability. For a discussion of roles and capabilities, see [“Creating Roles” on page 4](#).

Use the User Manager to

- Reassign survey items to users. See [“Reassign a Survey to Another User” on page 77](#).
- Manage user attributes. See [“Add Attributes to a User” on page 78](#).
- Send e-mails to individual users. See [“Send an e-mail” on page 85](#).

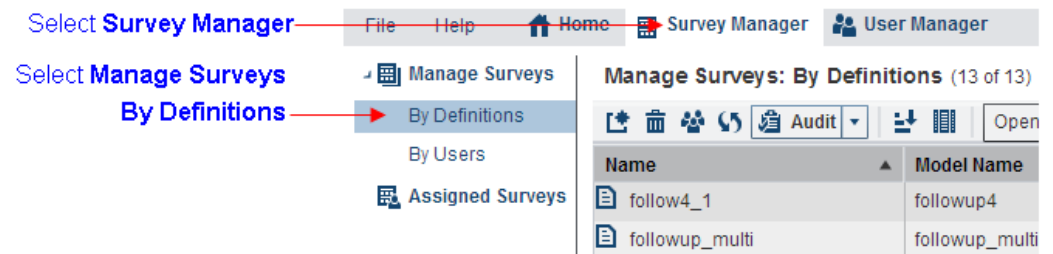


## Reassign a Survey to Another User

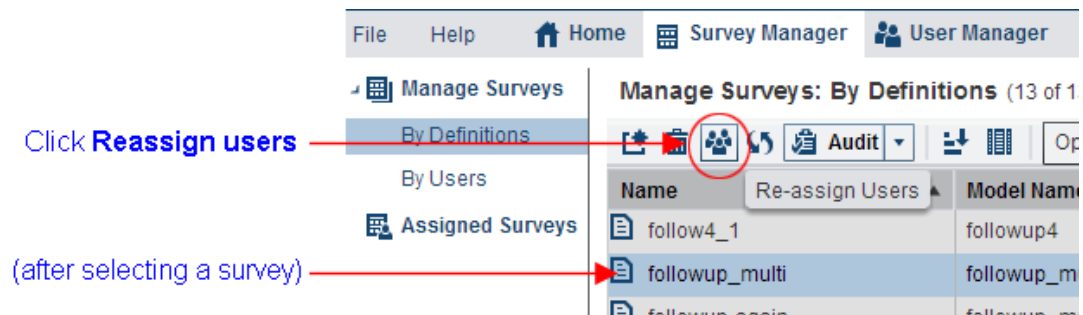
*Note:* Only the creator of a survey can reassign it.

To reassign the fields in a survey to a different user:

1. Select **Survey Manager**.
2. Select **Manage Surveys** ⇨ **By Definitions**.



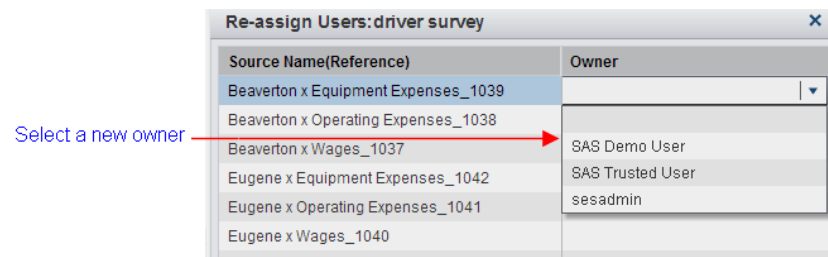
3. Select a survey.
4. Click the **Re-assign users** icon.



The Re-assign Users window appears.

5. Select a user from the drop-down list and click **OK**.

*Note:* Select **Notify users about survey assignments** to send an e-mail to the new user.

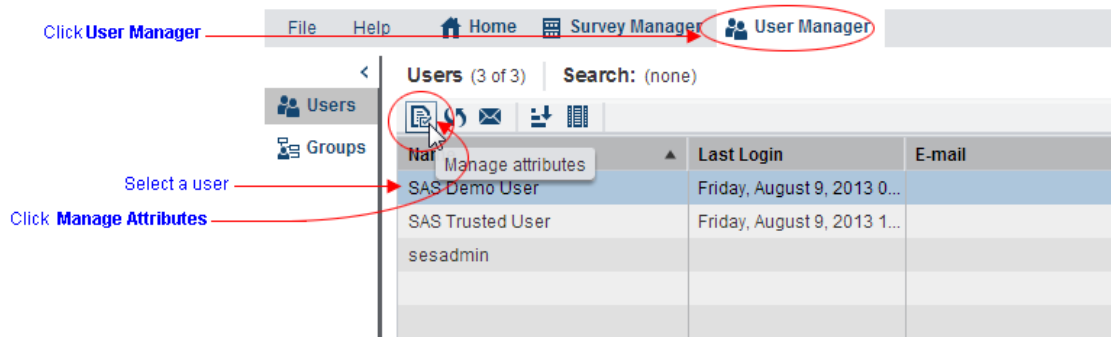


## Add Attributes to a User

You can add attributes to a survey taker so that the survey taker can take an aggregated survey. See “[Create an Aggregated Survey](#)” on page 64.

To add or delete attributes for a user:

1. Click **User Manager**.
2. Select a user.
3. Click **Manage Attributes**.

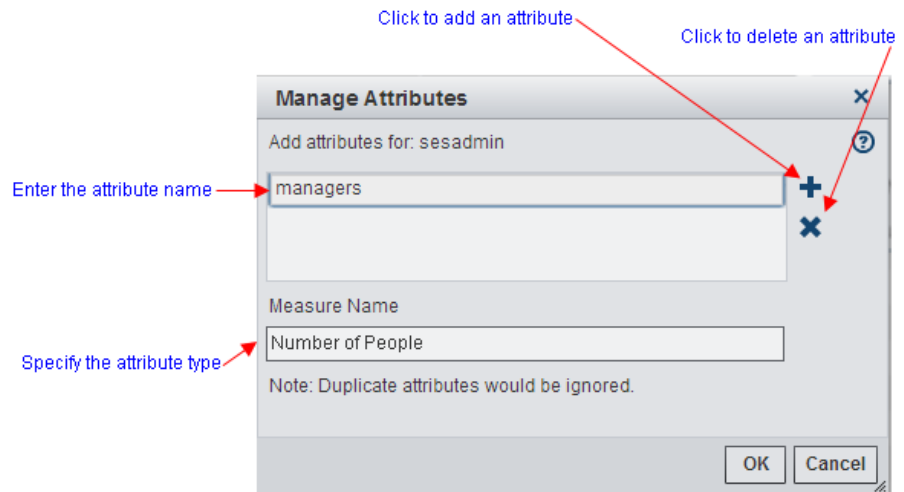


4. The Manage Attributes window appears.

- Click the plus button ( **+** ) to add an attribute. Then enter the attribute name.
- Select an attribute and click the X button ( **X** ) to delete an attribute.
- Specify the **Measure Name**.

*Note:* What you specify for the name does not make any difference in how the DQF is derived for an aggregated survey—it is only for your own documentation.

- Click **OK**.





## Chapter 9

# Administer Surveys

---

<b>Survey Administrator</b> .....	<b>81</b>
<b>Mark a Survey as Incomplete</b> .....	<b>82</b>
<b>Audit Surveys</b> .....	<b>83</b>
<b>View Approved Assignments</b> .....	<b>84</b>
<b>Send an e-mail</b> .....	<b>85</b>
<b>Enable E-mail Notification</b> .....	<b>86</b>
<b>Delete a Survey</b> .....	<b>88</b>
<b>Delete Exported Model Survey Data</b> .....	<b>88</b>
<b>Reassign a Survey</b> .....	<b>89</b>

---

## Survey Administrator

The administrator of a survey is the survey creator.

Among other administration tasks, a survey creator can do the following:

1. Reopen a completed survey.

On receiving notification that a survey has been submitted, a survey creator has the option of reopening the survey by marking it as **Incomplete**. If a survey creator reopens the survey, then:

- the survey status changes to **Incomplete**.
- the survey taker receives a notification.
- the survey taker retakes the survey and submits it again after saving all items as surveyed.

See “[Mark a Survey as Incomplete](#)” on page 82.

*Note:* Staging tables in the database are not rolled back if a survey creator reopens a survey.

2. Audit new assignments.

If the survey taker has added any new assignments, then a survey creator must audit the assignments to either approve or disapprove.

- If the survey creator approves, then the new assignment values, as entered by the survey taker, are written to the staging table in the database.
- If the survey creator disapproves, then the new assignment is removed from the Assignment staging table in the database—it is as though the assignment was never made.

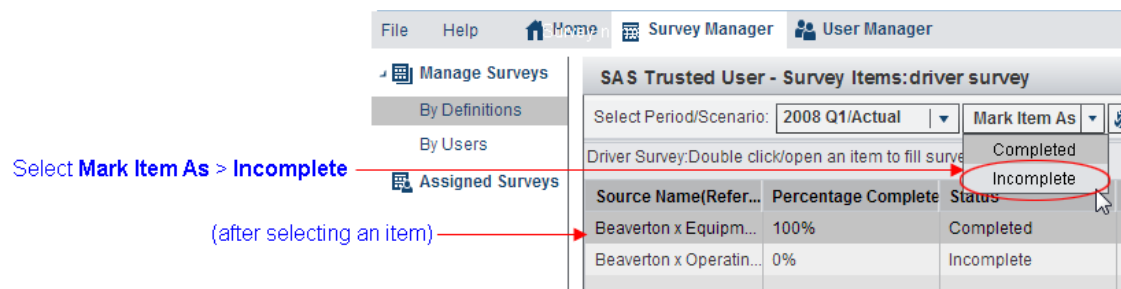
See “[Audit Surveys](#)” on page 83.

## Mark a Survey as Incomplete

When a survey taker submits a survey, the survey is marked as Completed. The survey creator is notified that the survey has been submitted, and the survey taker can no longer modify any responses. Once the survey creator is notified, the survey creator can mark survey items as Incomplete so that the survey taker can change the responses and resubmit the survey.

To mark a survey item as Incomplete:

1. Open the survey.
2. Select a survey item.
3. Select **Mark Item As** ⇨ **Incomplete**.



You are asked to confirm the status change, and you can choose to notify the survey taker.

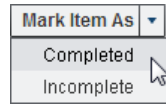
The 'Change Status' dialog box is shown. It contains the text 'Are you sure you want to change the survey status?' and a checked checkbox labeled 'Notify Survey Taker'. There is a text input field below the checkbox. At the bottom right, there are 'Yes' and 'No' buttons.

If you click Yes, the **Surveyed** status changes to No.

The 'Surveyed' status dropdown menu is shown. It contains four options: 'Yes', 'No', 'Yes', and 'Yes'. The 'No' option is circled in red.

Once the **Surveyed** status is No, the survey taker can retake the survey for that item.

*Note:* You can also mark a survey item as Completed, in which case the survey taker can no longer make changes to that item.




---

## Audit Surveys

When a survey creator selects the model option to allow a survey taker to add new destination accounts to a driver survey, and a user adds a new destination account, the survey creator must either approve or disapprove the addition. See [“Allow a Survey Taker to Add New Destination Accounts” on page 96](#).

- If the survey creator approves, then the new assignment values, as entered by the survey taker, are written to the Assignment staging table in the database.
- If the survey creator disapproves, then the new assignment is removed from the database staging table—it is as if the assignment was never made.

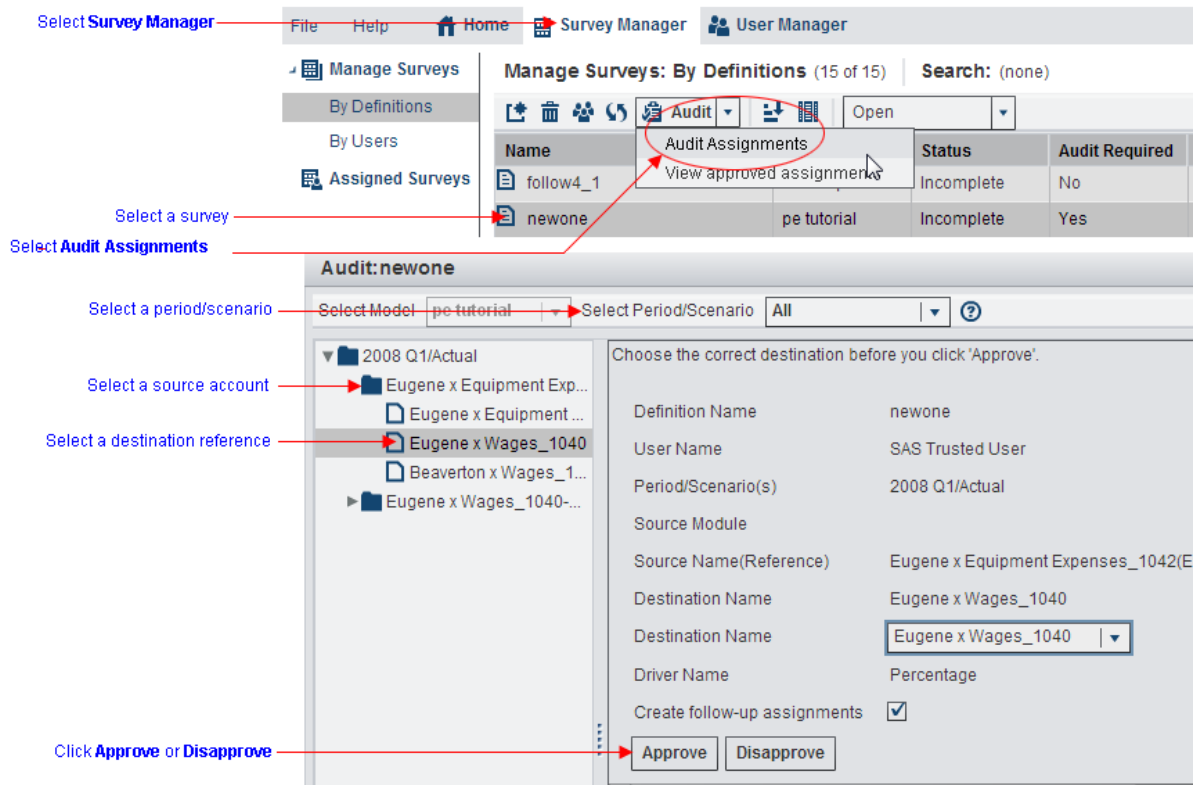
*Note:* A survey creator cannot subsequently change a decision to approve or disapprove. Disapproving a user’s addition has the same effect as though the survey taker had never made the new assignment at all.

To audit a survey:

1. Select the **Survey Manager**.
2. Select a survey to audit.
3. Select **Audit** ⇒ **Audit Assignments**.
4. Select a source account.
5. Select a destination account.
6. Select whether to **Create Follow Up Assignments**. See [“Create Follow-up Surveys” on page 63](#).
7. Click either **Approve** or **Disapprove**.

Once all the destination accounts have been approved, the **Audit Required** column changes to No.

8. Click **Close**.



## View Approved Assignments

As a survey creator, you can view approved assignments, but you cannot reverse a decision to disapprove them. See [“Audit Surveys” on page 83](#).

*Note:* You cannot view disapproved assignments. They are as though the survey taker never made them.

To view approved assignments:

1. Select the **Survey Manager**.
2. Select a survey.
3. Select **Audit** ⇒ **View approved assignments**.
4. Click **Close**.



## Send an e-mail

As a survey creator, you can send an e-mail to a survey taker in three different ways:

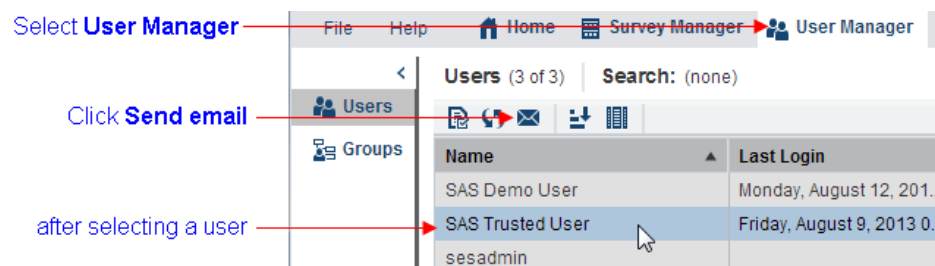
- To a selected user
- To a new survey taker
- To a reassigned survey taker

### To a selected user

To send an e-mail to a selected user:

1. Select **User Manager**.
2. Select a user—the recipient of the e-mail.
3. Click **Send e-mail**.

*Note:* The user's e-mail address must be stored in the SAS metadata server. See [“Creating Users”](#) on page 17.



### To a new survey taker

You can send an e-mail to each survey taker who is assigned a survey item when you create a survey.

Select **Notify users about survey assignment** on the New Definition – User Assignments window.

**New Definition**

User assignments Step 3 of 3

Assign owners for survey items

Source Name(Reference)	Owner
Beaverton x Equipment Expenses_1039	SAS Trusted User
Beaverton x Operating Expenses_1038	SAS Trusted User

☒ Notify users about survey assignment

### To a reassigned survey taker

You can send an e-mail when a survey item is reassigned to a different survey taker. Select **Notify users about survey assignment** on the Re-assign Users window. See [“Reassign a Survey to Another User”](#) on page 77.

**Re-assign Users:**

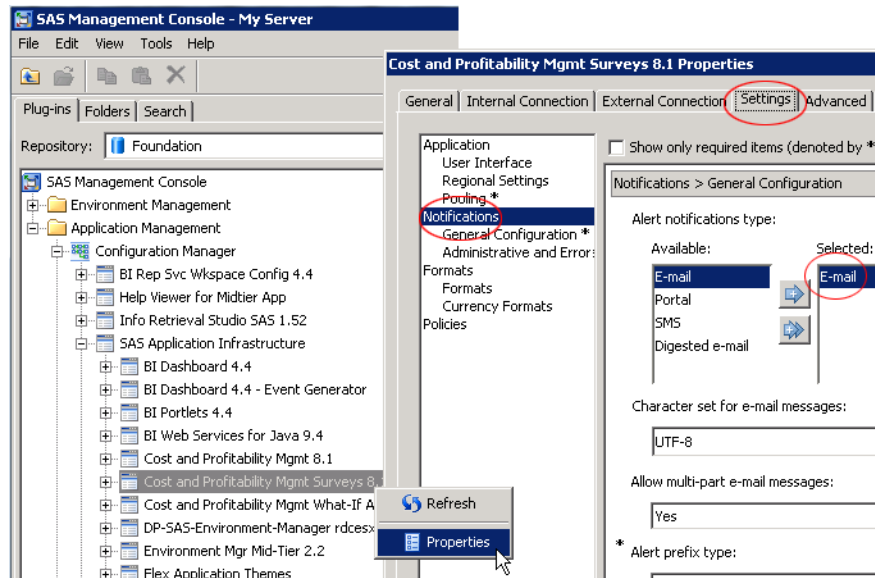
Source Name(Reference)	Owner
Beaverton x Equipment Expenses_1039	SAS Demo User
Beaverton x Operating Expenses_1038	SAS Demo User

☒ Notify users about survey assignment

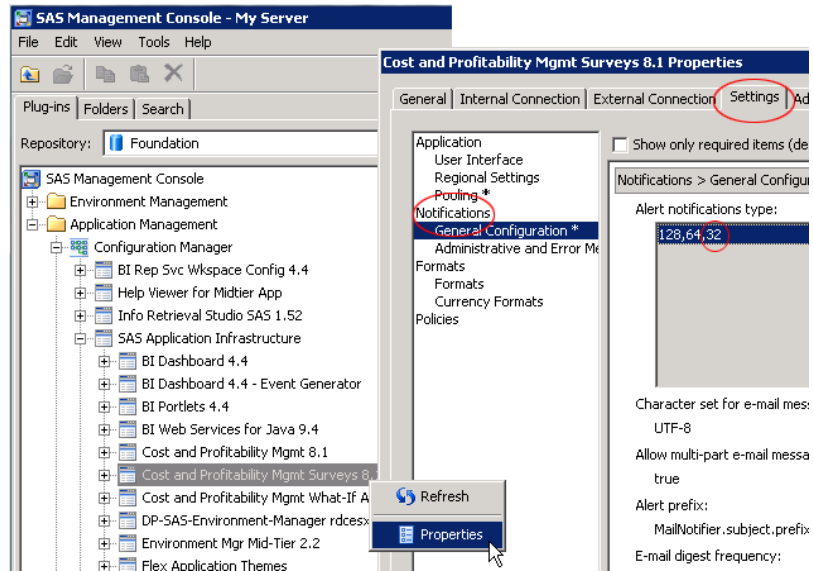
## Enable E-mail Notification

To enable e-mail notification for SAS Cost and Profitability Management Surveys, do the following:

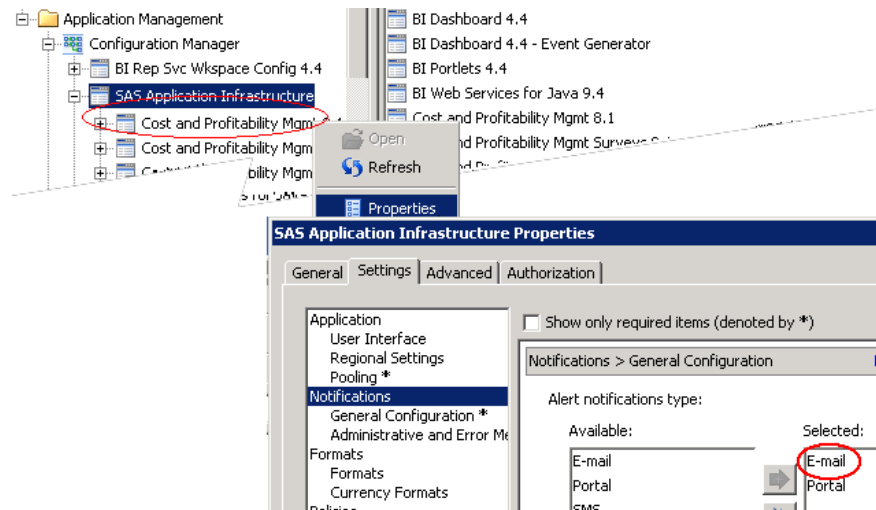
1. Log on to SAS Management Console as an administrator, and access the SAS Cost and Profitability Management middle-tier server.
2. Click the **Plug-Ins** tab.
3. Expand **Application Management**.
4. Expand **Configuration Manager**.
5. Expand **SAS Application Infrastructure**.
6. Right-click **Cost and Profitability Mgmt Surveys 8.1** and select **Properties**.
7. Click the **Settings** tab on the Properties window.
8. Select **Notifications**.
9. Ensure that one of the following is true:
  - E-mail is selected for Alert Notifications Type, as shown in the following picture.



- or, 32 is selected for Alert Notifications Type, as shown in the following picture.



*Note:* 32 indicates that E-mail has been selected at the SAS Application Infrastructure level (as shown in the following picture) so as to apply to all child components, including Cost and Profitability Management Surveys.

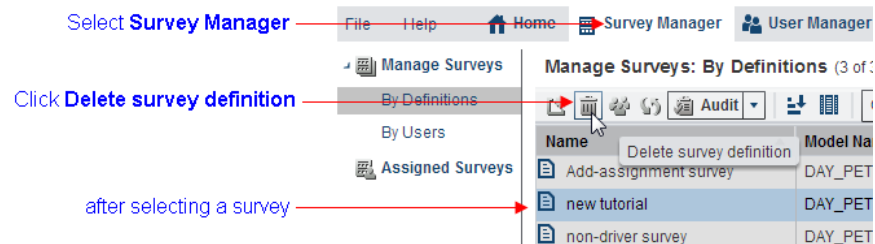


## Delete a Survey

To delete a survey:

1. Select **Survey Manager**.
2. Select a survey.
3. Click **Delete survey definition**.

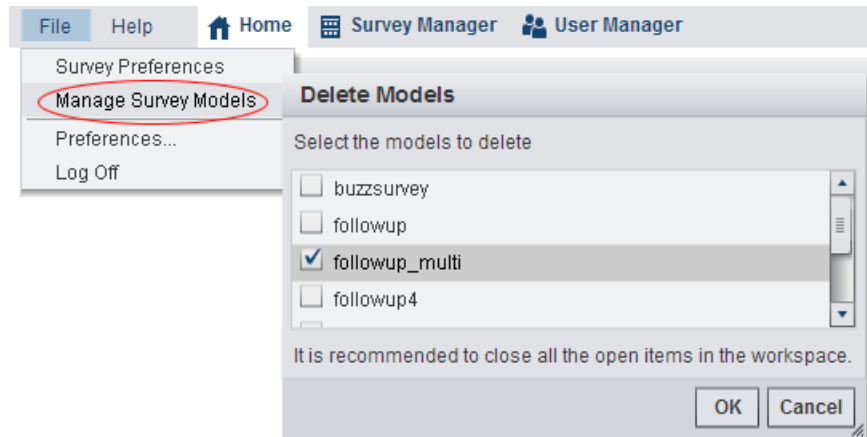
*Note:* Any data already written to staging tables is not rolled back.



## Delete Exported Model Survey Data

To delete model data that was previously exported from SAS Cost and Profitability Management for creating surveys:

1. Select **File** ⇒ **Manage Survey Models**  
The Delete Models dialog opens.
2. Select a model to delete and click **OK**.



---

## Reassign a Survey

See [“Reassign a Survey to Another User”](#) on page 77.



## Chapter 10

# Preferences

---

<b>Overview</b> .....	<b>91</b>
<b>Global Preferences</b> .....	<b>92</b>
<b>General Preferences</b> .....	<b>92</b>
<b>Application Preferences</b> .....	<b>93</b>
<b>Model Preferences</b> .....	<b>94</b>
Overview .....	94
Allow a Survey Taker to Add New Destination Accounts .....	96
Follow-up Driver Surveys .....	98

---

## Overview

You can select the following types of preferences:

### **Global Preferences**

Global preferences apply to all SAS Web applications that are displayed with the Adobe Flash Player. See [“Global Preferences” on page 92](#).

### **General Preferences**

General preferences apply to individual survey takers and affect the appearance, rather than the functionality, of the application. See [“General Preferences” on page 92](#).

### **Application Preferences**

Application preferences apply to every survey creator and to every survey taker and affect the functionality of the application. See [“Application Preferences” on page 93](#).

### **Model Preferences**

Model preferences apply to every survey creator and to every survey taker for a particular model. See [“Model Preferences” on page 94](#).

## Global Preferences

Global preferences apply to all SAS Web applications that are displayed with the Adobe Flash Player.

These preferences are set for each user.

To set global preferences, select **Files** ⇒ **Preferences**. Then select **Global Preferences**.

**Global Preferences** [Reset to Defaults](#)

User locale: Browser locale ▼  
 Locale changes take effect after you log off and log back on.

Theme: Default theme (SAS Corporate) ▼ ?

☐ Invert application colors

☐ Override settings for focus indicator

Color:

Thickness:  2 pixels

Opacity:  100%

Sample focus  
Adjacent item

☐ Reset to show all warning and confirmation messages Reset

### User locale

specifies the geographic region whose language and conventions are used in the applications. This setting might also apply to some SAS Web applications that are not displayed with the Adobe Flash Player.

*Note:* You must log off and back on for the **User locale** preference to take effect fully.

### Theme

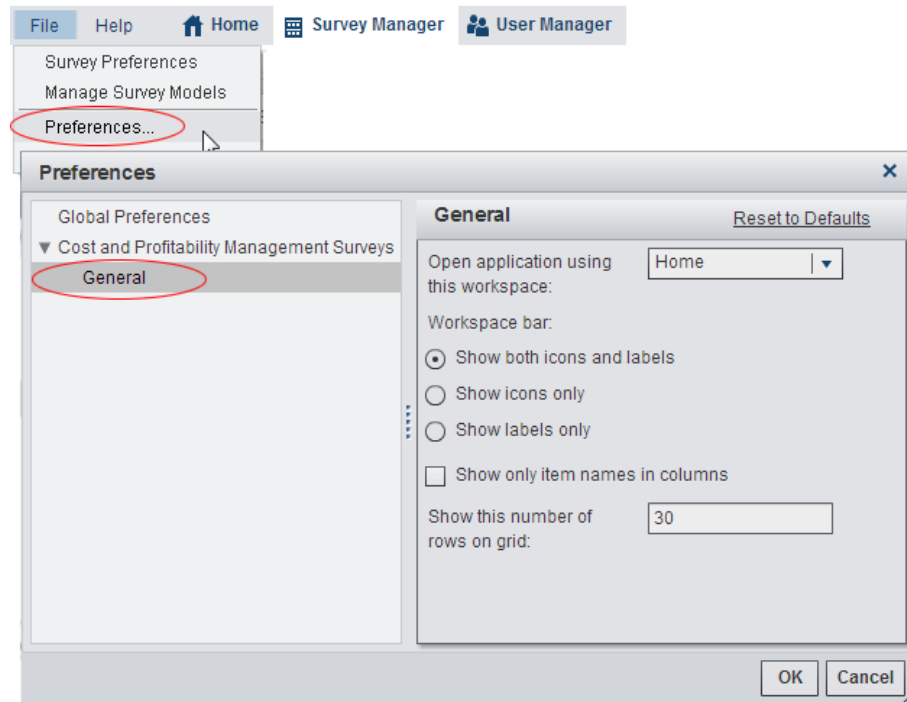
specifies the collection of colors, graphics, and fonts that appear in the applications.

## General Preferences

General preferences apply to individual survey takers and affect the appearance, rather than the functionality, of the application.

To set general preferences, select **Files** ⇒ **Preferences**. Then select **General**.



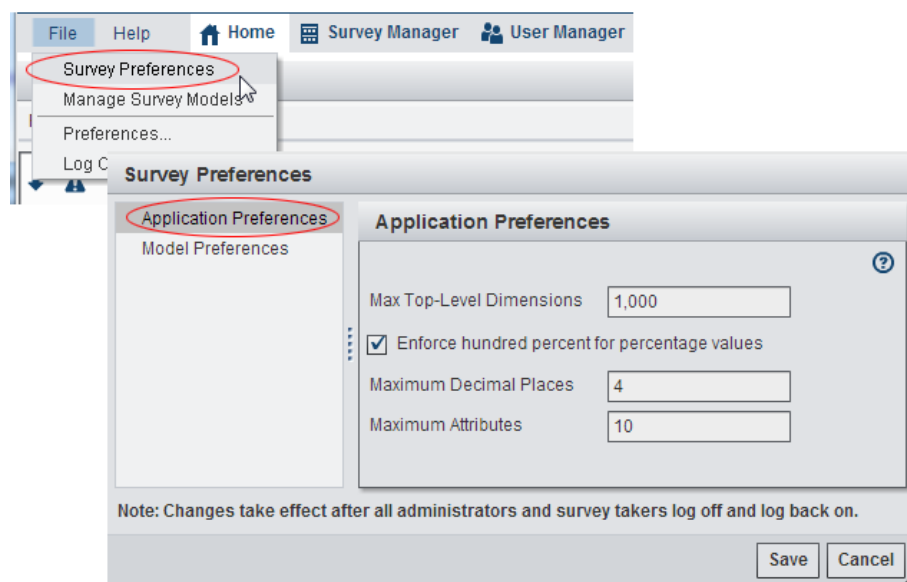


## Application Preferences

Application preferences apply to every survey creator and to every survey taker and affect the functionality of the application.

*Note:* You must log off and back on for the preferences to take effect fully.

To select application preferences, select **Files** ⇒ **Survey Preferences**. Then, select **Application Preferences**. (You must have Create Model capability.)



You can set the following Application preferences:

**Max Top-Level Dimensions**

specifies the number of entries in the **Top-level Dimension Reference** drop-down list. This drop-down list is presented when you create a survey:

- If you select Any Dimension, then all accounts in the module are presented on the next page for assignment to survey owners.
- If you select a particular dimension for the top-level dimension, then only those accounts in the module under the specified dimension can be assigned to survey owners.

The screenshot shows a 'Create a survey' dialog box with the following fields:

- Select Model:** buzzsurvey
- Select Period/Scenario:** 2014/Actual
- Select module:** Module1
- Type of Survey:** Drivers
- Define scope of query:**
  - Top-level Dimension Reference:** Any Dimension (highlighted with a red box and a blue arrow from the text 'Top-level dim ref')
  - Driver Name:** Any Driver

**Enforce hundred percent for percentage values**

specifies both of the following:

- When a user takes a driver survey involving a Percentage driver, the user's responses must total 100%.
- When a user takes an aggregated survey, percentage contributions to Driver Quantity Fixed (DQF) must total 100%. See [“Create an Aggregated Survey” on page 64](#).

**Maximum Decimal Places**

specifies the maximum number of decimal places that are accepted from survey taker responses. This option does not prevent a survey taker from entering more decimal places, but it determines how many decimal places are stored in the database.

**Maximum Attributes**

specifies how many attributes can be added to any particular survey taker for taking an aggregated survey. See [“Create an Aggregated Survey” on page 64](#).

*Note:* If you have already added attributes to users and then lowered the number of maximum attributes, attributes that exceed the maximum are not automatically removed from users. The preference applies only to the subsequent setting of attributes.

---

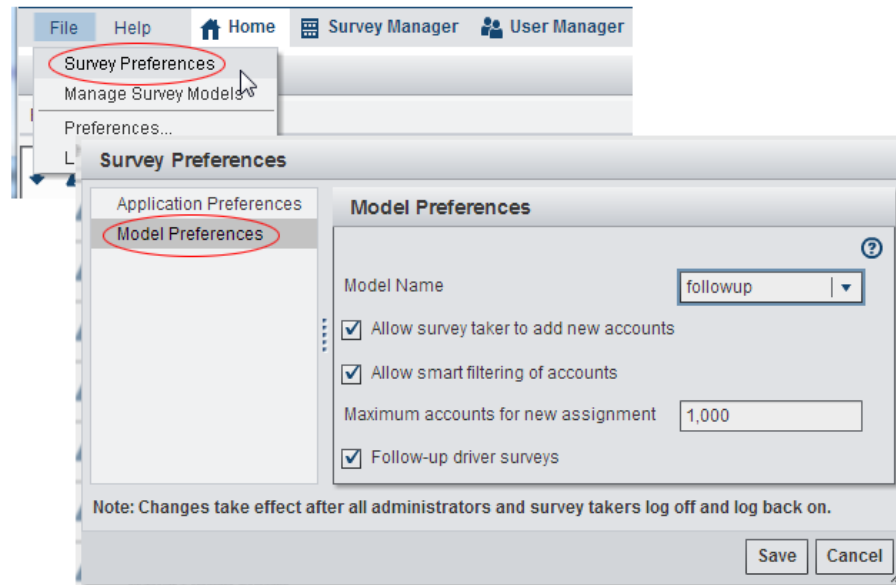
## Model Preferences

**Overview**

Model preferences apply to every survey creator and to every survey taker for a particular model.

*Note:* You must log off and back on for the preferences to take effect fully.

To select model preferences, select **Files** ⇒ **Survey Preferences**. Then select **Model Preferences**. (You must have Create Model capability.)



You can set the following Model preferences:

#### **Model Name**

specifies the model to which the following preferences apply.

#### **Allow Survey Taker To Add New Accounts**

allows survey takers, when taking a driver survey, to create assignments to additional accounts. The survey taker can then take a survey for those new accounts. See [“Allow a Survey Taker to Add New Destination Accounts” on page 96](#).

#### **Allow smart filtering of accounts**

Smart filtering affects the drop-down list that is presented to a survey taker who is adding destination accounts to a survey. See [“Allow a Survey Taker to Add New Destination Accounts” on page 96](#).

- When you allow smart filtering, the drop-down list contains only those accounts that are in the top-level dimension.

For example, suppose that the dimensions of an existing destination account are the following:

Region  
Channel  
Product

and whose dimension signature is **Paris X Retail X Purse**. The top level dimension, in this case, is Paris. If **Allow smart filtering of accounts** is selected, then only other destination accounts whose top-level dimension is also Paris are displayed for selection.

- When you disallow smart filtering (deselect the checkbox), the drop-down list contains accounts that are outside of the top-level dimension.

#### **Maximum accounts for new assignment**

specifies the maximum number of assignments to additional accounts that a survey taker can add to a survey.

**Follow-up driver surveys**

causes the automatic creation and assigning of surveys for driver surveys. Whenever you create a driver survey and assign it to a survey taker for a source account, then if there are assignments from the destination account to other accounts a follow-up driver survey is automatically created from the destination account to the other accounts and assigned to the same survey taker.

For more information, see [“Follow-up Driver Surveys” on page 98](#).

**Allow a Survey Taker to Add New Destination Accounts**

A survey taker who is taking a driver survey can create assignments that don’t already exist in the model.

*Note:* A survey taker can add new assignments provided that a survey creator has selected the **Allow Survey Taker to Add New Accounts** model option for the model. See [“Model Preferences” on page 94](#).

Each new assignment must be audited by the survey creator. After the survey creator has approved the addition, the survey values for the assignment are saved to a staging table. See [“Audit Surveys” on page 83](#).

Each new assignment that a survey taker creates, along with the new values specified, is saved in the Assignment staging table. See [“Assignment table” on page 122](#).

*Note:*

- A survey taker can add destination accounts only in the same module as already existing destination accounts.
- A survey taker can add destination accounts only for source accounts that already have an existing assignment to some destination account. If an account has no assignments, then even if it has a driver assigned it will not have been exported with survey data and is not available for adding destination accounts.

**Detailed Example**

As previously discussed, to create an assignment using the SAS Cost and Profitability Management user interface, you

- add a driver to the source account
- add potential destination accounts to the right assignments pane
- click on the destination account

The following picture shows how to create an assignment between Equipment Expenses (EPDEE) and Land Distribution (OPDAD).

Click here

Display Name	Reference	DrvName	IntsctnName	Reference
RESOURCE (PRIMARY PANE)			Eugene x Parcel Delivery x Land Distribution	EPDLD
USA			Oakland x Parcel Delivery x Air Distribution	OPDAD
California			Oakland x Parcel Delivery x Land Distribution	OPDLD
Oregon			Los Angeles x Customer Service x Expedite	LACSES
Beaverton				
Eugene				
Customer Service				
Parcel Delivery				
Equipment Expenses	EPDEE	Percentage		

to create this assignment

Display Name	Reference	DrvName	IntsctnName	Reference
RESOURCE (PRIMARY PANE)			Eugene x Parcel Delivery x Land Distribution	EPDLD
USA			Oakland x Parcel Delivery x Air Distribution	OPDAD
California			Oakland x Parcel Delivery x Land Distribution	OPDLD
Oregon			Los Angeles x Customer Service x Expedite	LACSES
Beaverton				
Eugene				
Customer Service				
Parcel Delivery				
Equipment Expenses	EPDEE	Percentage		
Operating Expenses	EPDNE	Allocated Cost		

As an example, the next picture shows how a survey taker can create an assignment from the source account, Beaverton x Operating Expenses, to the new destination account, Eugene x Operating Expenses, while taking a Driver survey, as follows:

- 1. Double-click the source account (Beaverton x Operating Expenses) to open it.

1. Double-click the source account

Source Name(Refer...	Percentage Complete	Status	New Assignments
Beaverton x Equipment E...	0%	Incomplete	
Beaverton x Operating Ex...	0%	Incomplete	

The list of its existing destination accounts is displayed.

- 2. Click the **Add assignments** button (+).

2. Click **Add assignments**.

Fill Quantities For: Beaverton x Operating Expenses\_1038

Save As + Add assignments

Period/Scenario(s):2008 Q1 in terms of Dollars

Measure Name	Dollars	Surveyed
Beaverton x Air Distribution_2020	83,000.0000	No
Beaverton x Expedite Package Shipments...	8,000.0000	No
Beaverton x Inspect_2019	54,000.0000	No

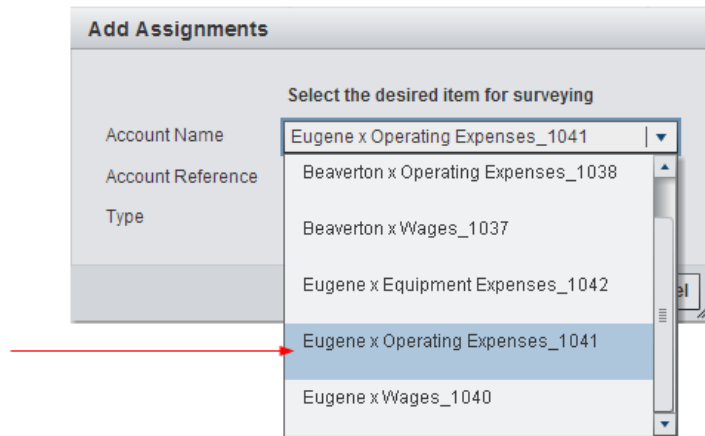
The Add assignments window appears.

- 3. Select a new destination account from the drop-down list of accounts, and then click **OK**.

*Note:* The items in the drop-down list are controlled by the **Allow smart filtering of accounts** Model preference. If this option checked and there are no accounts underneath the top-level account, then the drop-down list will not display any

accounts. In that case, you can turn off **Allow smart filtering of accounts** to see additional accounts. See “[Model Preferences](#)” on page 94.

3. Select a new destination account



4. The new destination account is added to the list of destinations..

4. The destination account is added to the list

The 'Fill Quantities For: Beaverton x Operating Expenses\_1038' window is shown. It has a 'Save As' button and a search bar. Below is a table with columns: Measure Name, Dollars, and Surveyed. The table lists several measures, including 'Eugene x Operating Expenses\_1041' which is circled in red. A red arrow points to the checkbox in the first column of this row.

Measure Name	Dollars	Surveyed
Beaverton x Air Distribution_2020	83,000.0000	No
Beaverton x Expedite Package Shipments...	8,000.0000	No
Beaverton x Inspect_2019	54,000.0000	No
Eugene x Operating Expenses_1041	0.0000	No

## Follow-up Driver Surveys

If you choose this option, then whenever you create a driver survey if the following condition holds:

- You assign, to a survey taker, the survey from source account AAA to a destination account BBB, and there is an assignment from account BBB to another account CCC

then the following occurs:

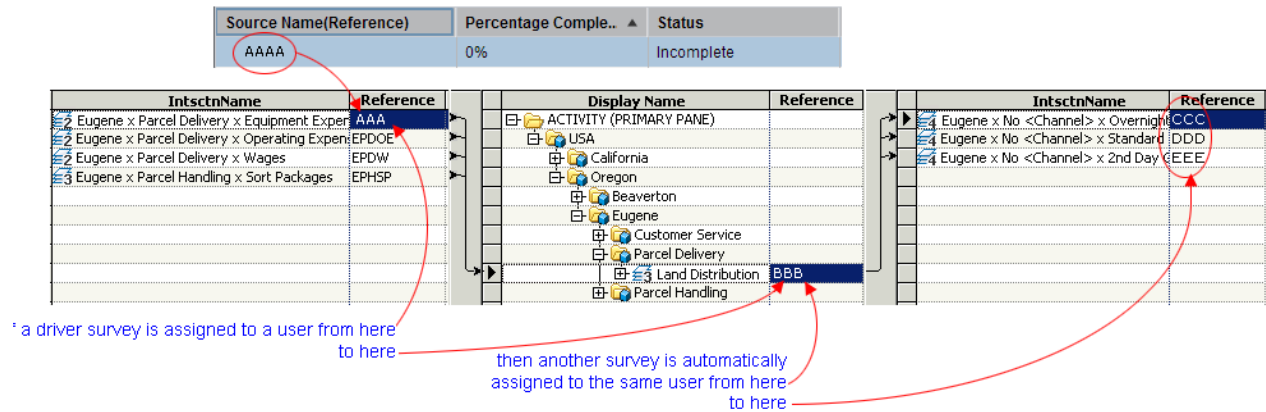
- When the survey from AAA to BBB is submitted, then another survey is automatically created from account BBB to account CCC.

See “[Submit a Survey](#)” on page 75.

- The new survey is automatically assigned to the same survey taker.

For example, in the following picture you can see that there is an assignment from AAA to BBB. And, there is an assignment from BBB to three other accounts (CCC, DDD, and EEE). So, if you create a Driver survey for the module containing AAA and assign the survey to John, then when the survey is submitted, another survey is automatically

created from BBB to the three other accounts (CCC, DDD, and EEE). This automatically created survey is assigned to John.



*Note:* Accounts using the Evenly Assigned driver are not included in automatic follow-up because Evenly Assigned drivers do not require user input to determine the allocation of costs.





## Chapter 11

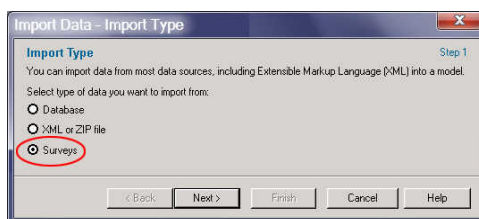
# Importing Survey Data back into the Model

Importing Survey Data .....	101
-----------------------------	-----

## Importing Survey Data

To import survey data, do the following.

1. Select **File** ⇒ **Import** ⇒ **Model Data**.
2. Select **Surveys**, and then click **Next**.



3. On the Import Data – Model window, select an existing model to update with survey data. You cannot create a new model from survey data.

- a. Select an existing model to update with survey data.

*Note:* You cannot create a new model from survey data.

- b. Select the survey data to import.

*Note:* Make sure that the survey data is for the correct model. If the data is from a different model than the one from which data was exported, the import can corrupt the existing model. See [Step 1 on page 47](#).

- c. Select whether you want to do periodic import.

Periodic import allows you to import only the periods that have changed in a model. For more information, see “Incremental Cube Generation” in Chapter 64 of *SAS Cost and Profitability Management: User’s Guide*.

- d. Click **Next**.

**Import Data - Model** Step 2 of 6

**Model**

Choose the model you want to import the data into.

Enter new model name or Select the existing model you want to import into:

☐ New model:

\* Model name:

\* Model Reference:

☒ Existing model:

(Select Model)

Select survey model to update an existing model:

Select one of the option to import new data when importing model structures and data:

Options

☒ Update all data in the model, then import new data

☐ Periodic import

☐ Remove all data in the model, then import new data

< Back Next > Finish Cancel Help

4. On the Import Data – Select Tables window, select tables from the survey data being imported and map them to tables in the model being updated.

The tables being imported correlate to the tables that were previously exported. See [Step 3 on page 48](#).

**Import Data - Select Tables** Step 3 of 6

**Select Tables**

Select the tables from your source database and map them to tables in the SAS Cost And Profitability Management schema. One or more source tables can be mapped to a destination table. At least one table must be checked to move to the next step.

Source Table	Target Table
<input checked="" type="checkbox"/> WM1005_Account	Account
<input checked="" type="checkbox"/> WM1005_Assignment	Assignment
<input checked="" type="checkbox"/> WM1005_EnteredCostElement	EnteredCostElement
<input checked="" type="checkbox"/> WM1005_ValueAttribute	ValueAttribute
<input checked="" type="checkbox"/> WM1005_ValueAttribAssn	ValueAttributeAssociation

Select All Clear All

< Back Next > Finish Cancel Help

5. On the Import Data – Options window, for each table being imported, select whether you want to identify accounts by their dimension signature.

**Note:** The import will work regardless of what you choose, so you can simply click **Next**.

**Import Data - Options** Step 4 of 6

**Options**

Select the options for this import. You can identify accounts either by dimension signature information or by account reference number. Indicate for each of the tables below whether or not dimension signature information will be used to identify accounts.

If you choose to identify accounts by dimension signature information for a table, you must provide at least one pair of dimension reference and dimension member reference field mapping for that table.

Table	Dimension Signature Required
WM1001_Account	<input checked="" type="checkbox"/>
WM1001_Assignment	<input checked="" type="checkbox"/>
WM1001_EnteredCostElement	<input checked="" type="checkbox"/>
WM1001_ValueAttribAssn	<input checked="" type="checkbox"/>

SAS Cost And Profitability Management can automatically create accounts in the model when valid dimension signature information is provided but the account does not already exist in the model.

☒ Automatically create accounts using dimension signatures

< Back Next > Finish Cancel Help

6. On the Import Data – Map Columns window, select the columns to be imported..

The columns being imported correlate to the columns that were previously exported. See [Step 4 on page 49](#). And see [Step 5 on page 49](#).

**Import Data - Map Columns** Step 5 of 6

**Map Columns**

Select the fields from the input tables and map them to fields in the output tables. You may also add default values for fields which are not included in the input tables.

Select a column from the source table and a column from the staging table to create a mapping. All required fields are marked with an asterisk (\*) and must be mapped. When applicable, select the number of required dimensions.

Dimensions:

Source	Target Column Name	Default
<input checked="" type="checkbox"/> Id		
<input checked="" type="checkbox"/> ModuleType	*ModuleType	
<input checked="" type="checkbox"/> ModuleId		
<input checked="" type="checkbox"/> Period	*Period	
<input checked="" type="checkbox"/> Scenario	*Scenario	
<input checked="" type="checkbox"/> Reference	*Reference	
<input checked="" type="checkbox"/> DriverName	DriverName	
<input checked="" type="checkbox"/> Name	Name	
<input checked="" type="checkbox"/> OutputQuantityUE	OutputQuantityUE	
<input checked="" type="checkbox"/> Revenue	Revenue	

WM1003\_Account WM1003\_Assignment

You can add a mapping for a column which does not exist by clicking on the Add button. You can specify a default value for this column.

Add Delete

< Back Next > Finish Cancel Help

7. Review your selections and click **Finish**.

**Import Data - Summary** Step 6 of 6

**Summary**

Please review your selections:

**Import File Type:**  
Surveys

**Source:**  
connection string

**Tables:**  
WM1003\_Account -> Account  
WM1003\_Assignment -> Assignment  
WM1003\_EnteredCostElement -> EnteredCostElement  
WM1003\_ExternalUnit -> ExternalUnit  
WM1003\_ValueAttribute -> ValueAttribute  
WM1003\_ValueAttribAssn -> ValueAttributeAssociation

**Model:**  
Type: Import in an existing model  
Name: Parcel Express Tutorial  
Reference: petul

☐ Save configuration as: Name:

☐ Save without running Description:

< Back Next > Finish Cancel Help



## Part 3

---

# Staging Tables for Import and Export

<i>Chapter 12</i>	
<b>About Staging Tables</b> .....	<i>107</i>
<i>Chapter 13</i>	
<b>Staging-Table Schemas</b> .....	<i>117</i>



## Chapter 12

# About Staging Tables

---

<b>Introduction</b>	<b>107</b>
<b>Staging Tables for Importing a New Model</b>	<b>108</b>
<b>Periodic Data versus Structural Data</b>	<b>108</b>
<b>Staging Table Dependencies</b>	<b>109</b>
<b>Naming Conventions</b>	<b>109</b>
General naming conventions	109
Attribute naming conventions	109
Dimension naming conventions	110
Dimension level naming conventions	110
Dimension member naming conventions	111
Driver naming conventions	112
Entered cost element naming conventions	112
Module naming conventions	112
Period naming conventions	112
Period level naming conventions	113
Scenario naming conventions	113
Scenario level naming conventions	113
Stage attribute naming conventions	114
Workspace item naming conventions	114
<b>Reference Conventions</b>	<b>114</b>
Account reference conventions	114
Attribute reference conventions	114
Dimension reference conventions	114
Dimension member reference conventions and dimension	
attribute reference conventions	114
Period reference conventions	115
Scenario reference conventions	115
Entered cost element reference conventions	115
<b>Data Types</b>	<b>115</b>

---

## Introduction

Staging tables define the SAS Cost and Profitability Management data schema. Each table corresponds to a specific structural or periodic aspect of a model, such as period, scenario, dimension, and so on. Use this data schema to create a database that will hold

the data that you want to import into a model. This data schema is also used by SAS Cost and Profitability Management to export models to XML files.

---

## Staging Tables for Importing a New Model

When you create a new model by importing data, you must import at least the following tables:

- Dimension
- DimensionMember
- DimensionLevel
- DimensionOrder

---

## Periodic Data versus Structural Data

Staging tables are distinguished by whether they contain periodic or structural data. Periodic data is model data which is stored separately for each period/scenario association. Structural data is the model data which is independent of any period/scenario association. It is data which is common to all period/scenario associations.

### Periodic data

The following staging tables contain periodic data:

- Account
- Assignment
- CurrencyRate
- DimMemberDimAttrAssociation
- DimMemberValueAttrAssociation
- ExternalUnit
- EnteredCostElement
- ValueAttributeAssociation
- ValueAttributePeriodicDef
- DimensionalAttributeAssociation

When importing data for an already-existing model, you can speed the import by importing only periodic data (see “Periodic Import” in Chapter 74 of *SAS Cost and Profitability Management: User’s Guide*).

### Structural data

The following staging tables contain structural data:

- Dimension
- DimensionMember
- DimensionLevel
- DimensionOrder



- Driver
- Model
- ValueAttributes
- Period
- PeriodLevel
- Scenario
- Scenariolevel

---

## Staging Table Dependencies

1. A dimension must exist before creating dimensional orders.
2. A dimensional order must exist before creating dimensional members
3. Dimensional members must exist before creating an account
4. A driver must exist before creating an assignment.
5. An account must exist before creating entered cost elements or assignments or adding attributes.
6. Value attributes must exist before creating a value attribute association

---

## Naming Conventions

### ***General naming conventions***

The name of any item must conform to the following rules:

- Names cannot contain this character: |
- Any item, such as a dimension, a driver, an attribute, and so on, that might become a dimension in a cube cannot have the reserved names All or None.
- Names are case insensitive. For example, the name My Model is the same as my model and mY mODEL.

In addition to the general naming conventions, there are more restrictive naming conventions for the following items.

### ***Attribute naming conventions***

In addition to the general naming conventions, attribute names must conform to these rules:

- Attribute names must be unique within a parent.
- Attribute names may contain up to 64 alphanumeric characters. However, attribute names longer than 50 characters are truncated to 50 characters when a cube is generated for Microsoft Analysis Services. SAS OLAP allows all 64 characters for attribute names.

- An attribute name cannot be the name of a numeric property.
- Attribute names may contain these characters, even though these characters are not valid in cubes:

. , ; ' ^ : ? \* & % \$ ! - + = ( ) [ ] { } / \

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

See “Stage Attributes” in Chapter 33 of *SAS Cost and Profitability Management: User’s Guide*.

### **Dimension naming conventions**

In addition to the general naming conventions, dimension names must conform to these rules:

- Dimension names may contain up to 64 alphanumeric characters. However, dimension names longer than 32 characters are truncated to 32 characters when a cube is generated for Microsoft Analysis Services. SAS OLAP allows all 64 characters for dimension names.
- Dimension names must be unique among all dimensions and dimension attributes.
- Dimension names must be unique.
- Dimension names may contain these characters, even though these characters are not valid in cubes:

. , ; ' ^ : ? \* & % \$ ! - + = ( ) [ ] { } / \

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

*Note:* “Module”, “Period”, “Scenario”, and “Driver” are reserved names and you cannot use them for either a dimension name or a dimension reference.

### **Dimension level naming conventions**

In addition to the general naming conventions, dimension level names must conform to these rules:

- Dimension level names may contain up to 64 alphanumeric characters. However, dimension level names longer than 50 characters are truncated to 50 characters when a cube is generated for Microsoft Analysis Services. SAS OLAP allows all 64 characters for dimension level names.
- Dimension level names must begin with an alphabetic character.
- Dimension level names cannot contain these characters:

/ \ |

- Dimension level names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

- *Note:* Because of the mechanism used by SAS Cost and Profitability Management to store dimension level names, some user-specified names will cause conflicts with the

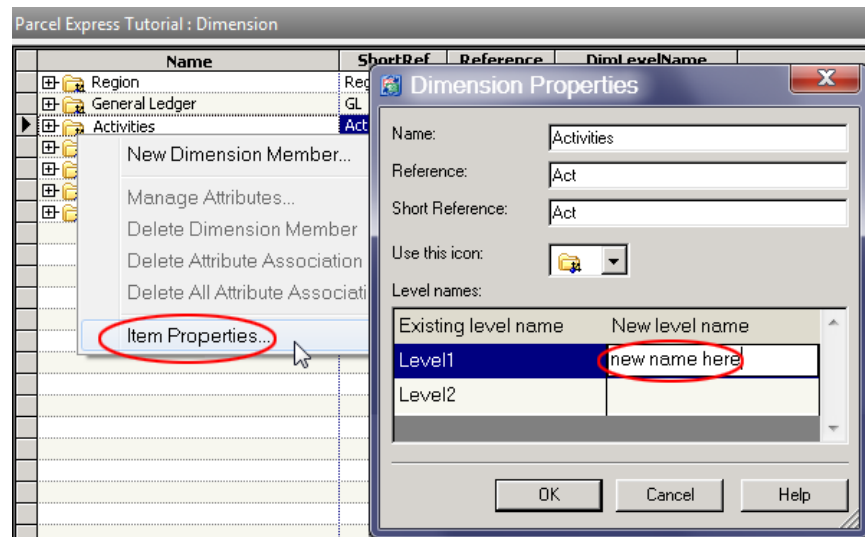
underlying database (regardless of whether Microsoft SQL Server or Oracle is used). These conflicts will appear as obscure error messages when calculating a model. Dimension level names that will cause conflicts are reserved words in the Microsoft SQL query language. Some of the more common reserved words are: level, group, function, drop, and join. To avoid dimension level name conflicts in cases where you need to use a word such as a level name, add a descriptive prefix or suffix. For example, using the name Level or LeVeL might cause errors, but the names Dept\_Level and Level\_02 are fine.

To change the name of a dimension level:

1. Open a model.
2. Select **Model** ⇒ **Dimensions**.
3. Right-click a dimension and select **Item Properties**.

The Dimension Properties window opens.

4. Type a new name, and then click **OK**.



### Dimension member naming conventions

In addition to the general naming conventions, dimension member names must conform to these rules:

- Dimension member names may contain up to 64 alphanumeric characters. However, dimension member names longer than 50 characters are truncated to 50 characters when a cube is generated for Microsoft Analysis Services. SAS OLAP allows all 64 characters for dimension member names.
- Dimension member names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore ( \_ ) when a cube is generated.

Dimension member names must be unique within a parent.

**Driver naming conventions**

In addition to the general naming conventions, driver names must conform to these rules:

- Driver names must be unique within all drivers.
- Driver names may contain up to 64 alphanumeric characters. However, driver names that are longer than 50 characters are truncated to 50 characters when a cube is generated.
- Driver names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

**Entered cost element naming conventions**

In addition to the general naming conventions, entered cost element names must conform to these rules:

- Entered cost element names may contain up to 64 alphanumeric characters.
- Entered cost element names must be unique under the same account in the same period/scenario association.
- Entered cost element names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

**Module naming conventions**

When renaming modules, the name must conform to these rules::

- Module names may contain up to 64 alphanumeric characters.
- Module names can contain the following characters: alphanumeric, underscores, embedded blanks.

**Period naming conventions**

In addition to the general naming conventions, period names must conform to these rules:

- Period names must be unique within all periods.
- Period names may contain up to 64 alphanumeric characters. However, period names that are longer than 50 characters are truncated to 50 characters when a cube is generated.
- Period names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

### ***Period level naming conventions***

In addition to the general naming conventions, period level names must conform to these rules:

- Period level names must be unique within all period levels.
- Period level names may contain up to 64 alphanumeric characters. However, period level names that are longer than 50 characters are truncated to 50 characters when a cube is generated.
- Period level names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

### ***Scenario naming conventions***

In addition to the general naming conventions, scenario names must conform to these rules:

- Scenario names must be unique within all scenarios.
- Scenario names may contain up to 64 alphanumeric characters. However, scenario names that are longer than 50 characters are truncated to 50 characters when a cube is generated.
- Scenario names may contain these characters, even though these characters are not valid in cubes:

. [ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

### ***Scenario level naming conventions***

In addition to the general naming conventions, scenario level names must conform to these rules:

- Scenario level names must be unique within all scenario levels.
- Scenario level names may contain up to 64 alphanumeric characters. However, scenario level names that are longer than 50 characters are truncated to 50 characters when a cube is generated.
- Scenario level names may contain these characters, even though these characters are not valid in cubes:

[ ]

Each of these characters will be replaced with an underscore (\_) when a cube is generated.

**Stage attribute naming conventions**

In addition to the general naming conventions and Attribute naming conventions, stage attribute names must conform to these rules:

- Stage names must begin with an alphabetic character (letter).
- You can change stage names as long as they retain their order when sorted. If the sort order changes them, you will need to regenerate the fact tables for all period/scenarios in the model.

**Workspace item naming conventions**

In addition to the general naming conventions, workspace item names must conform to these rules:

- Workspace item names cannot contain these characters:  
/ \ or |
- Workspace item names must be unique within a folder.
- Workspace item names may contain up to 64 alphanumeric characters.

---

## Reference Conventions

**Account reference conventions**

- Account references must be unique within a module for all period/scenario associations.

See “Stage Attributes” in Chapter 33 of *SAS Cost and Profitability Management: User’s Guide*.

**Attribute reference conventions**

- Attribute references must be unique within all attributes.

**Dimension reference conventions**

- Dimension references must be unique within all dimensions and dimension attributes.

*Note:* “Module”, “Period”, “Scenario”, and “Driver” are reserved names and you cannot use them for either a dimension name or a dimension reference.

**Dimension member reference conventions and dimension attribute reference conventions**

- Dimension member references and dimension attribute references must be unique within a dimension.

**Period reference conventions**

- Period references must be unique within all periods.

**Scenario reference conventions**

- Scenario references must be unique within all scenarios.

**Entered cost element reference conventions**

- Entered cost element references must be unique within a module for all period/scenario associations.

---

## Data Types

The SAS Cost and Profitability Management data schema uses the following data types:

**Alphanumeric**

is a text value that translates to values such as varchar, nvarchar, char, or nchar.

**Date**

is a regular binary date value that must be recognized by ADO.NET.

**DriverSignature**

is a driver type that is identified by a series of ModuleType and DriverName pairs.

**Float**

is an 8-byte floating point number.

**Integer**

is a 4-byte signed integer.

**Memo**

is a large text-based value that translates to the text or ntext data types in Microsoft SQL Server.

**Boolean**

is a Boolean value. The actual value may be a value other than True/False, such as Yes/No, bit, or a Char field. SAS Cost and Profitability Management will attempt to interpret Boolean values in other formats.

**DimensionSignature**

is a series of one or more pairs of the following dimension and dimension member reference information for each dimension, where (n) represents the dimension level:

DimRef (n)

DimMemberRef (n)

The number of pairs that is used for importing is the maximum number of structural dimensions that is used in the modules being imported. For example, if the Resource module uses two dimensions and the Cost Object module uses three dimensions, the number of DimRef and DimMemberRef pairs must be three. The following example is from a model that tracks the food preparation costs for an airline.

```
DimRef1="Customer"
```

```
DimMemberRef1="WF Portland to CHI"
DimRef2="Products"
DimMemberRef2="Omelette"
```

The dimensions that are included in the Dimension Signature data type are only structural dimensions. Dimension attributes are imported and exported using the DimensionAttributeAssociation table.

### **ExportDimensionSignature**

A series of sets of the following dimension information for each dimension, where (n) represents the dimension level:

```
DimName (n)
DimRef (n)
DimMemberName (n)
DimMemberRef (n)
DimLevel (n)
DimLevelName (n)
```

The following example is from a model that tracks the food preparation costs for an airline:

```
DimName1="Customer"
DimRef1="Customer"
DimMemberName1="WF Portland to CHI"
DimMemberRef1="WF Portland to CHI"
DimLevel1="1"
DimLevelName1="Customer L1"
DimName2="Products"
DimRef2="Products"
DimMemberName2="Omelette"
DimMemberRef2="Omelette"
DimLevel2="1"
DimLevelName2="Products L1"
```



## Chapter 13

# Staging-Table Schemas

---

<b>Table Headings</b> .....	<b>118</b>
<b>Account table</b> .....	<b>119</b>
<b>Assignment table</b> .....	<b>122</b>
<b>AssignmentNonUnique table</b> .....	<b>123</b>
<b>CurrencyRate table</b> .....	<b>124</b>
CurrencyRate table .....	124
Currency Codes .....	125
<b>Dimension table</b> .....	<b>127</b>
<b>DimensionAttributeAssociation table</b> .....	<b>128</b>
<b>DimensionLevel table</b> .....	<b>129</b>
<b>DimensionMember table</b> .....	<b>129</b>
<b>DimensionOrder table</b> .....	<b>130</b>
<b>DimMemberDimAttrAssociation</b> .....	<b>130</b>
<b>DimMemberValueAttrAssociation</b> .....	<b>131</b>
<b>Driver table</b> .....	<b>131</b>
<b>EnteredCostElement table</b> .....	<b>132</b>
<b>Model table</b> .....	<b>133</b>
Model table schema .....	133
Currency Code .....	134
AdvancedOptions Field .....	134
<b>Module table</b> .....	<b>137</b>
<b>Period table</b> .....	<b>137</b>
<b>PeriodLevel table</b> .....	<b>138</b>
<b>RollupAccount table</b> .....	<b>138</b>
<b>Scenario table</b> .....	<b>140</b>
<b>ScenarioLevel table</b> .....	<b>140</b>
<b>ValueAttribute table</b> .....	<b>141</b>
<b>ValueAttributeAssociation table</b> .....	<b>141</b>
<b>ValueAttributePeriodicDef table</b> .....	<b>142</b>

## Table Headings

All tables in the data schema contain the following columns:

This column	Holds this information
Field Name	Represents a value in a model. You can map an external field to this name.
Type of Data	Specifies the type of data stored in the field. See <a href="#">“Data Types” on page 115</a> .
Length	Defines the maximum length of the field. memo is for long text values that can be virtually unlimited; it translates to the Microsoft SQL Server text and ntext types.
Key	Specifies that the field uniquely identifies an item. These values are required whenever inserting, updating, or deleting an item. In some tables, there is more than one way to identify an item. In those cases, there is a key1 and key2. When identifying the item, you must specify all key1 fields or all key2 fields. See “Understanding keys” in Chapter 74 of <i>SAS Cost and Profitability Management: User’s Guide</i> .
Import	Identifies whether the field is required or optional for import. N/A means the field is not valid for import. Import allows the insertion of new items and the updating or merging of existing items. For updating or merging, only key fields are required.  In some tables there is a Required1 and Required2. The user must provide all Required1 fields or all Required2 fields together with all Required fields for inserting a new item.
Export	Indicates whether the field is optional or required for archiving a model by exporting it to staging tables. (See “Overview” in Chapter 76 of <i>SAS Cost and Profitability Management: User’s Guide</i> .)

The following information applies to all the tables:

- All Reference and Name field string comparisons are not case sensitive.
- The identifier for periods and scenarios is Period.Reference and Scenario.Reference.
- In XML, a database null value is represented by \$NULL\$. Null values and empty strings are handled in the same way. Null numbers are typically imported to a model as is.
- The Reference and Name fields for hierarchical items (such as Period, Scenario, and DimMember) must comply with the naming conventions (see the online help).
- The fields that are noted as Required for archive are the fields that are exported by default.
- When a table is exported to a database, most of the names of the exported fields are the same as the names listed in this section. However, some names are changed when exported; those changes are noted in the Field Name column of each table. SAS Cost

and Profitability Management attempts to match the destination column names with the source column names as closely as possible, so this name change is necessary only when a database imposes certain naming limitations (such as a limited number of characters).

## Account table

This table specifies the dimension intersections in the Resource module, the Activity module, and the Cost Object module. The combination of the columns DimRef<n> and DimMemberRef<n> for each valid dimension is the dimension signature.

**CAUTION:**

If you are using Oracle as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	N/A	Optional	–
ModuleReference	AlphaNumeric	64	key1, key2	Required	Required	Resource   "Activity"   "CostObject"
Period	AlphaNumeric	64	key1, key2	Required	Required	–
Scenario	AlphaNumeric	64	key1, key2	Required	Required	–
DimSignature	DimensionSignature	–	key1	Required	Required	There is no column of this name. Dimension signature contains dimension members involved in the account creation. So DimRef(n) and DimmemberRef(n) are mainly required for dimension signature. DimName, DimMemberName, Dimlevel, DimLevelname are optional
DimRef	AlphaNumeric	64	–	–	Required	–
DimName	AlphaNumeric	64	–	–	Optional	–
DimMemberRef	AlphaNumeric	64	–	–	Required	–
DimMemberName	AlphaNumeric	64	–	–	Optional	–
DimLevel	Integer	–	–	–	Optional	–
DimLevelName	AlphaNumeric	64	–	–	Optional	–

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Reference	AlphaNumeric	64	key2	Required	Required	Account must exist to use Reference as the key, otherwise there is no way to hang a dimension reference to a dimension signature. Value must be unique within module. Default value will be generated if blank or not supplied.
DriverName	AlphaNumeric	64	—	Optional	Optional	—
Name	AlphaNumeric	64	—	Optional	Optional	Default generated if blank or not supplied
OutputQuantityUE	Float	—	—	Optional	Optional	—
PeriodicNote	Memo	—	—	Optional	Optional	—
PublishName	AlphaNumeric	—	—	Optional	Optional	—
Revenue	Float	—	—	Optional	Optional	—
SoldQuantity	Float	—	—	Optional	Optional	—
TDQUE	Float	—	—	Optional	Optional	—
UnitOfMeasure	AlphaNumeric	64	—	Optional	Optional	—
ValueAttributes	Memo/Float	—	—	N/A	Optional	There is no column of this name. If user want to see the associated attributes (text or numeric) with the account then user can defines these attributes. These attributes will come as column in the staging table.
AllocatedCost	Float	—	—	N/A	Optional	—
AssignedCost	Float	—	—	N/A	Optional	—
AssignedIdleCost	Float	—	—	N/A	Optional	—
AssignedIdleQuantity	Float	—	—	N/A	Optional	—
AssignedNonReciprocalCost	Float	—	—	N/A	Optional	—
AssignedReciprocalCost	Float	—	—	N/A	Optional	—
Cost	Float	—	—	N/A	Optional	—
DrivableCost	Float	—	—	N/A	Optional	—
DrivenCost	Float	—	—	N/A	Optional	—
DrivenQuantity	Float	—	—	N/A	Optional	—
DriverRate	Float	—	—	N/A	Optional	—

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
EnteredCost	Float	–	–	N/A	Optional	–
IdleCost	Float	–	–	N/A	Optional	–
IdlePercentage	Float	–	–	N/A	Optional	–
IdleQuantity	Float	–	–	N/A	Optional	–
OutputQuantity	Float	–	–	N/A	Optional	–
Profit	Float	–	–	N/A	Optional	–
ReceivedAllocatedCost	Float	–	–	N/A	Optional	–
ReceivedCost	Float	–	–	N/A	Optional	–
ReceivedDrivenCost	Float	–	–	N/A	Optional	–
ReceivedNonReciprocalCost	Float	–	–	N/A	Optional	–
ReceivedReciprocalCost	Float	–	–	N/A	Optional	–
TDQ	Float	–	–	N/A	Optional	Total Driver Quantity
TDQBasic	Float	–	–	N/A	Optional	Total Driver Quantity Basic
TDQCalculated	Float	–	–	N/A	Optional	Total Driver Quantity Calculated
UnassignedCost	Float	–	–	N/A	Optional	–
UnassignedQuantity	Float	–	–	N/A	Optional	–
UnitCost	Float	–	–	N/A	Optional	–
UnitProfit	Float	–	–	N/A	Optional	–
UnitRevenue	Float	–	–	N/A	Optional	–
UsedCost	Float	–	–	N/A	Optional	–
UsedQuantity	Float	–	–	N/A	Optional	–
IsBehavior	Yes/No	–	–	Optional	Optional	To indicate accounts marked as behavior. It is used to Link to Profitability Management
UnitCostEntered	–	–	–	Optional	Optional	As there will be no "ExternalUnit" module in 8.1 so we will export this property as part of Account table
ForWhatIfAnalysis	Yes/No	–	–	Optional	Optional	Added in 8.1 to indicate accounts marked for What-If analysis
ModelNote	AlphaNumeric	2048	–	Optional	Optional	Added in 8.1
ReciprocalId	Float	–	–	Optional	Optional	Added in 8.1

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
IsReciprocal	Yes/No	–	–	Optional	Optional	Added in 8.1
SoldCost	Float	–	–	Optional	Optional	Added in 8.1

## Assignment table

This table specifies assignments. The accounts in an assignment can be specified with the columns `SourceReference` and `DestinationReference` or with the columns `SourceDimRef<n>` and `SourceDimMemberRef<n>`, and `DestinationDimRef<n>` and `DestinationDimMemberRef<n>`.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key1, key2	Required	Required	–
Scenario	AlphaNumeric	64	key1, key2	Required	Required	–
SourceModuleReference	AlphaNumeric	64	key1, key2	Required	Required	Module reference of source account
SourceDimSignature	Signature	–	key1	Required 1	Required	"Source" is the prefix for all the DimSignature fields. Source account will be created if the AutoCreateAccount="1" attribute is included in the Config XML.
SourceReference	AlphaNumeric	64	key2	Required 2	Required	–
SourceCost	Float	–	–	–	Optional	–
DestinationModuleReference	AlphaNumeric	64	key1, key2	Required	Required	Module reference of destination account
DestinationDimSignature	Signature	–	key1	Required 1	Required	"Destination" is the prefix for all the DimSignature fields. Destination account will be created if the AutoCreateAccount=1 attribute is included in the Config XML.
DestinationReference	AlphaNumeric	64	key2	Required 2	Required	–

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
DriverName	AlphaNumeric	64	—	Optional	Optional	Source account's driver will be changed to this driver if it doesn't match already. Ignored if null or empty string. If AutoCreateAccount="1" attribute is included in the Config XML then this will be created as a Basic driver.
AllocatedCost	Float	—	—	Optional	Required	—
DriverQuantityFixed	Float	—	—	Optional	Required	—
DriverQuantityVariable	Float	—	—	Optional	Required	—
DriverWeightFixed	Float	—	—	Optional	Required	>=0? Ignored unless source account uses Weighted Driver. (Value stored in FixedWeight field)
DriverWeightVariable	Float	—	—	Optional	Required	(same as DriverWeightFixed)
Cost	Float	—	—	N/A	Optional	—
DriverQuantityBasic	Float	—	—	N/A	Optional	—
DriverQuantityCalculated	Float	—	—	N/A	Optional	—
IdleDriverQuantity	Float	—	—	N/A	Optional	—
IdleDriverQuantityUE	Float	—	—	Optional	Required	—
IdleCost	Float	—	—	—	Optional	—
ReciprocalId	Float	—	—	Optional	Optional	Added in 8.1
IsReciprocal	Yes/No	—	—	Optional	Optional	Added in 8.1

---

## AssignmentNonUnique table

This table specifies the driver quantities on destination accounts for all drivers where the driver quantity type is nonunique. The driver quantity on a destination account can be specified with the column DestinationReference or with the columns DimRef<n> and DimMemberRef<n>.

### CAUTION:

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key1, key2	Required	Required	–
Scenario	AlphaNumeric	64	key1, key2	Required	Required	–
DestinationModuleReference	AlphaNumeric	64	key1, key2	Required	Required	Module reference
DestinationDimSignature	Signature	–	key1	Required 1	Required	–
DestinationReference	AlphaNumeric	64	key2	Required 2	Required	–
DriverQuantityFixed	Float	–	–	Optional	Optional	–
DriverQuantityVariable	Float	–	–	Optional	Optional	–
DriverWeightFixed	Float	–	–	Optional	Optional	(see Assignment table)
DriverWeightVariable	Float	–	–	Optional	Optional	(see Assignment table)
SourceDimSignature	Signature	–	key1	Optional	Optional	–
DriverName	AlphaNumeric	64	key1, key2	Required	Required	–

---

## CurrencyRate table

### *CurrencyRate table*

This table specifies currency exchange rates for each period/scenario association.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key	Required	Required	–
Scenario	AlphaNumeric	64	key	Required	Required	–



Field Name	Type of Data	Max Len	Key	Import	Export	Notes
CurrencyFrom	AlphaNumeric	64	–	Required	Required	Currency code defined in CurrencyDefinition
CurrencyTo	AlphaNumeric	64	–	Required	Required	Currency code defined in CurrencyDefinition
Rate	Float	–	–	Optional	Required	CurrencyTo = CurrencyFrom * Rate

## Currency Codes

The following table lists the three-letter currency codes that are used in the following tables:

- The Model table
- The CurrencyRate table
- The XML calculate configuration schema

Currency	Code	Currency	Code	Currency	Code
Afghani	AFA	Guinea-Bissau Peso	GWP	Norwegian Krone	NOK
Albania Lek	ALL	Guyana Dollar	GYD	Nuevo Sol	PEN
Algerian Dinar	DZD	Hong Kong Dollar	HKD	Ouguiya	MRO
Argentine Peso	ARS	Iceland Krona	ISK	Pa'anga	TOP
Ariary	MGA	Indian Rupee	INR	Pakistan Rupee	PKR
Armenian Dram	AMD	Iranian Rial	IRR	Pataca	MOP
Aruban Guilder	AWG	Iraqi Dinar	IQD	Peso Uruguayo	UYU
Australian Dollar	AUD	Irish Punt	IEP	Philippine Peso	PHP
Azerbaijani Manat	AZM	Isle of Man Pounds	IMP	Portuguese Escudo	PTE
Bahamian Dollar	BSD	Italian Lira	ITL	Pound Sterling	GBP
Bahraini Dinar	BHD	Jamaican Dollar	JMD	Pula	BWP
Baht	THB	Jersey Pounds	JEP	Qatari Rial	QAR
Balboa	PAB	Jordanian Dinar	JOD	Quetzal	GTQ
Barbados Dollar	BBD	Kenyan Shilling	KES	Rand	ZAR
Belarus Ruble	BYR	Kina	PGK	Rial Omani	OMR
Belgian Franc	BEF	Kip	LAK	Riel	KHR
Belize Dollar	BZD	Kroon	EEK	Rufiyaa	MVR

Currency	Code	Currency	Code	Currency	Code
Bermudian Dollar (Bermuda Dollar)	BMD	Kuna	HRK	Rupiah	IDR
Bolivar	VEB	Kuwaiti Dinar	KWD	Russian Ruble	RUB
Boliviano	BOB	Kwacha	MWK	Rwanda Franc	RWF
Brazilian Real	BRL	Kwacha	ZMK	Saudi Riyal	SAR
Brunei Dollar	BND	Kwanza	AOA	Schilling	ATS
Burundi Franc	BIF	Kyat	MMK	Seychelles Rupee	SCR
Canadian Dollar	CAD	Lari	GEL	Singapore Dollar	SGD
Cape Verde Escudo	CVE	Latvian Lats	LVL	Slovak Koruna	SKK
Cayman Islands Dollar	KYD	Lebanese Pound	LBP	Solomon Islands Dollar	SBD
Cedi	GHC	Lempira	HNL	Som	KGS
CFA Franc BCEAO	XOF	Leone	SLL	Somali Shilling	SOS
CFA Franc BEAC	XAF	Leu	ROL	Somoni	TJS
CFP Franc	XPF	Leva	BGN	South Korean Won	KRW
Chilean Peso	CLP	Liberian Dollar	LRD	Spanish Peseta	ESP
Colombian Peso	COP	Libyan Dinar	LYD	Sri Lanka Rupee	LKR
Comoro Franc	KMF	Lilangeni	SZL	St Helena Pound	SHP
Cordoba Oro	NIO	Lithuanian Litas	LTL	Sucre	ECS
Costa Rican Colon	CRC	Loti	LSL	Sudanese Dinar	SDD
Cuban Peso	CUP	Luigini	SPL	Surinam Guilder	SRG
Cyprus Pound	CYP	Luxembourg Franc	LUF	Suriname Dollars	SRD
Czech Koruna	CZK	Malagasy Franc	MGF	Swedish Krona	SEK
Dalasi	GMD	Malaysian Ringgit	MYR	Swiss Franc	CHF
Danish Krone	DKK	Mali franc	MLF	Syrian Pound	SYR
Denar	MKD	Maltese Lira	MTL	Tajik Ruble	TJR
Deutsche Mark	DEM	Manat	TMM	Taka	BDT
Dinars	CSD	Markka	FIM	Tala	WST
Djibouti Franc	DJF	Mauritius Rupee	MUR	Tanzanian Shilling	TZS
Dobra	STD	Metical	MZM	Tenge	KZT

Currency	Code	Currency	Code	Currency	Code
Dominican Peso	DOP	Mexican Peso	MXN	Timor Escudo	TPE
Dong	VND	Moldovan Leu	MDL	Tolar	SIT
Drachma	GRD	Moroccan Dirham	MAD	Trinidad and Tobago Dollar	TTD
East Caribbean Dollar	XCD	Naira	NGN	Tugrik	MNT
Egyptian Pound	EGP	Nakfa	ERN	Tunisian Dinar	TND
Ekwele	CQE	Namibia Dollar	NAD	Turkish Lira	TRL
El Salvador Colon	SVC	Nepalese Rupee	NPR	UAE Dirham	AED
Ethiopian Birr	ETB	Netherlands	ANG	Uganda Shilling	UGX
euro	EUR	Netherlands Guilder	NLG	Ukrainian Hrivna	UAH
Falkland Pound	FKP	New Dinar	YUM	US Dollar	USD
Fiji Dollar	FJD	New Israeli Sheqel	ILS	Uzbekistan Sum	UZS
Forint	HUF	New Kwanza	AON	Vatu	VUV
French Franc	FRF	New Lei	RON	Yemeni Rial	YER
Gibraltar Pound	GIP	New Taiwan Dollar	TWD	Yen	JPY
Gourde	HTG	New Zaire	ZRN	Yuan Renminbi	CNY
Guarani	PYG	New Zealand Dollar	NZD	Zimbabwe Dollar	ZWD
Guernsey Pounds	GGP	Ngultrum	BTN	Zloty	PLN
Guinea Franc	GNF	North Korean Won	KPW		

## Dimension table

This table specifies the dimensions in the model. This table must include the dimensions that are required for building the structure of all the modules. This table includes the dimension attributes if dimension attributes are used in the model. Do not include numeric attributes, text attributes, and Boolean attributes.

### CAUTION:

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Reference	AlphaNumeric	64	key	Required	Required	–
Name	AlphaNumeric	64	–	Required	Required	–
ShortReference	AlphaNumeric	18	–	Optional	Required	Added in Abm 7.2

## DimensionAttributeAssociation table

This table specifies the accounts that are associated with dimension attributes. The association can be specified with the column ItemReference or with the columns DimRef<n> and DimMemberRef<n>.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key1, key2	Required	Required	–
Scenario	AlphaNumeric	64	key1, key2	Required	Required	–
ItemModuleReference	AlphaNumeric	–	key1, key2	Required	Required	Module reference
ItemDimSignature	Signature	–	key1	Required 1	Required	–
ItemReference	AlphaNumeric	–	key2	Required 2	Required	–
AttributeDimRef	AlphaNumeric	–	key1, key2	Required	Required	Non-structural dimensions
AttributeDimName	AlphaNumeric	–	–	–	Optional	–
AttributeDimMemberRef	AlphaNumeric	–	key1, key2	Required	Required	–
AttributeDimMemberName	AlphaNumeric	–	–	–	Optional	–
IsSystem	Number	–	–	Optional	Optional	To find out whether the associations are system generated or user defined.

---

## DimensionLevel table

This table specifies the name of each level in a dimension hierarchy.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
DimRef	AlphaNumeric	64	key	Required	Required	–
LevelNo	Integer	–	key	Required	Required	>=1
Name	AlphaNumeric	64	–	Required	Required	–

---

## DimensionMember table

This table specifies the hierarchy for each dimension.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
DimRef	AlphaNumeric	64	key	Required	Required	–
DimName	AlphaNumeric	–	–	–	Optional	–
Reference	AlphaNumeric	64	key	Required	Required	–
Name	AlphaNumeric	64	–	Required	Required	–
ParentReference	AlphaNumeric	64	–	Required	Required	Must be an existing DimensionMember Reference in the model or in this table. Root element can be identified as one of: empty string, Null, "All".
DimLevel	Integer	–	–	Optional	Optional	Hierarchy level override. Defaults to ParentLevel+1. Ignored if null or <=0.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
DimLevelName	AlphaNumeric	64	–	Optional	Optional	Hierarchy level override. Ignored if null. This takes precedent over DimLevel if this is non-null and DimLevel is also included.
DisplayOrder	Float	–	–	Optional	Optional	–

---

## DimensionOrder table

This table specifies the order of the dimensions in the modules. The records in this table must be sorted according to module type and sequence number.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
ModuleReference	AlphaNumeric	64	key	Required	Required	Module reference
SequenceNumber	Integer (>=1)	–	key	Required	Required	Order of dimension in module structure
DimRef	AlphaNumeric	64	–	Required	Required	Dimension Reference
DimName	AlphaNumeric	64	–	–	Optional	Dimension Name

---

## DimMemberDimAttrAssociation

This table stores attributes that are attached to dimension members.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Period	AlphaNumeric	64	key	Required	Required	–
Scenario	AlphaNumeric	64	key	Required	Required	–

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
DimMemberRef	AlphaNumeric	64	key	Required	Required	Dimension member reference
DimRef	AlphaNumeric	64	key	Required	Required	Dimension reference
AttributeDimRef	AlphaNumeric	64	key	Required	Required	Attribute dimension reference
AttributeDimMemberRef	AlphaNumeric	64	key	Required	Required	Attribute dimension member reference

---

## DimMemberValueAttrAssociation

This table stores the value of attributes that are attached to dimension members.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Period	AlphaNumeric	64	key	Required	Required	–
Scenario	AlphaNumeric	64	key	Required	Required	–
DimMemberRef	AlphaNumeric	64	key	Required	Required	Dimension member reference
DimRef	AlphaNumeric	64	key	Required	Required	Dimension reference
AttributeReference	AlphaNumeric	64	key	Required	Required	Attribute reference
NumericValue	Float	–	–	Optional	Optional	Numeric value of the associated attribute
StringValue	Memo	–	–	Optional	Optional	Numeric value of the associated attribute

---

## Driver table

This table specifies the drivers.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Name	AlphaNumeric	64	key	Required	Required	–
DriverType	AlphaNumeric	64	–	Required	Required	The drivers supported in 8.1 are of type "Percentage", "Evenly Assigned", "Standard"
UniqueDriverQuantities	Yes/No	–	–	Optional	Optional	Default=Yes ( for DriverType<>"Center")
Formula	Memo	2048	–	Optional	Optional	Default=""
UseFixedQuantities	Yes/No	–	–	Optional	Optional	Default based on DriverType
UseVariableQuantities	Yes/No	–	–	Optional	Optional	Default based on DriverType
UseWeightedQuantities	Yes/No	–	–	Optional	Optional	Default based on DriverType
SequenceNumber	Integer	–	–	Optional	Optional	Default=1
FixedDriverQuantityOverride	AlphaNumeric	64	–	Optional	Optional	Property API name
VariableDriverQuantityOverride	AlphaNumeric	64	–	Optional	Optional	Property API name
IdleFlowMethod	AlphaNumeric	64	–	Optional	Optional	"DontAssign"   "EvenlyAssign"   "UseDriverQuantities"   "UserEntered"   "UserProportion", Default="DontAssign"
UserEnteredCostAllocation	Yes/No	–	–	Optional	Required	Default=No
RuleFormula	AlphaNumeric	2048	–	Optional	Optional	–
UserRuleFormula	Yes/No	–	–	Optional	Optional	–

---

## EnteredCostElement table

This table specifies the entered cost elements and the account that is associated with each entered cost element. The account that is associated with an entered cost element can be specified with the column AccountReference or with the columns DimRef<n> and DimMemberRef<n>.

### CAUTION:

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.



Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key	Required	Required	–
Scenario	AlphaNumeric	64	key	Required	Required	–
ModuleReference	AlphaNumeric	64	key	Required	Required	Module reference
AccountReference	AlphaNumeric	64	–	Required 1	Required	–
AccountDimSignature	Signature	–	–	Required 2	Required	There is no column of this name. Dimension signature contains dimension members involved in the account creation. So DimRef(n) and DimmemberRef(n) are mainly required for dimension signature. DimName, DimMemberName, Dimlevel, DimLevelname are optional
Reference	AlphaNumeric	64	key	Required	Required	–
Name	AlphaNumeric	64	–	Required	Required	–
EnteredCost	Numeric	–	–	Optional	Optional	Default=0.0
UnitCostEntered	Numeric	–	–	Optional	Optional	Added in 8.1
ForWhatIfAnalysis	Yes/No	–	–	Optional	Optional	Added in 8.1 to indicate entered cost elements marked for What-If analysis.

## Model table

### Model table schema

This table specifies the model name, the description, and the default values for the model. SAS Cost and Profitability Management uses this table for export only.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Name	AlphaNumeric	64	–	Optional	Optional	Use can enter from UI.
Description	AlphaNumeric	64	–	Optional	Optional	–
DefaultOutputQuantity	Float	–	–	Optional	Optional	=0 or =1

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
BaseCurrency	AlphaNumeric	64	–	Optional	Optional	Currency code defined in CurrencyDefinition. Can only be defined when creating a new model.
AdvancedOptions	Memo	–	–	Optional	Optional	Information about default drivers for modules, numeric attributes and stages include in the model. It is stored in XML format
Reference	AlphaNumeric	8	–	Optional	Optional	Use can enter from UI.
ColumnLayout	AlphaNumeric	256	–	Optional	Optional	Not in use in 8.1. Because column layout can be stored seperately for each module (N-module) and the dimension view. So the column layouts are part of AdvancedOptions XML itself

## Currency Code

For the currency code used in the BaseCurrency field, see “[Currency Codes](#)” on page 125.

## AdvancedOptions Field

### Overview

When you import data into a model, you can write XML that specifies the advanced options of the model's properties. You specify this XML in the AdvancedOptions field of the Model table.

### Sample XML for the Advanced Options Field

The following example illustrates the use of the XML AdvancedOptions field:

```
<AdvancedOptions>
  <Modules>
    <Module Type="ExternalUnit" DefaultDriver="Stamdard" />
    <Module Type="Resource" DefaultDriver="Evenly Assigned" />
    <Module Type="Activity" DefaultDriver="Evenly Assigned" />
    <Module Type="CostObject" DefaultDriver="Evenly Assigned" />
  </Modules>
  <Cube Type="Common">
    <NumericAttributes>
      <NumericAttribute Reference="FI" />
    </NumericAttributes>
  </Cube>
  <Cube Type="DetailedContribution">
    <StageDefinition Type="Attribute" IncludeExternalUnitAsStage="true">
      <Stage Reference="S1R" CostFlow="In" />
      <Stage Reference="S2PA" CostFlow="In" />
      <Stage Reference="S3I" CostFlow="In" />
    </StageDefinition>
  </Cube>
</AdvancedOptions>
```

```

    <Stage Reference="S4S" CostFlow="In" />
    <Stage Reference="S5D" CostFlow="In" />
    <Stage Reference="S6F" CostFlow="In" />
    <Stage Reference="S7f" CostFlow="In" />
  </StageDefinition>
</Cube>
</AdvancedOptions>

```

The following is an example of the StageDefinition element with a Type of Module:

```

<StageDefinition Type="Module">
  <Stage Reference="ExternalUnit" CostFlow="Out" />
  <Stage Reference="Resource" CostFlow="Out" />
  <Stage Reference="Activity" CostFlow="Out" />
  <Stage Reference="CostObject" CostFlow="In" />
</StageDefinition>

```

### **XML AdvancedOptions structure**

The following structure shows the elements used in the XML AdvancedOptions field and their relationships to each other. All of the elements are required.

```

<AdvancedOptions>
  <Modules>
    <Module attributes/>
  </Modules>
  <Cube Type="Common">
    <NumericAttributes>
      <NumericAttribute attributes/>
    </NumericAttributes>
  </Cube>
  <Cube Type="DetailedContribution">
    <StageDefinition attributes>
      <Stage attributes/>
    </StageDefinition>
  </Cube>
</AdvancedOptions>

```

### **AdvancedOptions element (required)**

AdvancedOptions is the root element. This element has no attributes.

### **Modules element (required)**

This element contains elements that specify information for each module in a model. This element has no attributes.

### **Module element (required)**

This element contains attributes that specify the type of module and its default driver.

Attribute	Required	Values	Description
Type	Required (one Type attribute for each module)	"Resource" "Activity" "CostObject" "ExternalUnit"	Modules in a model

Attribute	Required	Values	Description
DefaultDriver	Required	String	Name of the default driver

### ***Cube Type=Common element (required)***

This element contains an attribute that specifies information that is common to all cubes.

### ***NumericAttributes element (required)***

This element contains elements that specify the numeric attributes that are included in cubes. This element has no attributes.

### ***NumericAttribute element (required)***

This element contains attributes that specify information about the numeric attributes that are included in cubes.

Attribute	Required	Values	Description
Reference	Required	String	Reference of the numeric attribute

### ***Cube Type=DetailedContribution element (required)***

This element contains an attribute that specifies information about the Multi-stage Contributions cube.

### ***StageDefinition element (required)***

This element contains attributes that specify how stages are defined.

Attribute	Required	Values	Description
Type	Required	"Module" "Attribute"	Method used to define stages
IncludeExternalUnit AsStage	Required when Type is "Attribute"	"true" "false"	Indicates whether external units are included in the stages

### ***Stage element (required)***

This element contains attributes that specify how stages are defined.

Attribute	Required	Values	Description
Reference	Required	String	Reference of each stage  When the StageDefinition Type is "Module": "Resource" "Activity" "CostObject" "ExternalUnit"  When the StageDefinition Type is "Attribute": Reference of the attribute
CostFlow	Required	"In" "Out"	Type of cost to use for a stage

**Related Topics:**

- Example XML AdvancedOptions field

---

## Module table

**CAUTION:**

If you are using Oracle as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Name	AlphaNumeric	64	—	Required	Required	Module Name in N-Module
Reference	AlphaNumeric	64	key	Required	Required	Module Reference in N-Module
ModuleOrder	Float	—	—	Required	Required	Module's order

---

## Period table

This table specifies the period names, the descriptions, and the start dates and the end dates.

**CAUTION:**

If you are using Oracle as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Reference	AlphaNumeric	64	key	Required	Required	–
ParentReference	AlphaNumeric	64	–	Required	Required	Root element can be identified as one of: Empty string, Null, "All"
Name	AlphaNumeric	64	–	Required	Required	–
StartDate	Date	–	–	Required	Required	May need to change if parent is in different date range
EndDate	Date	–	–	Required	Required	May need to change if parent is in different date range
Description	Memo	–	–	Optional	Optional	–

---

## PeriodLevel table

This table specifies the level names for period hierarchies.

**CAUTION:**

If you are using Oracle as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
LevelNo	Integer	–	key	Required	Required	>=1
Name	AlphaNumeric	64	–	Required	Required	–

---

## RollupAccount table

This table specifies the rollup accounts that are at a higher level than the leaf accounts. The rollup accounts' numeric properties are the rollup of the corresponding properties of the descendant leaf accounts, based on the dimensional hierarchy.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
ModuleReference	AlphaNumeric	64	key	–	Optional	Module reference
Period	AlphaNumeric	64	key	–	Optional	–
Scenario	AlphaNumeric	64	key	–	Optional	–
DimSignature	Signature	–	key	–	Optional	There is no column of this name. Dimension signature contains dimension members involved in the account creation. So DimRef and DimmemberRef are mainly required for dimension signature. DimName, DimMemberName, Dimlevel, DimLevelname are optional. Only a single Dim/DimMember pair can be specified.
DimRef	AlphaNumeric	64	–	–	Optional	–
DimName	AlphaNumeric	64	–	–	Optional	–
DimMemberRef	AlphaNumeric	64	–	–	Optional	–
DimMemberName	AlphaNumeric	64	–	–	Optional	–
DimLevel	Integer	–	–	–	Optional	–
DimLevelName	AlphaNumeric	64	–	–	Optional	–
AllocatedCost	Float	–	–	–	Optional	–
AssignedCost	Float	–	–	–	Optional	–
AssignedNonReciprocalCost	Float	–	–	–	Optional	–
AssignedReciprocalCost	Float	–	–	–	Optional	–
Cost	Float	–	–	–	Optional	–
DrivenCost	Float	–	–	–	Optional	–
EnteredCost	Float	–	–	–	Optional	–
IdleCost	Float	–	–	–	Optional	–
IdlePercentage	Float	–	–	–	Optional	–
Profit	Float	–	–	–	Optional	–
ReceivedAllocatedCost	Float	–	–	–	Optional	–

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ReceivedCost	Float	–	–	–	Optional	–
ReceivedDrivenCost	Float	–	–	–	Optional	–
ReceivedNonReciprocalCost	Float	–	–	–	Optional	–
ReceivedReciprocalCost	Float	–	–	–	Optional	–
Revenue	Float	–	–	–	Optional	–
UnassignedCost	Float	–	–	–	Optional	–
SoldCost	Float	–	–	–	Optional	Added in 8.1
ForWhatIfAnalysis	Yes/No	–	–	Optional	Optional	Added in 8.1 to indicate rollup accounts marked for What-If analysis

---

## Scenario table

This table specifies the scenario names and the descriptions.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Reference	AlphaNumeric	64	key	Required	Required	–
ParentReference	AlphaNumeric	64	–	Required	Required	Root element can be identified as one of: empty string, Null, "All"
Name	AlphaNumeric	64	–	Required	Required	–
Description	Memo	–	–	Optional	Optional	–

---

## ScenarioLevel table

This table specifies the level names for scenario hierarchies.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.



Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
LevelNo	Integer	–	key	Required	Required	>=1
Name	AlphaNumeric	64	–	Required	Required	–

---

## ValueAttribute table

This table specifies the attribute hierarchy for numeric attributes , text attributes , and Boolean attributes.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Reference	AlphaNumeric	64	key	Required	Required	–
Name	AlphaNumeric	64	–	Required	Required	–
ParentReference	AlphaNumeric	64	–	Required	Required	–
Type	AlphaNumeric	64	–	Required	Required	"Tagged"   "Numeric"   "Text"   "Center"
UnitOfMeasure	AlphaNumeric	64	–	Optional	Optional	Only applicable for "Numeric"

---

## ValueAttributeAssociation table

This table specifies the accounts that are associated with numeric attributes, text attributes, and Boolean attributes. The association can be specified with the column ItemReference or with the columns DimRef<n> and DimMemberRef<n>.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
Period	AlphaNumeric	64	key1, key2	Required	Required	–
Scenario	AlphaNumeric	64	key1, key2	Required	Required	–
ItemModuleReference	AlphaNumeric	–	key1, key2	Required	Required	Module reference
ItemDimSignature	Signature	–	key1	Required 1	Required	–
ItemReference	AlphaNumeric	64	key2	Required 2	Required	–
AttributeReference	AlphaNumeric	–	key1, key2	Required	Required	–
NumericValue	Float	–	–	Optional	Optional	–
StringValue	Memo	–	–	Optional	Optional	–
IsSystem	Number	–	–	Optional	Optional	To find out whether the associations are system generated or user defined.

## ValueAttributePeriodicDef table

This table specifies the default value and formula for the numeric attributes.

**CAUTION:**

**If you are using Oracle** as your database, then both the staging table name and the names of all the columns in the staging table must be uppercase only.

Field Name	Type of Data	Max Len	Key	Import	Export	Notes
ModelName	AlphaNumeric	64	–	–	Optional	–
Period	AlphaNumeric	64	key	Required	Required	–
Scenario	AlphaNumeric	64	key	Required	Required	–
Reference	AlphaNumeric	64	key	Required	Required	–
DefaultValue	AlphaNumeric	1024	–	Optional	Optional	–
Formula	AlphaNumeric	1024	–	Optional	Optional	Default=""
ForWhatIfAnalysis	Yes/No	–	–	Optional	Optional	Added in 8.1 to indicate numeric attributes marked for What-If analysis

## Part 4

---

# Export Registered Tables

*Chapter 14*

**Working with Registered Tables** ..... 145

*Chapter 15*

**Registered Table Schemas** ..... 153



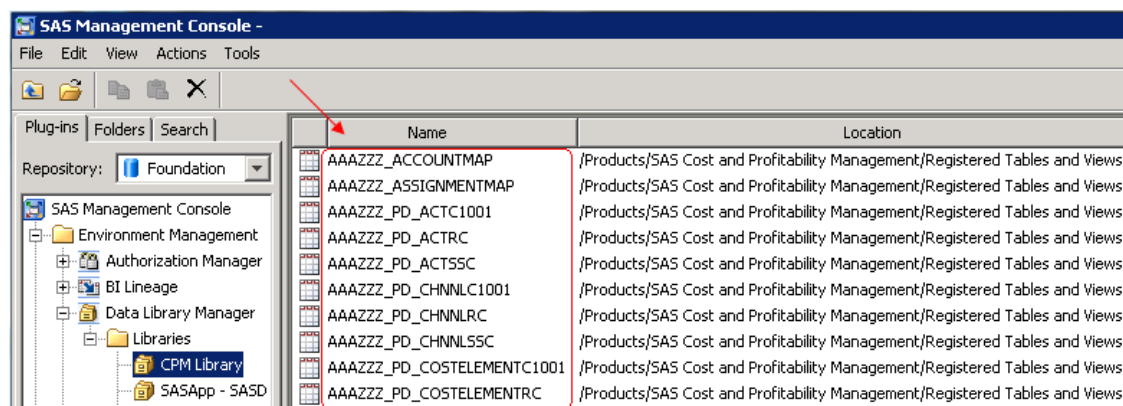
## Chapter 14

# Working with Registered Tables

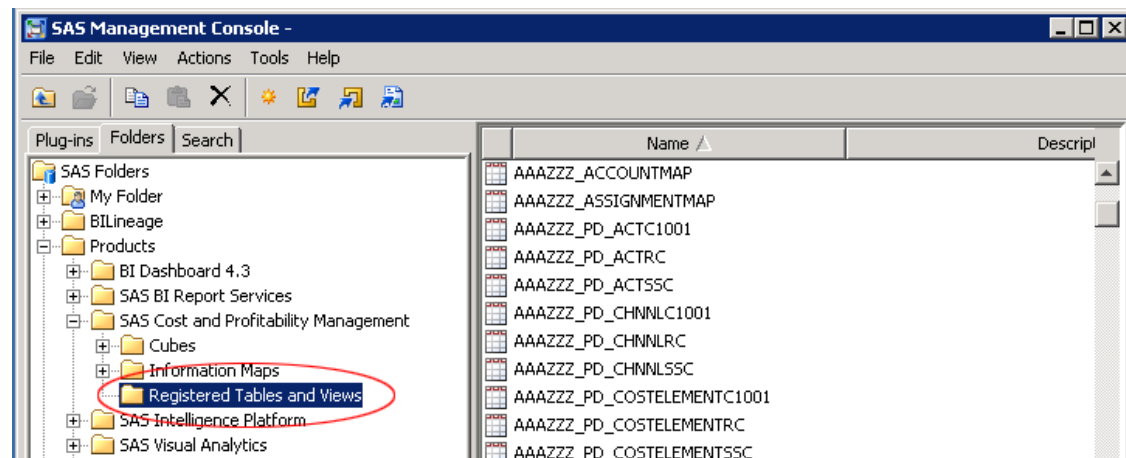
Overview .....	145
Model-Structure Tables .....	146
Model-Independent Views .....	147
Fact-Table Related Views .....	148
How To Export and Register Tables .....	150

## Overview

You can create database tables and views that mirror the memory-mapped files used by SAS Cost and Profitability Management. Each such database table or view is registered in SAS metadata and can be accessed by other SAS programs as well as your own. To see metadata for the registered tables and views, from the **Plug-ins** tab of SAS Management Console expand **Environment Management/Data Library Manager/Libraries/CPM Library**. You see a list of registered tables and views.



Additional metadata is stored in the **Folders** tab under **SAS Folders/Products/SAS Cost and Profitability Management/Registered Tables and Views**.



There are three categories of registered tables and views. Of these three categories, only the first contains tables that you have to export explicitly:

#### Model-structure tables

These tables describe the structure of a model. You must export these tables explicitly. See [“Model-Structure Tables” on page 146](#).

#### Model-independent views

These views are exported and registered automatically. See [“Model-Independent Views” on page 147](#).

#### Fact-table related views

These views are related to fact table generation. They are exported and registered automatically during cube generation. See [“Fact-Table Related Views” on page 148](#).

## Model-Structure Tables

The following tables describe the structure of a model. You must create these tables explicitly. See [“How To Export and Register Tables” on page 150](#).

Table Name	Description
<i>ModelRef_ACCOUNTMAP</i>	A join of <i>ModelRef_PV_ACCOUNT</i> with other registered tables. Includes details for multiple periods, dimensional signature, and all numeric properties. See <a href="#">“modelRef_ACCOUNTMAP” on page 156</a> .
<i>ModelRef_ASSIGNMENTMAP</i>	A join of <i>ModelRef_PV_ASSIGNMENT</i> with other registered tables. Includes details for source and destination dimensional signatures, and all entered and calculated properties. See <a href="#">“modelRef_ASSIGNMENTMAP” on page 159</a> .
<i>ModelRef_PV_ACCOUNT</i>	Details for multiple periods, dimensional signature, and all numeric properties. See <a href="#">“modelRef_PV_ACCOUNT” on page 159</a> .

Table Name	Description
<i>ModelRef_PV_ASSIGNMENT</i>	Details for source and destination dimensional signatures, and all entered and calculated properties. See “ <a href="#">modelRef_PV_ASSIGNMENT</a> ” on page 162.
<i>ModelRef_PV_ASSIGNMENTTEXT</i>	Details for source and destination dimensional signatures, and all entered and calculated properties. See “ <a href="#">modelRef_PV_ASSIGNMENTTEXT</a> ” on page 164.
<i>ModelRef_PV_ATTRIBUTE</i>	Details defining the attribute, attribute types, formulas for calculations, and default values. See “ <a href="#">modelRef_PV_ATTRIBUTE</a> ” on page 165.
<i>ModelRef_PV_ATTRIBUTEVALUE</i>	Attribute attachment to the model accounts and the numeric value for the attribute. See “ <a href="#">modelRef_PV_ATTRIBUTEVALUE</a> ” on page 166.
<i>ModelRef_PV_DIMENSION</i>	Details defining the model's dimensions. See “ <a href="#">modelRef_PV_DIMENSION</a> ” on page 167.
<i>ModelRef_PV_DRIVER</i>	Details defining the drivers, driver types, and formulas for rules assignment and for calculations. See “ <a href="#">modelRef_PV_DRIVER</a> ” on page 167.
<i>ModelRef_PV_ENTEREDCE</i>	Details of all entered cost elements, with their corresponding account attachment and values. See “ <a href="#">modelRef_PV_ENTEREDCE</a> ” on page 168.

---

## Model-Independent Views

The following model-independent views are created automatically.

View Name	Description
PUBLICMODEL	Contains a list of models. See “ <a href="#">PUBLICMODEL</a> ” on page 154.
PUBLICMODELSTATUS	For each model, lists its period scenario associations and whether each one is calculated or not.. See “ <a href="#">PUBLICMODELSTATUS</a> ” on page 155.

View Name	Description
PUBLICPERIOD	Contains a list of periods with their start and end dates. See “PUBLICPERIOD ” on page 155.
PUBLICSCENARIO	Contains a list of scenarios. See “PUBLICSCENARIO ” on page 156.

## Fact-Table Related Views

The following model-specific views are related to fact table generation. These views are created and registered automatically during cube generation.

View Name	Description
<p><i>ModelRef_PD_COSTELEMENT&lt;suffix&gt;</i></p> <p>where <b>&lt;suffix&gt;</b> is RC, SSC, or C&lt;cube configuration ID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_COSTELEMENTRC ABC_PD_COSTELEMENTSSC ABC_PD_COSTELEMENTC1001</p>	<p>Details defining the types of cost elements.</p> <p>See “modelRef_PD_COSTELEMENT&lt;suffix&gt;” on page 169.</p>
<p><i>ModelRef_PD_dimShortRef&lt;suffix&gt;</i></p> <p>where <b>&lt;suffix&gt;</b> is RC, SSC, or C&lt;cube configuration ID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_REGIONRC ABC_PD_REGIONSSC ABC_PD_REGIONC1001</p>	<p>Single dimension-member details: level by level, noting ID, Reference, and Name.</p> <p>See “modelRef_PD_dimShortRef&lt;suffix&gt;” on page 169.</p>
<p><i>ModelRef_PD_DRIVER&lt;suffix&gt;</i></p> <p>where <b>&lt;suffix&gt;</b> is RC, SSC, or C&lt;cube configuration ID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_DRIVERRC ABC_PD_DRIVERSSC ABC_PD_DRIVERC1001</p>	<p>Driver ID and corresponding Driver Name.</p> <p>See “modelRef_PD_DRIVER&lt;suffix&gt; ” on page 170.</p>



View Name	Description
<p><i>ModelRef_PD_MODULE</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_MODULERC</p> <p>ABC_PD_MODULESSC</p> <p>ABC_PD_MODULEEC1001</p>	<p>Details defining the types of modules.</p> <p>See “<a href="#">modelRef_PD_MODULE&lt;suffix&gt;</a>” on page 171.</p>
<p><i>ModelRef_PD_PERIOD</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_PERIODRC</p> <p>ABC_PD_PERIODSSC</p> <p>ABC_PD_PERIODC1001</p>	<p>Details defining the periodic hierarchy.</p> <p>See “<a href="#">modelRef_PD_PERIOD&lt;suffix&gt;</a>” on page 172.</p>
<p><i>ModelRef_PD_SCENARIO</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_SCENARIORC</p> <p>ABC_PD_SCENARIOSSC</p> <p>ABC_PD_SCENARIOC1001</p>	<p>Details defining the types of available scenarios.</p> <p>See “<a href="#">modelRef_PD_SCENARIO&lt;suffix&gt;</a>” on page 172.</p>
<p><i>ModelRef_PD_YESNO</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_YESNORC</p> <p>ABC_PD_YESNOSSC</p> <p>ABC_PD_YESNOC1001</p>	<p>Dimensional definition for Boolean values: Text strings.</p> <p>See “<a href="#">modelRef_PD_YESNO&lt;suffix&gt;</a>” on page 173.</p>

View Name	Description
<p><i>ModelRef_PF_&lt;suffix&gt;</i></p> <p>where <b>&lt;suffix&gt;</b> is RC, SSC, or MSC_C&lt;cubeID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PF_RC</p> <p>ABC_PF_SSC</p> <p>ABC_PF_MSC_C1001</p> <p>If you have checked <b>Also load tables into a library for the SAS LASR Analytic Server</b>, then, in addition to the tables using the naming convention above, tables in star-schema format are also generated that use the following naming convention:</p> <p><i>ModelRef_PF_&lt;suffix&gt;</i></p> <p>where <b>&lt;suffix&gt;</b> is RCSTAR, SSCSTAR, or MSC_C&lt;cubeID&gt;STAR for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PF_RCSTAR</p> <p>ABC_PF_SSCSTAR</p> <p>ABC_PF_MSC_C1001STAR</p>	<p>Fact Table: Stages and member IDs for each step through contribution. Source table for cube generation.</p> <p>See “<i>modelRef_PF_&lt;suffix&gt;</i>” on page 174 .</p>

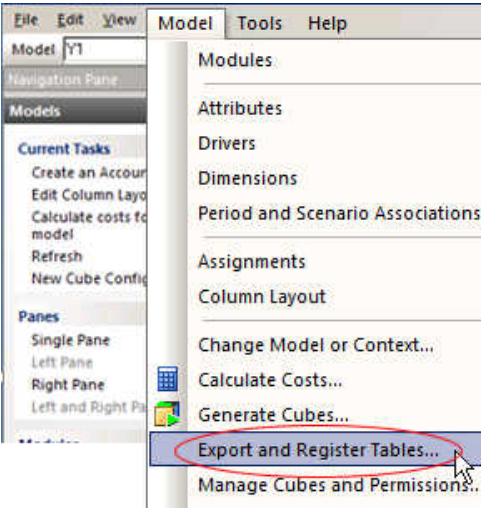
---

## How To Export and Register Tables

Model data is stored in memory-mapped files rather than in a database. You can create database tables and views that mirror the memory-mapped files and are registered in SAS Metadata so as to be available to other SAS programs or for your own use.

To export database tables and views:

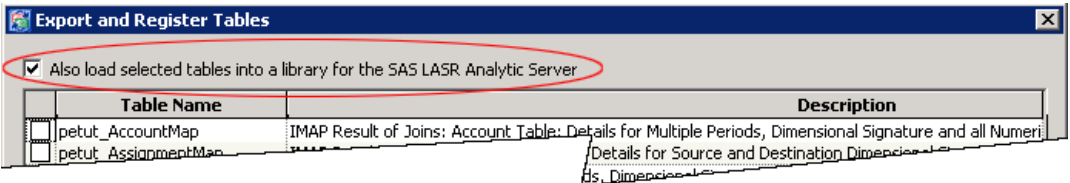
1. Select **Model** ⇒ **Export and Register Tables**.



The Export and Register Tables dialog box opens.

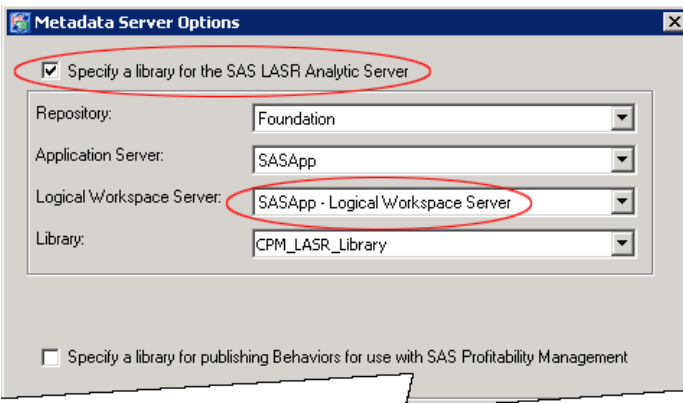
2. Select whether to. **Also load selected tables into a library for the SAS LASR Analytic Server.**

With this option, tables are created and registered in a SAS metadata library that is enabled for the SAS LASR Analytic Server.



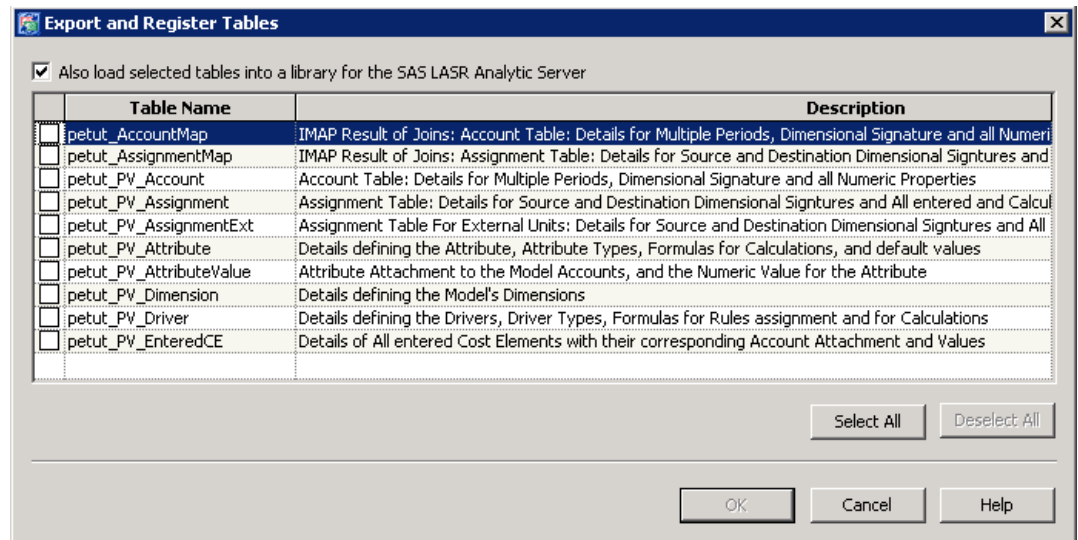
*Note:*

- For the option **Also load selected tables into a library for the SAS LASR Analytic Server** to be available, you must have specified this library on the Metadata Server Options dialog. See “Metadata Server Options” in Chapter 57 of *SAS Cost and Profitability Management: User’s Guide*.
- On the Metadata Server Options dialog, make sure that you have selected **SASApp – Logical Workspace Server** as shown in the picture below.



- You must also have authorization to write to the folder that is enabled for library. If you do not have those rights, the option is disabled. See [“Authorize Users for the LASR-Enabled Metadata Library”](#) on page 296.

3. Select the tables that you want to export.



4. Click **OK**.

To see metadata for the registered tables and views, from the **Plug-ins** tab of SAS Management Console expand **Environment Management/Data Library Manager/Libraries/CPM Library**. Additional metadata is stored in the **Folders** tab under **SAS Folders/Products/SAS Cost and Profitability Management/Registered Tables and Views**.

*Note:*

- Model-independent tables are exported and registered automatically. You do not have to select them in order for them to be exported, so they are not listed in this dialog. See [“Model-Independent Views” on page 147](#).
- Fact-table tables are exported automatically when you generate a model, so you do not have to select them here. See [“Fact-Table Related Views” on page 148](#). But, you do have to specify in a cube configuration if you also want the fact-table tables to be exported for the SAS LASR Analytic Server. See “Cube Configuration: Select a Model and General Options” in Chapter 58 of *SAS Cost and Profitability Management: User's Guide*.

## Chapter 15

# Registered Table Schemas

---

Column Descriptions for All Tables . . . . .	154
PUBLICMODEL . . . . .	154
PUBLICMODELSTATUS . . . . .	155
PUBLICPERIOD . . . . .	155
PUBLICSCENARIO . . . . .	156
modelRef_ACCOUNTMAP . . . . .	156
modelRef_ASSIGNMENTMAP . . . . .	159
modelRef_PV_ACCOUNT . . . . .	159
modelRef_PV_ASSIGNMENT . . . . .	162
modelRef_PV_ASSIGNMENTTEXT . . . . .	164
modelRef_PV_ATTRIBUTE . . . . .	165
modelRef_PV_ATTRIBUTEVALUE . . . . .	166
modelRef_PV_DIMENSION . . . . .	167
modelRef_PV_DRIVER . . . . .	167
modelRef_PV_ENTEREDCE . . . . .	168
modelRef_PD_dimShortRef<suffix> . . . . .	169
modelRef_PD_COSTELEMENT<suffix> . . . . .	169
modelRef_PD_DRIVER<suffix> . . . . .	170
modelRef_PD_MODULE<suffix> . . . . .	171
modelRef_PD_PERIOD<suffix> . . . . .	172
modelRef_PD_SCENARIO<suffix> . . . . .	172
modelRef_PD_YESNO<suffix> . . . . .	173
modelRef_PF_<suffix> . . . . .	174

## Column Descriptions for All Tables

### Field Name

The field name that is provided in the registered table.

### Field Type

The definition of each field type (numeric or text field).

### Join Field

The specific ID fields to join between tables. These fields define keys that can be used to link multiple tables for reporting. This column identifies the corresponding join field.

### Properties Reference

The specific corresponding property. See Chapter 87, “Alphabetic List of Properties,” in *SAS Cost and Profitability Management: User’s Guide*.



### Model View Columns

The specific property that is displayed in the model column view.

## PUBLICMODEL

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Name	Text			Model

The following is a sample PublicModel:

PUBLICMODEL ▾		
	 ID	 NAME
1	1002	AM2
2	1009	Test_AM2
3	1011	FirstE2EScenario
4	1012	FourthSceETOE
5	1000	OLAPView_TC01
6	1001	CreateNewM
7	1018	FifthE2EScenario
8	1019	SecondE2EScen...
9	1020	yubope
10	1021	pecopy
11	1022	Parcel Express b...
12	1023	pe_murali

## PUBLICMODELSTATUS

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
CalculationStatus	Text			Calculation status not viewable in the user interface
CubeStatus	Text			Cube status not viewable in the user interface
ModelID	Number			ID values not viewable in the user interface
Period	Text		Name	Name
Scenario	Text		Name	Name

The following is a sample PublicModeStatus:

PUBLICMODELSTATUS ▾				
	MODELID	PERIOD	SCENARIO	CALCULATIONSTATUS
1	1012	E2E2009	ETE2_Actual	CALCULATED
2	1019	E2E2009	ETE2_Actual	NOT CALCULATED
3	1020	2008 Q1	Actual	CALCULATED
4	1021	2008 Q1	Actual	CALCULATED
5	1022	2008 Q1	Actual	CALCULATED
6	1023	2008 Q1	Actual	CALCULATED
7	1020	2008 Q1	Planned	CALCULATED
8				
9				

## PUBLICPERIOD

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Description	Text			Description
EndDate	Number			EndDate
ID	Number	PeriodID		ID values not viewable in the user interface
Name	Text		Name	Name
ParentID	Number	PeriodID		ID values not viewable in the user interface
Reference	Text		Reference	Reference

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
StartDate	Number			StartDate

The following is a sample PublicPeriod:

PUBLICPERIOD ▾							
	ID	PARENTID	NAME	REFERENCE	STARTDATE	ENDDATE	DESCRIPTION
1	25	0	E2E2009	E2E2009	01JAN2009:00:0...	12JAN2009:00:0...	Set start and end...
2	27	3	2008 Q1	2008 Q1	01JAN2008:00:0...	31MAR2008:00:0...	
3	20	0	2004	2004	01JAN2004:00:0...	31DEC2004:00:0...	
4	21	20	2004_Q4	2004_Q4	01OCT2004:00:0...	31DEC2004:00:0...	
5	22	21	2004_12	2004_12	01DEC2004:00:0...	31DEC2004:00:0...	
6	26	0	SCE_04	SCE_04	01DEC2004:00:0...	31DEC2004:00:0...	
7	1						

## PUBLICSCENARIO

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Description	Text			Description
ID	Number	ScenarioID		ID values not viewable in the user interface
Name	Text		Name	Name
ParentID	Number	ScenarioID		ID values not viewable in the user interface
Reference	Text		Reference	Reference

The following is a sample PublicScenario:

PUBLICSCENARIO ▾					
	ID	PARENTID	NAME	REFERENCE	DESCRIPTION
1	23	0	ETE2_Actual	ETE2_Actual	Add ETE2_Actual...
2	25	0	Plan	Plan	
3	20	0	Planning	Planning	Tested by Song fr...
4	21	20	What_if1	What_if1	
5	22	20	What_if2	What_if2	
6	24	0	SCE_04	SCE_04	
7	1	0	Actual	Actual	Actual
8	2	0	Budget	Budget	

## modelRef\_ACCOUNTMAP

This public registered table is the join of the following other registered tables:



- “modelRef\_PV\_ACCOUNT ” on page 159
- “PUBLICPERIOD ” on page 155
- “PUBLICSCENARIO ” on page 156
- “modelRef\_PV\_DRIVER ” on page 167
- “modelRef\_PD\_dimShortRef<suffix>” on page 169

Field Name	From Registered Table	Field Type	Join Field	Properties Reference	Model View Columns
AllocatedCost	modelRef_PV_Account	Number		Allocated Cost	Allocated Cost
AssignedCost	modelRef_PV_Account	Number		Assigned Cost	Assigned Cost
AssignedIdleCost	modelRef_PV_Account	Number		Assigned Idle Cost	Assigned Idle Cost
AssignedIdleQuantity	modelRef_PV_Account	Number		Assigned Idle Quantity	Assigned Idle Quantity
AssignedNonReciprocalCost	modelRef_PV_Account	Number		Assigned Non-Reciprocal Cost	Assigned Non-Reciprocal Cost
Cost	modelRef_PV_Account	Number		Cost	Cost
CostReceived	modelRef_PV_Account	Number		Received Cost	Received Cost
Dim<dimension_short_ref>	modelRef_PV_Account	Number			Specific dimensions in the model - one row per dimension in the model
DrivableCost	modelRef_PV_Account	Number		Drivable Cost	Drivable Cost
DrivenCost	modelRef_PV_Account	Number		Driven Cost	Driven Cost
DrivenQuantity	modelRef_PV_Account	Number		Driven Quantity	Driven Quantity
DriverID	modelRef_PV_Account	Number	Driver.ID		ID values not viewable in the user interface
DriverRate	modelRef_PV_Account	Number		Driver Rate	Driver Rate
HasAssignments	modelRef_PV_Account	Number		Has Assignments	Has Assignments
HasEnteredCost	modelRef_PV_Account	Number		Has Entered Cost	Has Entered Cost
HasIdleCost	modelRef_PV_Account	Number		Has Idle Cost	Has Idle Cost
HasNotes	modelRef_PV_Account	Number		Has Notes	Has Notes
HasUsedCost	modelRef_PV_Account	Number		Has Used Costs	Has Used Costs
IdleCost	modelRef_PV_Account	Number		Idle Cost	Idle Cost
IdlePercentage	modelRef_PV_Account	Number		Idle Percentage	Idle Percentage
IdleQuantity	modelRef_PV_Account	Number		Idle Quantity	Idle Quantity
Measure	modelRef_PV_Account	Text		Unit of Measure	Unit of Measure

Field Name	From Registered Table	Field Type	Join Field	Properties Reference	Model View Columns
ModuleID	<i>modelRef_PV_Account</i>	Number			ID values not viewable in the user interface
ModuleOrder	<i>modelRef_PV_Account</i>	Number		Module Order	Module Order
Name	<i>modelRef_PV_Account</i>	Text		Name	Name
Note	<i>modelRef_PV_Account</i>	Text		Periodic Note	Periodic Note
OutputQuantity	<i>modelRef_PV_Account</i>	Number		Output Quantity	Output Quantity
PeriodID	<i>modelRef_PV_Account</i>	Number	Period.ID		ID values not viewable in the user interface
Profit	<i>modelRef_PV_Account</i>	Number		Profit	Profit
PublishName	<i>modelRef_PV_Account</i>	Text			ID values not viewable in the user interface
Reference	<i>modelRef_PV_Account</i>	Text		Reference	Reference
Revenue	<i>modelRef_PV_Account</i>	Number		Revenue	Revenue
ScenarioID	<i>modelRef_PV_Account</i>	Number	Scenario.ID		ID values not viewable in the user interface
SoldQuantity	<i>modelRef_PV_Account</i>	Number		Sold Quantity	Sold Quantity
TotalDriverQuantity	<i>modelRef_PV_Account</i>	Number		Total Driver Quantity (TDQ)	Total Driver Quantity (TDQ)
TotalDriverQuantityBasic	<i>modelRef_PV_Account</i>	Number		Total Driver Quantity Basic (TDQBasic)	Total Driver Quantity Basic (TDQBasic)
TotalDriverQuantityCalculated	<i>modelRef_PV_Account</i>	Number		Total Driver Quantity Calculated (TDQCalc)	Total Driver Quantity Calculated (TDQCalc)
Type	<i>modelRef_PV_Account</i>	Text		Type	Type
UnassignedCost	<i>modelRef_PV_Account</i>	Number		Unassigned Cost	Unassigned Cost
UnassignedQuantity	<i>modelRef_PV_Account</i>	Number		Unassigned Quantity	Unassigned Quantity
UnitCost	<i>modelRef_PV_Account</i>	Number		Unit Cost	Unit Cost
UnitCostEntered	<i>modelRef_PV_Account</i>	Number		Entered Unit Cost	Ent Unit Cost
UnitProfit	<i>modelRef_PV_Account</i>	Number		Unit Profit	Unit Profit
UnitRevenue	<i>modelRef_PV_Account</i>	Number		Unit Revenue	Unit Revenue
UsedCost	<i>modelRef_PV_Account</i>	Number		Used Cost	Used Cost
UsedQuantity	<i>modelRef_PV_Account</i>	Number		Used Quantity	Used Quantity

Field Name	From Registered Table	Field Type	Join Field	Properties Reference	Model View Columns
UserOutputQuantity	modelRef_PV_Account	Number		Output Quantity UE	Output Quantity UE
Name	PUBLICPERIOD	Text		Name	Name
StartDate	PUBLICPERIOD	Number			StartDate
Name	PUBLICSCENARIO	Text		Name	Name
Name	modelRef_PV_Driver	Text		Name	Name
Name	modelRef_PV_PD_dimShortRef	Text		Name	Name
Reference	modelRef_PV_PD_dimMemRef	Text		Name	Name

---

## modelRef\_ASSIGNMENTMAP

This registered table is the join of the following other registered tables:

- [“modelRef\\_PV\\_ASSIGNMENTTEXT ” on page 164](#)
- [“PUBLICPERIOD ” on page 155](#)
- [“PUBLICSCENARIO ” on page 156](#)
- [“modelRef\\_PV\\_DRIVER ” on page 167](#)

---

## modelRef\_PV\_ACCOUNT

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
AllocatedCost	Number		Allocated Cost	Allocated Cost
AssignedCost	Number		Assigned Cost	Assigned Cost
AssignedIdleCost	Number		Assigned Idle Cost	Assigned Idle Cost
AssignedIdleQuantity	Number		Assigned Idle Quantity	Assigned Idle Quantity
AssignedNonReciprocalCost	Number		Assigned Non-Reciprocal Cost	Assigned Non-Reciprocal Cost
Cost	Number		Cost	Cost
CostReceived	Number		Received Cost	Received Cost

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Dim<dimension_short_ref>	Number			Specific dimensions in the model - one row per dimension in the model
DrivableCost	Number		Drivable Cost	Drivable Cost
DrivenCost	Number		Driven Cost	Driven Cost
DrivenQuantity	Number		Driven Quantity	Driven Quantity
DriverID	Number	Driver.ID		ID values not viewable in the user interface
DriverRate	Number		Driver Rate	Driver Rate
HasAssignments	Number		Has Assignments	Has Assignments
HasEnteredCost	Number		Has Entered Cost	Has Entered Cost
HasIdleCost	Number		Has Idle Cost	Has Idle Cost
HasNotes	Number		Has Notes	Has Notes
HasUsedCost	Number		Has Used Costs	Has Used Costs
IdleCost	Number		Idle Cost	Idle Cost
IdlePercentage	Number		Idle Percentage	Idle Percentage
IdleQuantity	Number		Idle Quantity	Idle Quantity
Measure	Text		Unit of Measure	Unit of Measure
ModuleID	Number			ID values not viewable in the user interface
ModuleOrder	Number		Module Order	Module Order
Name	Text		Name	Name
Note	Text		Periodic Note	Periodic Note
OutputQuantity	Number		Output Quantity	Output Quantity
PeriodID	Number	Period.ID		ID values not viewable in the user interface
Profit	Number		Profit	Profit
PublishName	Text			ID values not viewable in the user interface
Reference	Text		Reference	Reference
Revenue	Number		Revenue	Revenue

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ScenarioID	Number	Scenario.ID		ID values not viewable in the user interface
SoldQuantity	Number		Sold Quantity	Sold Quantity
TotalDriverQuantity	Number		Total Driver Quantity (TDQ)	Total Driver Quantity (TDQ)
TotalDriverQuantityBasic	Number		Total Driver Quantity Basic (TDQBasic)	Total Driver Quantity Basic (TDQBasic)
TotalDriverQuantityCalculated	Number		Total Driver Quantity Calculated (TDQCalc)	Total Driver Quantity Calculated (TDQCalc)
Type	Text		Type	Type
UnassignedCost	Number		Unassigned Cost	Unassigned Cost
UnassignedQuantity	Number		Unassigned Quantity	Unassigned Quantity
UnitCost	Number		Unit Cost	Unit Cost
UnitCostEntered	Number		Entered Unit Cost	Ent Unit Cost
UnitProfit	Number		Unit Profit	Unit Profit
UnitRevenue	Number		Unit Revenue	Unit Revenue
UsedCost	Number		Used Cost	Used Cost
UsedQuantity	Number		Used Quantity	Used Quantity
UserOutputQuantity	Number		Output Quantity UE	Output Quantity UE
UserTotalDriverQuantity	Number		Total Driver Quantity UE (TDQUE)	Total Driver Quantity UE (TDQUE)

The following is a sample PV\_Account:

PEBUZZ_PV_ACCOUNT ▾														
	ID	PERIOD ID	SCENARIO ID	ALLOCATED COST	ASSIGNED COST	ASSIGNED IDLE COST	ASSIGNED IDLE QUANTITY	ASSIGNED NON-RECIPIENT ALLOC COST	COST	COST RECEIVED	COST RECEIVED ASSIGNED	COST RECEIVED BOC	DRIVEN QUANTITY	DRIVEN
1	1275	27	1	0	2...	.	.	12...	12...	0	.	.	.	.
2	1276	27	1	0	6...	.	.	16...	16...	0	.	.	.	.
3	1277	27	1	0	0...	.	.	10...	10...	0	.	.	.	.
4	1278	27	1	0	3...	.	.	13...	13...	0	.	.	.	.
5	1279	27	1	0	8...	.	.	28...	28...	0	.	.	.	.
6	1280	27	1	0	2...	.	.	12...	12...	0	.	.	.	.
7	1281	27	1	0	6...	.	.	36...	36...	0	.	.	.	.

## modelRef\_PV\_ASSIGNMENT

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
AllocatedCost	Number		Allocated Cost	Allocated Cost
AssignedIdleQuantity	Number		Assigned Idle Quantity	Assigned Idle Quantity
Cost	Number		Cost	Cost
DestinationID	Number	Account.ID		ID values not viewable in the user interface
DrivenCost	Number		Driven Cost	Driven Cost
DrivenQuantity	Number		Driven Quantity	Driven Quantity
FixedQuantity	Number		Driver Quantity Fixed	Driver Quantity Fixed
FixedWeight	Number		Driver Weight Fixed	Driver Weight Fixed
IdleCost	Number		Idle Cost	Idle Cost

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
PeriodID	Number	Period.ID		ID values not viewable in the user interface
QuantityBasic	Number		Driver Quantity Basic	Driver Quantity Basic
QuantityCalculated	Number		Driver Quantity Calculated	Driver Quantity Calculated
ScenarioID	Number	Scenario.ID		ID values not viewable in the user interface
SourceID	Number	Account.ID		ID values not viewable in the user interface
UsedCost	Number		Used Cost	Used Cost
UserIdleQuantity	Number		Idle Driver Quantity UE	Idle Driver Quantity UE
VariableQuantity	Number		Driver Quantity Variable	Driver Quantity Variable
VariableWeight	Number		Driver Weight Variable	Driver Weight Variable

The following is a sample PV\_Assignment:

PE_PV_ASSIGNMENT ▾																		
	PERIOD ID	SCENARIO ID	SOURCE ID	DESTINATION ID	ALLOCATED COST	ASSIGNED IDLE QUANTITY	COST	DRIVEN COST	DRIVEN QUANTITY	FIXED QUANTITY	FIXED WEIGHT	IDLE COST	QUANTITY BASIC	QUANTITY CALCULATED	USED COST	USER IDLE QUANTITY	VARIABLE QUANTITY	VARIABLE WEIGHT
1	27	25	12...	12...	.	.	18...	18...	9309	0	.	0	9309	9309	187...	.	0.0...	.
2	27	25	12...	12...	.	.	60...	60...	25	25	.	0	25	25	604...	.	0	.
3	27	25	11...	11...	.	.	55...	55...	285...	0	.	0	2853...	285...	554...	.	0.5	.
4	27	25	11...	12...	.	.	26...	26...	106...	0	.	0	1064...	106...	264...	.	0.45	.
5	27	25	12...	12...	.	.	32...	32...	106...	0	.	0	1064...	106...	325...	.		.
6	27	25	12...	11...	.	.	73...	73...	3200	0	.	0	3200	3200		.		.
7	27	25	12...	12...	.	.	680...	680...			.					.		.
8	27	25	12...	12...	.	.	11...	11...			.					.		.

---

**modelRef\_PV\_ASSIGNMENTTEXT**

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
AllocatedCost	Number		Allocated Cost	Allocated Cost
AssignedIdleQuantity	Number		Assigned Idle Quantity	Assigned Idle Quantity
Cost	Number		Cost	Cost
DestinationID	Number	Account.ID		ID values not viewable in the user interface
DestinationName	Text		Name	Name
DestinationReference	Text		Reference	Reference
DrivenCost	Number		Driven Cost	Driven Cost
DriverID	Number			ID values not viewable in the user interface
DriverQuantity	Number			Driver Quantity
FixedQuantity	Number		Driver Quantity Fixed	Driver Quantity Fixed
FixedWeight	Number		Driver Weight Fixed	Driver Weight Fixed
IdleCost	Number		Idle Cost	Idle Cost
PeriodID	Number	Period.ID		ID values not viewable in the user interface
QuantityBasic	Number		Driver Quantity Basic	Driver Quantity Basic
QuantityCalculated	Number		Driver Quantity Calculated	Driver Quantity Calculated
ScenarioID	Number	Scenario.ID		ID values not viewable in the user interface
SourceID	Number	Account.ID		ID values not viewable in the user interface
SourceName	Text		Name	Name
SourceReference	Text		Reference	Reference
UsedCost	Number		Used Cost	Used Cost
UserIdleQuantity	Number		Idle Driver Quantity UE	Idle Driver Quantity UE
VariableQuantity	Number		Driver Quantity Variable	Driver Quantity Variable



Field Name	Field Type	Join Field	Properties Reference	Model View Columns
VariableWeight	Number		Driver Weight Variable	Driver Weight Variable

The following is a sample PV\_AssignmentExt:

PE_PV_ASSIGNMENTEXT ▾																							
	PERIODID	SCENARIOID	DRIVERID	SOURCEID	SOURCEREFFERENCE	SOURCENAME	DESTINATIONID	DESTINATIONREFERENCE	DESTINATIONNAME	ALLOCATEDCOST	ASSIGNEDIDLEQUANTITY	COST	DRIVENCOST	DRIVENQUANTITY	FIXEDQUANTITY	FIXEDWEIGHT	IDLECOST	QUANTITYBASIC	QUANTITYCALCULATED	USED COST	USERIDLEQUANTITY	VARIABLEQUANTITY	VARIABLEWEIGHT
1	27	25	10...	1270	LA...	Air...	1237	LA...	Sta...			18...	18...	93...	0		0	93...	93...	18...		0.0...	
2	27	25	10...	1228	BC...	Res...	1242	BS	Sto...			60...	60...	25	25		0	25	25	60...		0	
3	27	25	10...	1169	OP...	Equi...	1184	OP	Air...			55...	55...	28...	0		0	28...	28...				
4	27	25	10...	1170	LA...	Air...	1237	LA...	Sta...			26...	26...	10...	0		0	10...	10...				

## modelRef\_PV\_ATTRIBUTE

This view is new for attributes on a dimension member.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Name	Text		Name	Name
Reference	Text		Reference	Reference
ParentID	Number			ID values not viewable in the user interface
Type	Text			Type
DimensionID	Number			ID values not viewable in the user interface
ReferenceID	Number			ID values not viewable in the user interface
Measure	Text		Unit of Measure	UoM

The following is a sample PV\_Attribute:

PE_PV_ATTRIBUTE ▾									
	ID	NAME	REFERENCE	PARENTID	TYPE	DIMENSIONID	REFERENCEID	MEASURE	
1	1069	Completed Expedit...	Completed Expedit...	0	32	.	.	Completed Requ...	
2	1070	Average Time To...	Average Time To...	0	32	.	.	Hours	
3	1071	Number of Inspe...	Number of Inspe...	0	32	.	.	Inspections	
4	1072	Inspections Pass...	Inspection...	0	32	.	.		
5	1073	Inspections Fail...	Inspection...	0	32	.	.		

## modelRef\_PV\_ATTRIBUTEVALUE

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Attribute	Text		Name	Name
ItemID	Number	Account.ID		ID values not viewable in the user interface
NumericValue	Number			Specific Attribute Value
PeriodID	Number	Period.ID		ID values not viewable in the user interface
ScenarioID	Number	Scenario.ID		ID values not viewable in the user interface
StringValue	Text			Specific Attribute Value

The following is a sample PV\_AttributeValue:

PE_PV_ATTRIBUTEVALUE ▾						
	PERIODID	SCENARIOID	ITEMID	ATTRIBUTE	NUMERICVALUE	STRINGVALUE
1	27	1	1091	Completed Expedit...	2000	
2	27	1	1091	Average Time To...	0.5	
3	27	25	1165	Capacity Varianc...	-0.1568	
4	27	25	1165	Units Available	2	
5	27	25	1167	Capacity Available	3500	
6	27	25	1167	Capacity Provided		
7	27	25	1167	Capacity Used		
8	27					
9						

## modelRef\_PV\_DIMENSION

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Name	Text		Dimension Name	Dimension Name

The following is a sample PV\_Dimension:

PE_PV_DIMENSION ▾		
	ID	NAME
1	1001	Activities
2	1002	Channel
3	1003	General Ledger
4	1004	External Units
5	1005	Organization
6	1006	Products and Se...
7	1007	Region
8	1008	Stages
9	1009	Fixed_Variable
10	1010	Customer Value
11	1011	Importance

## modelRef\_PV\_DRIVER

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Formula	Text		Driver Formula	Formula
ID	Number	Account.DriverID and Assignment.DriverID		ID values not viewable in the user interface
ImplementationType	Text		Driver Type	Driver Type
Name	Text		Driver Name	Driver Name
QuantityType	Text			Driver Checkbox: Shared or Unique
RuleFormula	Text		Rule Formula	Rule Formula
UseRuleFormula	Number		Use Rule Formula	Driver Checkbox: Use Rule Formula

The following is a sample PV\_Driver:

PE_PV_DRIVER ▾														
	ID	NAME	IMPLEMENTATION TYPE	QUANTITY TYPE	FORMULA	USE FIXED QUANTITIES	USE VARIABLE QUANTITIES	USE WEIGHTED QUANTITIES	FIXED OVERRIDE	VARIABLE OVERRIDE	IDLEFLOW METHOD	SEQUENCE NO	USER ENTERED COST ALLOCATION	USER RULE FORMULA
1	1	Evenly Assigned	Ev...	UN...		1	0	0	0	0	Do...	1	0	0
2	2	Percentage	Per...	UN...		1	0	0	0	0	Do...	1	0	0
3	1085	# of Complaints	Ba...	UN...		1	0	0	0	0	Do...	1	0	0
4	1086	FTE	Ba...	UN...		1	0	0	0	0	Do...	1	0	0
5	1087	Allocated Cost	Ba...	SH...		0	0	0	0	0	Us...	1	1	0
6	1088	Consumption D...	Ba...	UN...		1	1	0	0	0	Do...	1	0	0
7	1089	# of Packages	Ba...	UN...		1	1	0	0	0	Do...	1	0	0
8	3	Bill of Cost	Bill...	UN...		1	1	0	0	0	Do...	1	0	0
9	1090	# of Expedite R...	Cal...	UN...	('A...	1	0	0	0	0	Do...	1	0	0
10	4	Sales volume	Sal...	SH...		0	1	0	0	0	Do...	1	0	0

## modelRef\_PV\_ENTEREDCE

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Cost	Number		Cost	Cost
Name	Text		Name	Name
ParentID	Number	Account.ID		ID values not viewable in the user interface
PeriodID	Number	Period.ID		ID values not viewable in the user interface
Reference	Text		Reference	Reference
ScenarioID	Number	Scenario.ID		ID values not viewable in the user interface

The following is a sample PV\_EnteredCE:

PE_PV_ENTEREDCE ▾						
	PERIODID	SCENARIOID	PARENTID	NAME	REFERENCE	COST
1	27	1	1151	Equipment Expe...	BPDEE	52300
2	27	1	1156	Equipment Expe...	OPHEE	33900
3	27	1	1160	Equipment Expe...	EPDEE	21500
4	27	1	1163	Equipment Expe...	EPHEE	
5	27	1	1165	Equipment Expe...	ECSEE	
			1167	Equipment Expe...		
			1168	Equipment Expe...		

## modelRef\_PD\_dimShortRef<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.

For example:

ABC\_PD\_REGIONRC

ABC\_PD\_REGIONSSC

ABC\_PD\_REGIONC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
All_#####	Text			Specific model dimension (where ##### is the dimension name)
ID	Number			ID values not viewable in the user interface
Level1	Text		Dimension Level Name	Dimension Level Name
DisplayOrder	float			

The following is a sample <modRef>\_PD\_<dimRef><suffix>:

PE_PD_CHNL ▾								
	ID	DISPLAYORDER	ALL_CHANNEL	ID_0	DISPLAYORDER_0	LEVEL1	ID_1	DISPLAYORDER_1
1	0	0	ALL	0	0	ALL	0	0
2	1	2	ALL	0	0	NONE	1	1
3	1015	11	ALL	0	0	Customer Pick Up	1015	10
4	1017	21	ALL	0	0	Drop Box	1017	20
5	1037	31	ALL	0	0	Storefront	1037	30

## modelRef\_PD\_COSTELEMENT<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.

For example:

ABC\_PD\_COSTELEMENTRC  
 ABC\_PD\_COSTELEMENTSSC  
 ABC\_PD\_COSTELEMENTC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
CostElementName	Number		Name	Name
ID	Number			ID values not viewable in the user interface

The following is a sample `<modRef>_PD_CostElement`:

PE_PD_COSTELEMENT ▾		
	ID	COSTELEMENTNAME
1	0	All
2	4	EnteredCostElement
3	5	Assignment
4	11	InternalCostElement
5	12	ExternalCostElement

## **`modelRef_PD_DRIVER<suffix>`**

where `<suffix>` is RC, SSC, or C<cubeID> for a multi-stage contributions cube.



For example:

ABC\_PD\_DRIVERRC  
 ABC\_PD\_DRIVERSSC  
 ABC\_PD\_DRIVERC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Name	Text		Driver Name	Driver Name

The following is a sample `<modRef>_PD_Driver`:

PE_PD_DRIVER ▾		
	 ID	 DRIVERNAME
1	-2	None
2	0	All
3	1	Evenly Assigned
4	2	Percentage
5	3	Bill of Cost
6	4	Sales volume
7	1085	# of Complaints
8	1086	FTE
9	1087	Allocated Cost
10	1088	Consumption Driv...
11	1089	# of Packages
12	1090	# of Expedite Req...

## modelRef\_PD\_MODULE<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.



For example:

ABC\_PD\_MODULERC  
 ABC\_PD\_MODULESSC  
 ABC\_PD\_MODULEC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
ModuleName	Text		Module Name	Module

The following is a sample <modRef>\_PD\_Module:

PE_PD_MODULE ▾		
	 ID	 MODULENAME
1	-2	None
2	0	External Unit
3	1	Resource
4	2	Activity
5	3	Cost Object
6	4	Profitability

modelRef\_PD\_PERIOD<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.

For example:

ABC\_PD\_PERIODRC  
ABC\_PD\_PERIODSSC  
ABC\_PD\_PERIODC1001

Note: To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Period_L1 Name	Text			
.....				
Period_Ln Name	Text			

The following is a sample <modRef>\_PD\_Period:

PE_PD_PERIOD ▾																								
	ID	STARTDATE	YEAR	ID	STARTDATE	QUARTER	ID	STARTDATE	MONTH	ID	STARTDATE	PERIOD_L4	ID	STARTDATE	PERIOD_L5	ID	STARTDATE	PERIOD_L6	ID	STARTDATE	PERIOD_L7	ID	STARTDATE	
1	25	01JAN2009:00:0...	E2...	25	01...	E2...	25	01...	E2...	25	01...	E2...	25	01...	E2...	25	01...	E2...	25	01...	E2...	25	01...	
2	20	01JAN2004:00:0...	20...	20	01...	20...	20	01...	20...	20	01...	20...	20	01...	20...	20	01...	20...	20	01...	20...	20	01...	
3	26	01JAN2004:00:0...	SC...	26	01...	SC...	26	01...	SC...	26	01...	SC...	26	01...	SC...	26	01...	SC...	26	01...	SC...	26	01...	
4	1	01JAN2006:00:0...	20...	1	01...	20...	1	01...	20...	1	01...	20...	1	01...	20...	1	01...	20...	1	01...	20...	1	01...	
5	2	01JAN2007:00:0...	20...	2	01...	20...	2	01...	20...	2	01...	20...	2	01...	20...	2	01...	20...	2	01...	20...	2	01...	
6	3	01JAN2008:00:0...	20...	3	01...	20...	3	01...	20...	3	01...	20...	3	01...	20...	3	01...	20...	3	01...	20...	3	01...	

modelRef\_PD\_SCENARIO<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.



For example:

ABC\_PD\_SCENARIORC  
 ABC\_PD\_SCENARIOSSC  
 ABC\_PD\_SCENARIOC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
Scenario_L1 Name	Text			
.....				
Scenario_Ln Name	Text			

The following is a sample <modRef>\_PD\_Scenario:

PE_PD_SCENARIO ▾																		
	ID	TOP_SCENARIO	SCENARIO 1		PERSONAL SCENARIO	SCENARIO 2		SCENARIO 3	SCENARIO 4		SCENARIO 5	SCENARIO 6		SCENARIO 7	SCENARIO 8		SCENARIO 9	SCENARIO 10
			ID 1	SCENARIO 1 L2		ID 2	SCENARIO 2 L4		ID 3	SCENARIO 3 L5		ID 4	SCENARIO 4 L6		ID 5	SCENARIO 5 L7		
1	23	ETE2_Actual	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23 ET...	23
2	25	Plan	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25 Pla...	25
3	20	Planning	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20 Pla...	20
4	24	SCE_04	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24 SC...	24
5	1	Actual	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1 Act...	1
6	2	Budget	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2 Bu...	2
7	21	Planning	20 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21 Wh...	21
8	22	Planning	20 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22 Wh...	22

## modelRef\_PD\_YESNO<suffix>

where <suffix> is RC, SSC, or C<cubeID> for a multi-stage contributions cube.

For example:

ABC\_PD\_YESNORC  
 ABC\_PD\_YESNOSSC  
 ABC\_PD\_YESNOC1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
ID	Number			ID values not viewable in the user interface
YesNoName	Text			Yes or No Value

The following is a sample `<modRef>_PD_YesNo`:

PE_PD_YESNO ▾			
	ID		YESNONAME
1	0	No	
2	1	Yes	

## ***modelRef\_PF\_<suffix>***

where **<suffix>** is RC, SSC, or MSC\_<cubeID>.

For example:

ABC\_PF\_RC  
 ABC\_PF\_SSC  
 ABC\_PF\_MSC\_C1001

*Note:* To display the cubeID for a cube, select **Model** ⇒ **Manage Cubes and Permissions**, select a cube, and then click **Internal names**.

This view is created automatically when a fact table is generated.

**Table 15.1** Custom MSC Cube

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
PeriodID	Number			
ScenarioID	Number			
S1Dim<dimRef>	Text			One row for each dimension in the cube
Cost	Number		Cost	
Output	Number			
Revenue	Number			
SoldQuantity	Number			
UsedTotalDriverQuantity	Number			
UserOutputQuantity	Number			

Field Name	Field Type	Join Field	Properties Reference	Model View Columns
Attribute<dimID>	Number			One row for each dimensional attribute in the cube
TotalDriverQuantity	Number			



## **Part 5**

---

# Backup and Restore

### *Chapter 16*

### **Backup and Restore Utility** ..... 179



## Chapter 16

# Backup and Restore Utility

---

<b>Overview</b> .....	<b>179</b>
<b>Backup</b> .....	<b>180</b>
Syntax .....	180
Usage Notes .....	180
<b>Restore</b> .....	<b>181</b>
Syntax .....	181
Usage Notes .....	182

---

## Overview

Because models are stored in the form of memory-mapped files rather than in a database, you cannot back up your models by backing up a database. SAS Cost and Profitability Management provides a command-line batch utility, **AbmMdlMigration.exe**, to back up and restore all your models.

AbmMdlMigration is installed in the following directory:

```
<SASHome>\SASFoundation\9.4\abmiomsvr\sasexe
```

For example:

```
C:\Program Files\SASHome\SASFoundation\9.4\abmiomsvr\sasexe
```

*Note:* In future releases of SAS Cost and Profitability Management, AbmMdlMigration will also be used to migrate models from one version of the product to another.

To obtain command line help for AbmMdlMigration, enter the following batch command:

```
AbmMdlMigration -h
```

or simply:

```
AbmMdlMigration
```

## Backup

### Syntax

```
AbmMdlMigration -backup
-src="source_folder"
-dst="destination_folder"
-appcfg="model_config_folder"
```

*Note:* Although the syntax and the following example are shown on multiple lines, the command must be entered on a single line.

For example:

```
AbmMdlMigration -backup
-src="C:\SAS\Config\Lev1\SASApp\ABMServer\Models"
-dst="C:\temp"
-appcfg="C:\SAS\Config\Lev1\SASApp\ABMServer\config"
```

-src

The complete path of the folder with the model data to be backed up

-dst

The complete path of the folder where model data is to be backed up

-appcfg

The complete path of the folder that contains settings.xml and ABMLogConfig.xml files. The settings.xml file contains information about models. The ABMLogConfig.xml file contains the log-related configuration information.

### Usage Notes

- The administrator who backs up the models must have read and write permission to the source, destination, and configuration directories.
- Before running a backup or restore, you should stop the Object.Spawner. To do so, select **Start** ⇒ **Administrative Tools** ⇒ **Server Manager**. Then, from the Server Manager window:
  1. Select **Services**.
  2. Right-click **SAS [Config-Lev1] Object Spawner**.
  3. Select **Stop**.

After running the backup or restore, start the Object.Spawner.

- Run the following commands from the command prompt to start and stop the Object Spawner and to check the status of the Object Spawner in a Linux environment:

```
./ObjectSpawner.sh stop    Stops the Object Spawner service
./ObjectSpawner.sh start    Starts the Object Spawner service
./ObjectSpawner.sh status   Checks the status of the Object Spawner service
```

*Note:* Before running these commands, navigate to the directory where the Object Spawner is deployed. For example, navigate to the **SASConfig/Lev1/**



**ObjectSpawner** directory where the Object Spawner is deployed to run the related commands.

- Error and warning messages are written to the log file. Failure messages are written to the terminal. To preserve the messages, you can spool terminal output to a file. For example:

```
AbmMdlMigration -backup
-src="source_folder"
-dst="destination_folder"
-appcfg="model_config_folder"
> message.file
```

- For each model that is backed up, AbmMdlMigration creates, in the destination directory, a binary file whose name is M<model\_ID>.bin.

*Note:* You can display the ID of a model by right-clicking the model in Workspace Manager and selecting **Item Properties**. The model ID is displayed along with other information.

- Refer to the log file for information about which models were backed up and why. For example, you can not back up model 'M1002' because the file is empty.
- To back up models when SAS Cost and Profitability Management is deployed on a Linux environment, provide the user with read, write, and execute permission to the source, destination, and log directories by entering the following command from the command prompt:

```
sudo chmod -Rf 777 <Path of the directory>
```

For example, suppose you want to provide the cfigsas1 user with read, write, and execute permission to the log directory. In this case, enter the following command from the command prompt:

```
sudo chmod -Rf 777 /install/users/cfigsas1/SASConfig/Lev1/SASApp/ABMServer/Logs
```

- When you back up models, ensure that you also back up the database and verify that the database is in synch with the models.

---

## Restore

### Syntax

```
AbmMdlMigration -restore
-src="source_folder"
-dst="destination_folder"
-appcfg="model_config_folder"
```

*Note:* Although the syntax and the following example are shown on multiple lines, the command must be entered on a single line.

For example:

```
AbmMdlMigration -restore
-src="C:\temp"
-dst="C:\SAS\Config\Lev1\SASApp\ABMServer\Models"
-appcfg="C:\SAS\Config\Lev1\SASApp\ABMServer\config"
```

- src  
The complete path of the folder with the model data to be restored
- dst  
The complete path of the folder to where model data is to be restored
- appcfg  
The complete path of the folder that contains settings.xml and ABMLogConfig.xml files. The settings.xml file contains information about models. The ABMLogConfig.xml file contains the log-related configuration information.

## Usage Notes

- The administrator who restores the models must have read and write permission to the source, destination, and configuration directories.
- Before running a backup or restore, you should stop the Object.Spawner. To do so, select **Start** ⇒ **Administrative Tools** ⇒ **Server Manager**. Then, from the Server Manager window:
  1. Select **Services**.
  2. Right-click **SAS [Config-Lev1] Object Spawner**.
  3. Select **Stop**.

After running the backup or restore, start the Object.Spawner.

- Run the following commands from the command prompt to start and stop the Object Spawner and to check the status of the Object Spawner in a Linux environment:

```
./ObjectSpawner.sh stop    Stops the Object Spawner service
./ObjectSpawner.sh start    Starts the Object Spawner service
./ObjectSpawner.sh status  Checks the status of the Object Spawner service
```

*Note:* Before running these commands, navigate to the directory where the Object Spawner is deployed. For example, navigate to the **SASConfig/Lev1/ObjectSpawner** directory where the Object Spawner is deployed to run related commands.

- When you run AbmMdlMigration to restore models, if a model to be restored already exists, then before restoring the model AbmMdlMigration makes a copy of the already-existing model in a folder named M<model\_ID>.ORIG. If anything goes wrong with the restoration, then you can use the model file in M<model\_ID>.ORIG to revert to a known state of the model.

*Note:* You can display the ID of a model by right-clicking the model in Workspace Manager and selecting **Item Properties**. The model ID is displayed along with other information.

- Error and warning messages are written to the log file. Failure messages are written to the terminal. To preserve the messages, you can spool terminal output to a file. For example:

```
AbmMdlMigration -backup
-src="source_folder"
-dst="destination_folder"
-appcfg="model_config_folder"
> message.file
```

- Refer to the log file for information about which models were restored and why. For example, you can not restore the model 'M1002' because the file is empty.
- To restore models when SAS Cost and Profitability Management is deployed on a Linux environment, provide the user with read, write, and execute permission to the source, destination, and log directories by entering the following command from the command prompt:

```
sudo chmod -Rf 777 <Path of the directory>
```

For example, suppose you want to provide the cfigsas1 user with read, write, and execute permission to the log directory. In this case, enter the following command from the command prompt:

```
sudo chmod -Rf 777 /install/users/cfigsas1/SASConfig/Levl/SASApp/ABMServer/Logs
```

- When you restore the models, ensure that the database has already been restored and that it is in synch with the models.



## Part 6

---

# Using the API

<i>Chapter 17</i>	
<b>Programming the API</b> .....	187
<i>Chapter 18</i>	
<b>API Functions</b> .....	191
<i>Chapter 19</i>	
<b>XML Passed to the API</b> .....	201
<i>Chapter 20</i>	
<b>Easy API</b> .....	245



## Chapter 17

# Programming the API

---

<b>Overview</b> .....	<b>187</b>
<b>Progress reports</b> .....	<b>188</b>
<b>Log files</b> .....	<b>189</b>
<b>Writing a program in Java</b> .....	<b>190</b>

---

## Overview

The SAS Cost and Profitability Management Web Services Integration API enables other applications or 3rd party systems to seamlessly integrate SAS Cost and Profitability Management into their operational systems. Users or consultants can write code or script to automate their business reporting processes, which may include periodically obtaining data from various financial data sources, cleansing and importing data into the SAS Cost and Profitability Management system, computing the activity costs, or building OLAP cubes for contribution or profitability analysis.

SAS ETL and Analytical tools combined with the SAS Cost and Profitability Management Integration API provide numerous options for users to integrate, automate, and analyze data that resides on any system.

SAS Cost and Profitability Management Web Services Integration API on client is surfaced through a Microsoft COM/Automation interface. This API component on the client uses SOAP Protocol to communicate with the .NET webservices on server and hides all the complexities of sending operation requests and receiving result or logs at the completion of execution. Users who can run their integration components or code on Microsoft Windows Operating systems can effectively use this API.

*Note:* The API must be invoked from a machine on which the SAS Cost and Profitability Management client is installed.

### Prerequisites for the API

- SAS Cost and Profitability Management client and all the dependent components must be installed prior to using client integration API.
- SAS Cost and Profitability Management client should be in working condition (does not have to be currently running) to use this API functionality.

### Languages supported

You can invoke the API using the following languages:

- C#
- Java
- Visual C++

You can write a program that uses certain functions (such as importing data and exporting data) within SAS Cost and Profitability Management. You can use any COM-compliant language (such as C#, Visual Basic, or Visual C++) to access these functions, which are included in the SAS Cost and Profitability Management Web Services Integration API.

The SAS Cost and Profitability Management Web Services Integration API supports a COM dual interface: `vtable` (for early binding) and `IDispatch` (for late binding).

SAS Cost and Profitability Management must be installed on the computer on which you run your program.

*Note:* You can perform this task without first opening a model.

1. Review the information about importing model data or exporting model data.
2. If you are importing from a database:
  - a. Create a database or a database view that matches the data schema and then load the data to be imported.
  - b. Verify that the database user has the correct access.
3. If you are exporting to a database:
  - a. Create an empty database that matches the data schema.
  - b. Verify that the database user has the correct access.
4. Verify that the SAS Cost and Profitability Management client application is installed on the computer on which your program will run.
5. Verify that the XML import configuration, the XML export configuration, or the XML calculate configuration matches the appropriate data schema.
6. If you are using XML files, ensure that your program has access to the data:
  - a. For the XML import configuration, provide the user ID and password for the `DataSource` attribute of either the `StagingArea` element or the `StagingTable` element.
  - b. For the XML export configuration, provide the user ID and password for the `Connection` attribute of the `StagingArea` element.
7. Run your program.
8. Verify the results.

See [Chapter 18, “API Functions,” on page 191](#).

To receive a report of events that occur during the operation, your program can call the `GetOperationProgress` function.

---

## Progress reports

The functions that you can perform using the SAS Cost and Profitability Management Web Services Integration API run on the SAS Cost and Profitability Management server



and might require a considerable amount of time to complete. These functions regularly report progress to SAS Cost and Profitability Management. You can retrieve this progress report in the form of an XML string by invoking the `GetOperationProgress` function. Your program can read this string into the XML parser of your choice. Alternatively, your program can display the progress report information.

Here is the structure of the XML string:

```
<SasServicesStatus>
  <Operation Type="OPTYPE string" Status="OPSTATUS string" />
  <OperationMessages>
    <Message Text="Message text" />
    <Message Text="Message text" />
  </OperationMessages>
</SasServicesStatus>
```

The following table shows the possible values of the OPTYPE string:

This API function...	...has an OPTYPE string value of...
Calculate	Calculate
CopyModelData	Copy Model Data
ExportModelData	Export Model Data
ImportModelData	Import Model Data
Import CubeConfigurations	Import Cube Configurations
Export CubeConfigurations	Export Cube Configurations
Copy Paste Model	Copy Paste Model
Publish/Unpublish Periods/Scenarios	Publish Period Scenario
Export and Register Tables	Create Public Tables

The OPSTATUS string can have any of the following values:

- Succeeded
- Failed
- Not Complete

Messages typically contain progress information, such as Exporting accounts.

---

## Log files

The SAS Cost and Profitability Management Web Services Integration API creates two log files that contain information about the functions that were performed: `sasservices.log` and `sasoperations.log`. These log files are stored in the folder that contains your program's executable.

*Note:* These files can become large because new content is appended to the files. (The contents of these files are never overwritten.) Therefore, you should archive or delete these files periodically.

---

## Writing a program in Java

To use the SAS Cost and Profitability Management Web Services Integration API from Java, a Java Virtual Machine (JVM) must be installed on the computer on which the Java class is being executed. Also, a bridge is necessary for Java and COM to exchange data. SAS recommends Jawin.

Here is an example of the Java source code that you might use to access the `ExportModelData` function. Notice that Jawin converts an unsuccessful COM call to a Java exception.

```
try
{
    Ole32.CoInitialize();    // Initialize COM
    Application api = new _Application("new:SasServices.API");
    api.Connect( "MyUserName", "MyPassword" );
    String xmlConfig;
    String operationKey;
    // ... Build the configuration xml
    api.ExportModelData( xmlConfig, operationKey );
    // ... At some later time
    String xmlExportReport;
    api.GetOperationProgress( operationKey, xmlExportReport );
    api.Disconnect();
    Ole32.CoUninitialize(); // Shutdown COM
}

catch(COMException e)
{
    System.out.println("Got a COM Error: " + e);
}

catch(Exception e)
{
    e.printStackTrace();    // Handle an exception.
}
```

## Chapter 18

# API Functions

---

<b>AcquireModelLock</b> .....	<b>191</b>
<b>Calculate</b> .....	<b>192</b>
<b>CancelOperation</b> .....	<b>192</b>
<b>Connect</b> .....	<b>193</b>
<b>CopyModelData</b> .....	<b>194</b>
<b>CopyPasteModel</b> .....	<b>194</b>
<b>Disconnect</b> .....	<b>194</b>
<b>ExportCubeConfigurations</b> .....	<b>195</b>
<b>ExportModelData</b> .....	<b>195</b>
<b>ExportRegisterTables</b> .....	<b>196</b>
<b>GenerateCube</b> .....	<b>196</b>
<b>GetOperationLog</b> .....	<b>196</b>
<b>GetOperationProgress</b> .....	<b>197</b>
<b>ImportCubeConfigurations</b> .....	<b>197</b>
<b>ImportModelData</b> .....	<b>197</b>
<b>OnOperationComplete</b> .....	<b>198</b>
<b>PublishPeriodsScenarios</b> .....	<b>199</b>
<b>ReleaseModelLock</b> .....	<b>199</b>
<b>TerminateWriteLock</b> .....	<b>199</b>

---

## AcquireModelLock

This function locks the model in read or write mode.

Type	Argument	Data Type	Description
IN	ModelRef	BSTR	Reference of the model

---

Type	Argument	Data Type	Description
IN	ModelRef	BSTR	The type of lock to be acquired on the model: either read or write

---

## Calculate

This function calculates a model in the SAS Cost and Profitability Management database.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to calculate a model  See <a href="#">“XML to Calculate a Model”</a> on page 203.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## CancelOperation

This function cancels the import, export, or calculation operation in progress.

Type	Argument	Data Type	Description
IN	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## Connect

This function initializes a session with the SAS Cost and Profitability Management server.

Type	Argument	Data Type	Description
IN	ServerName	BSTR	SAS Cost and Profitability Management server name
IN	UserName	BSTR	SAS Cost and Profitability Management user with sufficient permissions to perform the operations in the program. A domain name may be required, such as when connecting to a server.
IN	Password	BSTR	Plain text password for the SAS Cost and Profitability Management user
IN	IWA	BSTR	Specifies whether to use Integrated Windows Authentication (IWA)
IN	metaport	BSTR	Port number of the metadata server
IN	metaServerName	BSTR	Machine name of the metadata server

---

## CopyModelData

This function copies model data from one period/scenario association to another.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to copy model data  See <a href="#">“XML to Copy Period/Scenario Data”</a> on page 209.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## CopyPasteModel

This function makes a copy, with a new name, of the specified model.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to copy and paste a model.  See <a href="#">“XML to Copy and Paste a Model”</a> on page 206.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## Disconnect

This function ends the session with the SAS Cost and Profitability Management server. This function has no arguments.

---

## ExportCubeConfigurations

This function exports cube configurations to an xml file.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to export cube configurations  See <a href="#">“XML to Export Cube Configurations”</a> on page 210.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## ExportModelData

This function exports model structures and data to a Microsoft SQL Server, Oracle, or Microsoft Access database.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to export a model  See <a href="#">“XML to Export a Model”</a> on page 211.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## ExportRegisterTables

This function exports and registers tables for use with other SAS products or with your own programs. See [Chapter 14, “Working with Registered Tables,”](#) on page 145.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to export and register tables.  See <a href="#">“XML to Export and Register Tables”</a> on page 221.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## GenerateCube

This function generates a cube and/or a fact table.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to generate a cube  See <a href="#">“XML to Generate a Cube”</a> on page 224.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## GetOperationLog

This function retrieves the operation log as XML.

Type	Argument	Data Type	Description
IN	OperationKey	BSTR	GUID string that uniquely identifies the operation



Type	Argument	Data Type	Description
OUT	Log	BSTR	The log in XML format

---

## GetOperationProgress

This function retrieves progress messages regularly reported by SAS Cost and Profitability Management operations in the form of an XML string progress report.

Type	Argument	Data Type	Description
IN	OperationKey	BSTR	GUID string that uniquely identifies this operation
OUT	xmlReport	BSTR	The reported progress of the operation

---

## ImportCubeConfigurations

This function imports cube configurations from a previously exported xml file.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to import cube configurations  See <a href="#">“XML to Import Cube Configurations”</a> on page 227.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## ImportModelData

This function imports model structures and data into a SAS Cost and Profitability Management model from a Microsoft SQL Server, Oracle, or Microsoft Access

database, or from an XML file that was previously exported from SAS Cost and Profitability Management.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to import a model  See <a href="#">“XML to Import a Model”</a> on page 229.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---

## OnOperationComplete

This function retrieves the status code at the end of an operation.

Type	Argument	Data Type	Description
IN	OperationKey	BSTR	GUID string that uniquely identifies the operation
OUT	Status	Integer	The status code for the operation. The status code may be one of the following:  0=OPSTATUS_FAILED  1=OPSTATUS_SUCCEEDED  2=OPSTATUS_UNDETERMINED

---

## PublishPeriodsScenarios

This function publishes periods and their associated scenarios.

Type	Argument	Data Type	Description
IN	XML	BSTR	XML to publish periods and scenarios.  See “XML to Publish Period/Scenario Associations” on page 242.
OUT	OperationKey	BSTR	GUID string that uniquely identifies the operation

---



---

## ReleaseModelLock

This function releases the lock on the read or write mode of the model.

Type	Argument	Data Type	Description
IN	ModelRef	BSTR	Reference of the model

---



---

## TerminateWriteLock

This function terminates a lock on the write mode of the model.

Type	Argument	Data Type	Description
IN	ModelRef	BSTR	Reference of the model

---



## Chapter 19

# XML Passed to the API

---

<b>Overview</b>	<b>203</b>
<b>XML to Calculate a Model</b>	<b>203</b>
Sample XML	203
Overview	203
XML Syntax	203
OROSCOMMAND Element (Required)	204
MODELCONTEXT Element (Required)	204
PeriodScenario Element (Required)	204
COMMANDPARAMS Element (Required)	205
CostFlow Element (Required)	205
<b>XML to Copy and Paste a Model</b>	<b>206</b>
Sample XML	206
Overview	207
XML Syntax	207
CopyPasteModel element (required)	207
MODELCONTEXT element (required)	208
COMMANDPARAMS element (required)	208
<b>XML to Copy Period/Scenario Data</b>	<b>209</b>
Sample XML	209
Overview	209
XML copy model data configuration structure	209
CopyModelData element (required)	209
ModelContext element (required)	209
<b>XML to Export Cube Configurations</b>	<b>210</b>
Sample XML	210
Overview	210
XML syntax	210
CubeConfigurationsExport element (required)	210
CubeConfiguration element (required)	211
<b>XML to Export a Model</b>	<b>211</b>
Sample XML	211
Overview	212
XML Syntax	213
OEExport Element (Required)	213
PeriodScenario Element (Required)	214
StagingArea Element (Required)	214
StagingTable Element (Required)	219
Column Element (Required)	219

ColumnFilter Element (Optional) . . . . .	219
![CDATA [expression]] Element (Required) . . . . .	220
DimensionColumn Element (Optional) . . . . .	220
DimensionFilter Element (Optional) . . . . .	220
Dimension Element (Required) . . . . .	220
DimensionMember Element (Optional) . . . . .	221
NestedObject Element (Optional) . . . . .	221
<b>XML to Export and Register Tables . . . . .</b>	<b>221</b>
Sample XML . . . . .	221
Overview . . . . .	222
XML Syntax . . . . .	222
PublishTablesInfo element (required) . . . . .	222
Publish element (optional) . . . . .	223
MODELCONTEXT element (required) . . . . .	223
Tables element (required) . . . . .	223
LasrInfo element (optional) . . . . .	223
<b>XML to Generate a Cube . . . . .</b>	<b>224</b>
Overview . . . . .	224
Sample XML . . . . .	224
XML Syntax . . . . .	224
OROSCOMMAND Element (Required) . . . . .	225
MODELCONTEXT Element (Required) . . . . .	225
PeriodScenario Element (Required) . . . . .	225
COMMANDPARAMS Element (Required) . . . . .	226
CubeConfig Element (Required) . . . . .	226
Configuring How Cubes Are Generated . . . . .	227
Errors in the Cube Configuration File . . . . .	227
<b>XML to Import Cube Configurations . . . . .</b>	<b>227</b>
Sample XML . . . . .	227
Overview . . . . .	228
XML syntax . . . . .	228
CubeConfigurations element (required) . . . . .	228
<b>XML to Import a Model . . . . .</b>	<b>229</b>
Sample XML . . . . .	229
Overview . . . . .	232
XML Syntax . . . . .	233
OEImport Element (Required) . . . . .	233
StagingArea Element (Required) . . . . .	234
StagingTable Element (Required) . . . . .	240
Column Element (Optional) . . . . .	241
<b>XML to Publish Period/Scenario Associations . . . . .</b>	<b>242</b>
Sample XML . . . . .	242
Overview . . . . .	242
XML syntax . . . . .	242
PublishPeriodScenarios element (required) . . . . .	242
ModelContext element (required) . . . . .	242
PeriodScenario element (required) . . . . .	243

---

## Overview

When you use certain functions in the SAS Cost and Profitability Management Web Services Integration API, you must provide an XML configuration string. This string may be included in your program, contained in a file, or entered through your program's user interface. Each XML configuration has a specific purpose and a unique structure.

*Note:* Some of the attributes in an XML configuration contain IDs to other data that resides on a specific SAS Cost and Profitability Management server, such as periods and scenarios. Therefore, do not use an XML configuration that was created for another server. If you use an XML configuration created for one server, these IDs might not correspond to any data on another server. Or, in the worst case, the IDs might correspond to the wrong data and the model might become corrupted.

---

## XML to Calculate a Model

### Sample XML

```
<OROSCOMMAND Version="2.0">
  <MODELCONTEXT Model="peAPI">
    <PeriodScenario PeriodId="9" ScenarioId="1"/>
  </MODELCONTEXT>
  <COMMANDPARAMS MessageLimit="50">
    <CostFlow SeqDriverLimit="-1" Force="1" DisableDriverRules="0"/>
  </COMMANDPARAMS>
</OROSCOMMAND>

<OROSCOMMAND Version="2.0">
  <MODELCONTEXT Model="MyFolder\peAPI">
    <PeriodScenario PeriodRef="2008 Q1" ScenarioRef="Actual" />
  </MODELCONTEXT>
  <COMMANDPARAMS MessageLimit="50">
    <CostFlow SeqDriverLimit="-1" Force="1" DisableDriverRules="0"/>
  </COMMANDPARAMS>
</OROSCOMMAND>
```

### Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to calculate a model, you must write XML that specifies what you want to calculate and how you want to calculate it.

Elements and attributes are case sensitive. String comparisons are not case sensitive.

### XML Syntax

The following structure shows the elements used in the XML and their relationships to each other.

```

<OROSCOMMAND Version="2.0">
  <MODELCONTEXT attributes>
    <PeriodScenario attributes/>
  </MODELCONTEXT>
  <COMMANDPARAMS attributes>
    <CostFlow attributes/>
  </COMMANDPARAMS>
</OROSCOMMAND>

```

### ***OROSCOMMAND Element (Required)***

OROSCOMMAND is the root element. This element has one version attribute. The OROSCOMMAND element must contain one MODELCONTEXT element and one COMMANDPARAMS element.

Attribute	Required	Values	Description
Version	Optional	"2.0"	If you omit the version number, then the command reverts to version 1 syntax which is supported but deprecated. Version 1 syntax is significantly different.

### ***MODELCONTEXT Element (Required)***

This element specifies the model to calculate. A MODELCONTEXT element can contain multiple PeriodScenario elements.

Attribute	Required	Values	Description
Model	Required	String	Full Workspace path of the model, excluding the Models Workspace folder.  <b>Examples:</b> "MyFolder\model1" "MyFolder\MySubFolder\model2"

### ***PeriodScenario Element (Required)***

Each PeriodScenario element specifies a period/scenario association to calculate. You can have multiple PeriodScenario elements.



Attribute	Required	Values	Description
PeriodId   PeriodRef	Required	Numeric if ID, String if Reference	Period
ScenarioId   ScenarioRef	Required	Numeric if ID, String if Reference	Scenario

### **COMMANDPARAMS Element (Required)**

This element configures the calculate operation. A COMMANDPARAMS element can have one CostFlow element.

Attribute	Required	Values	Description
MessageLimit	Optional	String that represents an integer between and including 0 and 2,147,483,648	Sets the maximum number of messages to include for each error/warning message type. Defaults to 50. If value is less than zero, all messages are included in the output. A message type is equivalent to a specific type of error or warning. The error/warning may apply to more than one item, so a message is normally generated for each item.

### **CostFlow Element (Required)**

This element specifies information about the calculation.

Attribute	Required	Values	Description
Diagnostic	Optional	"yes", "no"	Indicates whether detailed information about the calculation is returned. This information includes data such as the module that is being calculated and the amount of time that is required to calculate.

Attribute	Required	Values	Description
Force	Optional	"1" or "0"	<p>"1" means to perform calculations even if a system flag says that calculations are up to date.</p> <p>"0" means to perform calculations only if a system flag says that calculations are out of date.</p> <p>Omitting the Force attribute is the same as Force="0".</p>
DisableDriverRules	Optional	"1", "0"	<p>"1" means to ignore rule-based drivers. Do not generate assignments for any rule-based drivers. Specifying "1" for this option causes assignment generation not to take place, which means that assignments created by a previous calculation are retained.</p> <p>"0" is the default. It means that rule-based drivers are enabled and generate assignments.</p>
SeqDriverLimit	Optional	<p>"-1"</p> <p>"1" - "99"</p>	<p>Sequence number at which to stop the calculation. "-1" indicates that driver sequencing is not used</p>

## XML to Copy and Paste a Model

### Sample XML

The following example copies a model whose reference is OLDREF and pastes it into the root folder of the Models workspace with the name NEWNAME and reference NEWREF.

```
<CopyPasteModel Version="1.0">
```

```

<MODELCONTEXT
  Reference="NEWREF"
  Name="NEWNAME"
  ParentId="" <!-- root of Models workspace -->
  SourceReference="OLDREF"
  <COMMANDPARAMS IncFactData="0" IncCalcData="0"/>
</CopyPasteModel>

```

The following example copies a model whose ID is 1234 and pastes it into the folder named MyFolder\MySubFolder of the Models workspace with the name NEWNAME and reference NEWREF.

```

<CopyPasteModel Version="1.0">
  <MODELCONTEXT
    Reference="NEWREF"
    Name="NEWNAME"
    ParentId="MyFolder\MySubFolder"
    SourceId="1234"/>
  <COMMANDPARAMS IncFactData="1" IncCalcData="1"/>
</CopyPasteModel>

```

## Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to copy and paste a model, you must write XML that identifies the model to be copied and pasted.

String comparisons are not case sensitive.

## XML Syntax

The following structure shows the elements used in the export XML and their relationships to each other.

```

<CopyPasteModel Version="1.0">
  <MODELCONTEXT
    Reference="modelRef_new"
    Name="modelName_new"
    ParentId="folderPath_new"
    SourceReference="modelRef_old"
    SourceId="modelID_old"/>
  <COMMANDPARAMS IncFactData="0|1" IncCalcData="0|1"/>
</CopyPasteModel>

```

## CopyPasteModel element (required)

CopyPasteModel is the root element.

Attribute	Required	Values	Description
Version	Required	"1.0"	Version number of the function

**MODELCONTEXT element (required)**

This element identifies the model that is to be copied and the new model that is to be pasted, and where it is to be pasted.

Attribute	Required	Values	Description
Reference	Required	String	Reference of new model.
Name	Required	String	Name of the new model model.
ParentId	Required	String	Path of the Model Workspace folder in which the model is to be pasted.  <b>Examples:</b> "" (root of Models workspace) "MyFolder" "MyFolder\MySubFolder"
SourceReference	Required if SourceId is omitted	String	The reference of the model to be copied. This attribute is required if SourceId is omitted.
SourceId	Required if SourceReference is omitted	Integer	ID of the model to be copied. This attribute is required if SourceReference is omitted.  <i>Note:</i> To display the ID of a model, right-click the model in Workspace Manager and select <b>Item Properties</b> .

**COMMANDPARAMS element (required)**

This element specifies whether calculated data and fact table data is to be included with the pasted model.

Attribute	Required	Values	Description
IncFactData	Required	1 0	0=Do not include fact table data in the new model. 1=Include fact table data in the new model.
IncCalcData	Required	1 0	0=Do not include calculated data in the new model. 1=Include calculated data in the new model.

## XML to Copy Period/Scenario Data

### Sample XML

The following example copies model data from one period and scenario (IDs 1 and 3) to another period and scenario (IDs 6 and 2):

```
<CopyModelData>
  <ModelContext ModelId="1001" SrcPeriod="1" DestPeriod="6"
    SrcScenario="3" DestScenario="2" />
</CopyModelData>
```

### Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to calculate a model, you must write an XML CopyModelData configuration that specifies what you want to calculate (and how you want to calculate it).

String comparisons are not case sensitive.

### XML copy model data configuration structure

The following structure shows the elements used in the XML CopyModelData configuration and their relationships to each other.

```
<CopyModelData>
  <ModelContext attribute />
</CopyModelData>
```

*Note:* Although an element might be optional, if you include the element, there might be required elements within it.

### CopyModelData element (required)

CopyModelData is the root element. This element has no attributes. It must contain one ModelContext element.

### ModelContext element (required)

This element specifies the model and the model's period/scenario association to copy.

Attribute	Required	Values	Description
ModelId	Required	String	ID of the model
SrcPeriod	Required	String	ID of the period to copy from
SrcScenario	Required	String	ID of the scenario to copy from
DestPeriod	Required	String	ID of the period to copy to

Attribute	Required	Values	Description
DestScenario	Required	String	ID of the scenario to copy to

## XML to Export Cube Configurations

### Sample XML

```
<CubeConfigurationsExport Filename="c:\temp\peCubeConfigurationsExport.xml" DoFolders="True">
  <CubeConfiguration Name="Fred\PE_MSC63_Modules" />
  <CubeConfiguration Name="Fred\PE_MSC63_Stages" />
  <CubeConfiguration Name="Fred\PE_MSC64_Modules" />
  <CubeConfiguration Name="Fred\PE_MSC64_Stages" />
  <CubeConfiguration Name="Fred\PE_RC" />
  <CubeConfiguration Name="Fred\PE_SS" />
</CubeConfigurationsExport>
```

### Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to export cube configurations, you must write XML that specifies the cube configurations to be exported.

Elements and attributes are case sensitive. String comparisons are not case sensitive.

### XML syntax

The following structure shows the elements used in the XML and their relationships to each other.

```
<CubeConfigurationsExport attributes >
  <CubeConfiguration attribute />
</CubeConfigurationsExport>
```

### CubeConfigurationsExport element (required)

CubeConfigurationsExport is the root element. This element specifies the name of the file to be created during the export. The CubeConfigurationsExport element can contain multiple CubeConfiguration elements.

Attribute	Required	Values	Description
Filename	Required	String	Specifies the complete path, including the file name, of the file to be exported to. If the file already exists, the file is replaced.

Attribute	Required	Values	Description
DoFolders	Optional	"True", "False"	<p>"True" stores folder information in the exported file. When you reimport the file the folders are recreated.</p> <p>"False" causes no folder information to be stored in the exported file. On reimporting, all cube configurations are stored in the topmost Cube Configurations folder of the Workspace Manager. You should use this option only if no two cube configurations have the same name.</p>

### ***CubeConfiguration element (required)***

Each CubeConfiguration element specifies the name and path of a cube configuration to be exported.

Attribute	Required	Values	Description
Name	Required	String	Specifies the name and complete path (folders) of the cube configuration to be exported.

## **XML to Export a Model**

### ***Sample XML***

The following example exports model data from a model named Inflight to a database:

```
<OEExport Model="Inflight" Version="7.2" Type="Staging" MessageLimit="50" Filename="">
  <PeriodScenario Period="2000" Scenario="Actual" />
  <StagingArea
    JdbcDriverClass=""
    RepositoryName=""
    RepositoryId=""
    LibraryName=""
    LibraryID=""
```

```

LibraryReference=""
MetaWorkspaceServer=""
MetaWorkspaceServerId=""
WorkspaceServerName=""
Port=""
Engine=""
DBType="1"
DriverType="0"
HostName="D7920"
PortNumber="1433"
ServiceName="TestDBForStagingTables"
UserName="DBUser"
Password="xxxx"
AdvanceOptions="Security=SSPI;Persist Security Info=False;">
<StagingTable Name="Period" TableName="Period">
  <Column Name="ModelId" ColumnName="ModelId" />
  <Column Name="PeriodId" ColumnName="PeriodId" />
  <Column Name="Name" ColumnName="Name" />
  <Column Name="Reference" ColumnName="Reference" />
  <Column Name="ParentId" ColumnName="ParentId" />
  <Column Name="ParentReference" ColumnName="ParentReference" />
  <Column Name="StartDate" ColumnName="StartDate" />
  <Column Name="EndDate" ColumnName="EndDate" />
  <Column Name="Description" ColumnName="Description" />
</StagingTable>
<StagingTable Name="Scenario" TableName="Scenario">
  <Column Name="ModelId" ColumnName="ModelId" />
  <Column Name="ScenarioId" ColumnName="ScenarioId" />
  <Column Name="Name" ColumnName="Name" />
  <Column Name="Reference" ColumnName="Reference" />
  <Column Name="ParentId" ColumnName="ParentId" />
  <Column Name="ParentReference" ColumnName="ParentReference" />
  <Column Name="Description" ColumnName="Description" />
</StagingTable>
</StagingArea>
</OEExport>

```

The following example exports model data from a model named "Inflight" to XML instead (by replacing the previous first three lines):

```

<OEExport
  Model="Inflight"
  Version="6.2"
  Type="XML"
  MessageLimit="50"
  Filename="c:\temp\inflight.xml">
  <PeriodScenario Period="2000" Scenario="Actual" />
  <StagingArea DataSource="">

```

## Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to export information from SAS Cost and Profitability Management, you must write XML that identifies the model to be exported and the destination database or XML file. The XML can select all periods and scenarios to export or a subset of them. It can specify a whole model or specific tables (as defined in the data schema) to export.



String comparisons are not case sensitive.

## XML Syntax

The following structure shows the elements used in the export XML and their relationships to each other.

```

<OEExport attributes />
<PeriodScenario attributes />
<StagingArea attributes>
  <StagingTable attributes >
    <Column attributes />
    <ColumnFilter>
      <![CDATA[ expression ]]>
    </ColumnFilter>
    <DimensionColumn attributes />
    <DimensionFilter>
      <Dimension attribute />
      <DimensionMember attributes />
    </DimensionFilter>
  </StagingTable>
</StagingArea>
</OEExport>

```

*Note:* Although an element might be optional, if you include the element, there might be required elements within it.

## OEExport Element (Required)

OEExport is the root element. This element specifies general export information.

Attribute	Required	Values	Description
Model	Required	String	Full Workspace path to the model, excluding the Workspace folder and Models folder; the separator character between folders is the backslash (\); the path is not case sensitive
Version	Required	String	SAS Cost and Profitability Management version number

Attribute	Required	Values	Description
Type	Required	"Staging", "XML"	Format for the exported file; "Staging" is for a database, "XML" is for XML
MessageLimit	Optional	String	Maximum number of similar error messages to write to the log file; when this limit is reached, no other similar error messages are written; default is "50"
Filename	Required when Type is "XML"	String	Absolute path and file name for the exported XML file

### ***PeriodScenario Element (Required)***

This element specifies a period/scenario association. The OEEExport element can have multiple PeriodScenario elements.

Attribute	Required	Values	Description
Period	Required	String	Reference for the period
Scenario	Required	String	Reference for the scenario

### ***StagingArea Element (Required)***

This element specifies the beginning of the export information.

Attribute	Required	Values	Description
JdbcDriverClass	Optional	String	You must specify this attribute if you are using a JDBC driver and the database type is Other.  Otherwise you can omit this attribute.

Attribute	Required	Values	Description
RepositoryName	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
RepositoryId	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryName	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryID	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryReference	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
MetaWorkspaceServer	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
MetaWorkspaceServerId	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
WorkspaceServerName	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
Port	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
Engine	Optional	String	<p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
DBType	Required	String	<p>Specify one of the following numbers from 0 to 6:</p> <p>0 SAS</p> <p>1 SQLServer</p> <p>2 Oracle</p> <p>3 MySQL</p> <p>4 Microsoft Access</p> <p>5 Microsoft Excel</p> <p>6 Other</p>
DriverType	Required	String	<p>Specify either 0 or 1:</p> <p>0 JDBC</p> <p>1 SAS/ACCESS</p>
HostName	Optional	String	<p>For the following databases, specify the information indicated:</p> <p>SQLServer Machine name</p> <p>MySQL Machine name</p> <p>Oracle Machine name</p> <p>Microsoft Access Physical file path</p> <p>Microsoft Excel Physical file path</p> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
PortNumber	Optional	String	<p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p>
ServiceName	Optional	String	<p>For the following databases, specify the information indicated:</p> <p>SQLServer Database name</p> <p>MySQL Database name</p> <p>Oracle Oracle service name where the database instance is running.</p> <p>Otherwise you can omit this attribute.</p>
UserName	Optional	String	<p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p>
Password	Optional	String	<p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p> <p><i>Note:</i> The password, if specified, is encrypted.</p>

Attribute	Required	Values	Description
AdvanceOptions	Optional	String	Specify optional parameters for the following databases: <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> Otherwise you can omit this attribute.

### **StagingTable Element (Required)**

This element identifies the table and the columns of a data schema to export. The StagingArea element can have none or more StagingTable elements.

Attribute	Required	Values	Description
%> Name	Required	String	Name of the table
TableName	Optional	String	Name of the table in the exported file

### **Column Element (Required)**

This element corresponds to a column in a table of the data schema.

Attribute	Required	Values	Description
Name	Required	String	Name of the column
ColumnName	Optional	String	Name of the column in the exported file

### **ColumnFilter Element (Optional)**

This element identifies a filter that is based on the values of properties to filter the export data.

**![CDATA [expression]] Element (Required)**

This element specifies an expression that is used for the filter.

Attribute	Required	Values	Description
Expression	Required	String	Expression

**DimensionColumn Element (Optional)**

This element corresponds to a column that contains dimension information in a table of the data schema.

Attribute	Required	Values	Description
Name	Required	String	Name of the column that contains dimension information
ColumnName	Optional	String	Name of the column in the exported file
DimensionReference	Required for dimension information	For example: "Resource" "Activity" "Cost Object" "External Units"	Reference of the dimension

**DimensionFilter Element (Optional)**

This element identifies a filter that is based on a dimension to filter the export data.

**Dimension Element (Required)**

This element specifies a dimension that is used for the filter.

Attribute	Required	Values	Description
Reference	Required	String	Reference of the dimension



**DimensionMember Element (Optional)**

This element specifies the dimension member (or members) that is used for the filter.

Attribute	Required	Values	Description
Reference	Required	String	Reference of the dimension member
IncludeChildren	Required	"yes", "no"	Indicates whether the children of the dimension member are included

**NestedObject Element (Optional)**

This element specifies an item that is related to another item.

Attribute	Required	Values	Description
Relation	Required	"Attribute" "Destination" "Item" "Source"	Method by which the item is related to the other item

---

## XML to Export and Register Tables

**Sample XML**

The following example exports and registers tables for the model named "PubEx":

```
<PublishTablesInfo>
  <Publish PushToLASR="1"/>
  <MODELCONTEXT Reference="PubEx" MODEL="PubEx"/>
  <Tables>
    <Table Id="1" Name="PubEx_AccountMap"/>
    <Table Id="2" Name="PubEx_AssignmentMap"/>
    <Table Id="3" Name="PubEx_PV_Account"/>
    <Table Id="4" Name="PubEx_PV_Assignment"/>
    <Table Id="5" Name="PubEx_PV_AssignmentExt"/>
    <Table Id="6" Name="PubEx_PV_Attribute"/>
    <Table Id="7" Name="PubEx_PV_AttributeValue"/>
    <Table Id="8" Name="PubEx_PV_Dimension"/>
    <Table Id="9" Name="PubEx_PV_DimMember"/>
    <Table Id="10" Name="PubEx_PV_Driver"/>
    <Table Id="11" Name="PubEx_PV_EnteredCE"/>
  </Tables>
</PublishTablesInfo>
```

```

</Tables>
<LasrInfo
  ApplicationServerIdLASR="A1UVWXYZ.AT000002"
  MetaApplicationServerLASR="SASApp"
  LogicalServerIdLASR=""
  LogicalServerNameLASR="SASApp - Logical Workspace Server"
  LibraryIDLASR="A1UVWXYZ.B5000004"
  LibraryLASR="CPM_LASR_LIB"
  RepositoryIDLASR="A0000001.A1UVWXYZ"
  RepositoryLASR="Foundation"/>
</PublishTablesInfo>

```

## Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to export and register tables, you must write XML that identifies the model whose tables are to be exported and what tables to export and register.

String comparisons are not case sensitive.

## XML Syntax

The following structure shows the elements used in the export XML and their relationships to each other.

```

<PublishTablesInfo>
  <Publish PushToLASR="1|0"/>
  <MODELCONTEXT Reference="modelRef" MODEL="modelName"/>
  <Tables>
    <Table Id="ID" Name="tableName"/>
    <Table Id="ID" Name="tableName"/>
    ...
    <Table Id="ID" Name="tableName"/>
  </Tables>
  <LasrInfo
    ApplicationServerIdLASR="ID"
    MetaApplicationServerLASR="SASApp"
    LogicalServerIdLASR=""
    LogicalServerNameLASR="SASApp - Logical Workspace Server"
    LibraryIDLASR="ID"
    LibraryLASR="libraryName"
    RepositoryIDLASR="ID"
    RepositoryLASR="Foundation"/>
</PublishTablesInfo>

```

### ***PublishTablesInfo element (required)***

PublishTablesInfo is the root element. It contains:

- “Publish element (optional)” on page 223
- “MODELCONTEXT element (required)” on page 223
- “Tables element (required)” on page 223
- “LasrInfo element (optional)” on page 223

**Publish element (optional)**

This element is optional. It specifies whether the tables specified to be exported and registered should also be loaded into a library for the SAS LASR Analytic Server.

Attribute	Required	Values	Description
PushToLASR	Required	0 1	Defaults to "0" if the Publish element is omitted. "0"= Don't also load tables into a library for the SAS LASR Analytic Server. "1"=Also load tables into a library for the SAS LASR Analytic Server.

**MODELCONTEXT element (required)**

This element specifies the name and reference of the model whose tables are to be exported and registered.

Attribute	Required	Values	Description
Reference	Required	String	Model reference
MODEL	Required	String	Model name

**Tables element (required)**

This element specifies the tables to be exported and registered.

Attribute	Required	Values	Description
ID	Required	String	An arbitrary string used by the system to reference the table.
Name	Required	String	Name of the table to be exported. See <a href="#">“Model-Structure Tables”</a> on page 146.

**LasrInfo element (optional)**

This element identifies the LASR Server. The element is required only if PushToLASR="1" in the Publish element.

Attribute	Required	Values	Description
ApplicationServerIdLASR	Required	String	ID of LASR application server

Attribute	Required	Values	Description
MetaApplicationServerLASR	Required	String	Name of LASR meta application server
LogicalServerIdLASR	Required	String	Logical server ID of LASR
LogicalServerNameLASR	Required	String	Logical server name of LASR
LibraryIdLASR	Required	String	ID of LASR library
LibraryLASR	Required	String	Name of LASR library
RepositoryIdLASR	Required	String	ID of LASR repository
RepositoryLASR	Required	String	Name of LASR repository

## XML to Generate a Cube

### Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to generate a cube using a cube configuration, you must write XML that specifies what cube configuration you want to use.

Elements and attributes are case sensitive. String comparisons are not case sensitive.

### Sample XML

```
<OROSCOMMAND Version="2.0">
  <MODELCONTEXT ModelId="1242">
    <PeriodScenario PeriodId="13" ScenarioId="1"/>
    <PeriodScenario PeriodId="14" ScenarioId="1"/>
  </MODELCONTEXT>
  <COMMANDPARAMS MessageLimit="50" CubeAction="Generate">
    <CubeConfig Id="837"/>
  </COMMANDPARAMS>
</OROSCOMMAND>
```

### XML Syntax

The following structure shows the elements used in the XML to generate a cube and their relationships to each other.

```
<OROSCOMMAND Version="2.0">
  <MODELCONTEXT attributes>
    < PeriodScenario attributes/>
  </MODELCONTEXT>
```

```

<COMMANDPARAMSattributes>
<CubeConfig attributes/>
</COMMANDPARAMS>
</OROSCOMMAND>

```

### ***OROSCOMMAND Element (Required)***

OROSCOMMAND is the root element. This element has one version attribute. The OROSCOMMAND element must contain one MODELCONTEXT element and one COMMANDPARAMS element.

Attribute	Required	Values	Description
Version	Optional	"2.0"	If you omit the version number, then the command reverts to version 1 syntax which is supported but deprecated. Version 1 syntax is significantly different.

### ***MODELCONTEXT Element (Required)***

This element specifies the model for which to generate a cube. An MODELCONTEXT element can have multiple PeriodScenario elements.

Attribute	Required	Values	Description
Model   ModelID	Required	String if Model name; Numeric if ModelID	Full Workspace path of the model, excluding the Models Workspace folder if you use the Model Name.  Specifying the path is not necessary if you use the ModelID.

### ***PeriodScenario Element (Required)***

Each PeriodScenario element specifies a period/scenario association to generate. You can have multiple PeriodScenario elements.

Attribute	Required	Values	Description
PeriodId   PeriodRef	Required	Numeric if ID, String if Reference	Period

Attribute	Required	Values	Description
ScenarioId   ScenarioRef	Required	Numeric if ID, String if Reference	Scenario

### **COMMANDPARAMS Element (Required)**

This element configures the calculate operation. An COMMANDPARAMS element can have one CubeConfig element.

Attribute	Required	Values	Description
CubeAction	Optional	"Generate", "CountRows"	"Generate" generates a cube and/or fact table.  "CountRows" counts the rows in the cube to be generated.
MessageLimit	Optional	String that represents an integer between and including 0 and 2,147,483,648	Sets the maximum number of messages to include for each error/warning message type. Defaults to 50. If value is less than zero, all messages are included in the output. A message type is equivalent to a specific type of error or warning. The error/warning may apply to more than one item, so a message is normally generated for each item.

### **CubeConfig Element (Required)**

This element specifies the cube configuration to be used in the generation.

Attribute	Required	Values	Description
ID	Required	Numeric	ID of the cube configuration to be used for generating the cube and/or fact table.

## Configuring How Cubes Are Generated

When SAS Cost and Profitability Management generates cubes, it includes all dimensions to provide the most flexible data analysis. The cubes include redundant information so that the information is available from many data analysis perspectives. However, if your model contains a large amount of information, the process of generating the cubes and viewing them interactively on the OLAP page can take a long time.

To improve performance, a Modeler can create a configuration file that controls which dimensions are included in the cubes. This configuration file must be named `CubeConfig.xml` and it must be located in the Enterprise Server folder in the product installation folder (typically: `C:/Program Files/SAS/Cost and Profitability Management Solution/Enterprise Server`). If this configuration file is not in this location, SAS Cost and Profitability Management generates the cubes by using the default information.

The XML in the configuration file must conform to the XML cube configuration schema. (See: Example of the configuration file)

## Errors in the Cube Configuration File

When cubes are generated and the configuration file is processed, the following errors in the configuration file are reported:

- Failed to parse the configuration file

Although the configuration file exists in the appropriate location, the XML code within the configuration file contains one or more errors.

- Invalid cube name

The name that was specified for a cube is not valid. See: About the cube names.

The following errors in the configuration file are not reported:

- Invalid stage names

The name that was specified for a stage is not valid. See: About the stage names.

- Invalid dimension names

The name that was specified for a dimension is not valid. See: About dimension names.

- A cube configuration that does not specify any dimensions

The configuration file does not contain the required dimension element.

---

## XML to Import Cube Configurations

### Sample XML

```
<CubeConfigurations UploadFile="c:\temp\peCubeConfigurationsExport.xml" Action="rename"/>
```

## Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to import cube configurations, you must write XML to specify the file to be imported.

Elements and attributes are case sensitive. String comparisons are not case sensitive.

## XML syntax

The XML to import cube configurations consists of a single element.

```
< CubeConfigurations attributes/>
```

## CubeConfigurations element (required)

CubeConfigurations is the root and only element. It specifies the name of the file to be imported.

Attribute	Required	Values	Description
UploadFile	Required	String	Specifies the complete path, including the file name of the file to be imported.



Attribute	Required	Values	Description
Action	Optional "rename" is the default	"rename" "replace" "donotimport"	<p>"rename" If a cube configuration exists with the same name as a cube configuration being imported, the one that you are importing is renamed. This applies to every cube configuration being imported if multiple cube configurations are imported.</p> <p>"replace" If a cube configuration exists with the same name as a cube configuration being imported, then the existing cube configuration is replaced with the imported one.</p> <p>"donotimport" If a cube configuration exists with the same name as a cube configuration that you are importing, then the duplicate cube configuration is not imported. Leave the existing one in place.</p>

---

## XML to Import a Model

### Sample XML

The following example imports a model named TestModel into SAS Cost and Profitability Management:

```
<OEImport
  Model="TestModel"
  Action="Create"
  Version="7.2"
  Type="Staging"
  Description="TestModel import">
  <StagingArea
    JdbcDriverClass=" "
    RepositoryName=" "
```

```

RepositoryId=""
LibraryName=""
LibraryID=""
LibraryReference=""
MetaWorkspaceServer=""
MetaWorkspaceServerId=""
WorkspaceServerName=""
Port=""
Engine=""
DBType="1"
DriverType="0"
HostName="D920"
PortNumber="1433"
ServiceName="TestDBForStagingTables"
UserName="DBUser "
Password="xxxx"
AdvanceOptions="Security=SSPI;Persist Security Info=False;">
<StagingTable Name="Period" TableName="tbl_Period">
  <Column Name="Reference" />
  <Column Name="Name" />
  <Column Name="ParentReference" />
  <Column Name="StartDate" />
  <Column Name="EndDate" />
</StagingTable>
<StagingTable Name="PeriodLevel" TableName="tbl_PeriodLevel">
  <Column Name="LevelNo" />
  <Column Name="Name" />
</StagingTable>
<StagingTable Name="Scenario" TableName="tbl_Scenario">
  <Column Name="Reference" />
  <Column Name="Name" />
  <Column Name="ParentReference" ColumnName="ParentRef" />
</StagingTable>
<StagingTable Name="ScenarioLevel" TableName="tbl_ScenarioLevel">
  <Column Name="LevelNo" />
  <Column Name="Name" />
</StagingTable>
<StagingTable Name="CurrencyRate" TableName="tbl_CurrencyRate">
  <Column Name="Period" />
  <Column Name="Scenario" />
  <Column Name="CurrencyFrom" />
  <Column Name="CurrencyTo" />
  <Column Name="Rate" />
</StagingTable>
<StagingTable Name="Driver" TableName="tbl_Driver">
  <Column Name="Name" />
  <Column Name="DriverType" />
  <Column Name="UniqueDriverQuantities" ColumnName="UniqueQuantity" />
</StagingTable>
<StagingTable Name="Dimension" TableName="tbl_Dimension">
  <Column Name="Reference" ColumnName="DimRefnum" />
  <Column Name="Name" ColumnName="DimName" />
</StagingTable>
<StagingTable Name="DimensionOrder" TableName="tbl_DimViewOrder">
  <Column Name="ModuleType" />
  <Column Name="DimRef" ColumnName="DimRefnum" />

```

```

        <Column Name="SequenceNumber" ColumnName="SequenceNo" />
    </StagingTable>
    <StagingTable Name="DimensionLevel" TableName="tbl_DimLevel">
        <Column Name="LevelNo"/>
        <Column Name="DimRef" />
        <Column Name="Name" />
    </StagingTable>
    <StagingTable Name="DimensionMember" TableName="tbl_DimMember">
        <Column Name="DimRef" ColumnName="DimRefnum" />
        <Column Name="Name" />
        <Column Name="Reference" />
        <Column Name="ParentReference" />
        <Column Name="DimLevelName" />
        <Column Name="DimLevel" />
    </StagingTable>
    <StagingTable Name="Account" TableName="tbl_Account">
        <Column Name="ModuleType" ColumnName="Module" />
        <Column Name="Period" />
        <Column Name="Scenario" />
        <Column Name="DriverName" ColumnName="Driver" />
        <Column Name="Reference" />
        <Column Name="TDQUE" />
    </StagingTable>
    <StagingTable Name="Account" TableName="tbl_AccountUpdate">
        <Column Name="ModuleType" ColumnName="Module" />
        <Column Name="Period" />
        <Column Name="Scenario" />
        <Column Name="DriverName" ColumnName="Driver" />
        <Column Name="Reference" />
        <Column Name="TDQUE" />
    </StagingTable>
    <StagingTable Name="EnteredCostElement" TableName="tbl_EnteredCE">
        <Column Name="ModuleType" ColumnName="Module" />
        <Column Name="Period" />
        <Column Name="Scenario" />
        <Column Name="Reference" />
        <Column Name="EnteredCost" ColumnName="Cost" />
        <Column Name="Name" />
    </StagingTable>
    <StagingTable Name="Assignment" TableName="tbl_IBOCSourceDestinationByDimension">
        <Column Name="Period" />
        <Column Name="Scenario" />
        <Column Name="SourceModuleType" ColumnName="SrcModule" />
        <Column Name="DestinationModuleType" ColumnName="DestModule" />
        <Column Name="DriverQuantityFixed" ColumnName="FixedQuantity" />
        <Column Name="DriverQuantityVariable" ColumnName="VariableQuantity" />
        <Column Name="DriverName" ColumnName="Driver" />
    </StagingTable>
    <StagingTable Name="Assignment" TableName="tbl_AssignmentsSourceDestinationByDimension">
        <Column Name="Period" />
        <Column Name="Scenario" />
        <Column Name="SourceModuleType" ColumnName="SrcModule" />
        <Column Name="DestinationModuleType" ColumnName="DestModule" />
        <Column Name="DriverQuantityFixed" ColumnName="FixedQuantity" />
        <Column Name="DriverQuantityVariable" ColumnName="VariableQuantity" />
    </StagingTable>

```

```

<StagingTable Name="ExternalUnit" TableName="tbl_ExternalUnit">
  <Column Name="Period" />
  <Column Name="Scenario" />
  <Column Name="Reference" />
  <Column Name="UnitCostEntered" />
</StagingTable>
<StagingTable Name="AssignmentNonUnique" TableName="tbl_AssignmentNonUnique">
  <Column Name="Period" />
  <Column Name="Scenario" />
  <Column Name="DriverName" />
  <Column Name="DriverQuantityFixed" ColumnName="DriverQuantityFixed" />
  <Column Name="DriverQuantityVariable" ColumnName="DriverQuantityVariable" />
  <Column Name="DestinationModuleType" />
</StagingTable>
<StagingTable Name="ValueAttribute" TableName="tbl_Attribute">
  <Column Name="Type" />
  <Column Name="Reference" />
  <Column Name="Name" />
  <Column Name="ParentReference" ColumnName="Parent" />
</StagingTable>
<StagingTable Name="ValueAttributeAssociation" TableName="tbl_ValueAttributeAssociation">
  <Column Name="Period" />
  <Column Name="Scenario" />
  <Column Name="ItemModuleType" />
  <Column Name="AttributeReference" />
  <Column Name="Value" />
</StagingTable>
<StagingTable Name="DimensionAttributeAssociation" TableName="tbl_DimensionAttributeAssociation">
  <Column Name="Period" />
  <Column Name="Scenario" />
  <Column Name="ItemModuleType" />
  <Column Name="AttributeDimRef" />
  <Column Name="AttributeDimMemberRef" />
</StagingTable>
</StagingArea>
</OEImport>

```

## Overview

When you use the SAS Activity-Based Management Web Services Integration API to import data, you must write XML that maps the structures and periodic data in a database or XML file to the model database.

Create one XML import configuration for new structures. Create a different XML import configuration to delete or update existing structures. Do not add, delete, and update structures in the same XML import configuration.

For importing from an XML file, the configuration does not have to contain detailed table and column information. A sample XML import configuration can be as follows:

```
<OEImport Version="1.1" Type="XML" Action="Create" Model="Test" Filename="C:\temp\TestModel.xml"></OEImport>
```

The following elements and attributes specify the structures and periodic data to import into a model. Elements and attributes are case sensitive. String comparisons are not case sensitive.

Each XML can import into one model.

## XML Syntax

The following structure shows the elements used in the XML and their relationships to each other.

```
<OEImport attributes >
  <StagingArea attribute >
    <StagingTable attributes >
      <Column attributes >
    </StagingTable>
  </StagingArea>
</OEImport>
```

*Note:* Although an element might be optional, if you include the element, there might be required elements within it.

### OEImport Element (Required)

OEImport is the root element. This element specifies general import information. You can have one OEImport element in an XML import configuration.

If you include diagnostic information, you will see LoadCache elements within ImportTable elements.

Attribute	Required	Values	Description
Model	Required	String	Full Workspace path, excluding the Workspace Name folder and Models folder; the separator character between folders is the backslash (\); if the model does not exist, a new one is created; if the model exists, the import process adds and deletes structures and periodic data in the existing model; new periods and scenarios can be created

Attribute	Required	Values	Description
Action	Required	<blank> Create ReplaceAll	Inserts and merges data into a model  Create creates a new model  ReplaceAll removes all model-specific data from the existing model before processing information; periods and scenarios, which are not model-specific, are not removed
Version	Required	String	SAS Cost and Profitability Management version number
CustomSchema		1	Must always be 1
Type	Required	XML, Staging	Type of import
Filename	Required when Type is XML	String	Absolute path and file name for the exported XML file
Description	Optional	String	Description of the import

### **StagingArea Element (Required)**

This element defines the connection information. An OEImport element can have one or more StagingArea elements.

*Note:* The system generates a connection string when you use the Connection Information dialog to connect to a database.

**Connection Information** ✕

Select the method to attach to the target database. Then select the database type and enter the connection details.

☒ Use a JDBC driver  
☐ Use SAS/ACCESS

Database type: SAS datasets ▼

Repository: Foundation ▼

Application server: SASApp ▼

Logical workspace server: SASApp - Logical Workspace Server ▼

Library: ABM\_BevT\_Lib ▼

Test Connection
OK
Cancel

Attribute	Required	Values	Description
JdbcDriverClass	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using a JDBC driver and the database type is Other.</p> <p>Otherwise you can omit this attribute.</p>
RepositoryName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
RepositoryId	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryID	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
LibraryReference	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>



Attribute	Required	Values	Description
MetaWorkspaceServer	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
MetaWorkspaceServerId	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
WorkspaceServerName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
Port	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
Engine	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute if you are using either of the following:</p> <ul style="list-style-type: none"> <li>• SAS datasets</li> <li>• SAS/Access</li> </ul> <p>Otherwise you can omit this attribute.</p>
DBType	Required	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>Specify one of the following numbers from 0 to 6:</p> <p>0 SAS</p> <p>1 SQLServer</p> <p>2 Oracle</p> <p>3 MySQL</p> <p>4 Microsoft Access</p> <p>5 Microsoft Excel</p> <p>6 Other</p>
DriverType	Required	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>Specify either 0 or 1:</p> <p>0 JDBC</p> <p>1 SAS/ACCESS</p>

Attribute	Required	Values	Description
HostName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>For the following databases, specify the information indicated:</p> <p>SQLServer Machine name</p> <p>MySQL Machine name</p> <p>Oracle Machine name</p> <p>Microsoft Access Physical file path</p> <p>Microsoft Excel Physical file path</p> <p>Otherwise you can omit this attribute.</p>
PortNumber	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p>
ServiceName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>For the following databases, specify the information indicated:</p> <p>SQLServer Database name</p> <p>MySQL Database name</p> <p>Oracle Oracle service name where the database instance is running.</p> <p>Otherwise you can omit this attribute.</p>

Attribute	Required	Values	Description
UserName	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p>
Password	Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>You must specify this attribute for any of the following:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p> <p>Note: The password, if specified, is encrypted.</p>
AdvanceOptions	%>Optional	String	<p>% " class=TableCell style="width: 40%;"&gt;</p> <p>Specify optional parameters for the following databases:</p> <ul style="list-style-type: none"> <li>• SQLServer</li> <li>• Oracle</li> <li>• MySQL</li> </ul> <p>Otherwise you can omit this attribute.</p>

### StagingTable Element (Required)

This element maps a SAS Cost and Profitability Management model table to one of the tables in the source database or XML file.

When you import information from a database or XML file, all of the tables in the database must exist, but the following tables must contain information (other tables can be empty):

- Period
- Scenario

- Dimension
- DimensionOrder

You must define at least one dimension for each of these tables or the import will fail or an empty table will be created.

When you use the SAS Cost and Profitability Management Web Services Integration API to import information into a model, the XML import configuration can provide any StagingTable element as long as model items exist before they are referenced. For example, before importing an account, the period, scenario, driver, and dimension members that the account references must exist.

The DimensionOrder table is required when you create a model. When a model is created, it uses the information in the DimensionOrder table to determine which dimensions define the hierarchies in each module. If you are not creating a model, this table is ignored.

A StagingArea element can have one or more StagingTable elements. If there is more than one StagingTable element, each one must have a different value for the TableName attribute.

You can import Column elements and DimensionSignature elements from different source database tables; for example, you can import some accounts from one table and other accounts from a different table.

If a model table requires either a reference or dimension signature, and if both are supplied, then the dimension signature is used to identify an account.

Attribute	Required	Values	Description
Name	Required	String	Name of the model table to import into
TableName	Optional if it is the same as Name	String	Name of the table in the source database or XML file that has the information to import into a model

### **Column Element (Optional)**

This element maps a field in a SAS Cost and Profitability Management model table to one of the fields in the source database or XML file. A StagingTable element can have one or more Column elements.

Attribute	Required	Values	Description
Name	Required	String	Name of the model field to import into
ColumnName	Optional if it is the same as Name	String	Name of the field in the source database or XML file that has the information to import into a model

Attribute	Required	Values	Description
DefaultVal	Optional	String	Default value

## XML to Publish Period/Scenario Associations

### Sample XML

```
<PublishPeriodScenarios >
  <ModelContext ModelFullName="peAPI" >
    <PeriodScenario PeriodId="9" ScenarioId="1" Publish="true" />
  </ModelContext>
</PublishPeriodScenarios>

<PublishPeriodScenarios >
  <ModelContext ModelFullName="peAPI">
    <PeriodScenario PeriodRef="2008 Q1" ScenarioRef="Actual" Publish="false" />
  </ModelContext>
</PublishPeriodScenarios>
```

### Overview

When you use the SAS Cost and Profitability Management Web Services Integration API to publish period/scenario associations you must write XML to specify the associations to be published.

### XML syntax

The following structure shows the elements used in the XML to publish period/scenario associations and their relationships to each other.

Elements and attributes are case sensitive. String comparisons are not case sensitive.

```
<PublishPeriodScenarios >
  <ModelContext attributes>
    <PeriodScenario attributes>
  </ModelContext>
</ModelContext>
</PublishPeriodScenarios>
```

### **PublishPeriodScenarios element (required)**

PublishPeriodScenarios is the root element. It has no attributes. A PublishPeriodScenarios element can have one ModelContext element.

### **ModelContext element (required)**

This element specifies the model whose period/scenario associations are to be published. A ModelContext element can have multiple PeriodScenario elements.

Attribute	Required	Values	Description
ModeFullName	Required	String	Full Workspace path of the model, excluding the Models Workspace folder

### ***PeriodScenario element (required)***

Each PeriodScenario element specifies a period/scenario association—one period and one scenario. You can have multiple PeriodScenario elements.

Attribute	Required	Values	Description
PeriodId   PeriodRef	Required	Numeric if ID, String if Reference	Period
ScenarioId   ScenarioRef	Required	Numeric if ID, String if Reference	Scenario





## Chapter 20

# Easy API

---

<b>Using Easy API</b> .....	<b>245</b>
Overview .....	245
Create an XML File .....	246
Save Easy API Commands in a Text File .....	247
Invoke Easy API .....	250
<b>Example of Using Easy API</b> .....	<b>251</b>
Overview .....	251
1. Export January Tables .....	252
2. Run a SAS Job to Update the Tables .....	257
3. Re-import the Tables .....	258
4. Calculate the Model .....	261
5. Run Easy API .....	263

---

## Using Easy API

### Overview

Using Easy API, you can do in batch many of the same operations that you can do inside SAS Cost and Profitability Management. With Easy API, you can

- Acquire or release a read or read\write lock on a model
- Terminate a write lock on a model
- Import and export model data
- Calculate a model
- Generate a cube
- Copy and paste a model
- Copy model data from one period/scenario to another
- Import and export cube configurations
- Export and register tables

In addition, you can use Easy API to run SAS stored processes, an external SAS Enterprise Guide project, or any other executable that you want to invoke. So, for

example, you can use Easy API to export model data, invoke a SAS stored process to update the exported data, and finally import the updated data back into your model.

Operations are run in the order specified in your EasyAPI.txt file. Whatever operations you perform, Easy API synchronizes them so that the next operation to run does not begin until the previous one has finished. For example, a SAS program to update exported tables does not run until the tables have been exported.

Invoking Easy API to perform a SAS Cost and Profitability Management operation involves three steps:

1. “Create an XML File” on page 246

The XML file describes the operation to be performed.

2. “Save Easy API Commands in a Text File” on page 247

Easy API commands invoke SAS Cost and Profitability Management and pass an XML file to tell it what to do.

3. “Invoke Easy API” on page 250

Easy API uses the text file to run its commands.

## Create an XML File

SAS Cost and Profitability Management uses XML internally to encode the information that it needs for performing operations. Easy API uses the same XML to invoke SAS Cost and Profitability Management in batch to perform those operations. Following is sample XML to generate a cube. Notice that the XML specifies the model, periods, and cube configuration to be used in generating the cube.

```

<OROSCOMMAND Version="2.0">
  <MODELCONTEXT Reference="petut" ModelId="1011">
    <PeriodScenario ScenarioId="1" PeriodId="21"/>
  </MODELCONTEXT>
  <COMMANDPARAMS CubeAction="Generate" ForceCubeGenerate="1">
    <CubeConfig Id="76"/>
  </COMMANDPARAMS>
</OROSCOMMAND>

```

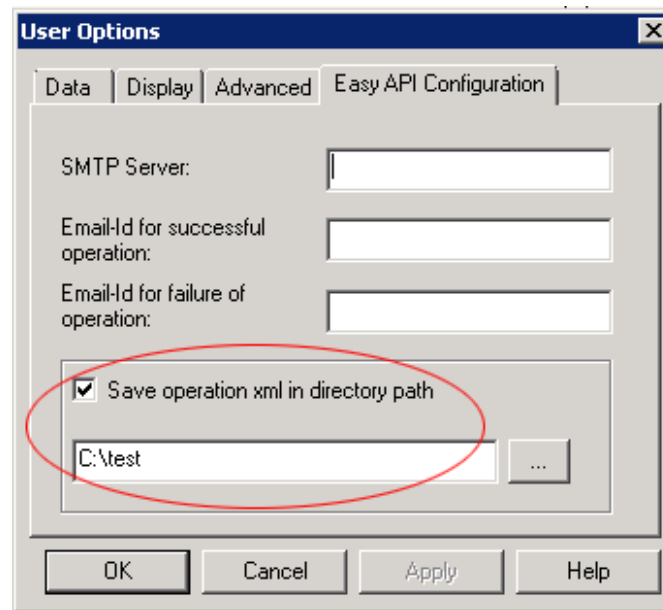
The diagram shows the XML structure with three annotations: 'model' points to the value 'petut' in the MODELCONTEXT tag; 'period' points to the value '21' in the PeriodId attribute of the PeriodScenario tag; and 'cube configuration' points to the value '76' in the CubeConfig tag.

Because Easy API uses exactly the same XML to invoke SAS Cost and Profitability Management that SAS Cost and Profitability Management itself uses internally, the easiest way for you to create the XML that you need to run Easy API is to ask SAS Cost and Profitability Management to create it.

To ask SAS Cost and Profitability Management to create XML:

1. Inside SAS Cost and Profitability Management, select **Tools** ⇒ **User Options**.
2. Click the **Easy API Configuration** tab.
3. Select **Save operation xml in directory path**.
4. Specify the directory path where the XML will be saved.

Now, when you perform an operation inside SAS Cost and Profitability Management, the XML for that operation is saved in a file in the directory that you specified.



You can modify the XML file to suit your purposes. For example, you might modify the XML file shown here to generate different periods for the same model or to generate the same periods for a different model.

For detailed information on the XML files, see [Chapter 19, “XML Passed to the API,”](#) on page 201.

### Save Easy API Commands in a Text File

The following table lists the Easy API commands and tells what each command does. Notice that each command takes one parameter which is either the path and name of an XML file, or the path and name of an external program.

Command and sample argument	What it does
<code>acquiremodellock "read   write" "modelRef"</code>	Acquires a read or write lock on the model. <i>Note:</i> If you do not issue <code>acquiremodellock</code> before performing an operation on a model, the system attempts to do it for you. It is recommended, however, that you explicitly acquire a lock on a model before performing an operation.
<code>releasemodellock "modelRef"</code>	Releases whatever lock (read or write) that is on the model. <i>Note:</i> If you do not issue <code>releasemodellock</code> , the system releases the lock on the model for you. It is recommended, however, that you explicitly release the lock on a model to ensure that it is available for other operations.
<code>terminatewritelock "modelRef"</code>	Terminates a write lock that is acquired by a modeler on the model.
<code>calculate "&lt;folder_path&gt;\your.xml"</code>	Calculate and/or generate a cube <i>Note:</i> The XML file that you use determines whether this command does a calculation or generates a cube.

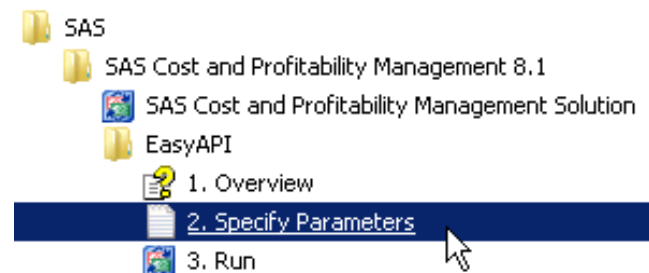
Command and sample argument	What it does
copypaste "<folder_path>\your.xml"	Copy model data and paste as a new model
copyperiod "<folder_path>\your.xml"	Copy model data from one period/scenario to another
export "<folder_path>\your.xml"	Export model data
export cube "<folder_path>\your.xml"	Export cube configurations
export and register tables "<folder_path>\your.xml"	Export database tables that are registered in the SAS Management Console for use with other SAS or non-SAS programs. See Chapter 82, "Export Registered Tables," in <i>SAS Cost and Profitability Management: User's Guide</i> .
import "<folder_path>\your.xml"	Import model data
import cube "<folder_path>\your.xml"	Import cube configurations
run "<folder_path>\your.xml"	Execute external programs including, but not limited to, SAS stored processes. For example, you can also run SAS Enterprise Guide vbscripts using this Run command.
// Comment	You can put ' or // in front of the command line to comment out (skip) a particular Easy API command.

*Note:* Refer to the PeriodDefinition and ScenarioDefinition database tables to obtain the values of the PeriodId and ScenarioId tags to used in the XML file.

To issue an Easy API command, put it in a text file named EasyAPI.txt residing in the following directory:

<installation\_directory>SASCostandProfitabilityManagementClient\8.1\EasyAPI\  
 For example: C:\Program Files\SASHome\x86\SASCostandProfitabilityManagementClient\8.1\EasyAPI\

An easy way to open EasyAPI.txt is by selecting **Start** ⇒ **All Programs** ⇒ **SAS** ⇒ **SAS Cost and Profitability Management 8.1** ⇒ **EasyAPI** ⇒ **2. Specify parameters** from the **Start** menu.



The following picture shows EasyAPI.txt as it appears on installation of SAS Cost and Profitability Management.

```

File Edit Format View Help
Use the following parameters:
SAS_environment_name
If you want to use IWA then mention "USE IWA" here.
If you are using IWA then mention metadata machine name here.
If you are using IWA then mention metadata port here.
If you are not using IWA then mention your login here.
If you are not using IWA then mention your password or your encoded password generated using EncodePassword.exe here.

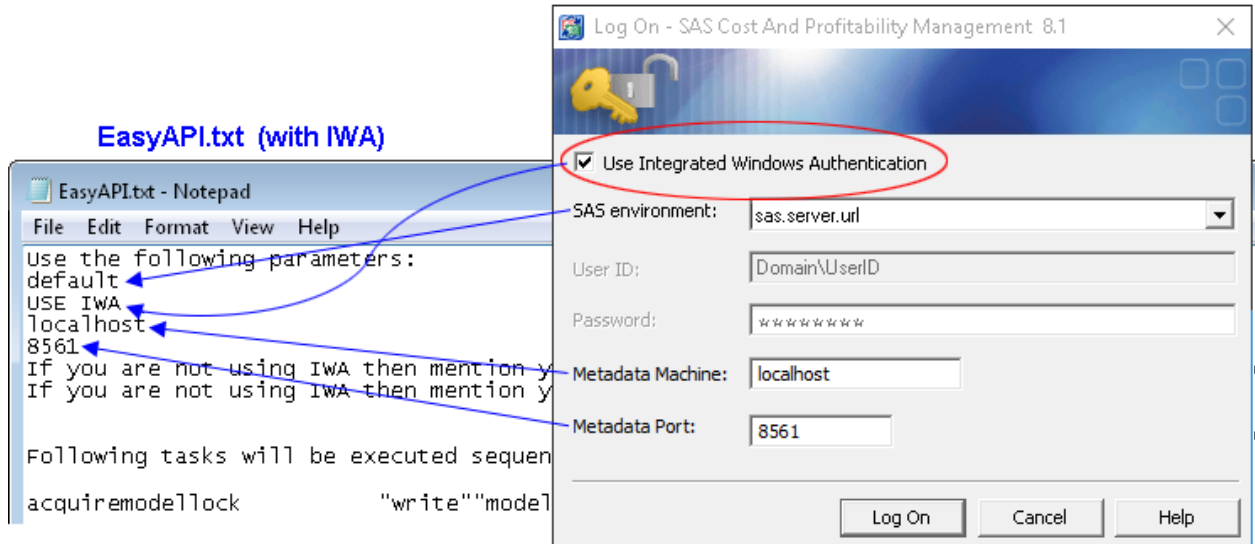
Following tasks will be executed sequentially. Please update entries and path locations for operations which you need

acquiremodellock      "<read/write>""<modelref>"
releasemodellock      "<modelref>"
export                 "Folder Location\Export_Period1.xml"
run                   "Folder Location\Update_Period1.sas"
import                "Folder Location\Import_Period1.xml"
calculate              "Folder Location\Calculate_Period1.xml"
export cube            "Folder Location\exportcubeconfig_Model_ConfigName.xml"
import cube            "Folder Location\importcubeconfig_Model_ConfigName.xml"
export and register tables "Folder Location\publishdata.xml"
copypaste              "Folder Location\copypaste.xml"

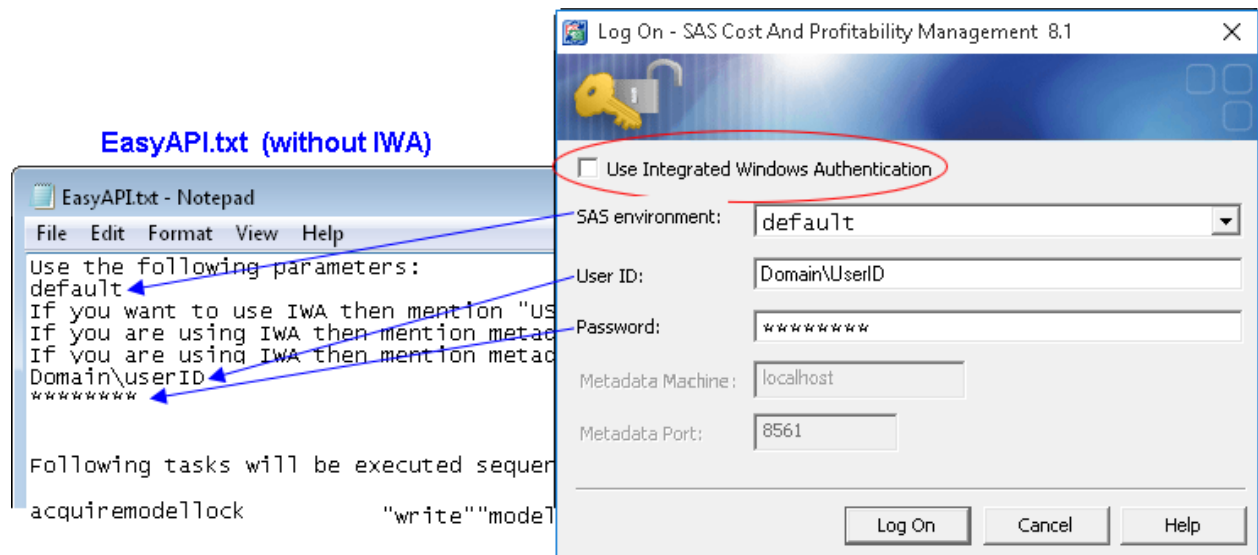
```

Make sure to include your login credentials in the txt file.

The following picture shows EasyAPI.txt for logging on with IWA.

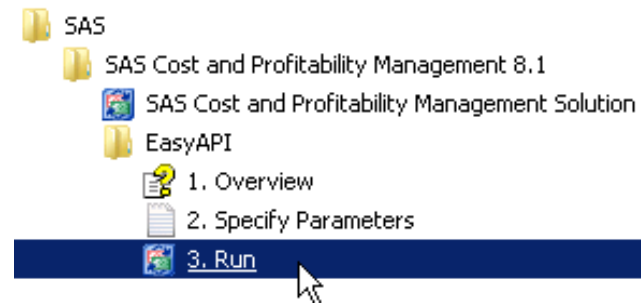


And the following picture shows EasyAPI.txt for logging on without IWA.



### Invoke Easy API

To invoke Easy API, select **Start** ⇒ **All Programs** ⇒ **SAS** ⇒ **SAS Cost and Profitability Management 8.1** ⇒ **EasyAPI** ⇒ **3. Run** from the **Start** menu.



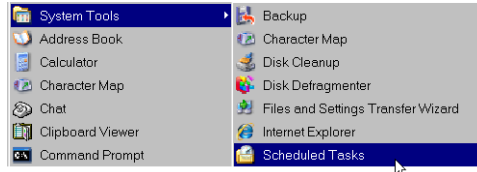
Easy API can e-mail you the results of its operation. To receive an e-mail with operation results:

1. Inside SAS Cost and Profitability Management, select **Tools** ⇒ **User Options**.
2. Click the **Easy API Configuration** tab.
3. Specify an **SMTP server** for sending the mail.
4. Specify an **Email Id for successful operation**.
5. Specify an **Email Id for failure of operation**.

Notes:

- Log files named EasyAPI.log or CutomEasyAPI.log are created in the Easy API Installation folder. You can also access the Easy API operations log in the Windows Event Viewer.
- You can also invoke Easy API by running EasyAPI.exe, which is installed in  
`<installation directory>SASCostandProfitabilityManagementClient\8.1\EasyAPI\.`

- By supplying a path argument to EasyAPI.exe, you can tell it to use a different txt file for Easy API commands, for example, **EasyAPI.exe "c:\MyPath\EasyAPI2.txt"**. If you don't supply a path argument, then Easy API uses EasyAPI.txt in its installation directory.
- You can use the Microsoft Windows Scheduled Tasks Wizard to schedule EasyAPI.exe to run automatically at selected intervals.



- In the EasyAPI.txt file, you can either store your password in clear text or you can encode it using EncodePassword.exe located at **<installation directory>\SASCostandProfitabilityManagementClient\8.1\EasyAPI\**. EncodePassword.exe produces an encoded string that you can paste into EasyAPI.txt. Easy API then decodes the password before performing Easy API operations.

---

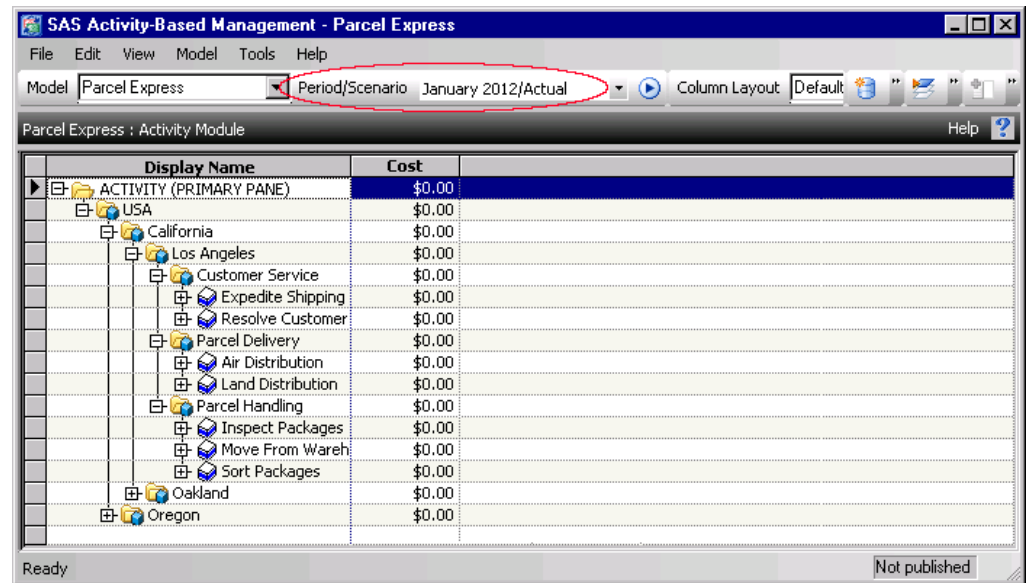
## Example of Using Easy API

### Overview

This section presents an example of using Easy API to automate performing the following tasks:

- [“1. Export January Tables” on page 252](#)
- [“2. Run a SAS Job to Update the Tables” on page 257](#)
- [“3. Re-import the Tables” on page 258](#)
- [“4. Calculate the Model” on page 261](#)
- [“5. Run Easy API” on page 263](#)

Notice in the following picture of a sample model that the cost is \$0 for January activities because the model hasn't been updated with data for the month. So, in this example, we export January tables, update them, and re-import them back into the model for calculation. The example shows how to automate all of these tasks using Easy API.



## 1. Export January Tables

### Turn on Writing of XML

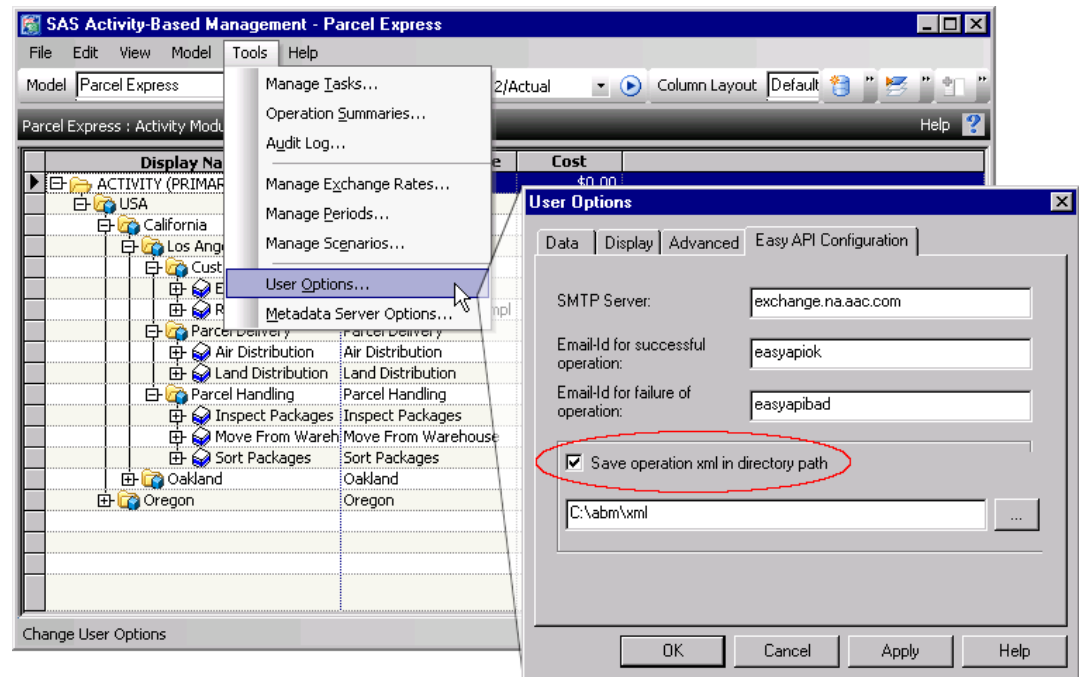
Before exporting the January tables, we must do one thing—ask SAS Cost and Profitability Management to write to a file the XML that it uses to perform operations. Because Easy API uses exactly the same XML to invoke SAS Cost and Profitability Management that SAS Cost and Profitability Management itself uses internally, the easiest way to create the XML that you need to run Easy API is to ask SAS Cost and Profitability Management to create it.

To ask SAS Cost and Profitability Management to create XML:

1. Inside SAS Cost and Profitability Management, select **Tools** ⇒ **User Options**.
2. Click the **Easy API Configuration** tab.
3. Select **Save operation xml in directory path**.
4. Specify the directory path where the XML will be saved.

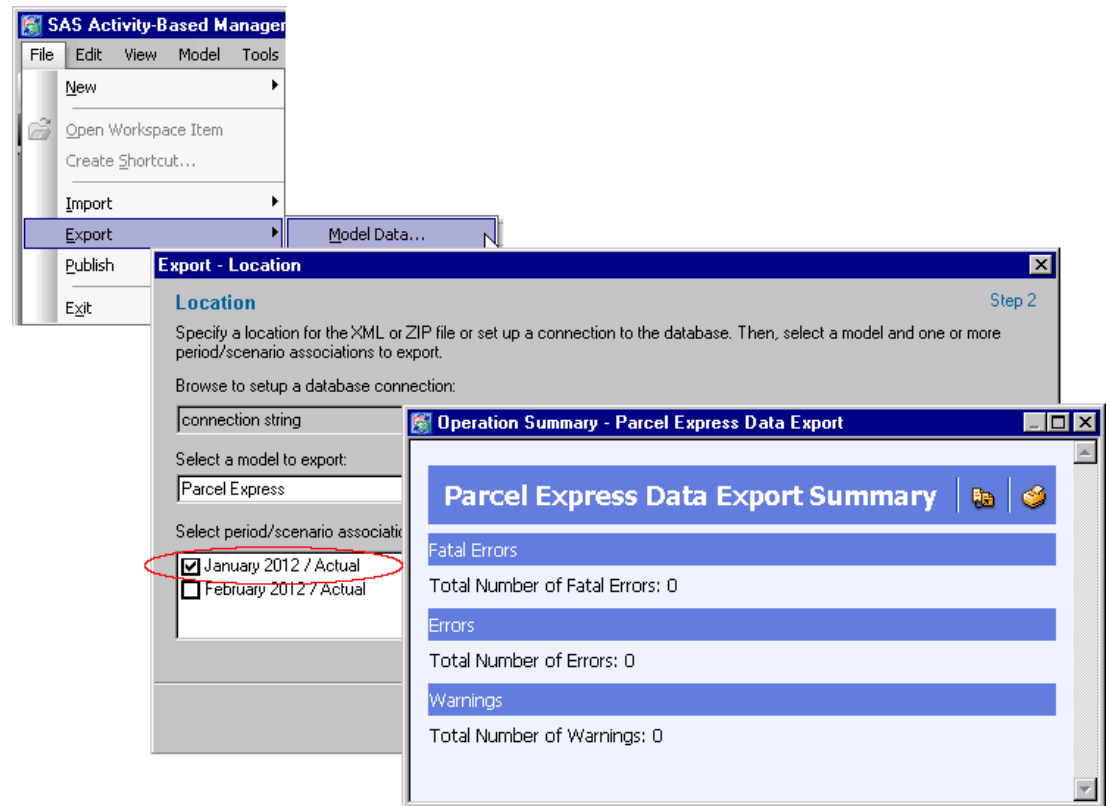
Now, when you perform an operation inside SAS Cost and Profitability Management, the XML for that operation is saved in a file in the directory that you specify.



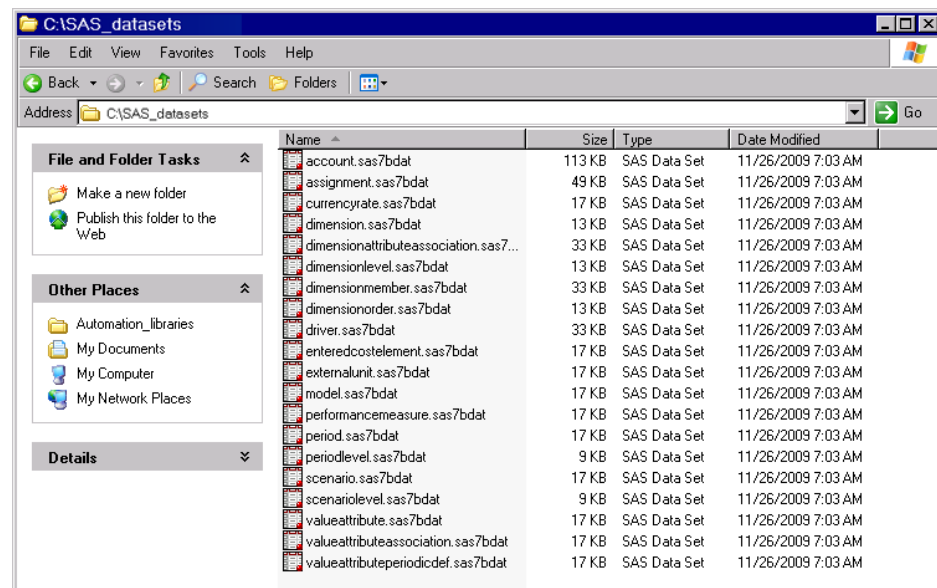


### Export Tables

Now we are ready to export the January tables. We don't show the export in any detail. Note, however, that the export has completed with no errors. It is important that you can successfully do inside SAS Cost and Profitability Management whatever task it is that you want to automate in batch with Easy API. Because Easy API calls SAS Cost and Profitability Management to perform tasks, if you can't perform the task inside SAS Cost and Profitability Management, then you won't be able to perform the same task outside SAS Cost and Profitability Management using Easy API.



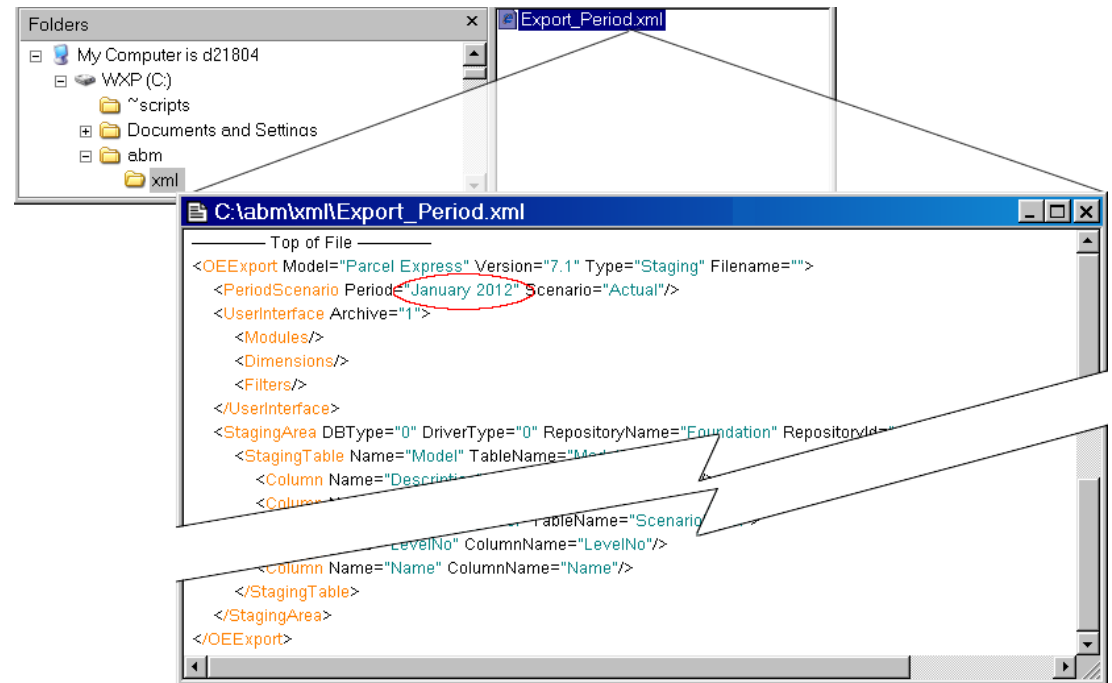
Because the export was successful, it produced the SAS tables shown in the following picture. Although this example shows SAS tables, you can use Easy API to automate exporting any type of database tables that you want.



### Modify the Export XML

Because we asked SAS Cost and Profitability Management to write to a file the XML that it uses to perform operations, you can see in the following picture the XML file that it produced during the export. The filename, however, has been changed for the sake of this example.

Notice the parameter that calls for exporting the January table. If you want to export the February table next when the export task is performed in batch using Easy API, then you can modify this XML to specify February instead.



### Automate the Export for Next Time

Having modified the XML file to export the period that you want, you next need to point Easy API to the modified file. Easy API uses a text file, EasyAPI.txt, for its commands. Each command takes one argument, which is usually the name and path of the XML file to be passed to SAS Cost and Profitability Management for performing an operation. For a list of Easy API commands, see [“Save Easy API Commands in a Text File” on page 247](#).

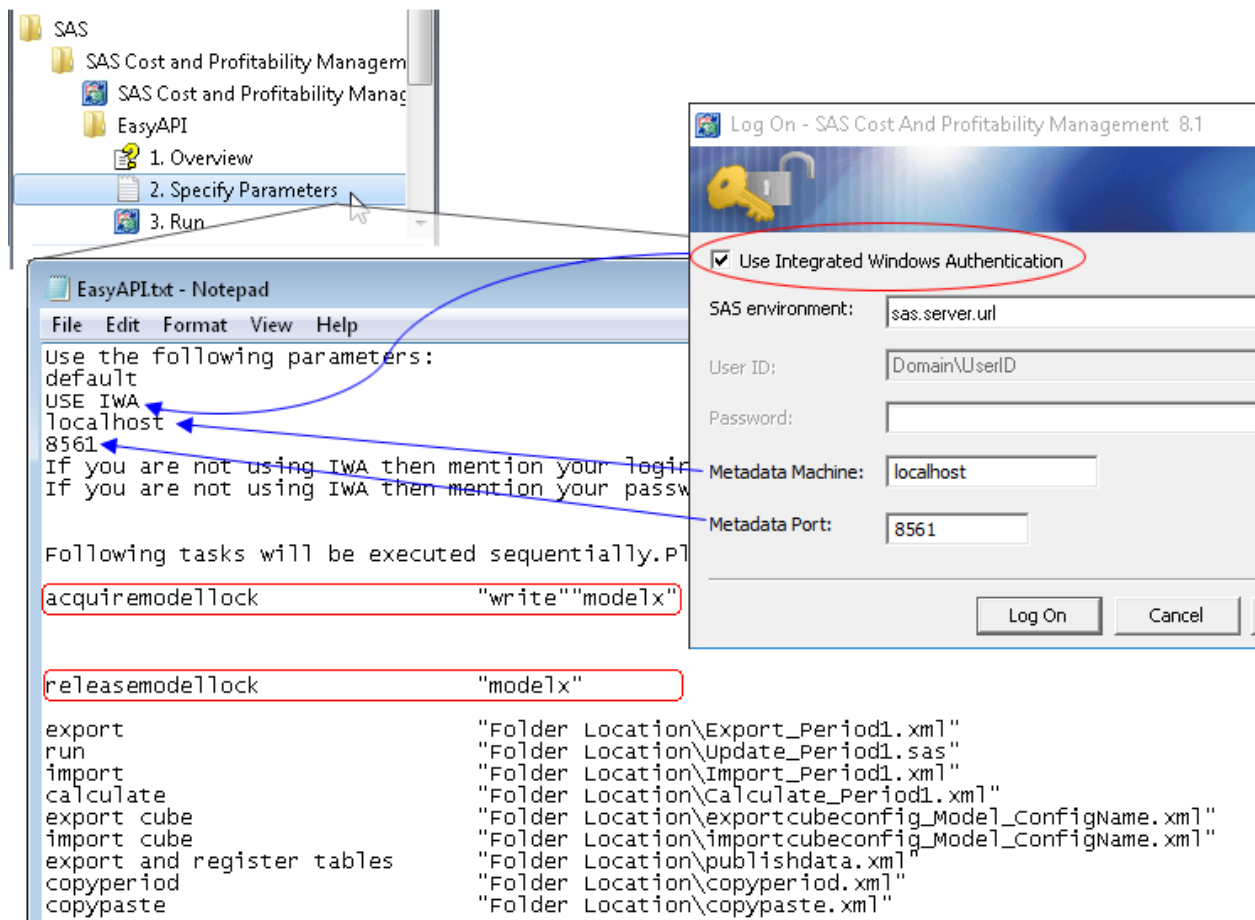
EasyAPI.txt is installed in the following directory:

```
<installation directory>\SASCostandProfitabilityManagementClient\8.1\EasyAPI
```

For example: C:\Program Files\SASHome\x86\SASCostandProfitabilityManagementClient\8.1\EasyAPI

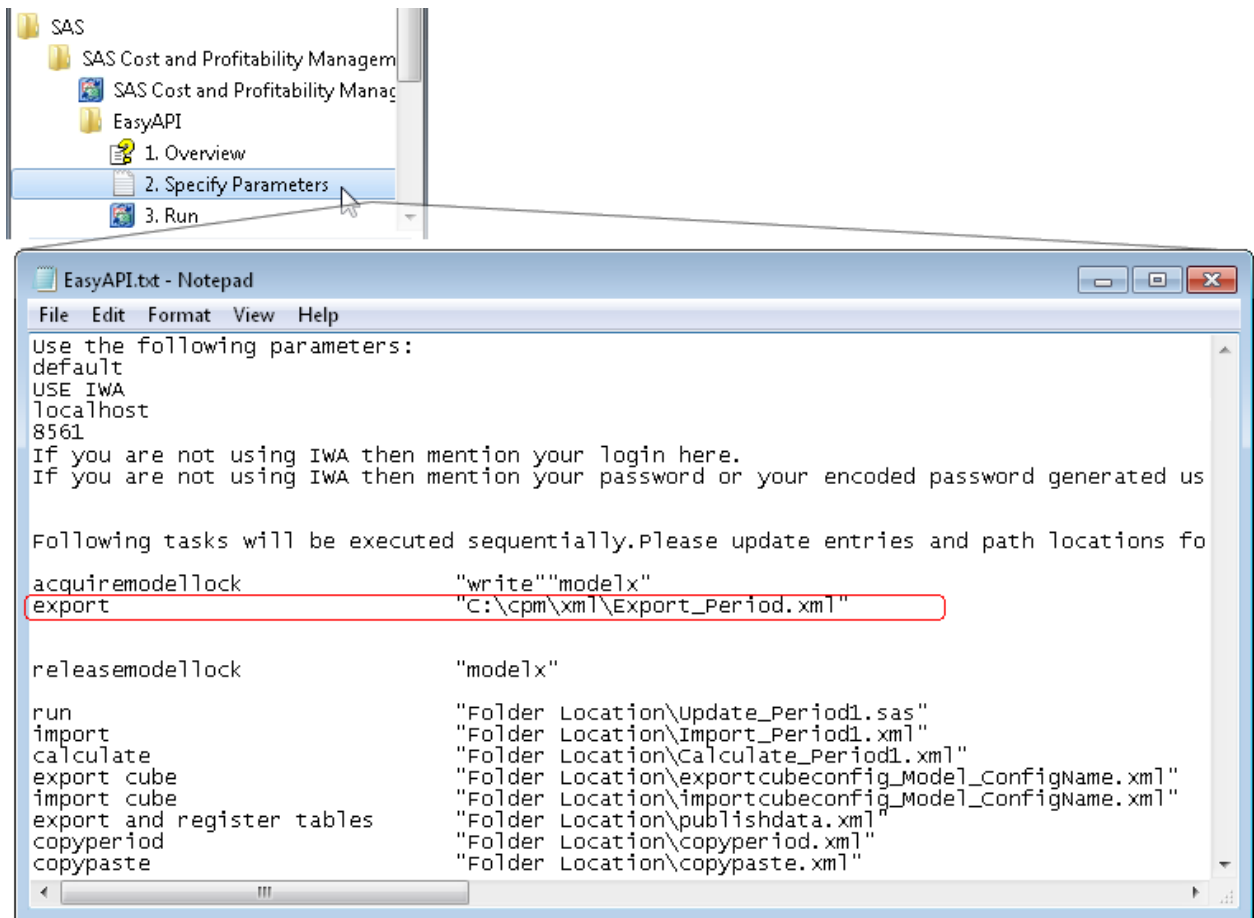
To open EasyAPI.txt, select **Start** ⇒ **All Programs** ⇒ **SAS** ⇒ **SAS Cost and Profitability Management 8.1** ⇒ **Easy API** ⇒ **2. Specify Parameters** from the **Start** menu.

The following picture shows EasyAPI.txt modified so as to log in to SAS Cost and Profitability Management using Integrated Windows Authentication (IWA). It also shows that the **acquiremodellock** and **releasemodellock** commands have been modified so as to acquire a **write** lock on **modelx** and later to release the lock. Space is added in between the two commands to allow further API commands to be inserted.



The following picture shows EasyAPI.txt with the following command inserted after the **acquiremodellock** command:

```
export "c:\cpm\xml\Export_Period.xml"
```



## 2. Run a SAS Job to Update the Tables

### Run the Job

Having exported the January tables, your next task is to run a SAS job to update the tables. We don't show any details of the SAS job, and you can run any program that you want to update the tables. Regardless of what program you use, you can automate its running with Easy API.

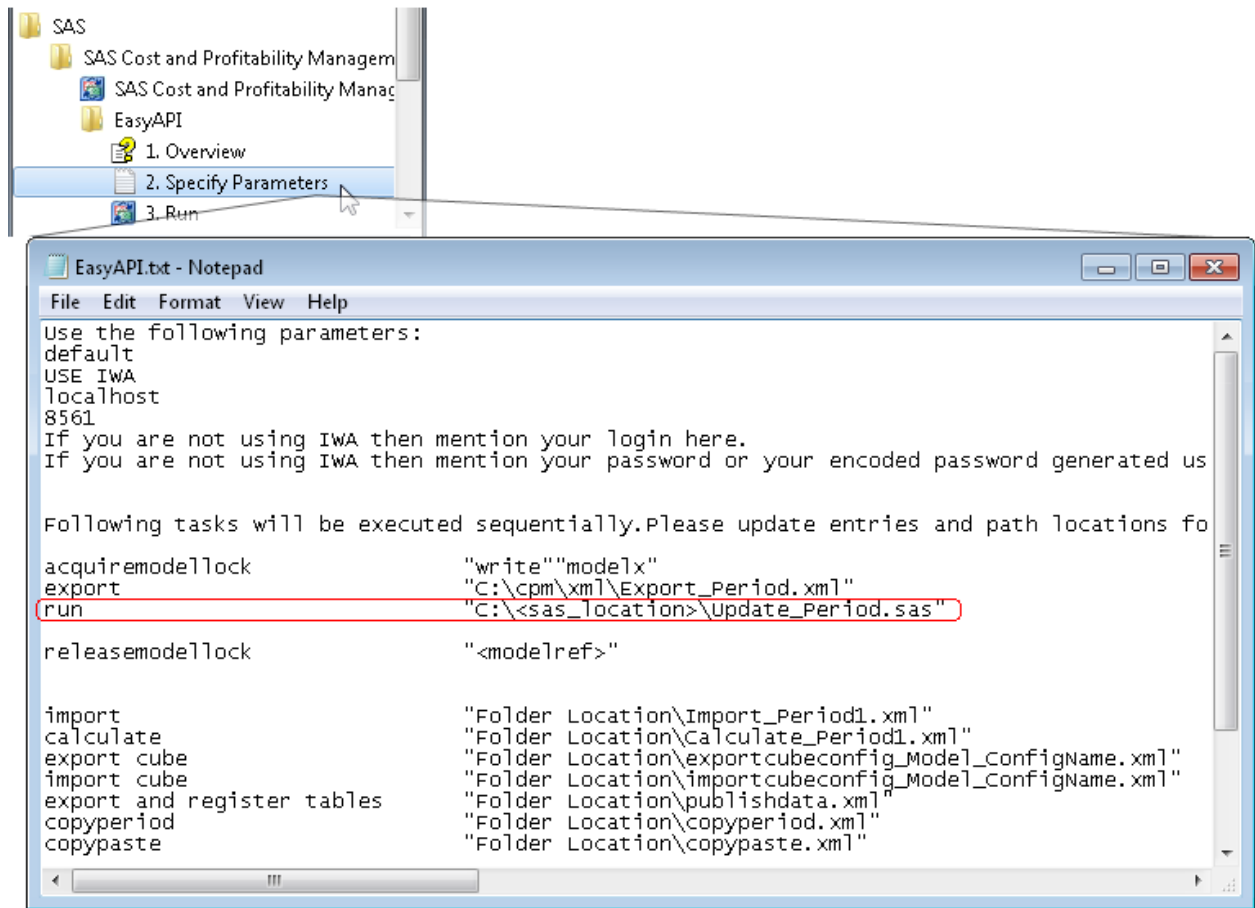
The following picture shows one of the tables after it was updated with entered costs for January.

	Period	Scenario	ModuleType	AccountDimRef1	AccountDimMemberRef1	EnteredCost
1	January 2012	Actual	Resource	Region	Cary City	100
2	January 2012	Actual	Resource	Region	South San Francisco City	200
3	January 2012	Actual	Resource	Region	Washington DC	240
4	January 2012	Actual	Resource	Region	Portland City	440

### Automate the SAS Job for Next Time

To automate running the job, all you have to do is modify EasyAPI.txt again to invoke the desired program. The following picture shows the following API command inserted:

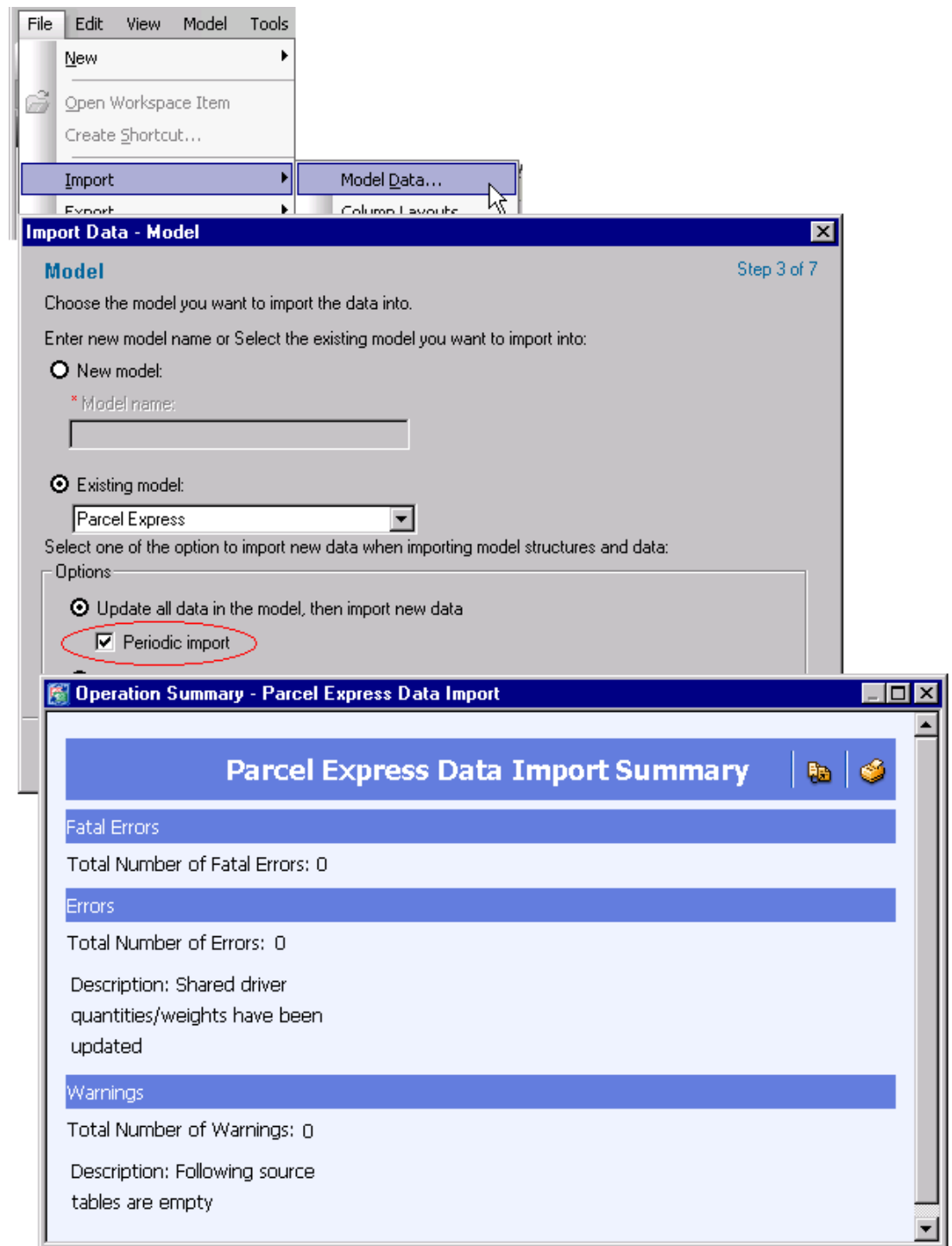
```
run "c:\<sas_location>\Update_Period.sas"
```



### 3. Re-import the Tables

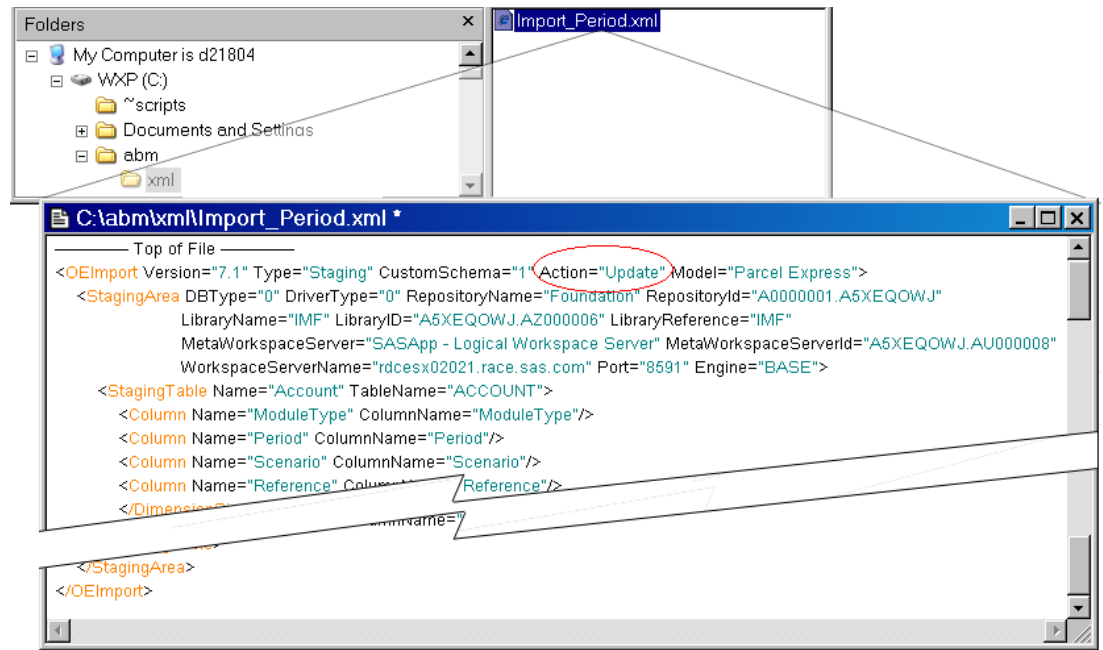
#### Re-import the Tables

Now that the tables have been updated, you can re-import them back into the model. In the following picture, note that we have selected **Periodic import**. This option causes only those staging tables that contain periodic data to be displayed in the Import Wizard for you to select for importing. (Non-periodic data does not change from one period to the next and, so, does not have to be updated and re-imported).



### Modify the Import XML

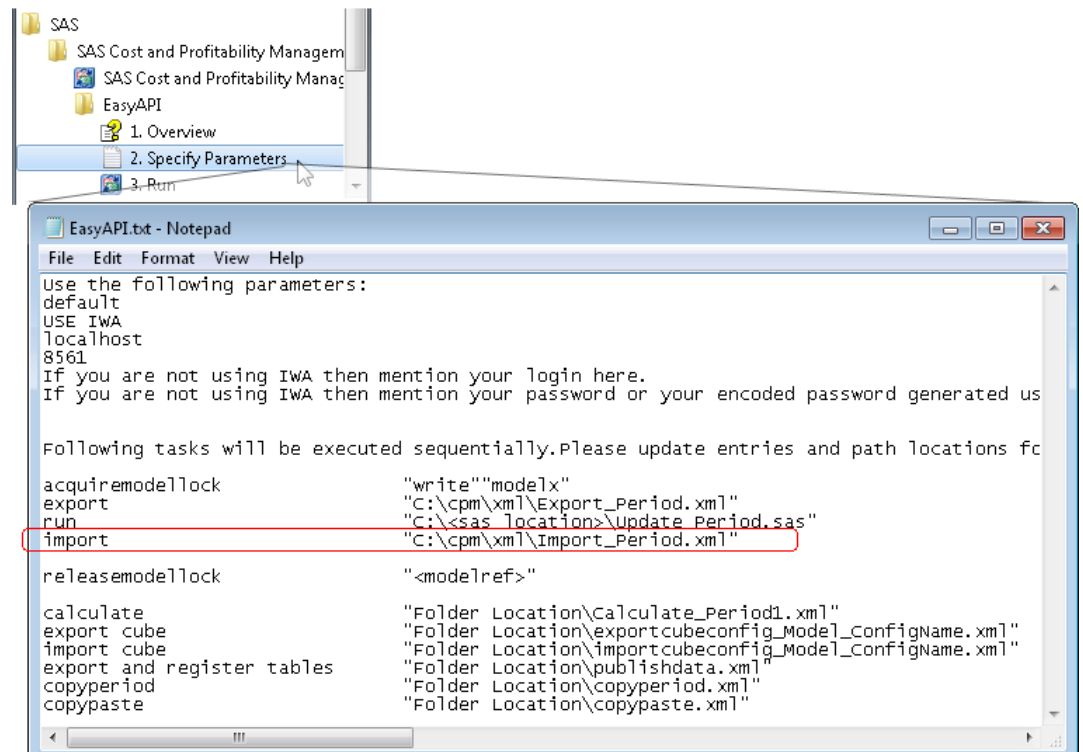
Once again, SAS Cost and Profitability Management has written to a file the XML that it used for performing the import. Notice that the `Activity=` parameter specifies `Update`. That means that the model is updated rather than a new model being created. Also notice that the file does not have a `Period=` parameter. The period information (January, in this case) is contained inside the tables being imported rather than inside the XML file. That means that you do not have to modify this XML file at all to automate the import with Easy API. You do, however, have to let Easy API know where the file is located, which we do next.



### Automate the Import for Next Time

To make Easy API perform the import, once again you must modify EasyAPI.txt. Notice in the following picture that a third command line has been inserted to specify

```
import "c:\cpm\xml\Import_Period.xml"
```

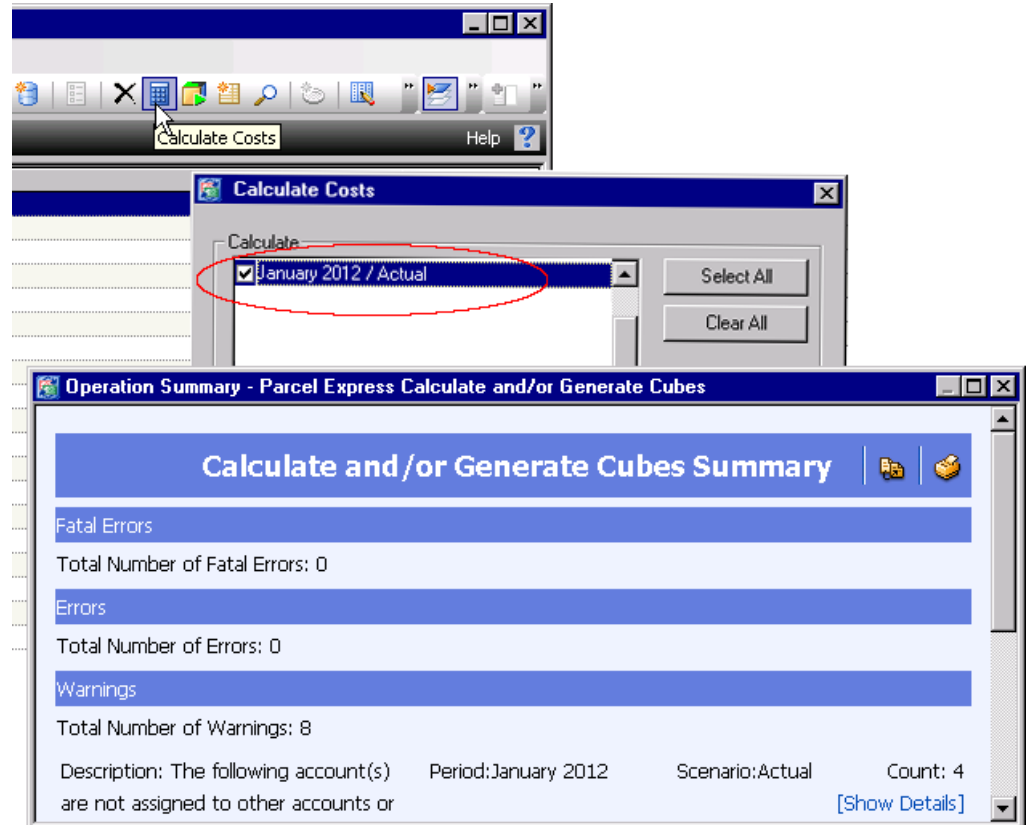




#### 4. Calculate the Model

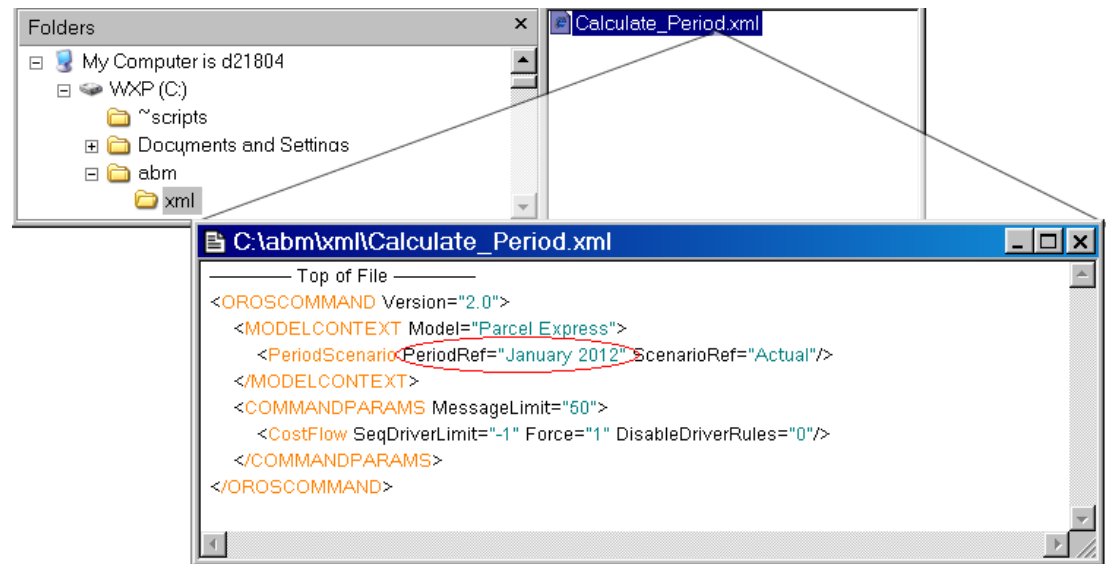
##### **Calculate the Model**

Once the updated tables have been re-imported back into the model, the model must be calculated. Assuming that the other periods have already been calculated, it is necessary to calculate only January.



##### **Modify the Calculate XML**

Looking at the XML that SAS Cost and Profitability Management produced for the calculation, you can see that to use it with Easy API, you only need to modify the XML to specify the period to be calculated.



*Note:* Suppose the calculation produces XML like the following which uses IDs.

```
<OROSCOMMAND Version="2.0">
  <MODELCONTEXT ModelId="1242">
    <PeriodScenario PeriodId="103" ScenarioId="1"/>
  </MODELCONTEXT>
  <COMMANDPARAMS MessageLimit="50">
    <CostFlow SeqDriverLimit="-1" Force="1" DisableDriverRules="0"/>
  </COMMANDPARAMS>
</OROSCOMMAND>
```

Then you can change the XML to use references instead:

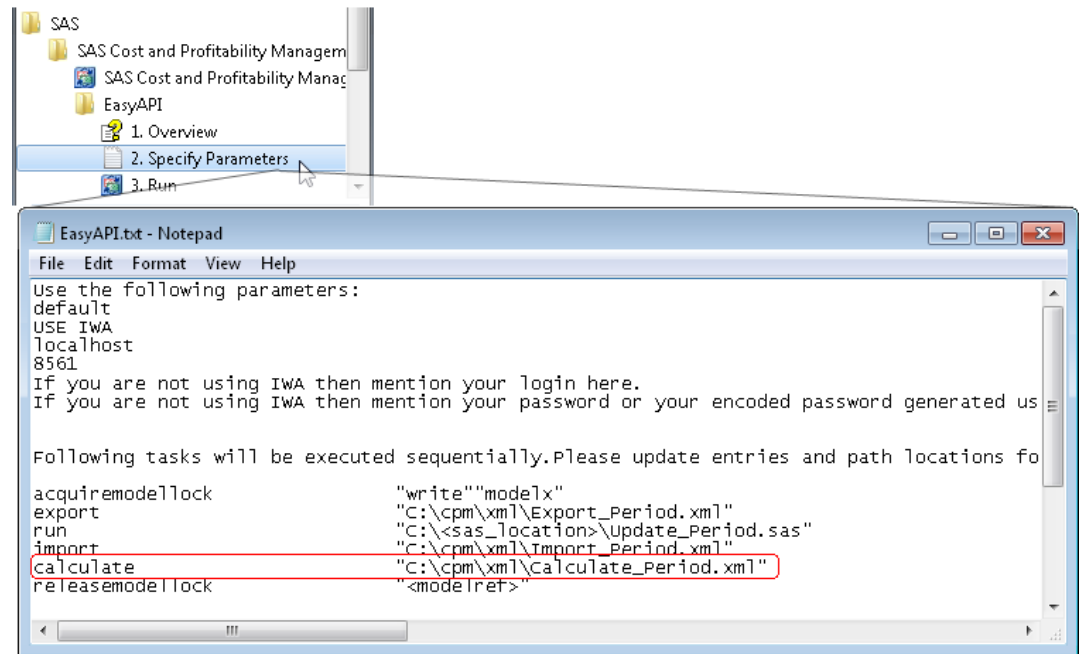
```
<OROSCOMMAND Version="2.0">
  <MODELCONTEXT Model="Parcel Express">
    <PeriodScenario PeriodRef="January 2012" ScenarioRef="Actual"/>
  </MODELCONTEXT>
  <COMMANDPARAMS MessageLimit="50">
    <CostFlow SeqDriverLimit="-1" Force="1" DisableDriverRules="0"/>
  </COMMANDPARAMS>
</OROSCOMMAND>
```

*Note:* Instead of specifying a period to calculate, you can select all periods by specifying **PeriodID="0"**.

### ***Automate the Calculate for Next Time***

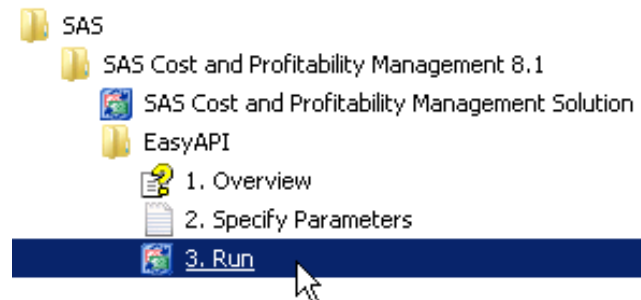
Once again, after modifying the XML file to calculate the period you want, you need to tell Easy API where the file is located. In the following picture, you can see that the following **calculate** command has been inserted:

```
Calculate "c:\cpm\xml\Calculate_Period.xml"
```



## 5. Run Easy API

Now that EasyAPI.txt has been modified and all the XML files are in place, we are ready to run a batch job. To run Easy API, select **Start** ⇒ **Programs** ⇒ **SAS** ⇒ **SAS Cost and Profitability Management 8.1** ⇒ **Easy API** ⇒ **3. Run** from the **Start** menu.



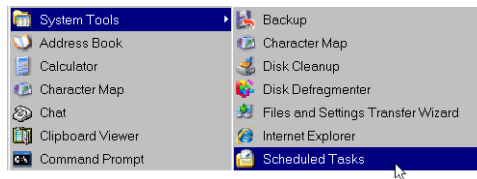
Easy API synchronizes all the operations so that one operation does not begin until the operation before it has completed. So, for example, the program to update exported tables is not launched until the export has completed. And the operation to re-import the updated tables does not begin until the update operation has completed.

Easy API can e-mail you the results of its operations. To receive an e-mail with operation results:

1. Inside SAS Cost and Profitability Management, select **Tools** ⇒ **User Options**.
2. Click the **Easy API Configuration** tab.
3. Specify an **SMTP server** for sending the mail.
4. Specify an **Email Id for successful operation**.
5. Specify an **Email Id for failure of operation**.

**Notes:**

- You can also invoke Easy API by running EasyAPI.exe, which is installed in `<installation directory>\SASCostandProfitabilityManagementClient\8.1\EasyAPI\`
- By supplying a path argument to EasyAPI.exe, you can tell it to use a different txt file for Easy API commands, for example, **EasyAPI.exe "c:\MyPath\EasyAPI2.txt"**. If you don't supply a path argument, then Easy API uses EasyAPI.txt in its installation directory.
- You can use the Microsoft Windows Scheduled Tasks Wizard to schedule EasyAPI.exe to run automatically at selected intervals.



## Part 7

---

# SAS Visual Analytics

<i>Chapter 21</i>	
<b>Introduction</b> .....	267
<i>Chapter 22</i>	
<b>Working with SAS LASR Analytic Server</b> .....	273
<i>Chapter 23</i>	
<b>Sample Reports for SAS Visual Analytics</b> .....	307



## Chapter 21

# Introduction

---

<b>SAS Visual Analytics</b> .....	<b>267</b>
<b>SAS LASR Analytic Server</b> .....	<b>267</b>
<b>The Process in a Nutshell</b> .....	<b>268</b>
Step 1: Establish System-Level Security for SAS Visual Analytics .....	268
Step 2: Give Users Capabilities for SAS Visual Analytics .....	268
Step 3: Create a SAS LASR Analytic Server in Metadata .....	268
Step 4: Create a LASR-Enabled Metadata Library .....	268
Step 5: Authorize Users for the LASR-Enabled Metadata Library .....	269
Step 6: Authorize Users to Manage LASR .....	269
Step 7: Start the SAS LASR Analytic Server .....	270
Step 8: Push Model Data to the LASR-Enabled Metadata Library .....	270
Step 9: Explore Your Data .....	271

---

## SAS Visual Analytics

SAS Visual Analytics provides a highly visual, drag-and-drop interface for exploring data that has been loaded into the SAS LASR Analytic Server. You can use SAS Visual Analytics to create and interact with reports and then to display those reports on a mobile device or on the web. You can explore your data using interactive visualizations such as charts, histograms, and tables. You can add filters and use drag-and-drop functionality to create a well-formatted report with tables, gauges, and graphs.

## SAS LASR Analytic Server

The SAS Visual Analytics Explorer client works on top of the SAS LASR Analytic Server. The SAS LASR Server is an in-memory analytic engine that can load large volumes of data into memory in a grid computing environment. Because each machine is responsible for loading only a section of the data, the loading time is reduced dramatically. Once the data is loaded, the SAS LASR Server distributes the data and workload among multiple machines to perform massively parallel processing. The server can process client requests at extraordinarily high speeds due to the combination of hardware and software that is designed for rapid concurrent access to tables in memory. The server can also be deployed on a single machine when the workload and data volumes do not demand a distributed computing environment.

---

## The Process in a Nutshell

Viewing SAS Cost and Profitability Management data with SAS Visual Analytics and the SAS LASR Analytic Server involves the following steps:

### **Step 1: Establish System-Level Security for SAS Visual Analytics**

There are two levels of security around SAS Visual Analytics:

- System level security is managed during installation of SAS Visual Analytics through the use, for example, of RSA private keys and access to directories. For information, refer to the SAS Visual Analytics deployment and administration documentation at <http://support.sas.com/documentation/onlinedoc/va/index.html>.
- Metadata level security for SAS Cost and Profitability Management users is managed by SAS Management Console through the metadata for users, libraries, and servers. Setting up metadata level security is described in the following steps.

### **Step 2: Give Users Capabilities for SAS Visual Analytics**

See “Give Users Capabilities for SAS Visual Analytics” on page 273 .

### **Step 3: Create a SAS LASR Analytic Server in Metadata**

In addition to a physical machine acting as a SAS LASR Analytic Server, you must create the metadata that describes the machine. You can either:

- “Create a New SAS LASR Analytic Server in Metadata ” on page 275
- “Use an Existing SAS LASR Analytic Server” on page 282

### **Step 4: Create a LASR-Enabled Metadata Library**

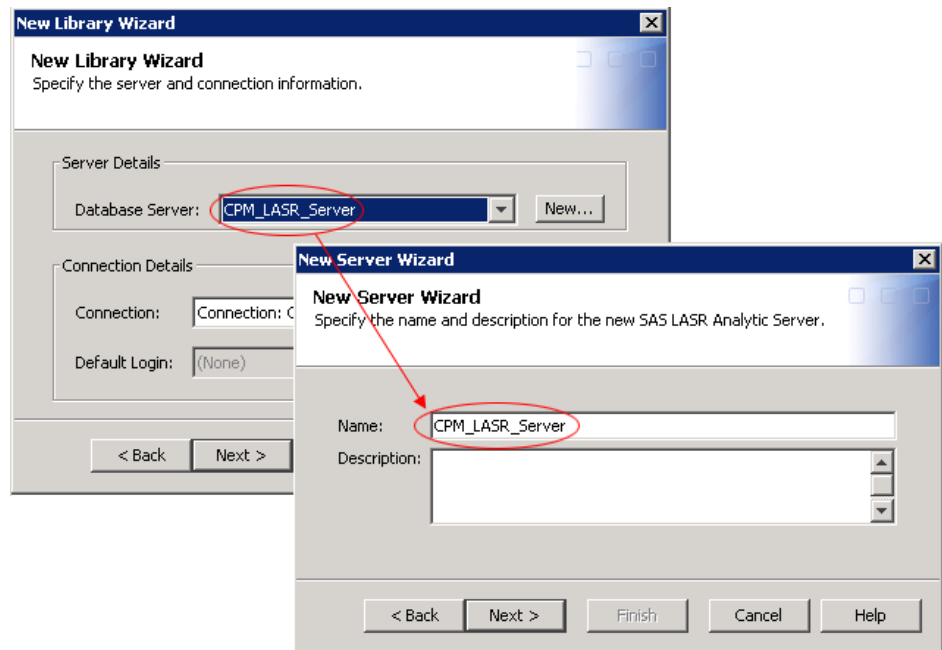
To view SAS Cost and Profitability Management data with SAS Visual Analytics, you export your data to a LASR-enabled metadata library that SAS Visual Analytics can use. You can either:

- “Create a New LASR-Enabled Metadata Library” on page 287
- “Use an Existing SAS LASR Analytic Server” on page 282

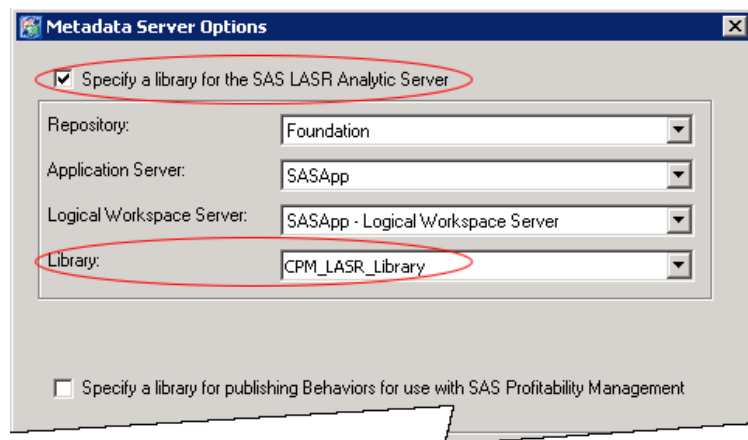
Whether you create a new library or use an existing one, you must do the following:

- Define a connection from this library to the SAS LASR Analytic Server whose metadata you created in the previous step (Step 2: Create a SAS LASR Analytic Server in Metadata). See [Step 6 on page 291](#).





- Identify this library to the SAS Cost and Profitability Management client using the client's Metadata Server Options dialog. See "Metadata Server Options" in Chapter 57 of *SAS Cost and Profitability Management: User's Guide*.



*Note:* A LASR library is displayed for selection only for a user who has Read/Write permission on the library. See [“Authorize Users for the LASR-Enabled Metadata Library”](#) on page 296.

### **Step 5: Authorize Users for the LASR-Enabled Metadata Library**

See [“Authorize Users for the LASR-Enabled Metadata Library”](#) on page 296.

### **Step 6: Authorize Users to Manage LASR**

See [“Authorize Users to Manage LASR”](#) on page 299.

### Step 7: Start the SAS LASR Analytic Server

SAS Visual Analytics works with data that has been loaded into a SAS LASR Analytic server. There are two ways that you can have an instance of SAS LASR Analytic Server to work with data from SAS Cost and Profitability Management:

- You use SAS Visual Analytics to start an instance of a SAS LASR Analytic Server. See “[Start the SAS LASR Analytic Server Using SAS Visual Analytics](#)” on page 302.
- You can use Base SAS to start an instance of a SAS LASR Analytic Server. The following is a sample command to start an instance of the SAS LASR Analytic Server on port=12345 with server=localhost,

```
LIBNAME VALIBLA SASIOLA start port=12345 Server=localhost;
```

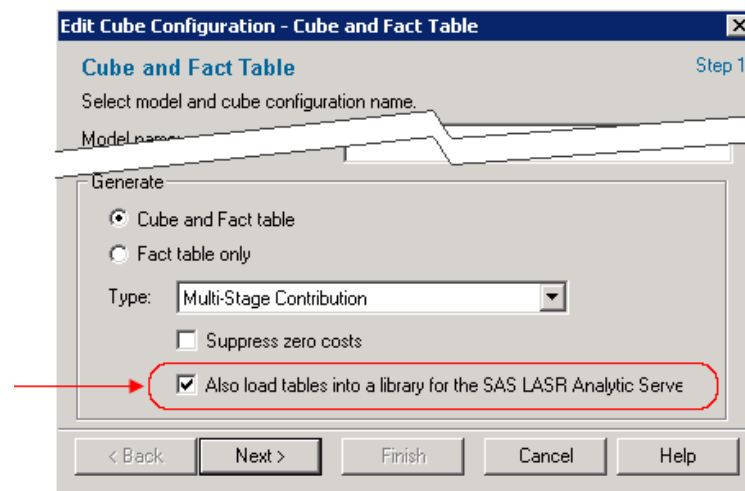
Please refer to Base SAS documentation, for example the documentation on support.sas.com at <http://support.sas.com/documentation/onlinedoc/base/index.html>.

### Step 8: Push Model Data to the LASR-Enabled Metadata Library

Using the SAS Cost and Profitability Management client, you use the following processes to push data into a LASR-enabled metadata library that is accessible by a SAS LASR Analytic Server for viewing with SAS Visual Analytics:

#### Generate cube or fact table data

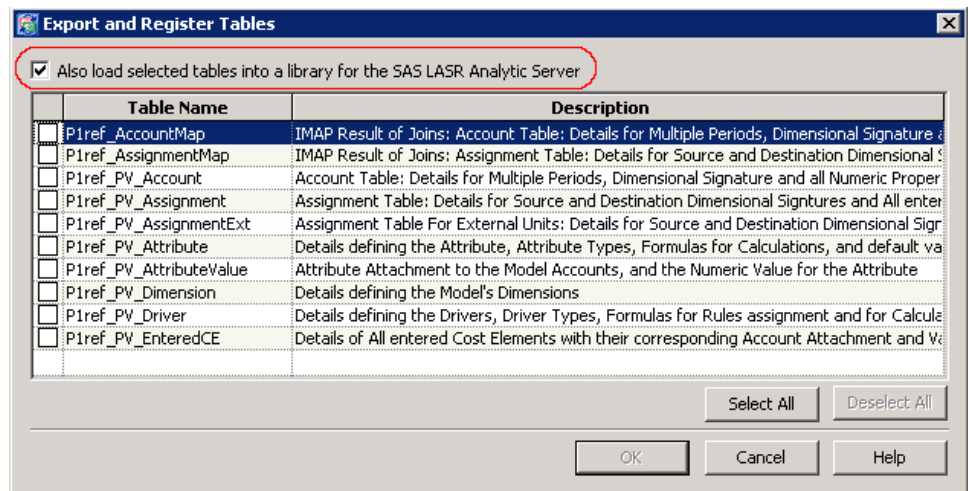
If you selected, in your cube configuration, the option to **Also load tables into a library for the SAS LASR Analytic Server**, then when you generate a cube or fact table, corresponding tables are automatically loaded into the specified LASR-enabled metadata library. See “Cube Configuration: Select a Model and General Options” in Chapter 58 of *SAS Cost and Profitability Management: User’s Guide*.



#### Export and register tables

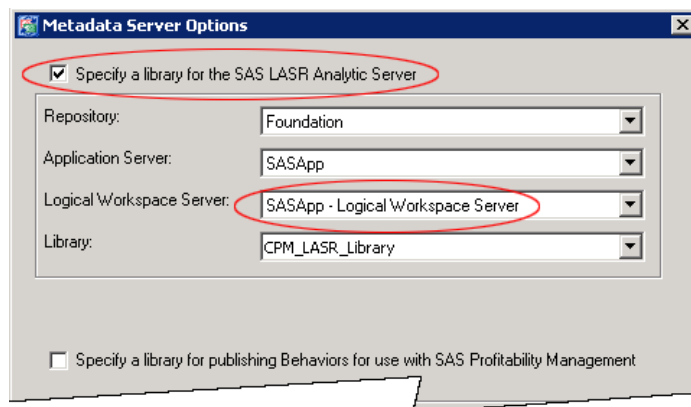
From the SAS Cost and Profitability Management client, select **Model** ⇒ **Export and Register Tables** to export tables to the LASR-enabled metadata library.

Make sure to select **Also load selected tables into a library for the SAS LASR Analytic Server**. See Chapter 14, “[Working with Registered Tables](#),” on page 145.



Note:

- For the option **Also load selected tables into a library for the SAS LASR Analytic Server** to be available, you must have specified this library on the Metadata Server Options dialog. See “Metadata Server Options” in Chapter 57 of *SAS Cost and Profitability Management: User’s Guide*.
- On the Metadata Server Options dialog, make sure that you have selected **SASApp – Logical Workspace Server** as shown in the picture below.



- You must also have authorization to write to the folder that is enabled for library. If you do not have those rights, the option is disabled. See “[Authorize Users for the LASR-Enabled Metadata Library](#)” on page 296.

## Step 9: Explore Your Data

See “[Explore the Data](#)” on page 305.



## Chapter 22

# Working with SAS LASR Analytic Server

---

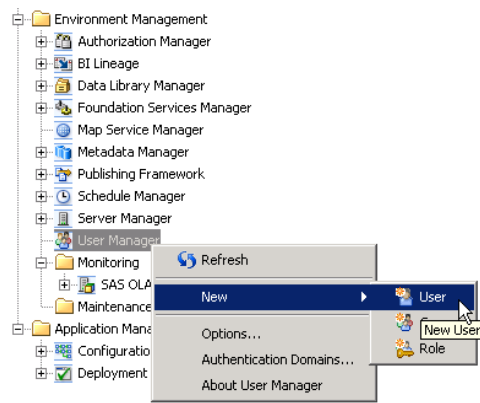
<b>Give Users Capabilities for SAS Visual Analytics</b> .....	<b>273</b>
<b>Create a New SAS LASR Analytic Server in Metadata</b> .....	<b>275</b>
<b>Use an Existing SAS LASR Analytic Server</b> .....	<b>282</b>
<b>Create a New LASR-Enabled Metadata Library</b> .....	<b>287</b>
<b>Use an Existing LASR-Enabled Metadata Library</b> .....	<b>294</b>
<b>Determine the Port Number of a LASR Server</b> .....	<b>295</b>
<b>Authorize Users for the LASR-Enabled Metadata Library</b> .....	<b>296</b>
<b>Authorize Users to Manage LASR</b> .....	<b>299</b>
Overview .....	299
Visual Analytics Data Administrators .....	300
Visual Analytics Users .....	301
Visual Data Builder Administrators .....	301
<b>Start the SAS LASR Analytic Server Using SAS Visual Analytics</b> .....	<b>302</b>
<b>Explore the Data</b> .....	<b>305</b>

---

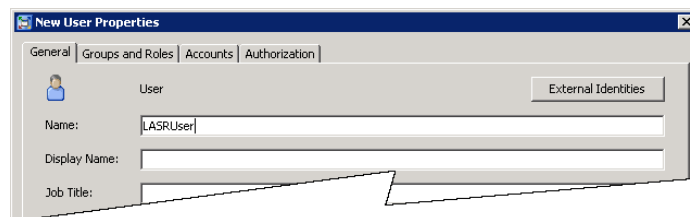
## Give Users Capabilities for SAS Visual Analytics

To work with SAS Visual Analytics and the SAS LASR Analytic Server, a user must have the appropriate capabilities in virtue of either:

- inheriting those capabilities by belonging to a group with those capabilities, or
  - being granted a role that includes those capabilities.
1. Open SAS Management Console, right click User Manager, and select **New** ⇒ **User**; or you can assign groups and roles to an existing user.



2. Name the new user (assuming that you are creating a new user rather than assigning groups and roles to an existing user).



3. Click the **Groups and Roles** tab, and make the user a member of the following groups:

Visual Analytics Data Administrators

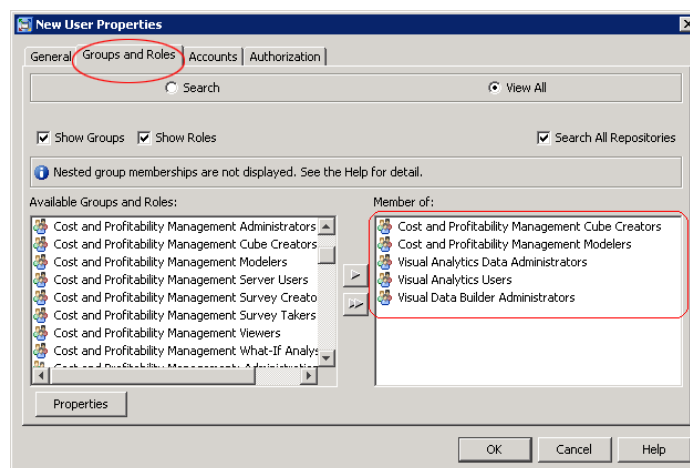
Visual Analytics Users

Visual Data Builder Administrators

And, make the user a member of Cost and Profitability Management groups as desired. The picture below shows making the user a member of:

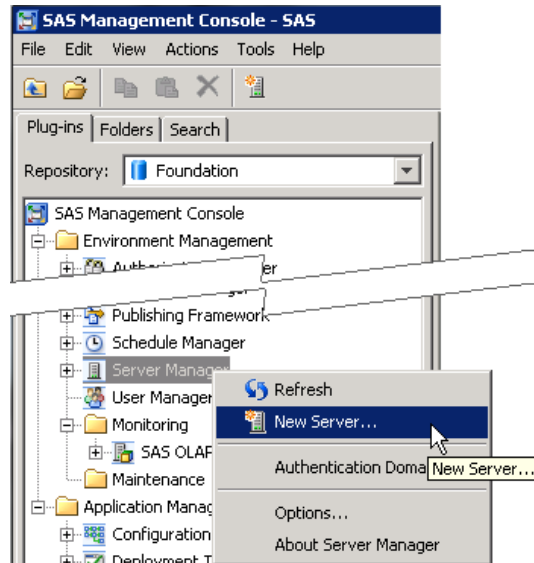
Cost and Profitability Management Cube Creators

Cost and Profitability Management Modelers

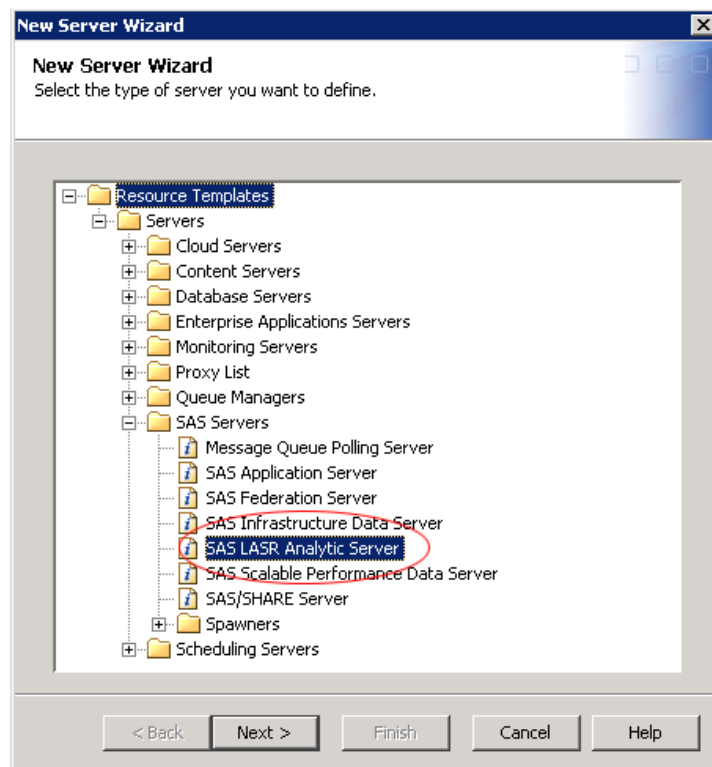


## Create a New SAS LASR Analytic Server in Metadata

1. In the **Plug-ins** tab of SAS Management Console, right-click **Environment Management** ⇒ **Server Manager** and select **New Server**.



2. Select **SAS LASR Analytic Server** under **SAS Servers**.



3. Name the new server, and then click **Next**.

You can choose any name that you want. Later, when you create a LASR-enabled library, you will use this name to establish a connection from that library to this server. (Step 6 on page 291)

4. Next, specify whether the SAS LASR Analytic Server is on a single, local machine or on a remote machine.

Specify the following server properties:

#### Single machine server

Select **Yes** if the SAS LASR Analytic Server installation is contained on a single machine.

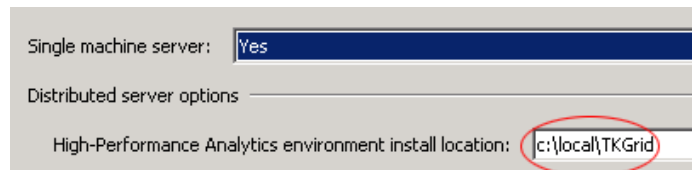
Select **No** if the SAS LASR Analytic Server is installed on a remote machine.

#### High-Performance Analytics environment install location

If you selected **Yes** for **Single machine server**, then:

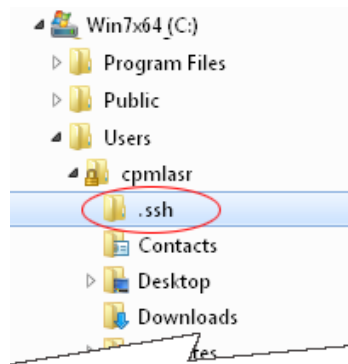
- specify an absolute path such as **C:\local\TKGrid**.
- Make sure that the path exists on that machine and that users have permission to access it..





If you selected **No** for **Single machine server**, then:

- Specify a relative path such as **/local/TKGrid**.
- Make sure that you have copied a **.ssh** folder to the Windows machine on which the SAS Cost and Profitability client software is installed. Copy the **.ssh** folder under the **Users** folder of any user that is to access the SAS LASR Analytic Server. The **.ssh** folder supports passwordless secure shell (SSH).

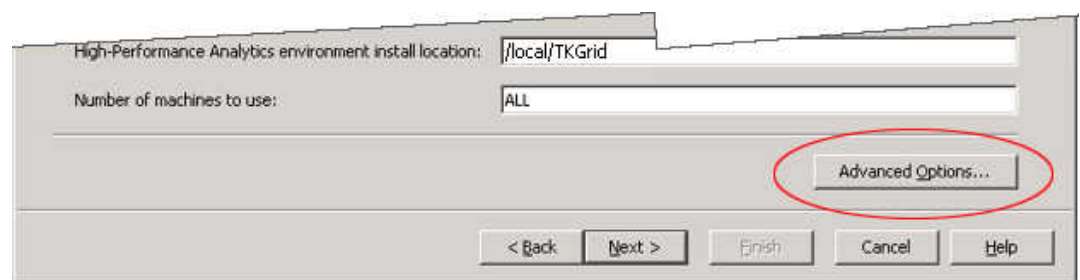


For information, refer to the SAS Visual Analytics deployment and administration documentation at <http://support.sas.com/documentation/onlinedoc/va/index.html>.

#### Number of machines to use

specifies the number of machines in the cluster to use for the server. Specify **ALL** to use all of the machines in the cluster.

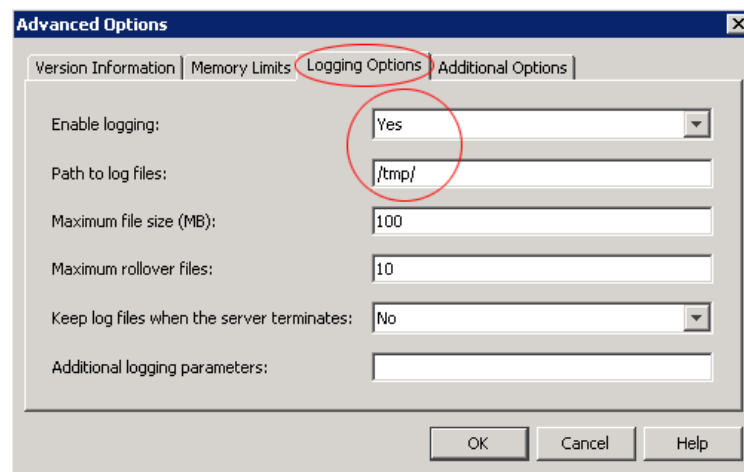
#### 5. Click **Advanced Options**.



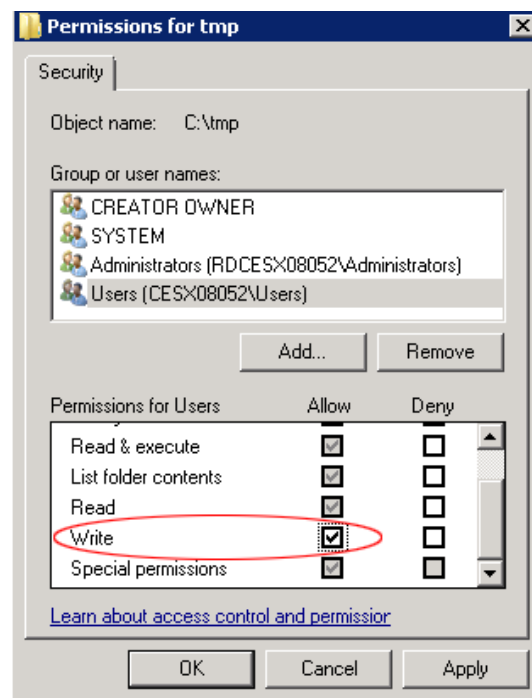
The Advanced Options window opens.

#### a. Click the **Logging Options** tab.

*Note:* This step is optional.



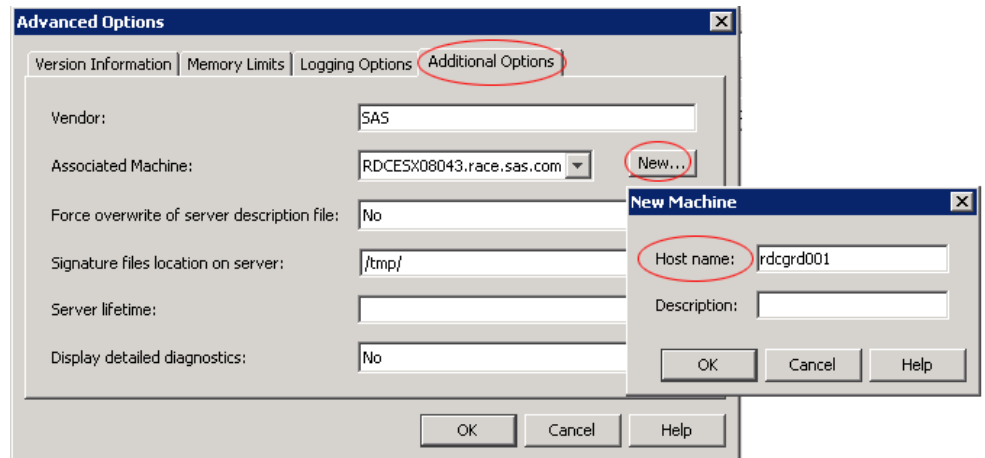
By default, **Enable logging** is set to No. If you want to enable logging, then make sure that the path that you specify for **Path to log files** is a directory that SAS Cost and Profitability Management users can write to.



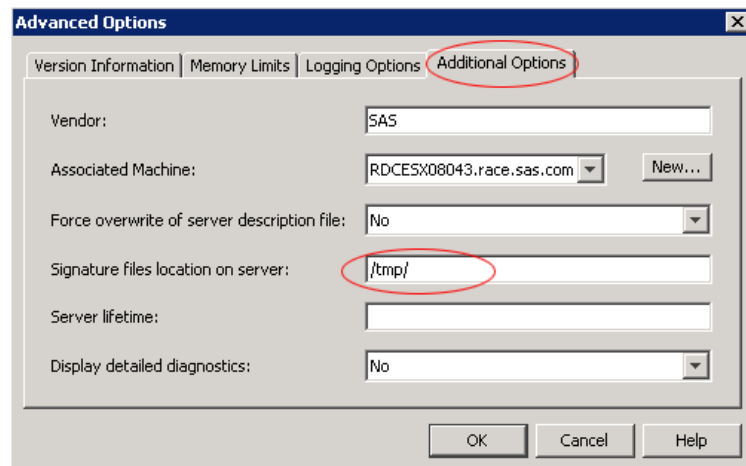
If the directory is on a local machine, then you can specify an absolute path such as **C:\tmp**. If the directory is on a remote machine, then you can specify a relative path such as **/tmp/**.

- b. Click the **Additional Options** tab.
  - i. Click **New** and provide the **Host name** of the LASR server.

*Note:* This step is not necessary for a LASR server installation on a single machine.



- ii. Make sure that the path that you specify for **Signature files location on server** is a directory that SAS Cost and Profitability Management users can write to. If the directory is on a local machine, then you can specify an absolute path such as **C:\tmp**. If the directory is on a remote machine, then you can specify a relative path such as **/tmp/**.



- c. Click **OK** to close the Advanced Options window.
- d. Click **Next** to go to the next New Server Wizard page.
6. On the next New Server Wizard page, Specify the following information:

#### Port number

Specify the port number that the SAS LASR Analytic Server uses to listen. See [“Determine the Port Number of a LASR Server” on page 295](#).

#### High-Performance Analytics environment host

specifies the location of the host machine in the SAS High-Performance Analytics environment.

#### Use LASR authorization service

specifies that the SAS LASR Authorization web service should be used and lists the URI for the service. The web service is provided by the SAS Visual Analytics software.

**New Server Wizard**  
Enter the connection properties.

Port number:

High-Performance Analytics environment host:

LASR authorization service location

☒ Use LASR authorization service:

[Advanced Options...](#)

< Back   Next >   Finish   Cancel   Help

Then, click **Next**.

7. Select the users and groups that have the capability to use the SAS LASR Analytic Server. You can select, for example, the following:
  - Cost and Profitability Management Modelers
  - Cost and Profitability Management Users
  - Cost and Profitability Management Administrators

**New Server Wizard**  
Set metadata-layer permissions.

Move identities that will start this LASR server to the Selected list.  
Each identity in the Selected list will be explicitly granted the Administer permission on this LASR server.

☐ Show users   ☒ Show groups

**Available:**

- BI Dashboard Administrators
- BI Dashboard Users
- BI Web Services Users
- Cost and Profitability Management Administrators
- Cost and Profitability Management Modelers
- Cost and Profitability Management Survey Takers
- Cost and Profitability Management Viewers
- Logical ABM Server Users
- PUBLIC
- SAS Administrators
- SAS General Servers
- SAS System Services
- SASUSERS
- Visual Analytics Users
- Cost and Profitability Management Cube Creators

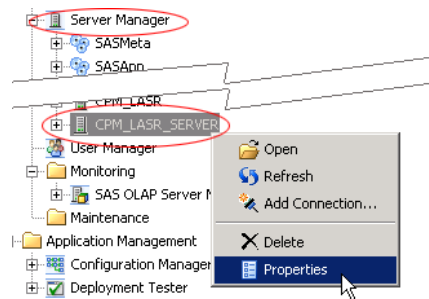
**Selected:**

- Visual Analytics Data Administrators
- Cost and Profitability Management Modelers**
- Cost and Profitability Management Users
- Cost and Profitability Management Administrator

< Back   Next >   Finish   Cancel   Help

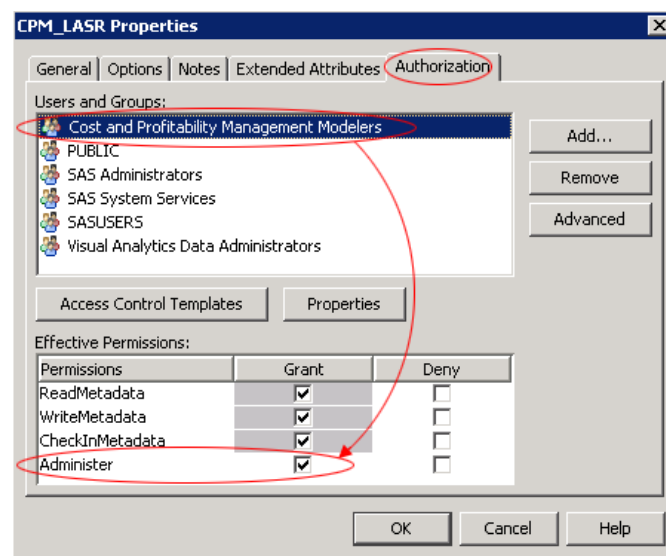
As it says on this dialog, "Each identity in the Selected list will be explicitly granted the Administer permission on this LASR server". Once the LASR server has been created, you can verify that the selected users or groups (for example, Cost and Profitability Management Modelers) has been granted Administer permission by doing the following:

- a. Right-click the newly created server under **Server Manager**, and select **Properties**.



- b. Click the **Authorization** tab and click a group, such as Cost and Profitability Management Modelers.

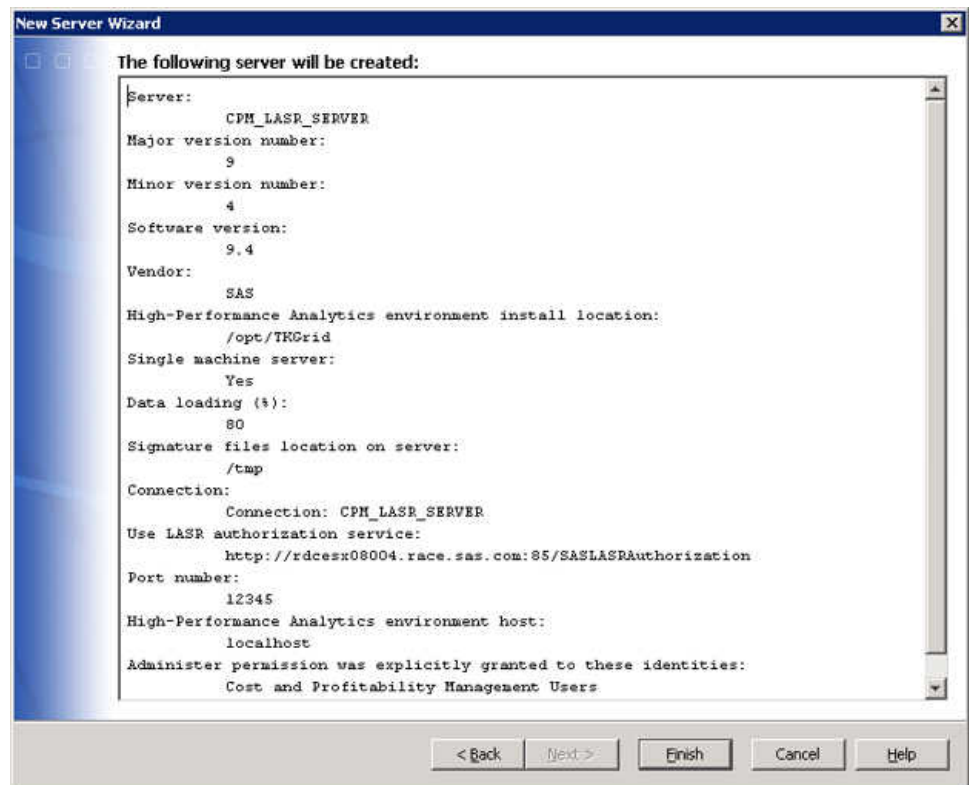
You can see that the Administer permission is granted to the group.



*Note:* You must also explicitly grant capabilities to individual users and groups. See [“Give Users Capabilities for SAS Visual Analytics” on page 273](#).

Click **Next**.

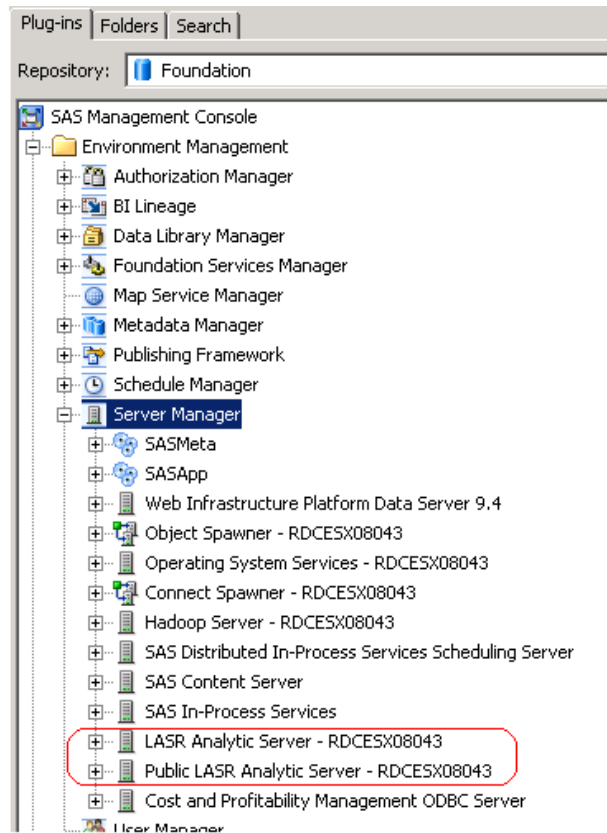
8. The summary page appears. If you are satisfied with your choices, then click **Finish**.



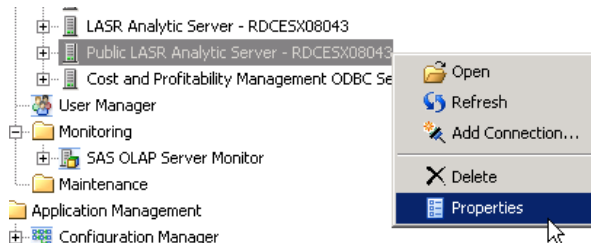
## Use an Existing SAS LASR Analytic Server

You can configure any existing SAS LASR Analytic server to receive your SAS Cost and Profitability Management data.

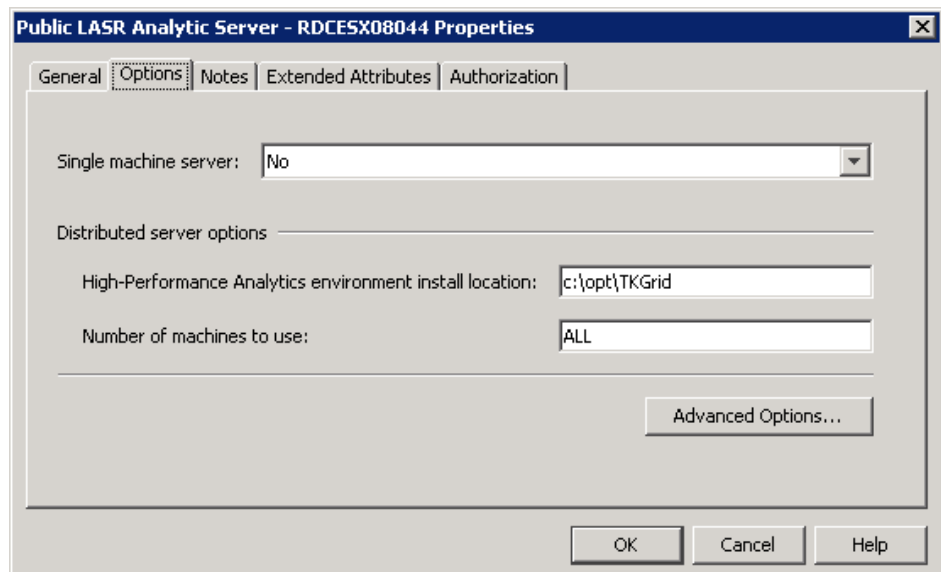
1. Look for SAS LASR Analytic Servers under the **Server Manager** folder of the **Plug-ins** tab of SAS Management Console.



2. Right-click a SAS LASR Analytic Server, and select **Properties**.



3. Click the **Options** tab.



Specify the following server properties:

#### Single machine server

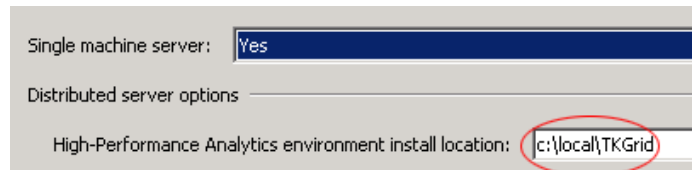
Select **Yes** if the SAS LASR Analytic Server installation is contained on a single machine.

Select **No** if the SAS LASR Analytic Server is installed on a remote machine.

#### High-Performance Analytics environment install location

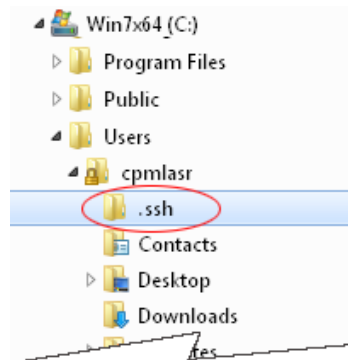
If you selected **Yes** for **Single machine server**, then:

- specify an absolute path such as **C:\local\TKGrid**.
- Make sure that the path exists on that machine and that users have permission to access it..



If you selected **No** for **Single machine server**, then:

- Specify a relative path such as **/local/TKGrid**.
- Make sure that you have copied a **.ssh** folder to the Windows machine on which the SAS Cost and Profitability client software is installed. Copy the **.ssh** folder under the **Users** folder of any user that is to access the SAS LASR Analytic Server. The **.ssh** folder supports passwordless secure shell (SSH).



For information, refer to the SAS Visual Analytics deployment and administration documentation at <http://support.sas.com/documentation/onlinedoc/va/index.html>.

#### Number of machines to use

specifies the number of machines in the cluster to use for the server. Specify **ALL** to use all of the machines in the cluster.

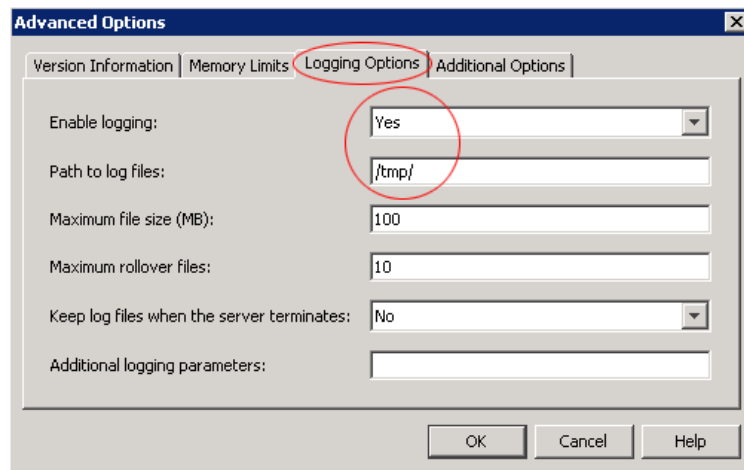
4. On the **Options** tab, select **Advanced Options**.

The Advanced Options window opens.

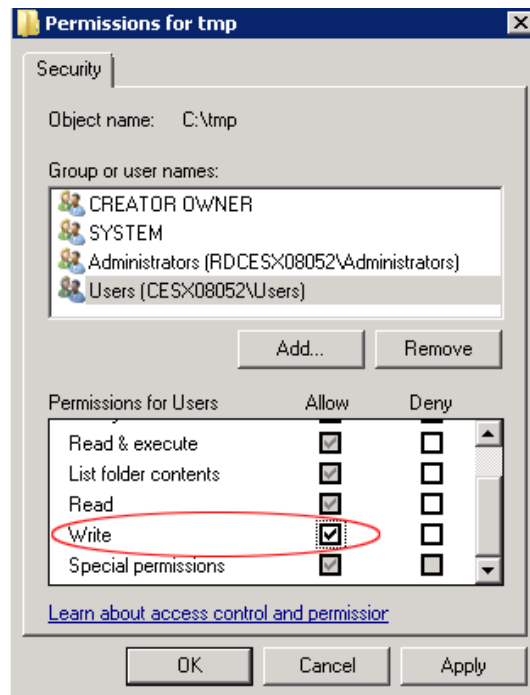
- a. Click the **Logging Options** tab.

*Note:* This step is optional.





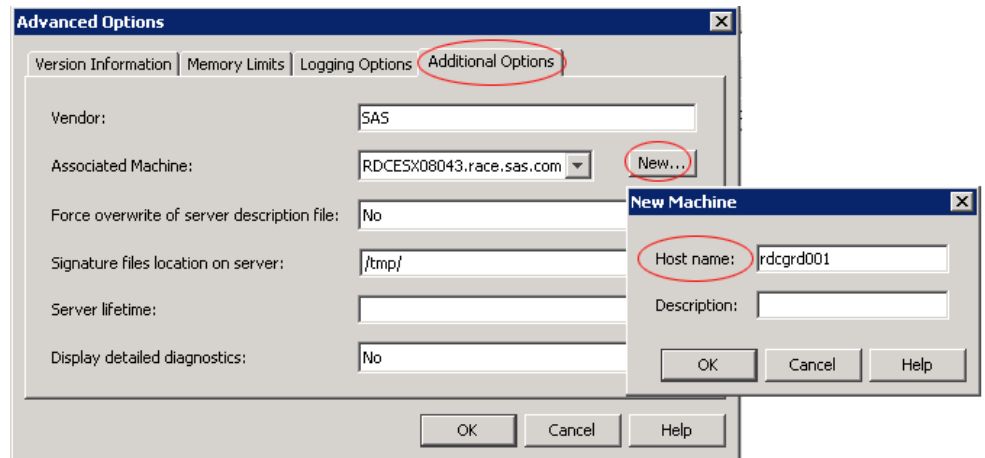
By default, **Enable logging** is set to No. If you want to enable logging, then make sure that the path that you specify for **Path to log files** is a directory that SAS Cost and Profitability Management users can write to.



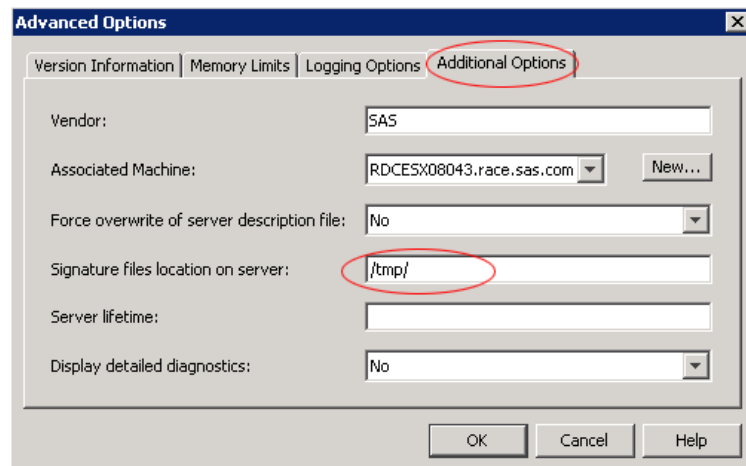
If the directory is on a local machine, then you can specify an absolute path such as **C:\tmp**. If the directory is on a remote machine, then you can specify a relative path such as **/tmp/**.

- b. Click the **Additional Options** tab.
  - i. Click **New** and provide the **Host name** of the LASR server.

*Note:* This step is not necessary for a LASR server installation on a single machine.



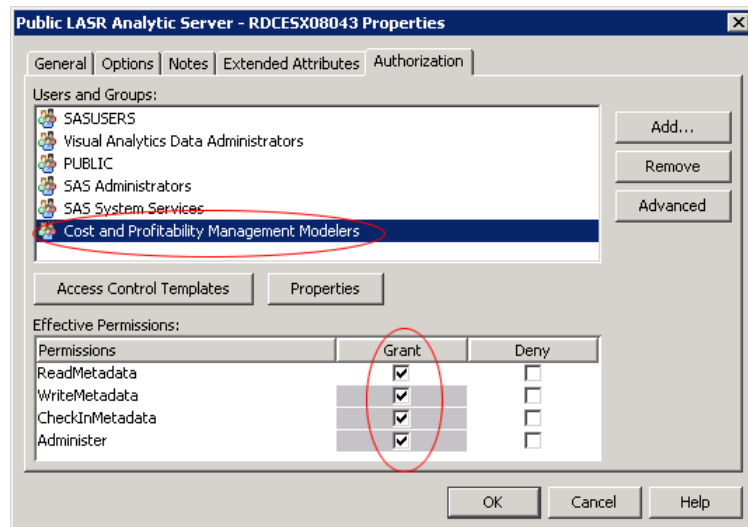
- ii. Make sure that the path that you specify for **Signature files location on server** is a directory that SAS Cost and Profitability Management users can write to. If the directory is on a local machine, then you can specify an absolute path such as **C:\tmp**. If the directory is on a remote machine, then you can specify a relative path such as **/tmp/**.



- c. Click **OK** to close the Advanced Options window.
  - d. Click **Next** to go to the next New Server Wizard page.
5. Click the **Authorizations** tab.

Add the group of users that is to have access to this server such as SAS Cost and Profitability Management Modelers and grant them the following permissions:

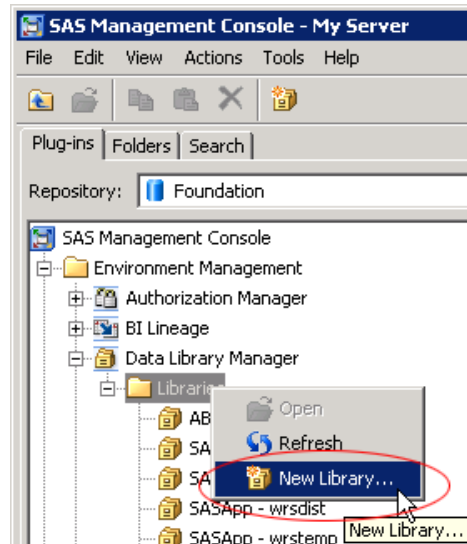
- ReadMetadata
- WriteMetadata
- CheckInMetadata (optional)
- Administer



## Create a New LASR-Enabled Metadata Library

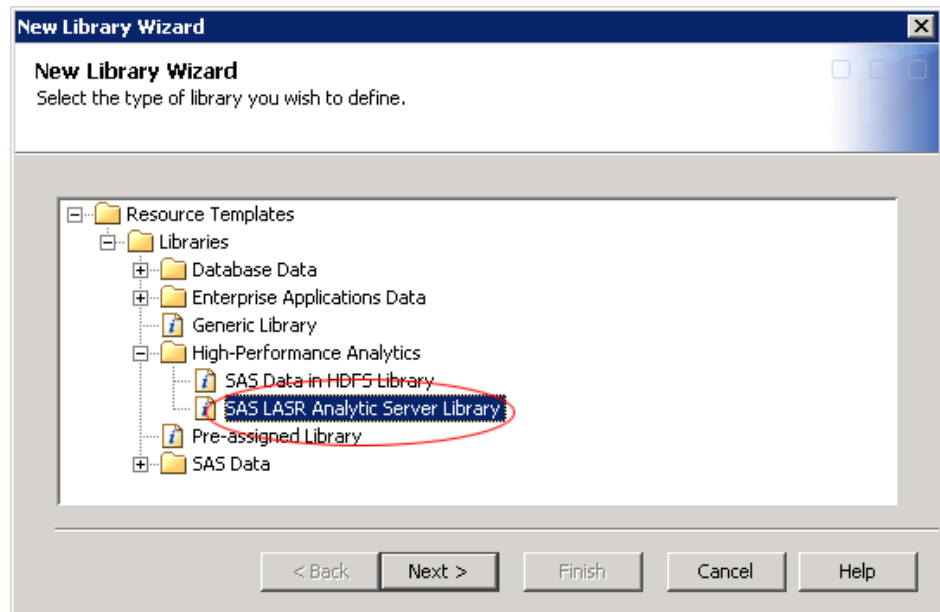
This section describes the steps for creating a metadata library that is enabled for the SAS LASR Analytic Server.

1. In the **Plug-ins** tab of SAS Management Console, right-click **Data Library Manager** ⇒ **Libraries** and select **New Library**.



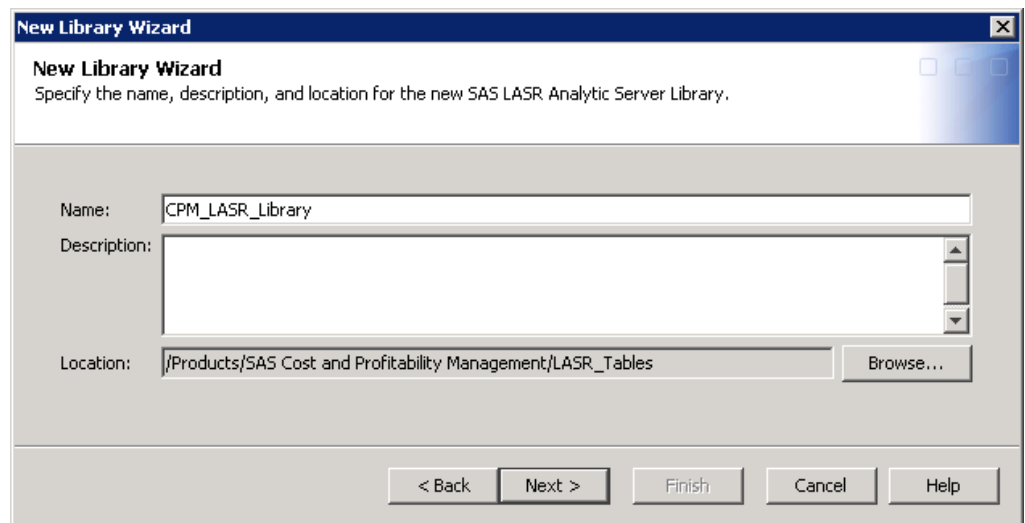
The New Library Wizard opens.

2. Select **SAS LASR Analytic Server Library** under **High-Performance Analytics**.



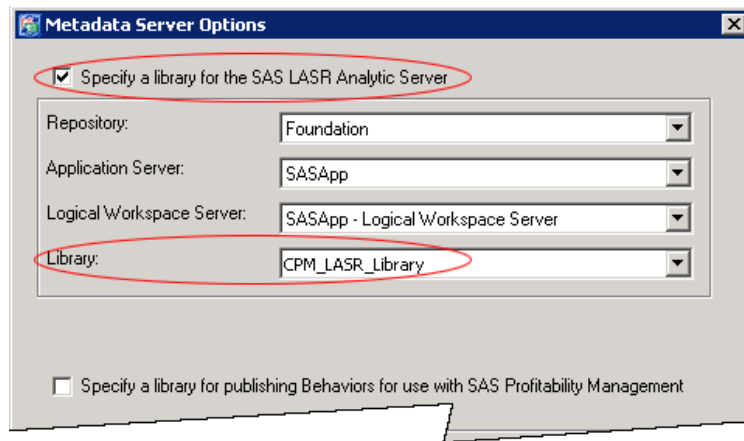
Click **Next**.

3. Name the new library and select the location of the folder in which the library definition is stored.



### Name

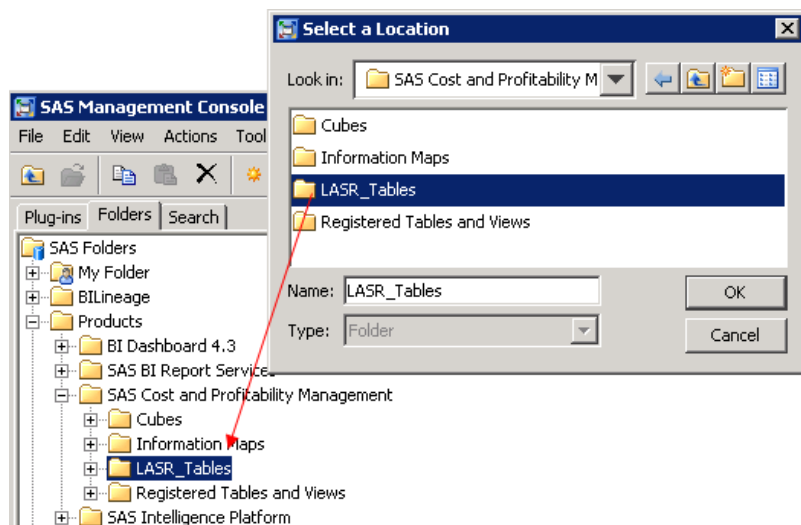
You use this name to identify the library to the SAS Cost and Profitability Management client with the client's Metadata Server Options dialog. See "Metadata Server Options" in Chapter 57 of *SAS Cost and Profitability Management: User's Guide*.



### Location

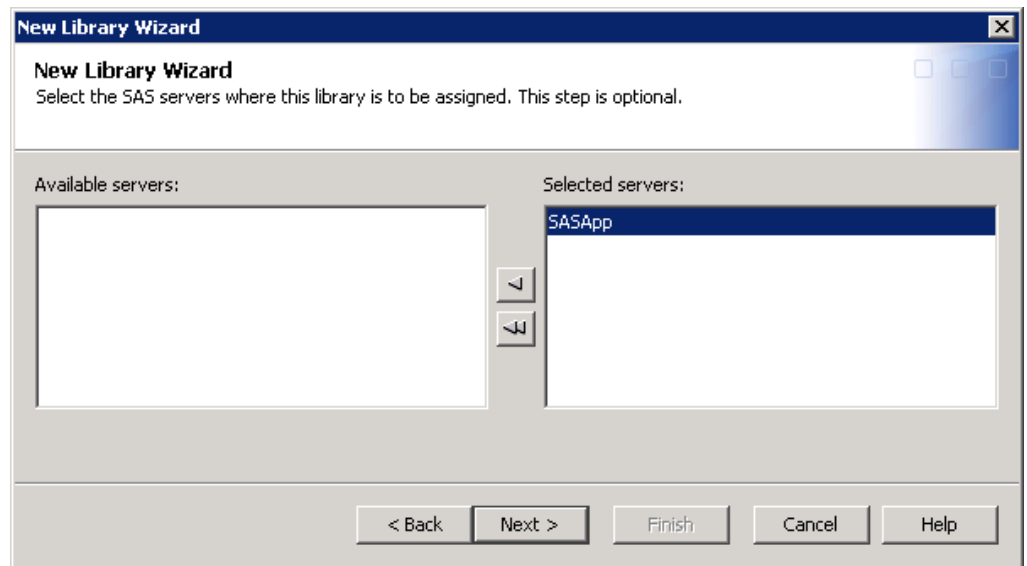
In selecting a location, you can select any folder for which SAS Cost and Profitability Management users have permissions. See [“Authorize Users for the LASR-Enabled Metadata Library”](#) on page 296.

In the picture below, the **LASR\_Tables** folder is selected under **SAS Cost And Profitability Management**.



Click **Next**.

4. Select **SASApp** as the server that will have access to the new library.



Click **Next**.

5. Specify the following information:

**Libref**

specifies the one- to eight-character name of the SAS library. The first character must be a letter, and all other characters can be either letters or numbers.

If you specify a libref that has already been assigned outside of SAS Management Console, you must also specify that the library is preassigned. Preassigned libraries include system libraries assigned by SAS and libraries that have already been assigned by a user. Click **Advanced Options** and select **Library is pre-assigned** in the **Pre-assign** tab. If you do not select this option, any queries that use this libref fail.

**Engine**

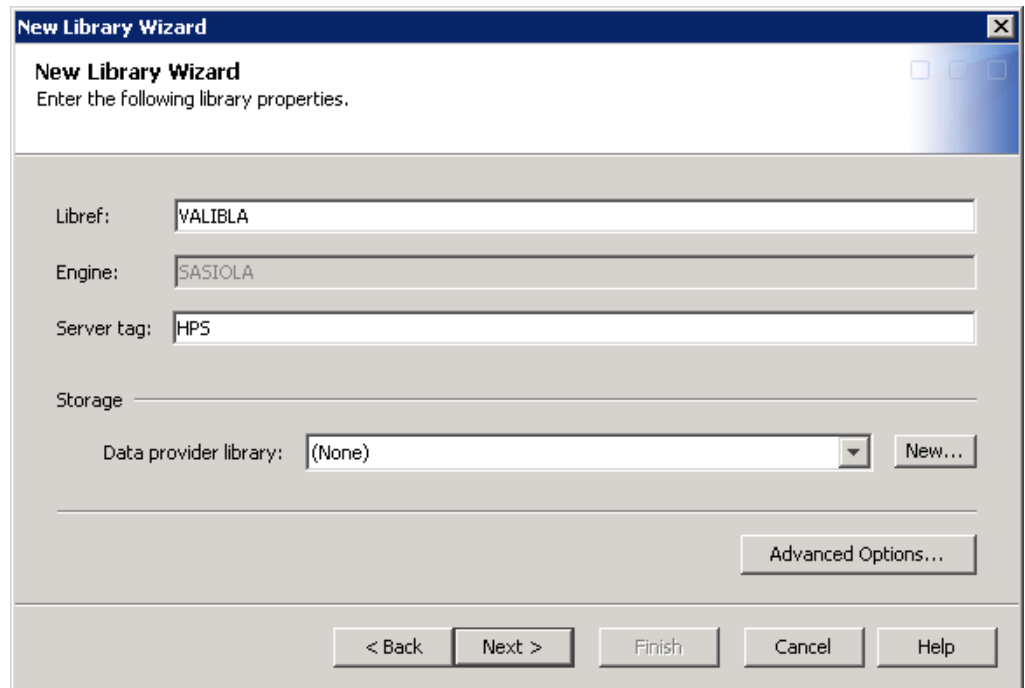
specifies the engine to use when accessing files in this library. The value is set to **SASIOLA** and cannot be changed.

**Server Tag**

specifies the tables that are associated with the library. This field corresponds to the TAG= option in the LIBNAME statement. If this field is not specified, a tag of WORK is used.

**Data Provider Library**

specifies the library from which LASR data can be reloaded. Only SAS BASE libraries can be selected.



**New Library Wizard**

Enter the following library properties.

Libref: VALIBLA

Engine: SASIOLA

Server tag: HPS

Storage

Data provider library: (None) New...

Advanced Options...

< Back Next > Finish Cancel Help

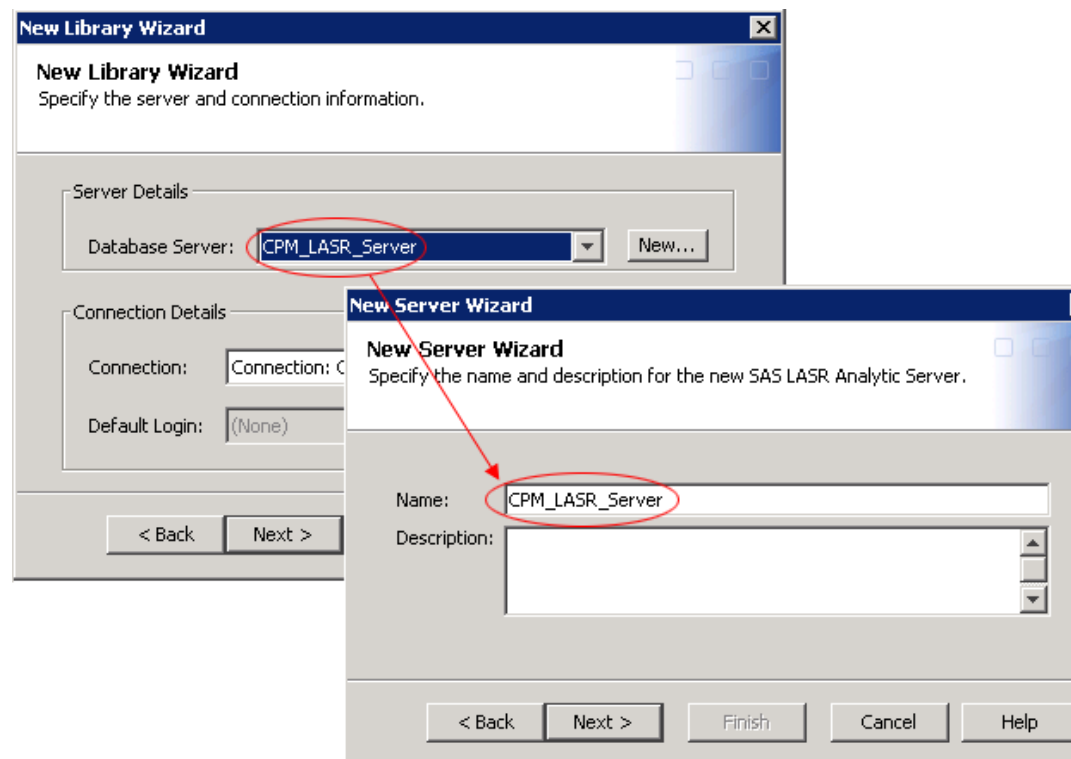
Click **Next**.

6. Specify the following information:

**Database Server**

specifies the server that contains the databases for the library you are defining.

The server must be one that you created using the New Server wizard (see [Step 3 on page 275](#)). Only servers corresponding to the selected library type are listed.



**New Library Wizard**

Specify the server and connection information.

Server Details

Database Server: CPM\_LASR\_Server New...

Connection Details

Connection: Connection: C

Default Login: (None)

< Back Next >

**New Server Wizard**

Specify the name and description for the new SAS LASR Analytic Server.

Name: CPM\_LASR\_Server

Description:

< Back Next > Finish Cancel Help

**Connection**

specifies the connection to the SAS LASR Analytic Server previously created.  
([Step 3 on page 275](#))

**Default Login**

specifies the user login that provides credentials for logging into the selected server using the specified connection. The pull-down menu on this field lets you select from the logins defined for the connection to the specified server. If logins have been defined, the menu on this field lets you select from the three most recently used logins or the **More Logins** option.

**New Library Wizard**  
Specify the server and connection information.

**Server Details**

Database Server: **CPM\_LASR\_SERVER** New...

**Connection Details**

Connection: **Connection: CPM\_LASR\_SERVER** New...

Default Login: **(None)**

< Back Next > Finish Cancel Help

Click **Next**.

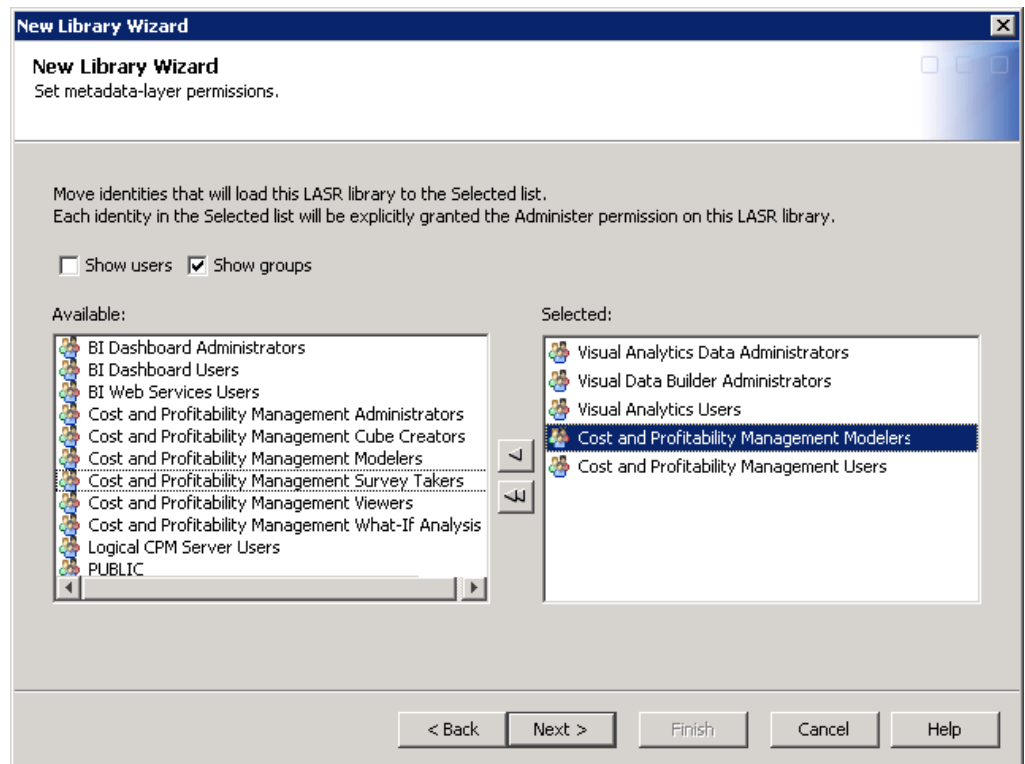
7. Select the users and groups that have capability to use the LASR library. The following two groups are automatically assigned:

- Visual Analytics Data Administrators
- Visual Data Builder Administrators

You can add the following groups:

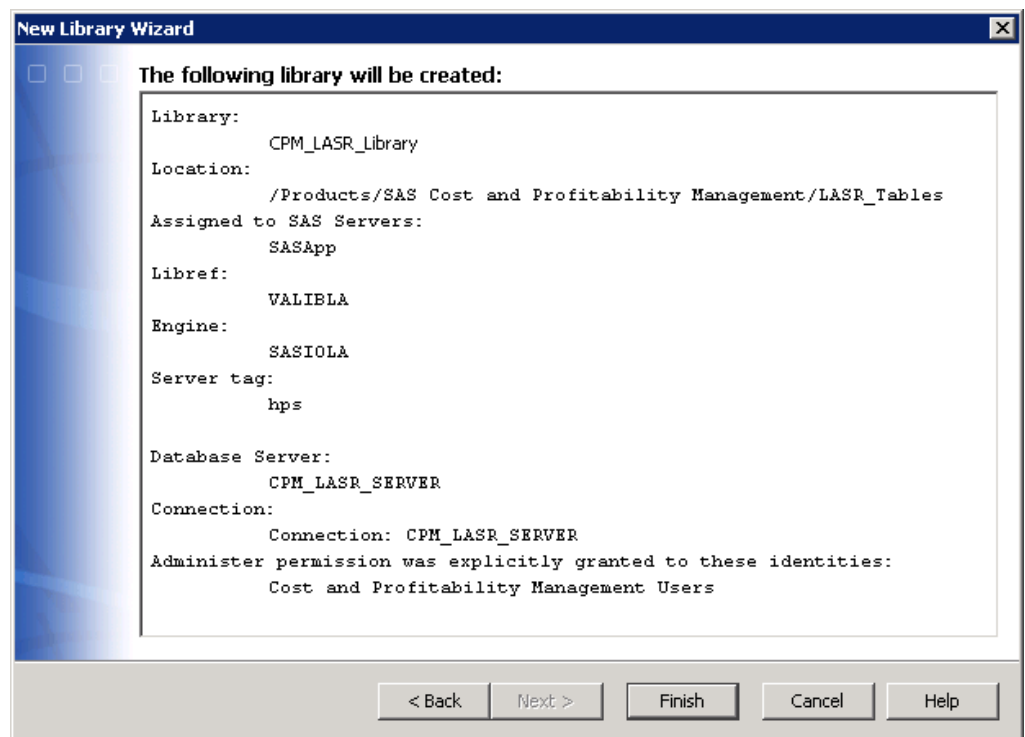
- Visual Analytics Users
- Cost and Profitability Management groups as desired, such as Cost and Profitability Management Modelers





Click **Next**.

8. Review your choices and click **Finish** if you are satisfied.



9. Give users authorization to use the library.

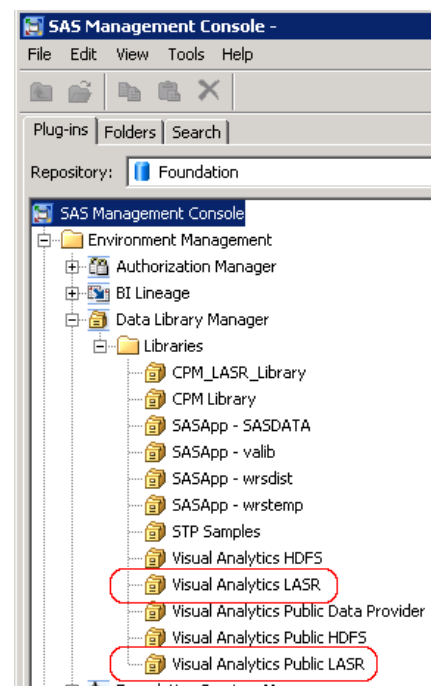
Once you have created a LASR-enabled metadata library, you must give SAS Cost and Profitability Management users authorization to use the library so that they can

export and register tables to the library. See “[Authorize Users for the LASR-Enabled Metadata Library](#)” on page 296.

## Use an Existing LASR-Enabled Metadata Library

Instead of creating a new library, you can select an existing LASR-enabled library. These libraries point to a SAS LASR Analytic Server instance which should have been already started.

You must explicitly authorize SAS Cost and Profitability Management users to work with this metadata library. See “[Authorize Users for the LASR-Enabled Metadata Library](#)” on page 296.



The SAS LASR Analytic Server instance can be either an SMP (symmetric multi-processing) installation or an MPP installation. An SMP LASR Analytic Server runs in a non-distributed environment, it does not involve multiple machines. An MPP LASR Analytic Server runs in a distributed computing environment in which the various server-side boxes are nodes in a grid. For more information, refer to support.sas.com, for example the documentation on [http://support.sas.com/documentation/onlinedoc/securedoc/index\\_lasrserver.html](http://support.sas.com/documentation/onlinedoc/securedoc/index_lasrserver.html).

### See Also

Chapter 82, “Export Registered Tables,” in *SAS Cost and Profitability Management: User’s Guide*

## Determine the Port Number of a LASR Server

The following is sample SAS code to determine the port number of a SAS LASR Analytic Server.

```
proc lasr create path="/tmp/";
           performance host="rdcgrd001"
                     install="/local/TKGrid"
                     nodes=all;

run;
```

In this sample code:

**path="/tmp"**

This is the **Path to log files**. See [Step 4a on page 284](#).

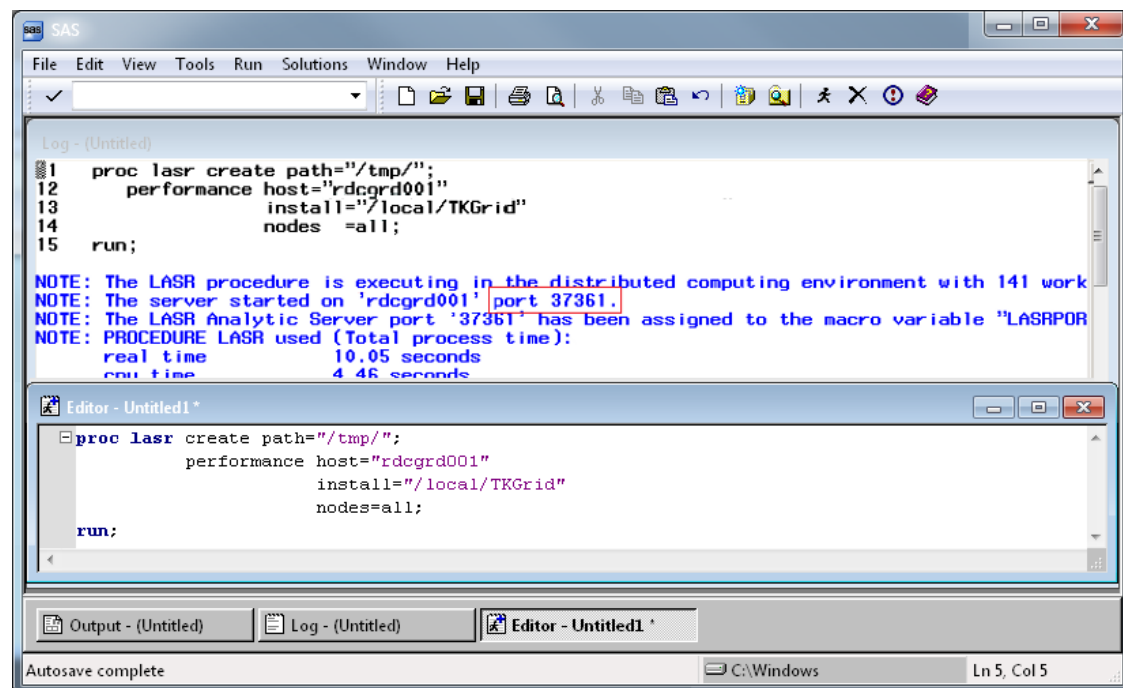
**host="rdcgrd001"**

This is the **Host name**. See [Step 4b.i on page 285](#).

**install="/local/TKGrid"**

This is the **High-Performance Analytics environment install location**. See [Step 4 on page 276](#).

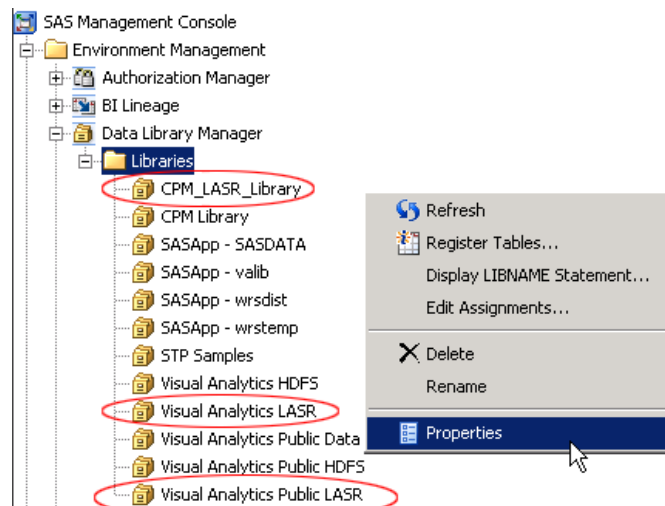
The following picture shows an example of running the sample code in Base SAS. You can see that the port number of the SAS Analytic LASR Server is displayed in the Log.



## Authorize Users for the LASR-Enabled Metadata Library

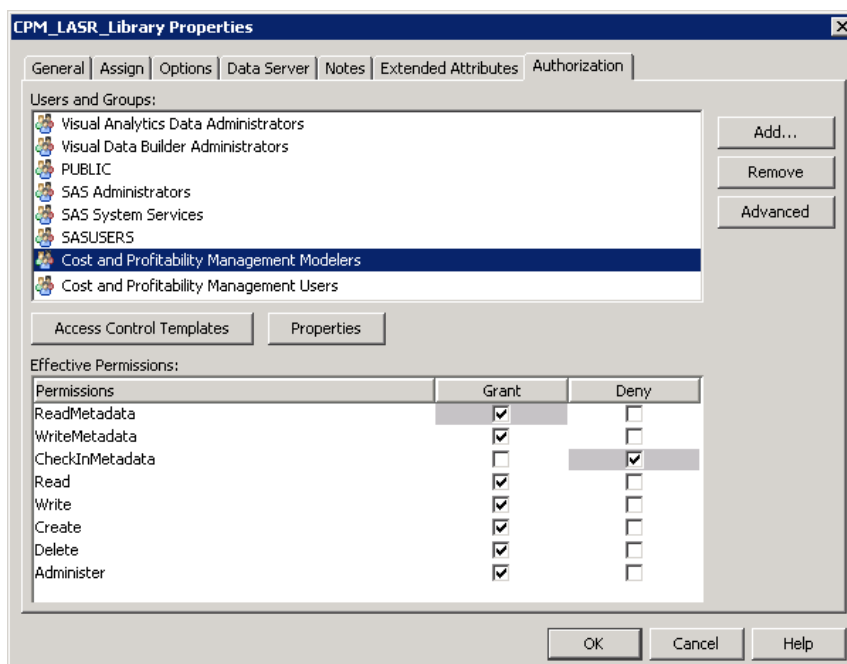
Whether you create a new LASR-enabled metadata library for your SAS Cost and Profitability Management data or use an existing LASR-enabled library, you must give users authorization to use the library. To give users authorization to the library, do the following:

1. In SAS Management Console, right-click your LASR-enabled metadata library and select **Properties**.



2. Click the **Authorization** tab, and grant to the group that will be working with this library, for example Cost and Profitability Management Modelers and Cost and Profitability Management Users, the following permissions:

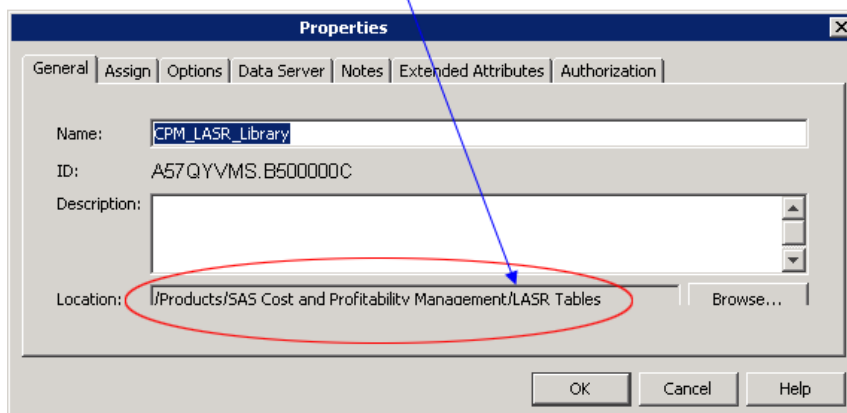
ReadMetadata  
WriteMetadata  
Read  
Write  
Create  
Delete  
Administer



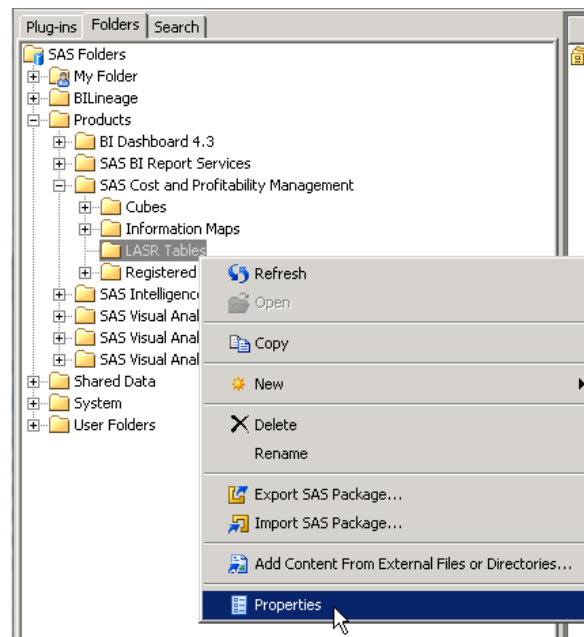
3. Grant the same permissions to the folder for the library.

a. Determine the folder

Grant permissions for this folder (whatever it is)

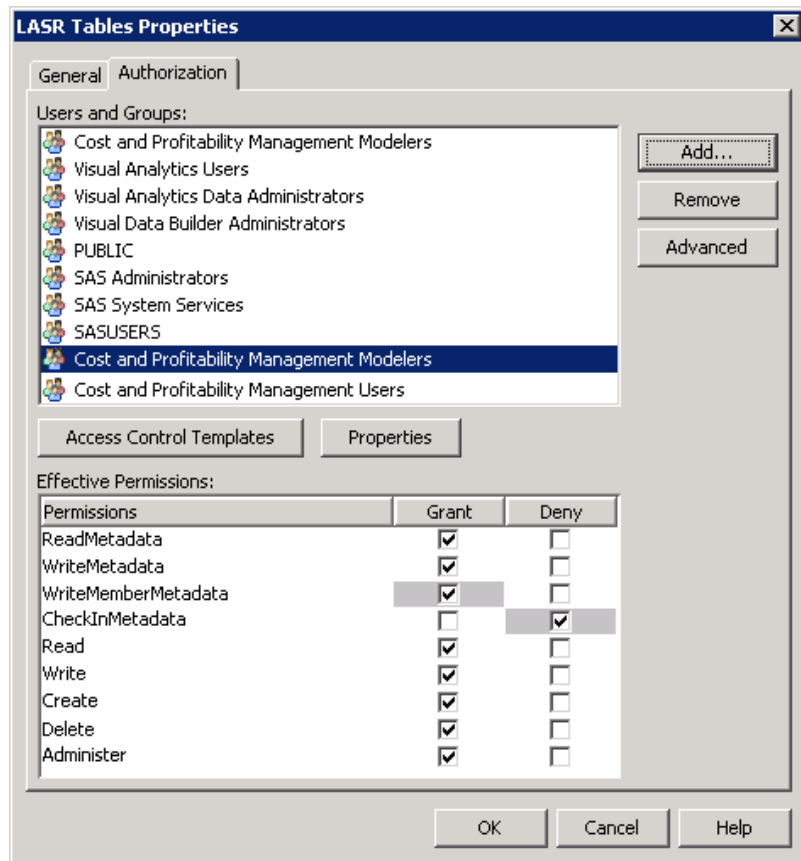


b. Right-click the folder and select properties



- c. Click the **Authorization** tab, and grant to the group that will be working with this folder, for example Cost and Profitability Management Modelers and Cost and Profitability Management Users, the following permissions:

ReadMetadata  
 WriteMetadata  
 WriteMemberMetadata  
 Read  
 Write  
 Create  
 Delete  
 Administer

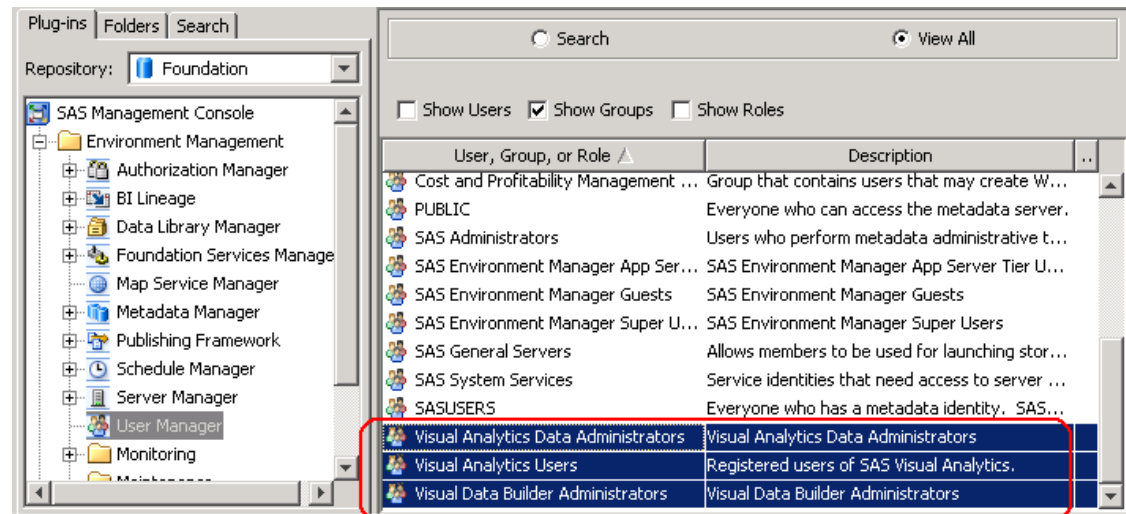


## Authorize Users to Manage LASR

### Overview

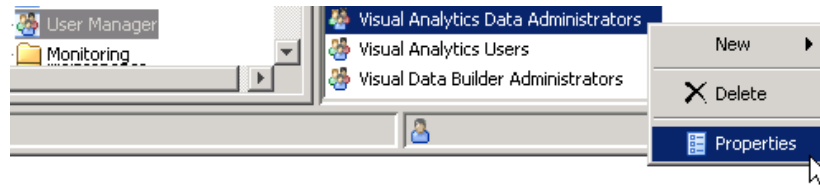
Users that you want to perform LASR functions must be granted permissions by the following SAS Visual Analytics groups:

- “Visual Analytics Data Administrators” on page 300
- “Visual Analytics Users” on page 301
- “Visual Data Builder Administrators” on page 301

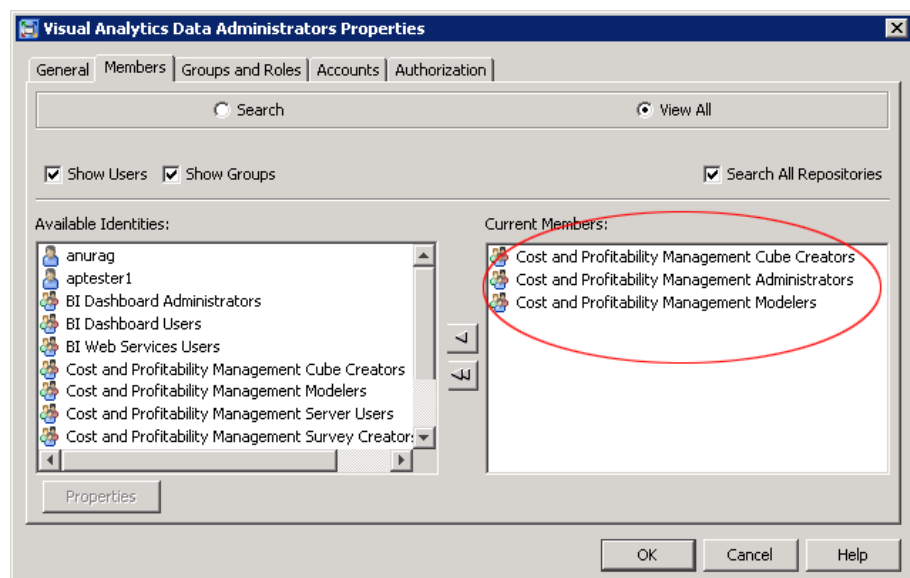


### Visual Analytics Data Administrators

1. Right-click **Visual Analytics Data Administrators** (under **User Manager**), and select **Properties**.



2. Click the **Members** tab and add Cost and Profitability groups that need to use LASR. The picture below shows adding the following groups:
  - Cost and Profitability Management Cube Creators
  - Cost and Profitability Management Administrators
  - Cost and Profitability Management Modelers

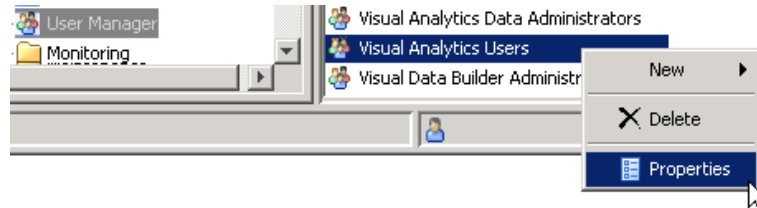




Granting these permissions allows the users to load data into LASR.

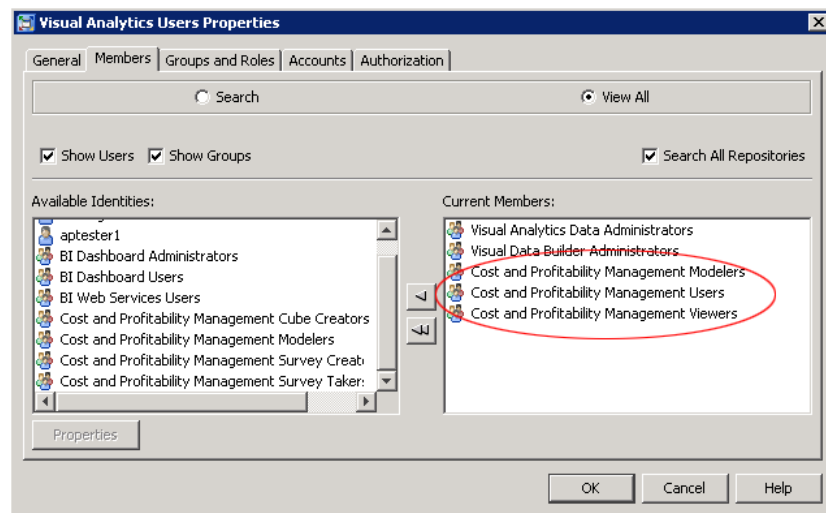
## Visual Analytics Users

1. Right-click **Visual Analytics Users** (under **User Manager**), and select **Properties**.



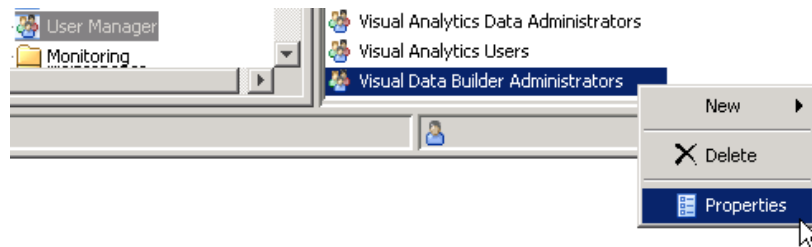
2. Click the **Members** tab and add Cost and Profitability groups that need to use LASR. The picture below shows adding the following groups:

- Cost and Profitability Management Modelers
- Cost and Profitability Management Users
- Cost and Profitability Management Viewers



## Visual Data Builder Administrators

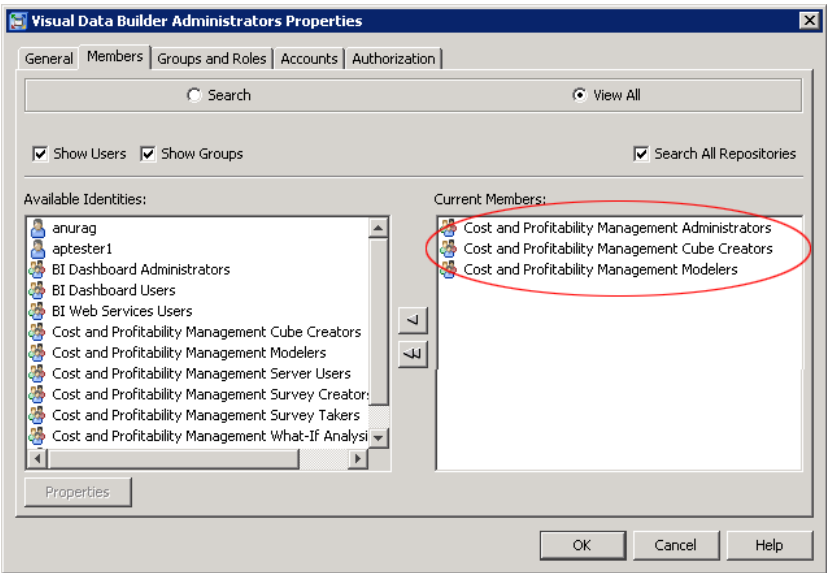
1. Right-click **Visual Data Builder Administrators** (under **User Manager**), and select **Properties**.



2. Click the **Members** tab add Cost and Profitability groups that need to use LASR. The picture below shows adding the following groups:

- Cost and Profitability Management Administrators

- Cost and Profitability Management Cube Creators
- Cost and Profitability Management Modelers



## Start the SAS LASR Analytic Server Using SAS Visual Analytics

The following steps describe using the SAS Visual Analytics Administrator.

1. Open the document **C:\SAS\Config\Lev1\Documents\Instructions.html** on the machine that is running SAS Visual Analytics.
2. In the document Instructions.html, find the section for SAS Visual Analytics Administrator. An example is shown in the following picture:

### SAS Visual Analytics Administrator

URL	<a href="http://rdcesx08027.race.sas.com:85/SASVisualAnalyticsAdministrator">http://rdcesx08027.race.sas.com:85/SASVisualAnalyticsAdministrator</a>
Log directory	<ul style="list-style-type: none"><li>• <a href="C:\SAS\Config\Lev1\Web\Logs\SASServer12_1\SASVisualAnalyticsAdministrator6.4.log">C:\SAS\Config\Lev1\Web\Logs\SASServer12_1\SASVisualAnalyticsAdministrator6.4.log</a></li></ul> <p>For more details about how to modify your log format and to see what logging entails, see "Administering Logging for SAS Web Applications" in the <i>SAS Intelligence Platform</i>.</p>

3. Click the URL to open SAS Visual Analytics Administrator.


**SAS Visual Analytics Administrator**

→ URL	<a href="http://rdcesx08027_race.sas.com:85/SASVisualAnalyticsAdministrator">http://rdcesx08027_race.sas.com:85/SASVisualAnalyticsAdministrator</a>
Log directory	<ul style="list-style-type: none"> <li><a href="#">C:\SAS\Config\Lev1\Web\Logs\SASServer12_1\SASVisualAnalyticsAdministrator6.4.log</a></li> </ul> <p>For more details about how to modify your log format and to see what logging entails, see "Administering Logging for SAS Web Applications" in <i>SAS Intelligence Platform</i>.</p>

The logon dialog appears.

- Log on to SAS Visual Analytics Administrator.

*Note:* The user logging on must be a member, directly or indirectly, of the **Visual Analytics: Administration** group. See “Give Users Capabilities for SAS Visual Analytics” on page 273.



The image shows a 'Log On to SAS®' dialog box with a dark blue background. It contains two input fields: 'User ID:' and 'Password:'. Below these fields is a blue 'Log On' button.

If your mid-tier is configured for IWA (Integrated Windows Authentication) then you can use IWA from your browser to log on if you have configured your browser to use IWA.

For information see “Support for Integrated Windows Authentication” in *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide, Third Edition*: <http://support.sas.com/documentation/cdl/en/bimtag/68217/HTML/default/viewer.htm#p1871e69gmwdr0n1o182krs1c10p.htm>.

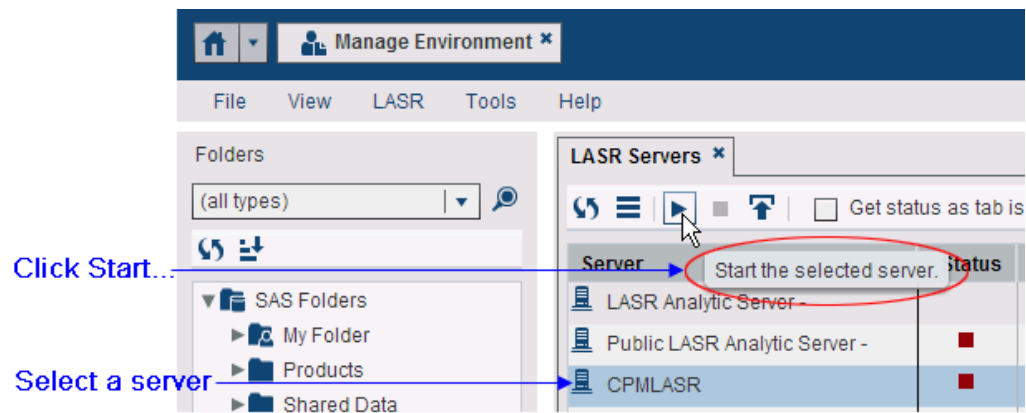
The SAS Visual Analytics Administrator application opens.

- Select **LASR** ⇒ **Manage Servers**.



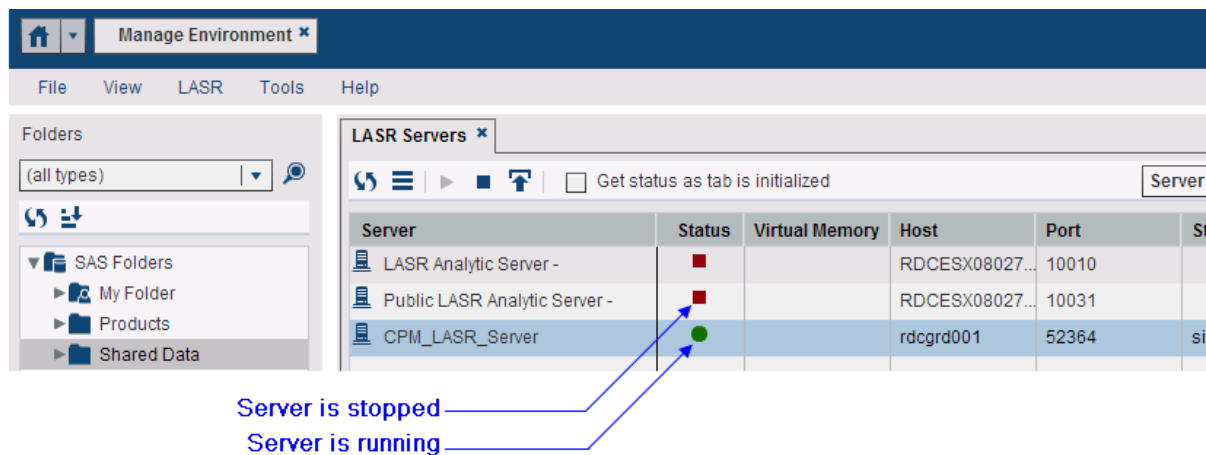
The **LASR Servers** tab opens.

- Select a server and click **Start the selected server**.

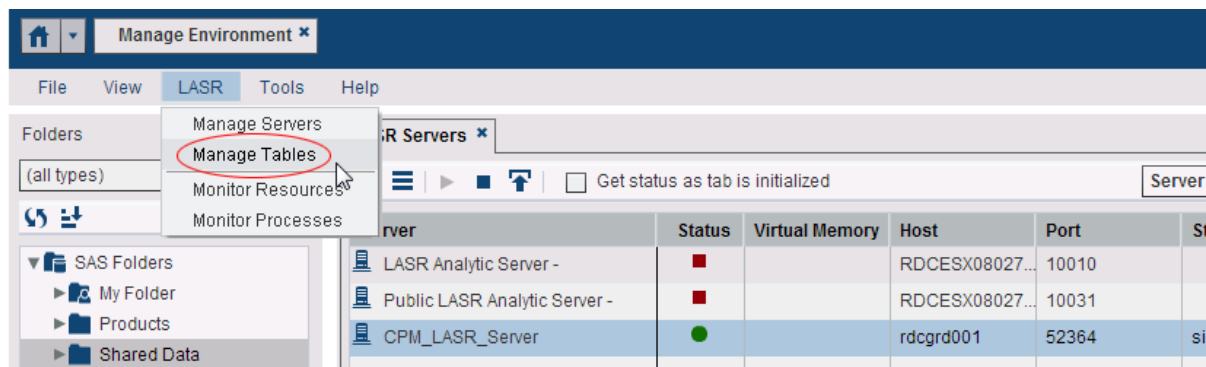


A green light ■ indicates that the server is running.

A red light ■ indicates that the server is stopped.



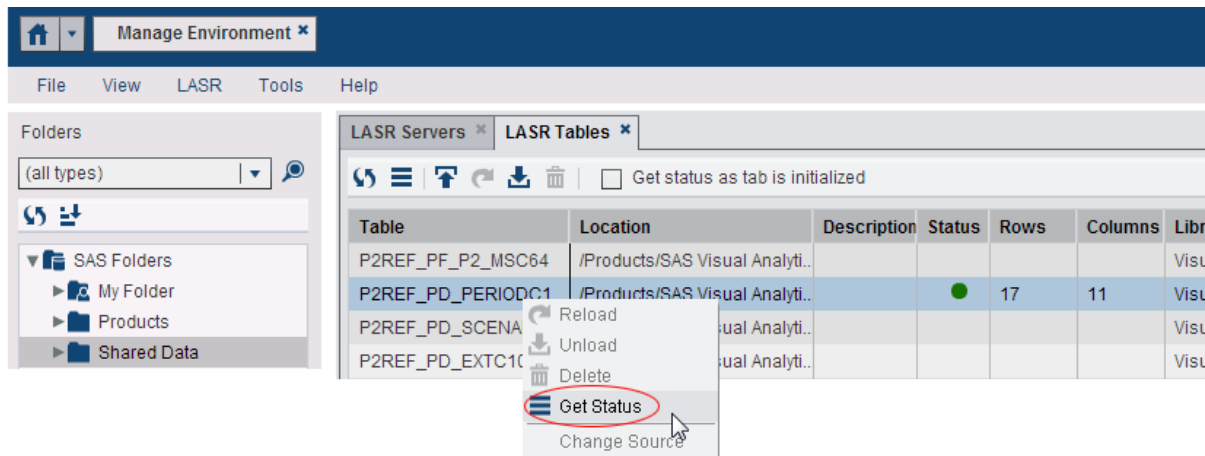
7. To see the tables on a server, select **LASR** ⇒ **Manage Tables**.



The **LASR Tables** tab opens.

8. To see the status of a table and its number of rows and columns, do either of the following:

- Click the table.
- Right-click the table and select **Get Status**.



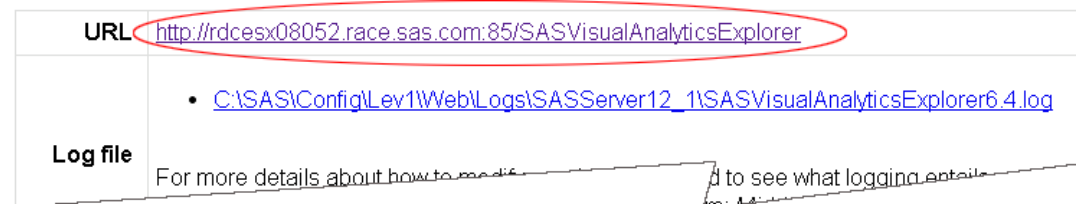
The status of the table is displayed as well as its number of rows and columns.

## Explore the Data

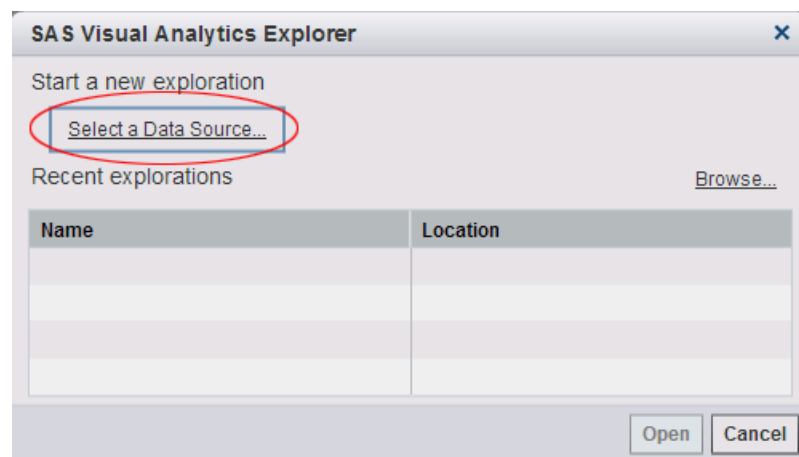
To explore your SAS Cost and Profitability Management data with SAS Visual Analytics:

1. Open the document **C:\SAS\Config\Lev1\Documents\Instructions.html** on the machine that is running SAS Visual Analytics.
2. In the document Instructions.html, find the section for SAS Visual Analytics Explorer. Then click the URL to open the Explorer.

### SAS Visual Analytics Explorer

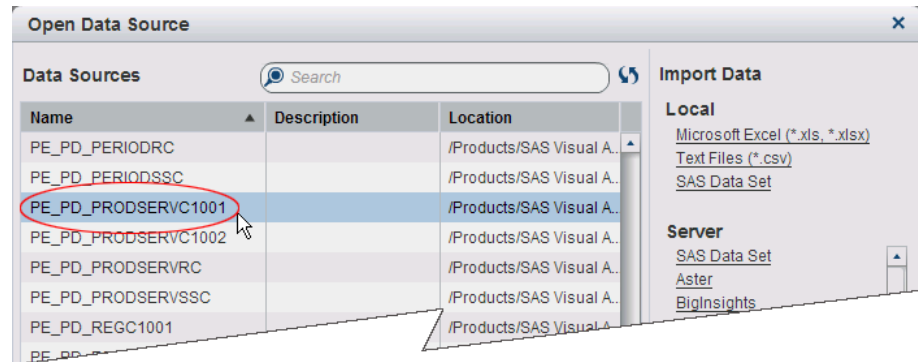


3. After logging on to the Explorer, click **Select a Data Source**.



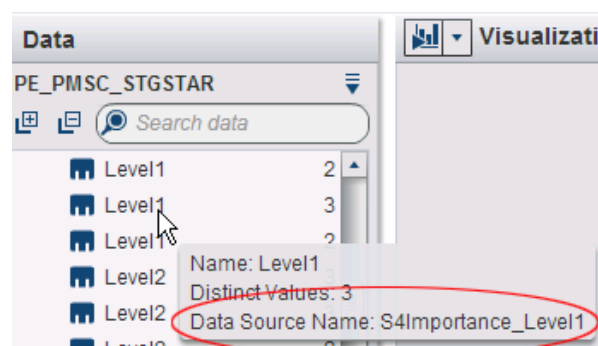
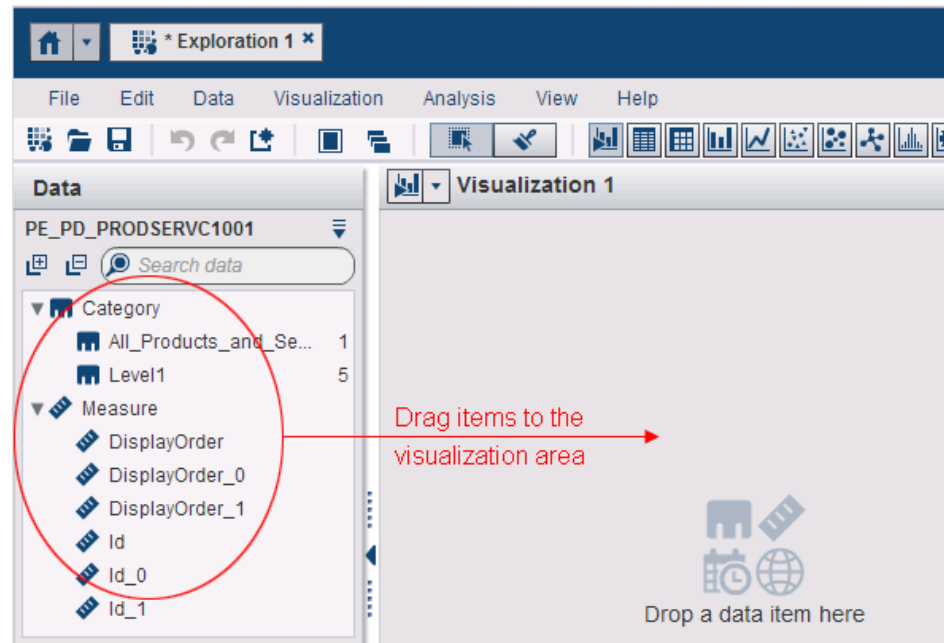
A list of available tables is displayed.

4. Select a table to view.



5. Drag items in the table to the Visualization area.

For further information, see the documentation on SAS Visual Analytics, for example the *SAS Visual Analytics User's Guide*, on [support.sas.com](http://support.sas.com).



You can control what is displayed for the label by using the LABELPREFIX option of PROC IMSTAT. When LABELPREFIX is specified, the dimension name is used as a prefix for all variables from dimension tables that have a label. For information, see the *SAS LASR Analytic Server Reference Guide* on [support.sas.com](http://support.sas.com).

## Chapter 23

# Sample Reports for SAS Visual Analytics

---

<b>Overview</b>	<b>308</b>
<b>Step 1 – Import the Models</b>	<b>311</b>
Overview	311
Procedure	311
<b>Step 2 – Import the Cube Configurations</b>	<b>316</b>
Overview	316
Procedure	317
<b>Step 3 – Start the SAS LASR Analytic Server</b>	<b>319</b>
Overview	319
Procedure	321
<b>Step 4 – Export and Register Tables</b>	<b>322</b>
Overview	322
Procedure	323
<b>Step 5 – Calculate the Models and Generate Cubes</b>	<b>327</b>
Overview	327
Procedure	330
<b>Step 6 – Import the Sample Reports</b>	<b>335</b>
Overview	335
Procedure	335
<b>Step 7 – View the Sample Reports</b>	<b>342</b>
Overview	342
Diagnosing Potential Problems	342
<b>View the ITBM_Dashboard Report</b>	<b>344</b>
Open the Report	344
Objects	346
Data	347
Report Tab 1: Resource	349
Report Tab 2: Activity	353
Report Tab 3: CostObject	354
Report Tab 4: TrendCost	355
<b>View the ITBM_Fact_View Report</b>	<b>357</b>
Overview	357
Report Tab: Step 2	357
Report Tab: Resource Contributions	359
Report Tab: Three Pane View	360

<b>View the ITBM_Account_View Report</b> .....	<b>363</b>
Overview .....	363
Report Tab: S3_ITIL – Activity Module .....	363
Report Tab: S6_Customer – Cost Objects .....	364
Report Tab: Act_Drivers – Activity Module Drivers .....	365

---

## Overview

SAS Cost and Profitability Management provides ten sample reports produced with SAS Visual Analytics to demonstrate how SAS Visual Analytics can provide insight into your models.

To view the sample reports, perform the following steps:

### Step 1 – Import the Models

The first step is to import four sample models. See [“Step 1 – Import the Models” on page 311](#). These instructions describe importing one particular model, ITBMnMOD, but the procedure is the same for all four of the following sample models:

Model name	Description
Banking	Cost flows in a bank.
FASHNMod	Cost flows for a large retailer.
Hospital	Cost flows in a hospital.
ITBMnMOD	Cost flows for an IT organization.

### Step 2 – Import the Cube Configurations

The next step is to import cube configurations for the sample models. See [“Step 2 – Import the Cube Configurations” on page 316](#).

Cube Configuration Name	Model Name
Bank_MSC	Banking
FASHNMSC	FASHNMod
Hosp_MSC	Hospital
ITNMod_MSC	ITBMnMOD

### Step 3 – Start the SAS LASR Analytic Server

The next step is to start the SAS LASR Analytic server in order to make SAS Cost and Profitability Management data available to SAS Visual Analytics. See [“Step 3 – Start the SAS LASR Analytic Server” on page 319](#).

### Step 4 – Export and Register Tables

In the next step you make the following SAS Cost and Profitability Management tables available to SAS Visual Analytics by exporting and registering the tables. See [“Step 4 – Export and Register Tables” on page 322](#).



**Table Name***ModelRef\_ACCOUNTMAP**ModelRef\_ASSIGNMENTMAP**ModelRef\_PV\_ACCOUNT**ModelRef\_PV\_ASSIGNMENT**ModelRef\_PV\_ASSIGNMENTTEXT**ModelRef\_PV\_ATTRIBUTE**ModelRef\_PV\_ATTRIBUTEVALUE**ModelRef\_PV\_DIMENSION**ModelRef\_PV\_DRIVER**ModelRef\_PV\_ENTEREDCE*

These instructions describe exporting and registering tables for one particular model, ITBMnMOD, but the procedure is the same for all four models.

**Step 5 – Calculate the Models and Generate Cubes**

In the next step you generate a cube for the sample models to make the following fact-table related views available to SAS Visual Analytics. See [“Step 5 – Calculate the Models and Generate Cubes” on page 327](#). These instructions describe generating a cube for one particular model, ITBMnMOD, but the procedure is the same for all four models.

*Note:* The following list shows all the possible tables that can be created during cube generation, but the only table that is used by the sample reports is the multi-stage contributions table.

**View Name***ModelRef\_PD\_COSTELEMENT<RC | SSC | C<cubeID>>**ModelRef\_PD\_dimShortRef<RC | SSC | C<cubeID>>**ModelRef\_PD\_DRIVER<RC | SSC | C<cubeID>>**ModelRef\_PD\_MODULE<RC | SSC | C<cubeID>>**ModelRef\_PD\_PERIOD<RC | SSC | C<cubeID>>**ModelRef\_PD\_SCENARIO<RC | SSC | C<cubeID>>**ModelRef\_PD\_YESNO<RC | SSC | C<cubeID>>**ModelRef\_PF\_<RC | SSC | C<cubeID>>*

**View Name**

*ModelRef\_PF\_<RCSTAR | SSCSTAR | MSC\_C<cubeID>STAR>*

**Step 6 – Import the Sample Reports**

The final step before viewing the sample reports is to import SAS package files that contain the reports. See [“Step 6 – Import the Sample Reports” on page 335](#). These instructions describe importing one set of sample reports for the ITBMnMOD model, but the procedure is the same for all the sample reports.

SAS Package File		Contained Reports
SQL	SQL_BankVA.spk	Bank_Account_View
		Bank_Dashboard
		Bank_Fact_View
Oracle	Orcl_FASHBI.spk	FASHBI_Acct_Analysis
SQL	SQL_FASHBI.spk	FASHBI_Dashboard
		FASHBI_Fact_Analysis
SQL	SQL_Hospital.spk	Hospital_View
Oracle	Orcl_ITBMVA.spk	ITBM_Account_View
PostgreSQL	Post_ITBMVA.spk	ITBM_Dashboard
SQL	SQL_ITBMVA.spk	ITBM_Fact_View

See [“Step 6 – Import the Sample Reports” on page 335](#).

**Step 7 – View the Sample Reports**

You can now view the sample reports. The following sections describe viewing sample reports for the ITBMnMOD model:

**ITBM\_Dashboard Report**

See [“View the ITBM\\_Dashboard Report” on page 344](#).

**ITBM\_Fact\_View Report**

See [“View the ITBM\\_Fact\\_View Report” on page 357](#).

**ITBM\_Account\_View Report**

See [“View the ITBM\\_Account\\_View Report” on page 363](#).

*Note:*

- Because there are minor differences in the way that the sample reports were built on the different databases supported by SAS Cost and Profitability Management, the pictures of the sample reports in this chapter might not match exactly the sample reports that you view depending on your database. For example, the sorting order for columns might be different as well as other minor differences.
- If you encounter a problem viewing the sample reports, see [“Diagnosing Potential Problems” on page 342](#).

## Step 1 – Import the Models

### Overview

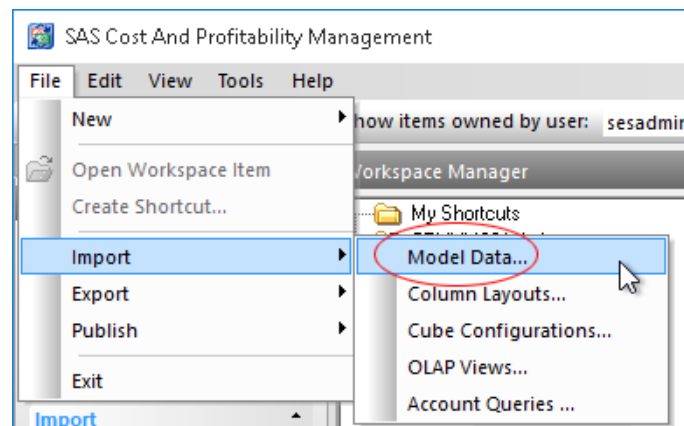
There are four sample models that you can import whose data was used to create the ten sample SAS Visual Analytics reports that are provided. The following table shows the name and reference of each of the models and the zip file that you import to import the model.

Model Name	Model Reference	Import File
Banking	Banking	Banking.zip
FASHNMod	FASHNMod	FASHBI.zip
Hospital	Hospital	Hospital.zip
ITBMnMOD	ITBMnMOD	ITBM.zip

### Procedure

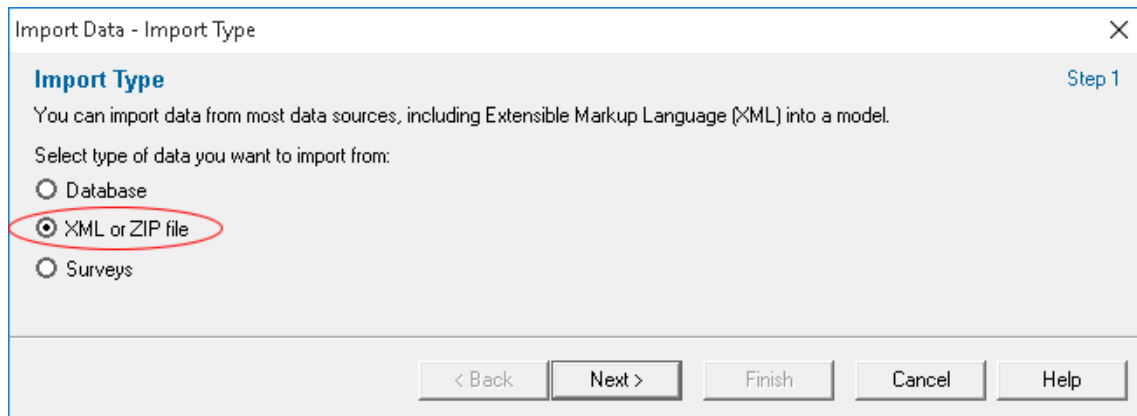
To import the sample models, do the following:

1. Log on to the SAS Cost and Profitability Management client.
2. Select **File** ⇒ **Import** ⇒ **Model Data**.



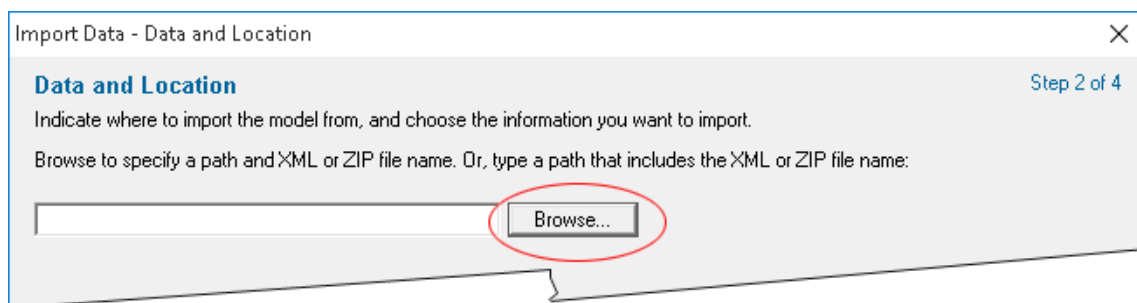
The Import Data – Import Type window opens.

3. Select **XML or ZIP File** and click **Next**.



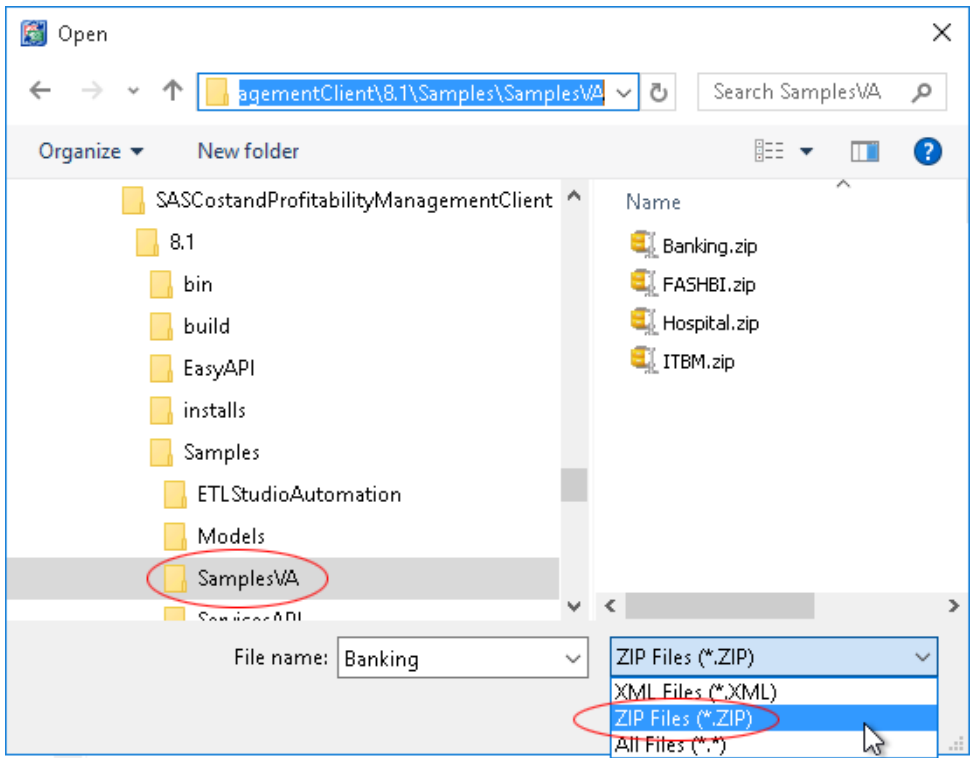
The Import Data – Data and Location window opens.

4. Click **Browse**.



The Open dialog appears.

5. Navigate to the location of the sample models and reports under the **SASHome** directory, for example: **C:\Program Files\SASHome\x86\SASCostandProfitabilityManagementClient\8.1\Samples\SamplesVA** and change the file type to **ZIP**.



Four .zip files are listed:

- Banking.zip
  - FASHN.zip
  - Hospital.zip
  - ITBM.zip
6. Select **ITBM.zip** and click **Open**. Then click **Next**.
- The Import Data – Model window opens.
7. Type the following model name and model reference:

Model Name	Model Reference
ITBMnMOD	ITBMnMOD

**Import Data - Model** Step 3 of 4

**Model**

Choose the model you want to import the data into.

Enter new model name or Select the existing model you want to import into:

☒ New model:

\* Model name:

\* Model Reference:

☐ Existing model:

Note: The content of the selected model (for all Periods/Scenarios) is deleted and replaced with the content of the selected XML or ZIP file. This assures that there is no conflict between the existing structure and the imported structure.

*Note:* You will also be importing a cube configuration for this model. So, it is important that the name and reference that you specify for the model match the name and reference specified for the model in the cube configuration. Otherwise, the cube configuration will not be used correctly when you attempt to generate a cube for the model.

Then click **Next**.

8. Click **Finish**.

**Import Data - Summary** Step 4 of 4

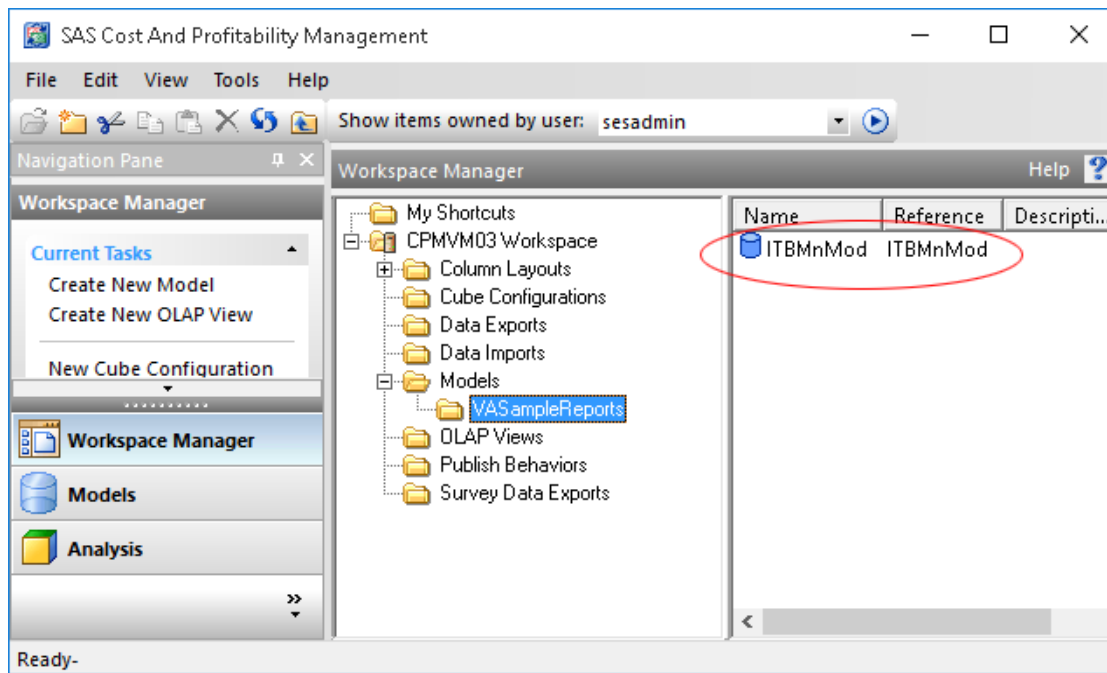
**Summary**

Please review your selections:

**Import File Type:**  
XML/Zip File

**Source:**  
C:\Program Files\SASHome\...9.1\Samples\Samples\VA\ITBM.zip

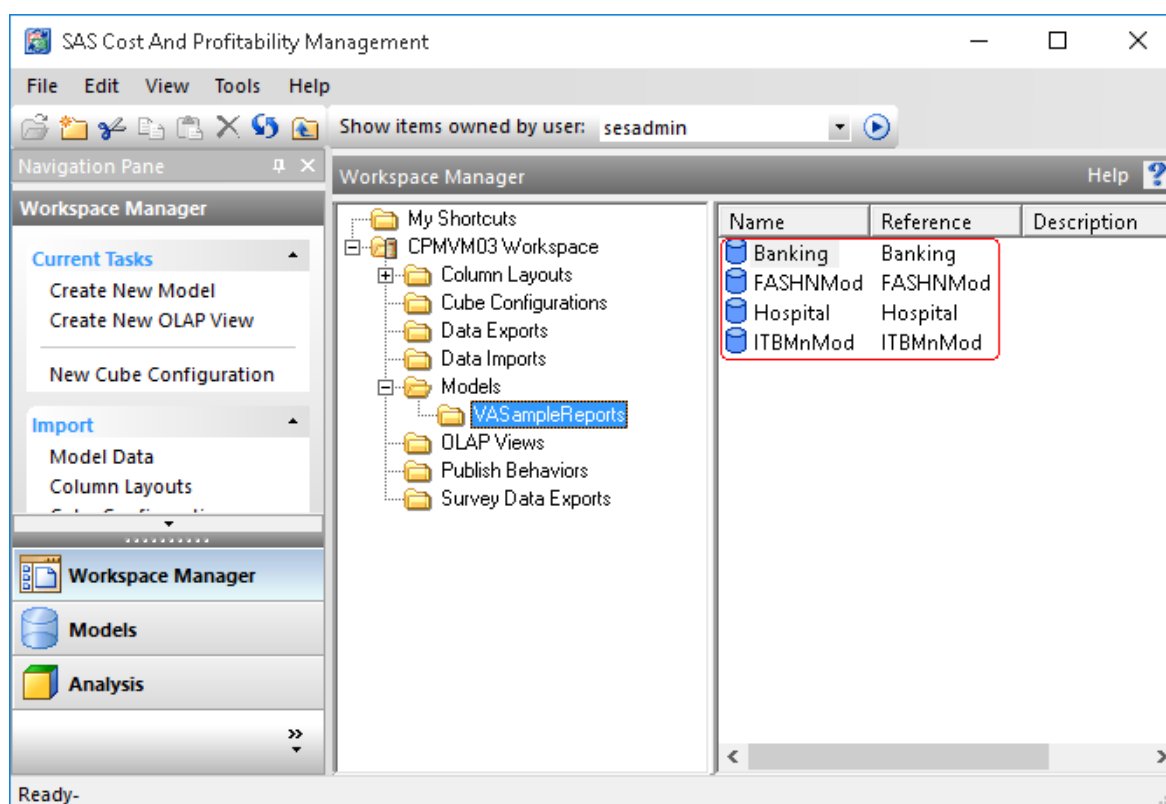
The **ITBMnMOD** model is imported.



9. Repeat the previous steps to import the three remaining models. The following table shows all four sample model files that you import and the name and reference to type for each imported model.

Model Name	Model Reference	Import File
Banking	Banking	Banking.zip
FASHNMod	FASHNMod	FASHBI.zip
Hospital	Hospital	Hospital.zip
ITBMnMOD	ITBMnMOD	ITBM.zip

*Note:* Make sure to type the model name and model reference as shown above so that each cube configuration that is to be imported matches its respective model.



*Note:* The picture shows the models in a folder named **VASampleReports**, but of course you can put the models wherever you like.

## Step 2 – Import the Cube Configurations

### Overview

In this step you import the following four cube configurations—one for each of the sample models.

Cube Configuration Name	Cube Configuration Reference	Model Name	Model Reference
Bank_MSC	Bank_MSC	Banking	Banking
FASHNMSC	FASHNMSC	FASHNMod	FASHNMod
Hosp_MSC	Hosp_MSC	Hospital	Hospital
ITNMod_MSC	ITNMod_MSC	ITBMnMOD	ITBMnMOD

Each of the cube configurations generates a multi-stage contribution cube. Each cube configuration also selects the option **Also load tables into a library for the SAS LASR Analytic Server** (see the picture below). Selecting this option causes tables that are related to fact table generation to be loaded into the library specified for the SAS LASR



Analytic server. For information on the SAS LASR Analytic server, see “Step 3 – Start the SAS LASR Analytic Server” on page 319.

**Edit Cube Configuration - Cube and Fact Table** Step 1

Select model and cube configuration name.

Model name:

Cube configuration name:

Cube configuration reference:

Description:

Generate

☒ Cube and Fact table  
☐ Fact table only

Type:

☐ Suppress zero costs

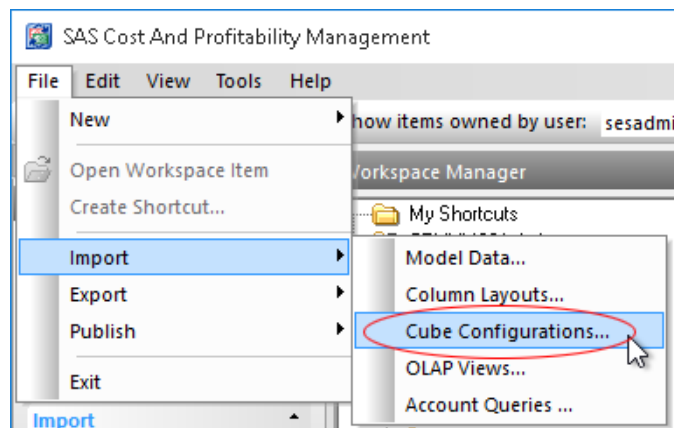
☒ Also load tables into a library for the SAS LASR Analytic Server

< Back   Next >   Finish   Cancel   Help

## Procedure

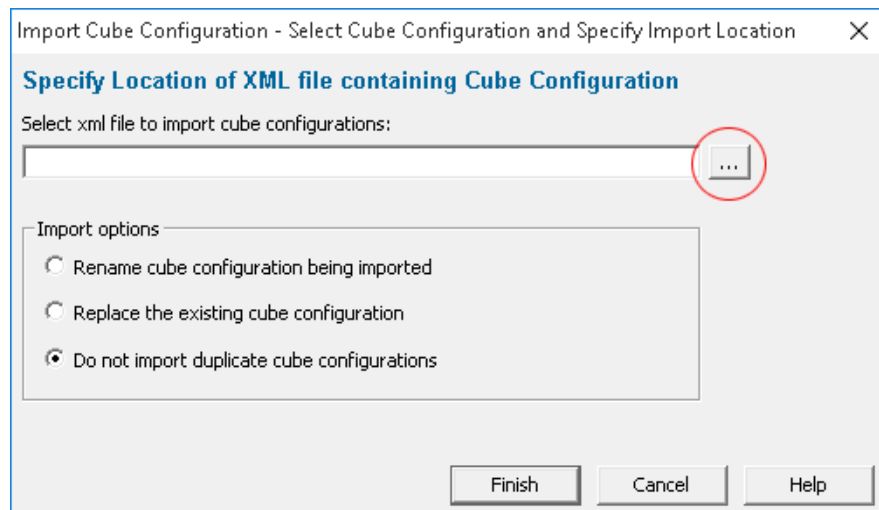
To import the cube configurations, do the following:

1. Select **File** ⇒ **Import** ⇒ **Cube Configurations**.



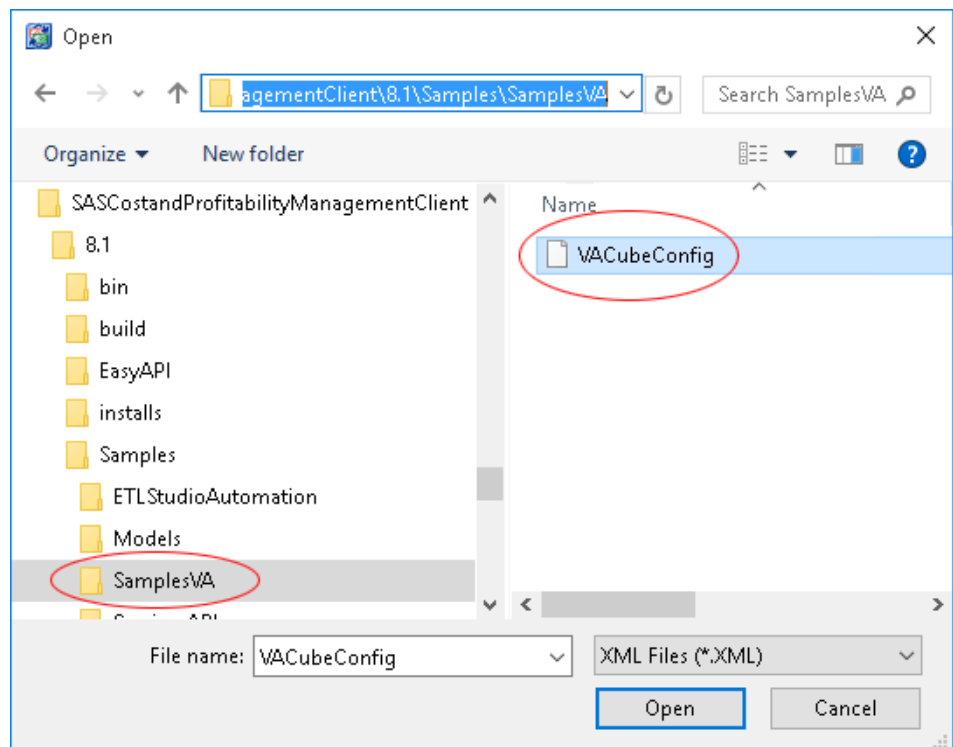
The Import Cube Configuration window opens.

2. Click the **Browse** button



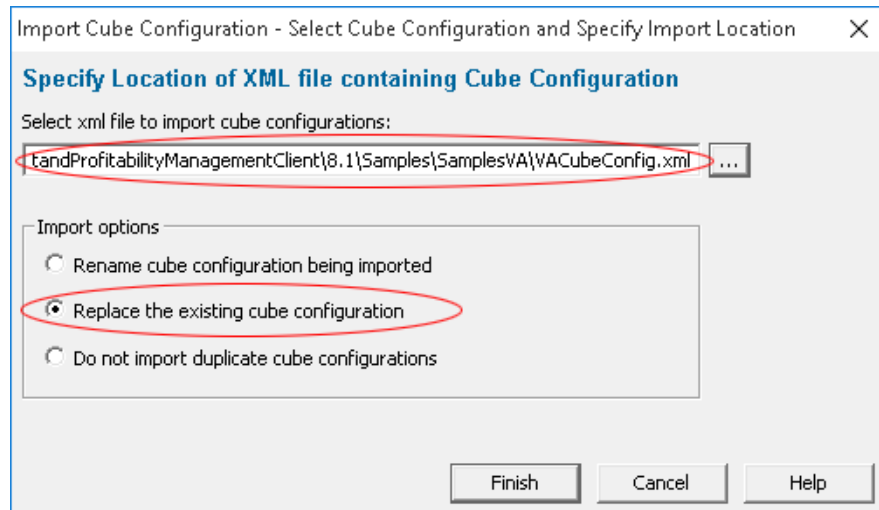
3. Navigate to the location of the sample models and reports under the **SASHome** directory, for example: **C:\Program Files\SASHome\x86\SASCostandProfitabilityManagementClient\8.1\Samples\SamplesVA**.

Select **VACubeConfig** and click **Open**.



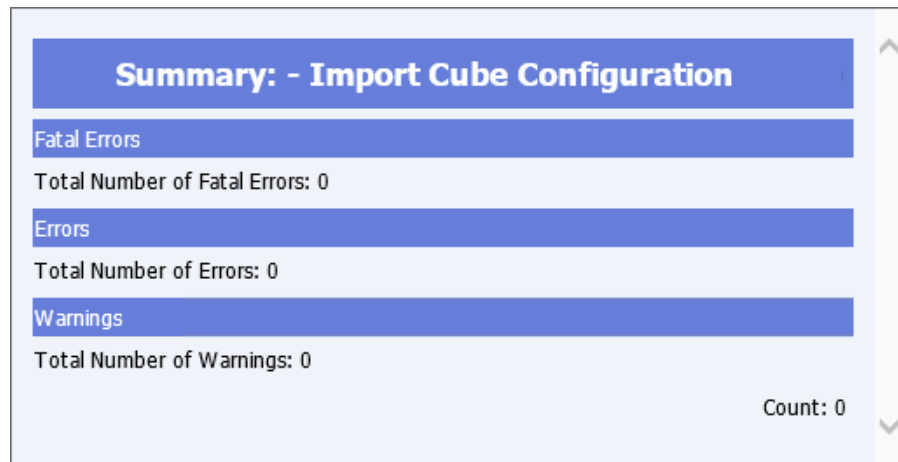
You are returned to the Import Cube Configuration window.

4. Select **Replace the existing cube configuration** to ensure that you use the correct cube configurations for the sample models.



5. Click **Finish**.

The cube configurations are imported.



## Step 3 – Start the SAS LASR Analytic Server

### Overview

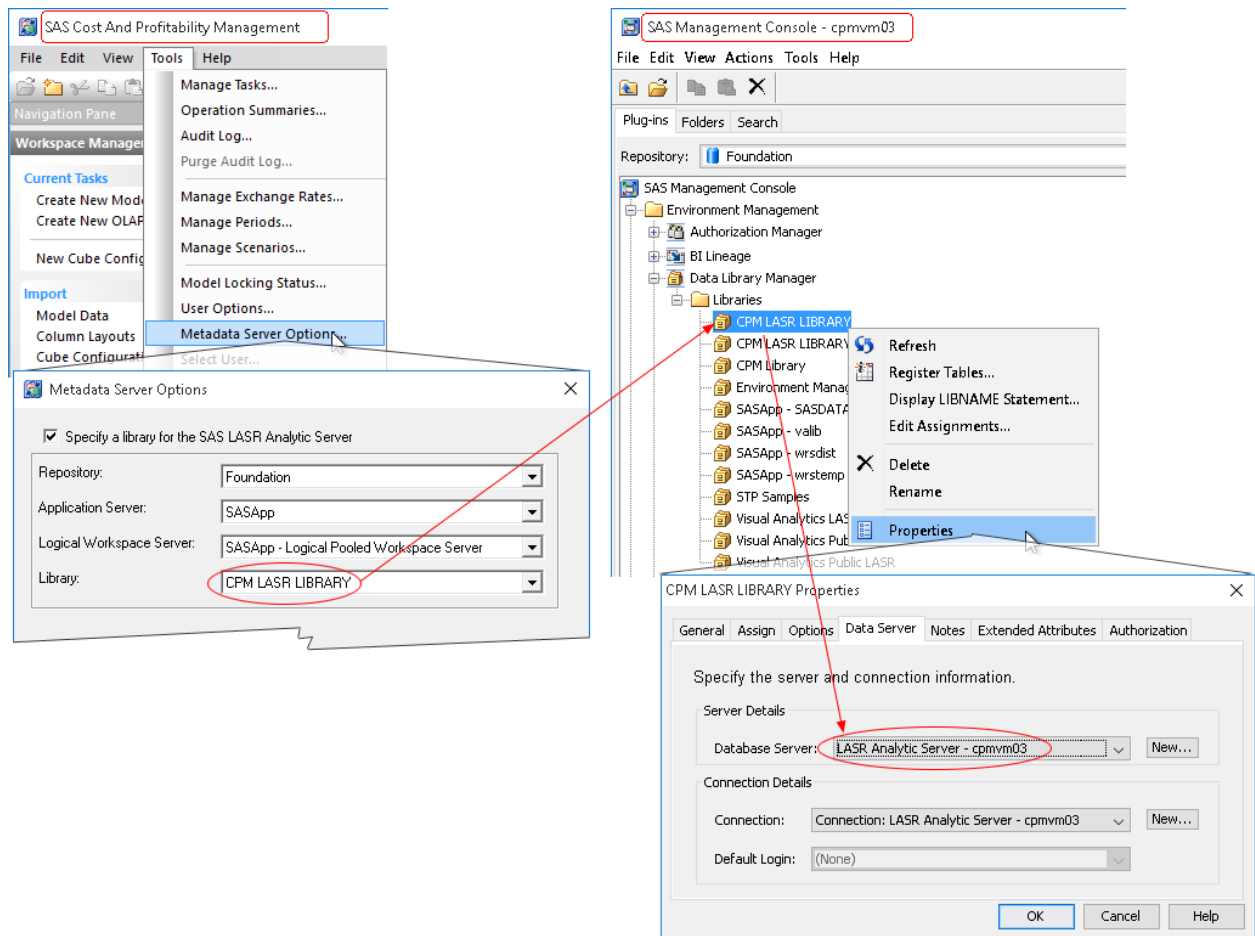
Before loading SAS Cost and Profitability Management tables into a LASR library for use with SAS Visual Analytics you must first start a SAS LASR Analytic server. The example in this chapter assumes that a SAS LASR Analytic server named **LASR Analytic Server - cpmvm03** is being used. The server that you use, of course, can be different. The following picture shows how to determine what SAS LASR Analytic server is being used for SAS Cost and Profitability Management.

1. In SAS Cost and Profitability Management, select **Tools** ⇒ **Metadata Server Options**.

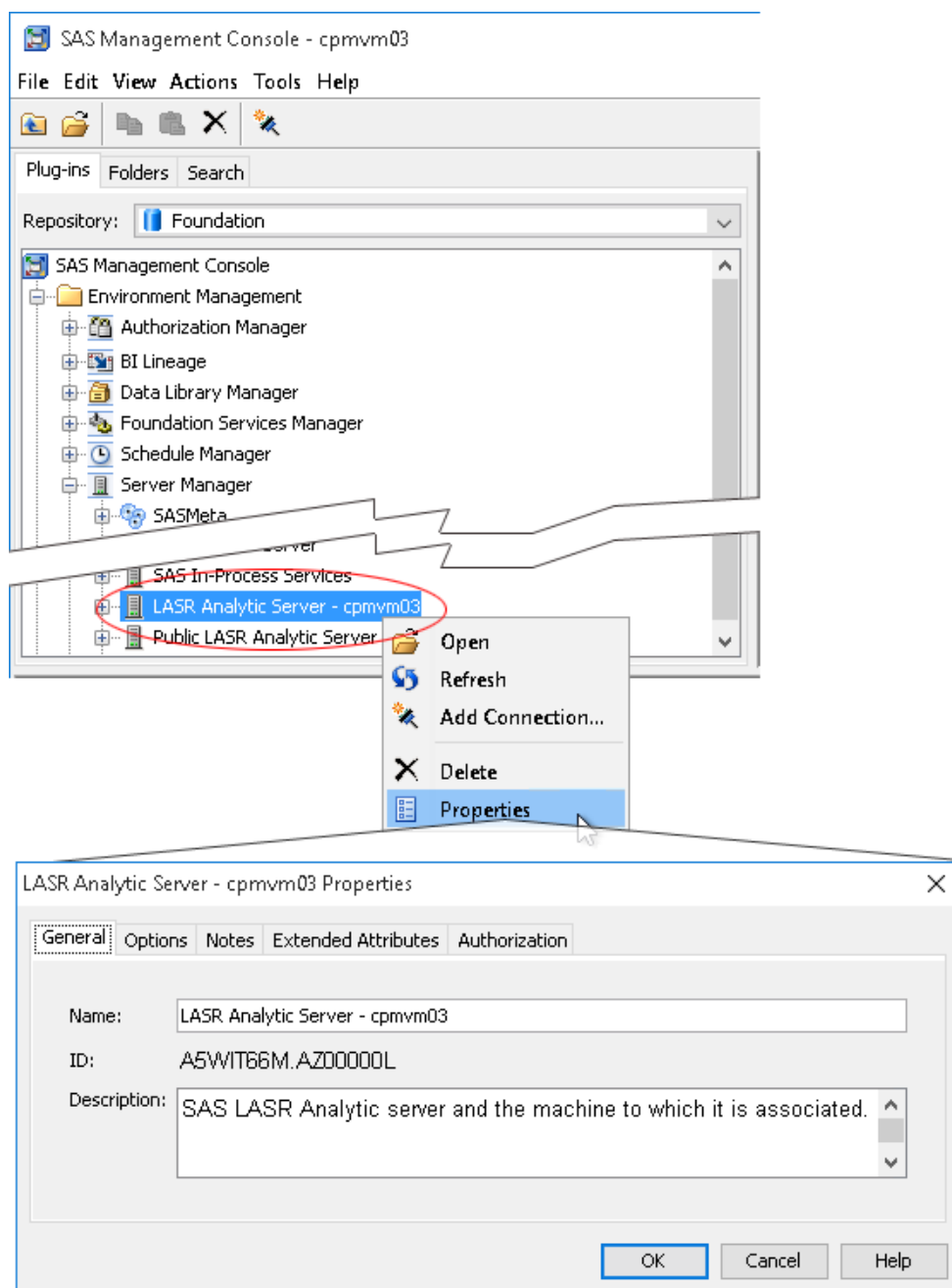
The picture shows that **Select a library for the SAS LASR Analytic Server** has been selected in the Metadata Server options window and that **CPM LASR LIBRARY** is specified for **Library**.

2. In SAS Management Console, select the **Plug-Ins** tab; right-click **CPM LASR LIBRARY** (under **Data Library Manager** ⇒ **Libraries**); and select **Properties**.

The picture shows that **LASR Analytic Server - cpmvm03** is selected for the **Database Server**.



The following picture shows the Properties window for the SAS LASR Analytic server **LASR Analytic Server - cpmvm03** that is being used with SAS Cost and Profitability Management.

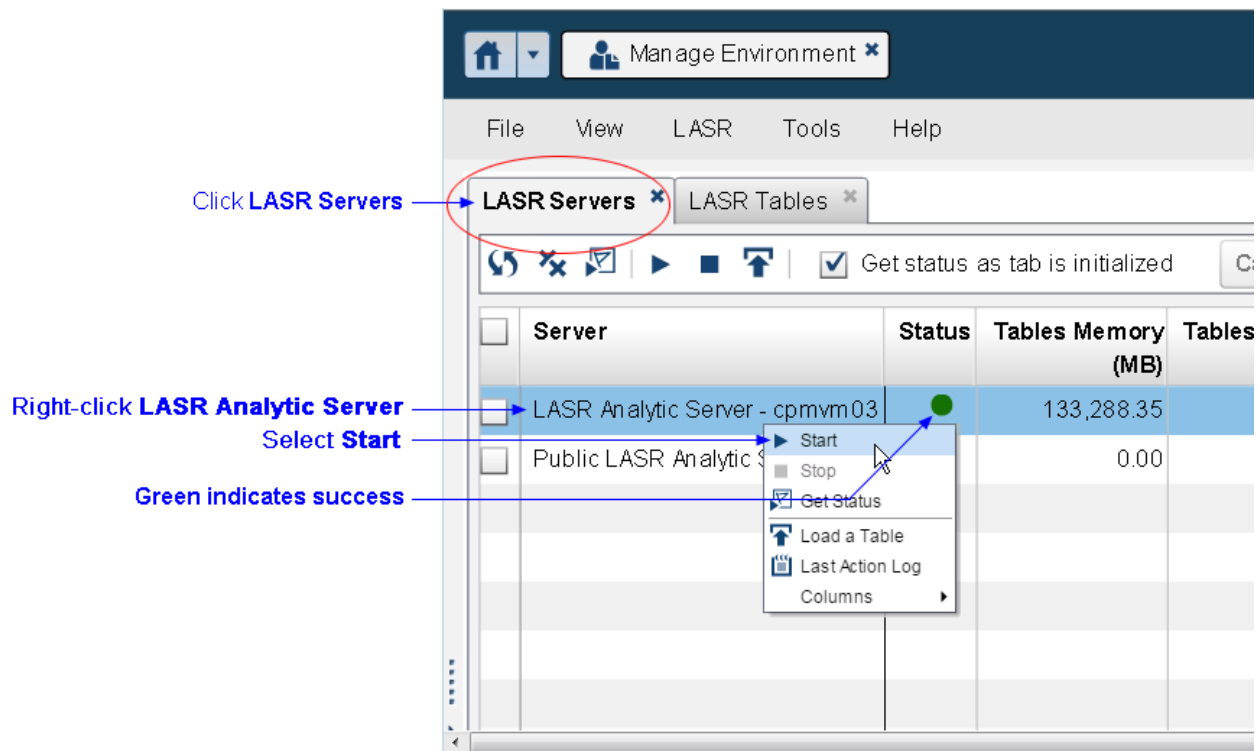


## Procedure

To start the SAS LASR Analytic server:

1. Start the **SAS Visual Analytics Administrator** from your browser. See [“Start the SAS LASR Analytic Server Using SAS Visual Analytics”](#) on page 302.
2. Click the **LASR Servers** tab.
3. Right-click **LASR Analytic Server** and select **Start**.

A green dot indicates that the server has started successfully.



## Step 4 – Export and Register Tables

### Overview

To produce reports, you must make SAS Cost and Profitability Management tables available to SAS Visual Analytics. The following tables describe the structure of a model. This section describes how to make these tables available to SAS Visual Analytics by exporting and registering the tables.

Table Name	Description
<i>ModelRef_ACCOUNTMAP</i>	A join of <i>ModelRef_PV_ACCOUNT</i> with other registered tables. Includes details for multiple periods, dimensional signature, and all numeric properties. See “ <a href="#">modelRef_ACCOUNTMAP</a> ” on page 156.
<i>ModelRef_ASSIGNMENTMAP</i>	A join of <i>ModelRef_PV_ASSIGNMENT</i> with other registered tables. Includes details for source and destination dimensional signatures, and all entered and calculated properties. See “ <a href="#">modelRef_ASSIGNMENTMAP</a> ” on page 159.
<i>ModelRef_PV_ACCOUNT</i>	Details for multiple periods, dimensional signature, and all numeric properties. See “ <a href="#">modelRef_PV_ACCOUNT</a> ” on page 159.

Table Name	Description
<i>ModelRef_PV_ASSIGNMENT</i>	Details for source and destination dimensional signatures, and all entered and calculated properties. See “ <a href="#">modelRef_PV_ASSIGNMENT</a> ” on page 162.
<i>ModelRef_PV_ASSIGNMENTTEXT</i>	Details for source and destination dimensional signatures, and all entered and calculated properties. See “ <a href="#">modelRef_PV_ASSIGNMENTTEXT</a> ” on page 164.
<i>ModelRef_PV_ATTRIBUTE</i>	Details defining the attribute, attribute types, formulas for calculations, and default values. See “ <a href="#">modelRef_PV_ATTRIBUTE</a> ” on page 165.
<i>ModelRef_PV_ATTRIBUTEVALUE</i>	Attribute attachment to the model accounts and the numeric value for the attribute. See “ <a href="#">modelRef_PV_ATTRIBUTEVALUE</a> ” on page 166.
<i>ModelRef_PV_DIMENSION</i>	Details defining the model's dimensions. See “ <a href="#">modelRef_PV_DIMENSION</a> ” on page 167.
<i>ModelRef_PV_DRIVER</i>	Details defining the drivers, driver types, and formulas for rules assignment and for calculations. See “ <a href="#">modelRef_PV_DRIVER</a> ” on page 167.
<i>ModelRef_PV_ENTEREDCE</i>	Details of all entered cost elements, with their corresponding account attachment and values. See “ <a href="#">modelRef_PV_ENTEREDCE</a> ” on page 168.

For further information on these tables, see [Chapter 14, “Working with Registered Tables,” on page 145](#). You will make other SAS Cost and Profitability Management tables available to SAS Visual Analytics when you generate a cube.

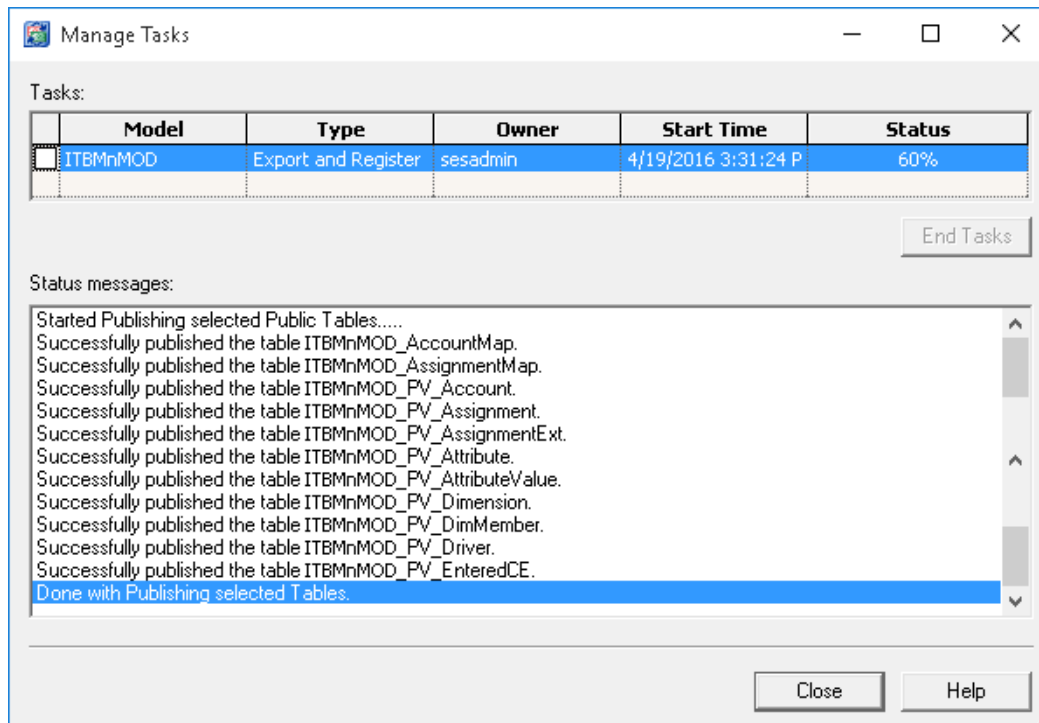
## Procedure

To export and register tables that describe the structure of your model:

1. In SAS Cost and Profitability Management client, open the model **ITBMnMOD**.
2. Select **Model** ⇒ **Export and Register Tables**.  
The Export and Register Tables window opens.
3. Select **Select All**, and then click **OK**.

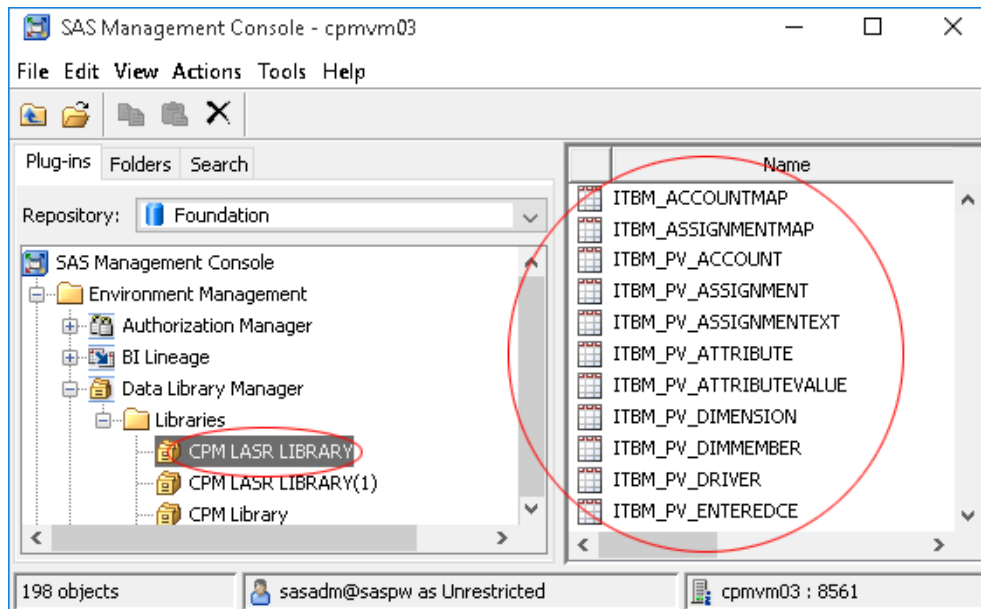
Table Name
AccountMap
AssignmentMap
PV_Account
PV_Assignment
PV_AssignmentExt
PV_Attribute
PV_AttributeValue
PV_Dimension
PV_DimMember
PV_Driver
PV_EnteredCE

The tables are exported and registered.

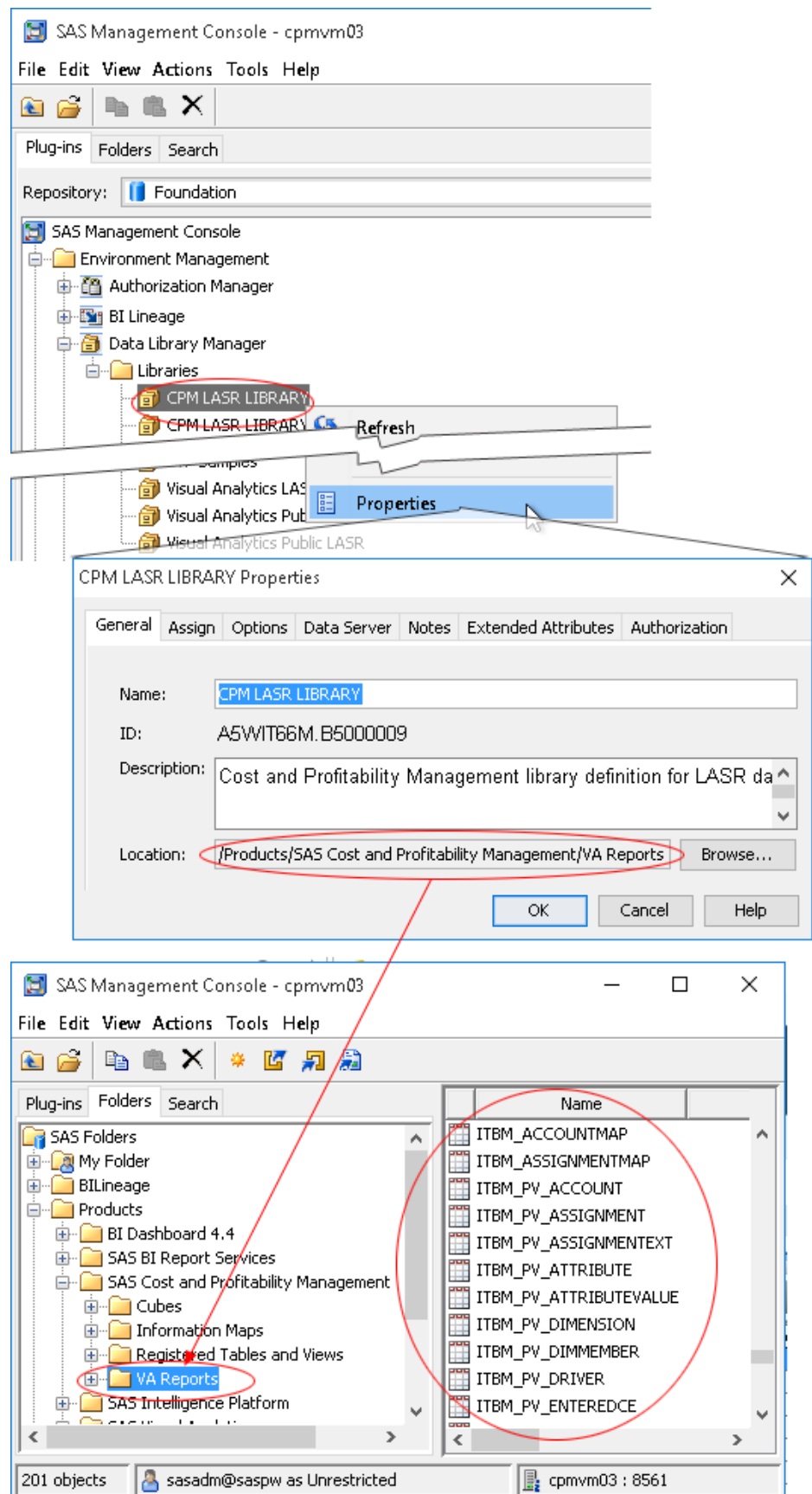




The tables are registered in the SAS Cost and Profitability Management LASR library named **CPM LASR LIBRARY**. See “[Step 3 – Start the SAS LASR Analytic Server](#)” on page 319.



And, similar metadata is added to the folder, **/Products/SAS Cost and Profitability Management/VA Reports**, specified for the library **CPM LASR LIBRARY**.



4. Repeat the steps above to export and register tables for each of the remaining models so that you have exported and registered tables for all four sample models:

Model name
Banking
FASHNMod
Hospital
ITBMnMOD

## Step 5 – Calculate the Models and Generate Cubes

### Overview

In a previous step you made some SAS Cost and Profitability Management tables available to SAS Visual Analytics by exporting and registering the tables. In this step you make some views available to SAS Visual Analytics by generating a fact table and cube for a model. The following model-specific views are related to fact table generation. These views are created and registered in metadata during cube generation.

*Note:* The following list shows all the possible tables that can be created during cube generation, but the only table that is used by the sample reports is the multi-stage contributions table.

View Name	Description
<i>ModelRef_PD_COSTELEMENT</i> < <b>suffix</b> >  where < <b>suffix</b> > is RC, SSC, or C< <b>cube configuration ID</b> > for a multi-stage contributions cube.  For example: ABC_PD_COSTELEMENTRC ABC_PD_COSTELEMENTSSC ABC_PD_COSTELEMENTC1001	Details defining the types of cost elements.  See “ <a href="#">modelRef_PD_COSTELEMENT&lt;suffix&gt;</a> ” on page 169.

View Name	Description
<p><i>ModelRef_PD_dimShortRef</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_REGIONRC</p> <p>ABC_PD_REGIONSSC</p> <p>ABC_PD_REGIONC1001</p>	<p>Single dimension-member details: level by level, noting ID, Reference, and Name.</p> <p>See “<a href="#">modelRef_PD_dimShortRef&lt;suffix&gt;</a>” on page 169.</p>
<p><i>ModelRef_PD_DRIVER</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_DRIVERRC</p> <p>ABC_PD_DRIVERSSC</p> <p>ABC_PD_DRIVERC1001</p>	<p>Driver ID and corresponding Driver Name.</p> <p>See “<a href="#">modelRef_PD_DRIVER&lt;suffix&gt;</a>” on page 170.</p>
<p><i>ModelRef_PD_MODULE</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_MODULERC</p> <p>ABC_PD_MODULESSC</p> <p>ABC_PD_MODULEC1001</p>	<p>Details defining the types of modules.</p> <p>See “<a href="#">modelRef_PD_MODULE&lt;suffix&gt;</a>” on page 171.</p>
<p><i>ModelRef_PD_PERIOD</i>&lt;<b>suffix</b>&gt;</p> <p>where &lt;<b>suffix</b>&gt; is RC, SSC, or C&lt;<b>cube configuration ID</b>&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_PERIODRC</p> <p>ABC_PD_PERIODSSC</p> <p>ABC_PD_PERIODC1001</p>	<p>Details defining the periodic hierarchy.</p> <p>See “<a href="#">modelRef_PD_PERIOD&lt;suffix&gt;</a>” on page 172.</p>

View Name	Description
<p><i>ModelRef_PD_SCENARIO</i>&lt;suffix&gt;</p> <p>where &lt;suffix&gt; is RC, SSC, or C&lt;cube configuration ID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_SCENARIORC</p> <p>ABC_PD_SCENARIOSSC</p> <p>ABC_PD_SCENARIOC1001</p>	<p>Details defining the types of available scenarios.</p> <p>See “<a href="#">modelRef_PD_SCENARIO&lt;suffix&gt;</a>” on page 172.</p>
<p><i>ModelRef_PD_YESNO</i>&lt;suffix&gt;</p> <p>where &lt;suffix&gt; is RC, SSC, or C&lt;cube configuration ID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PD_YESNORC</p> <p>ABC_PD_YESNOSSC</p> <p>ABC_PD_YESNOC1001</p>	<p>Dimensional definition for Boolean values: Text strings.</p> <p>See “<a href="#">modelRef_PD_YESNO&lt;suffix&gt;</a>” on page 173.</p>
<p><i>ModelRef_PF_</i>&lt;suffix&gt;</p> <p>where &lt;suffix&gt; is RC, SSC, or MSC_C&lt;cubeID&gt; for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PF_RC</p> <p>ABC_PF_SSC</p> <p>ABC_PF_MSC_C1001</p> <p>If you have checked <b>Also load tables into a library for the SAS LASR Analytic Server</b>, then, in addition to the tables using the naming convention above, tables in star-schema format are also generated that use the following naming convention:</p> <p><i>ModelRef_PF_</i>&lt;suffix&gt;</p> <p>where &lt;suffix&gt; is RCSTAR, SSCSTAR, or MSC_C&lt;cubeID&gt;STAR for a multi-stage contributions cube.</p> <p>For example:</p> <p>ABC_PF_RCSTAR</p> <p>ABC_PF_SSCSTAR</p> <p>ABC_PF_MSC_C1001STAR</p>	<p>Fact Table: Stages and member IDs for each step through contribution. Source table for cube generation.</p> <p>See “<a href="#">modelRef_PF_&lt;suffix&gt;</a>” on page 174 .</p>

## Procedure

To calculate the model **ITBMnMOD** and generate a cube:

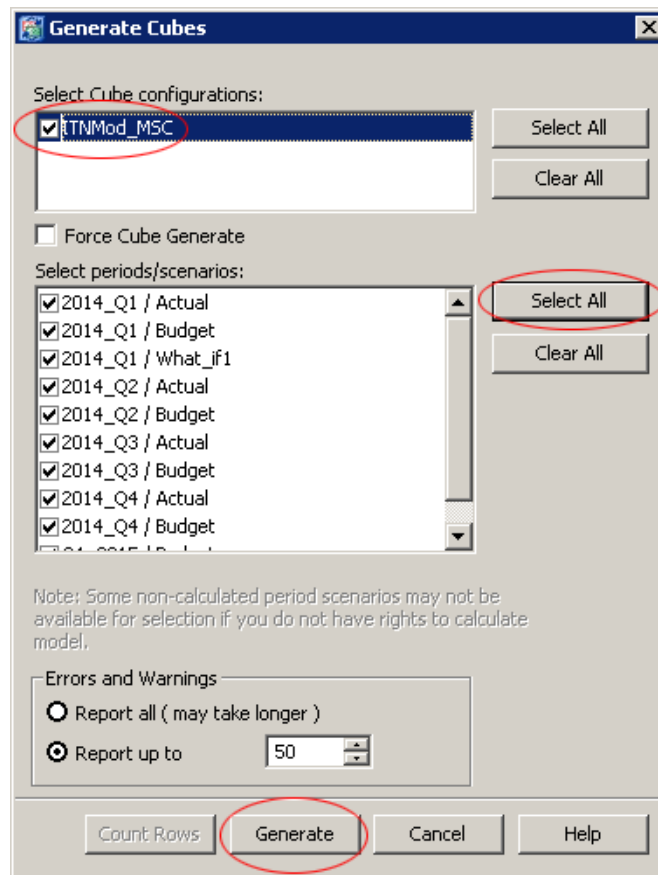
1. Make sure that the SAS LASR Analytic server is started. See [“Step 3 – Start the SAS LASR Analytic Server”](#) on page 319.
2. From the SAS Cost and Profitability Management client, make sure that the model **ITBMnMOD** is open and has a write lock.
3. Select **Model** ⇒ **Generate Cube**.

The Generate Cubes window opens.

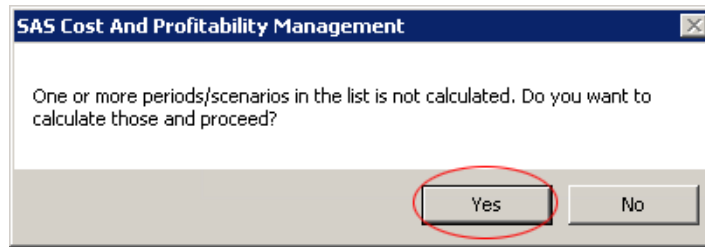
4. Select **ITNMod\_MSC** as your cube configuration.

(Or, select this cube configuration by its new name, such as **ITNMod\_MSC1**, if you chose to rename cube configurations during import. See [Step 4 on page 318](#).)

5. Click **Select All** to select all periods/scenarios.
6. Click **Generate**.

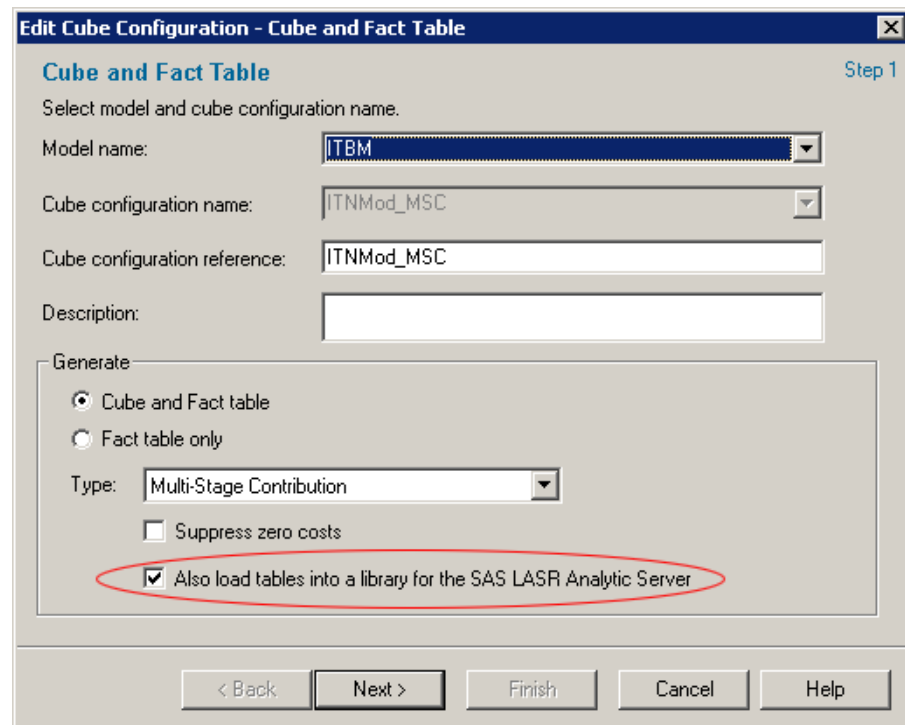


7. Click **Yes** when you are asked whether you want to calculate any periods/scenarios that have not yet been calculated.

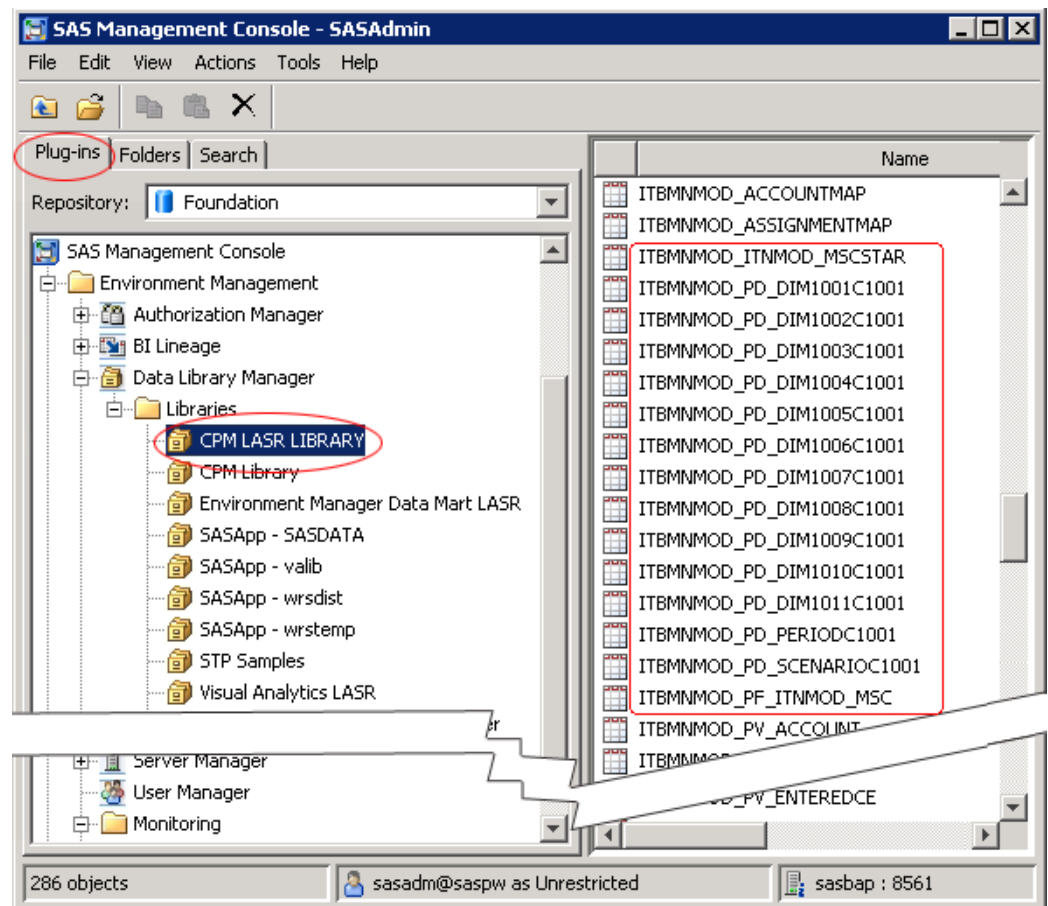


8. Repeat the previous steps to calculate and generate cubes for each of the remaining models.

*Note:* As you can see in the following picture, the option **Also load tables into a library for the SAS LASR Analytic Server** has been selected in the cube configuration **ITNMod\_MSC**. Selecting this option causes tables that are related to fact table generation to be loaded into the library specified for the SAS LASR Analytic server. This option is also selected in the three other cube configurations that you imported.

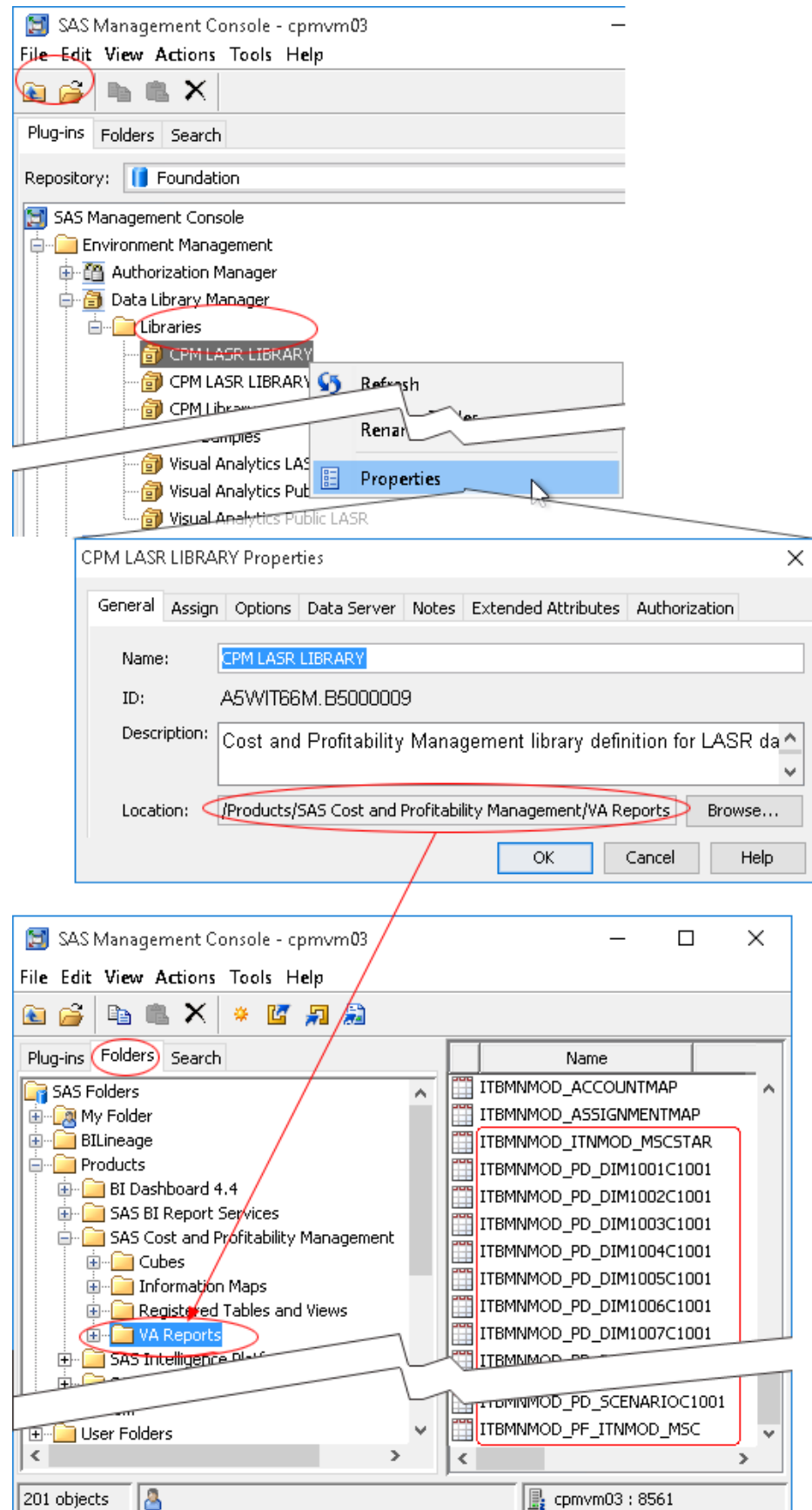


The following picture shows that metadata for the tables from cube generation is registered in the library specified for the SAS LASR Analytic server—**CPM LASR LIBRARY**. See “[Step 3 – Start the SAS LASR Analytic Server](#)” on page 319.



And, the following picture shows metadata for the same tables written to the folder assigned to that library.





You can also use the SAS Visual Analytics Administrator to verify that all of the model tables have been loaded into the LASR library.

1. Start the **SAS Visual Analytics Administrator**.
2. Click the **LASR Tables** tab.
3. Scroll down to see the tables for the model: **ITBMNMOD\_ACCOUNTMAP**, **ITBMNMOD\_ASSIGNMENTMAP**, and so on. You can notice the following (the following values can differ for your organization):

Location

**/Products/SAS Cost and Profitability Management/VA Reports**

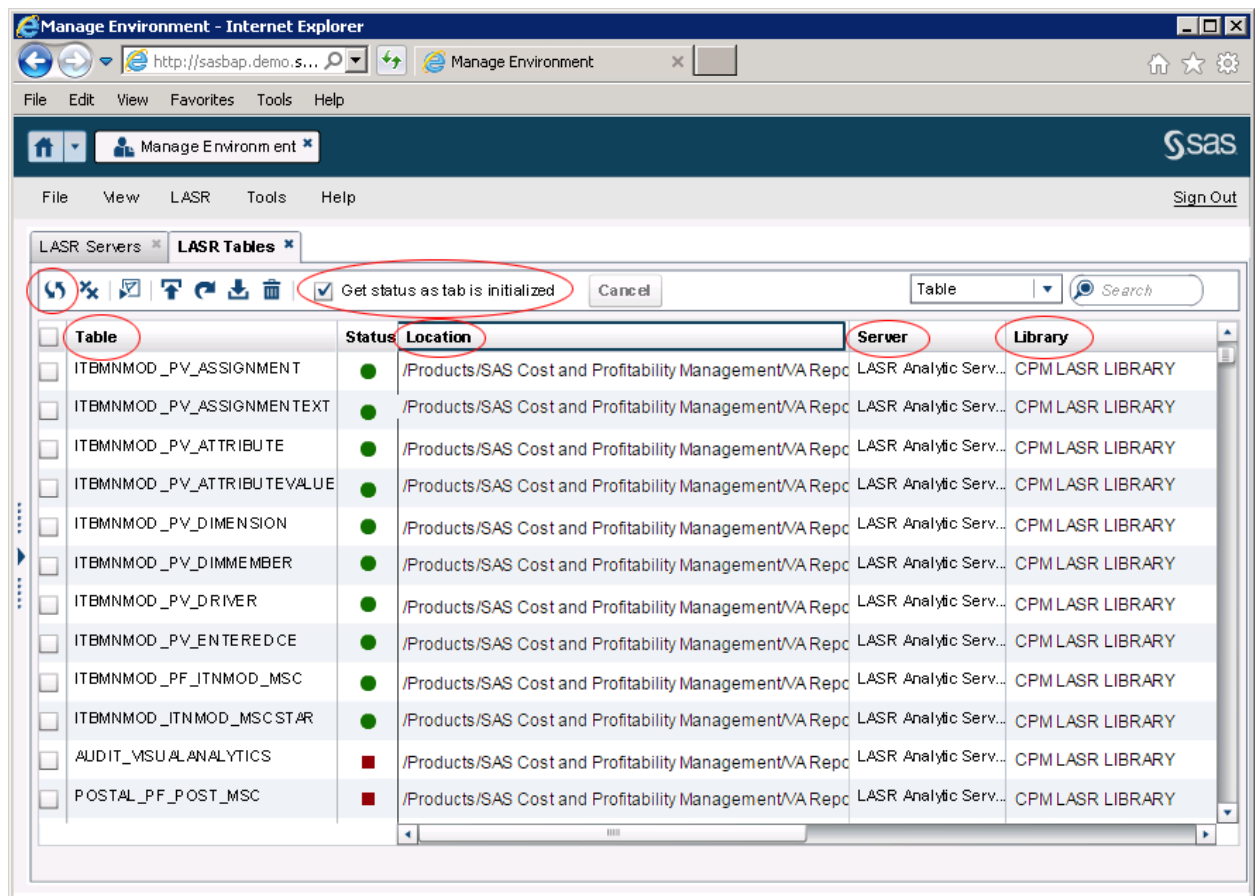
Server

**LASR Analytic Server — cpmvm05**

Library

**CPM LASR LIBRARY**

*Note:* If the LASR Server is stopped, then the tables are no longer in memory and you will have to re-export them again to load them into the LASR library. The **Status** column shows whether a table is loaded or not. Click the **Refresh** button to see whether a table is loaded. You can select **Get status as tab is initialized** so that the status of a table is automatically refreshed when the **LASR Tables** tab is opened.



## Step 6 – Import the Sample Reports

### Overview

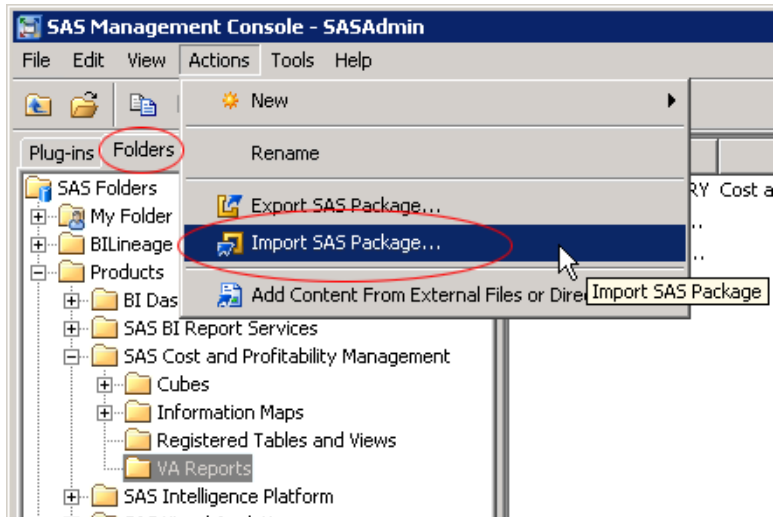
Now that you have started the SAS LASR Analytic server and loaded your SAS Cost and Profitability Management model data into a LASR library, you can view the sample reports in SAS Visual Analytics as soon as you have imported the reports. To do this you use SAS Management Console to import the appropriate SAS Package file depending on your database as shown in the following table:

SAS Package File		Contained Reports
SQL	SQL_BankVA.spk	Bank_Account_View
		Bank_Dashboard
		Bank_Fact_View
Oracle	Orcl_FASHBI.spk	FASHBI_Acct_Analysis
SQL	SQL_FASHBI.spk	FASHBI_Dashboard
		FASHBI_Fact_Analysis
SQL	SQL_Hospital.spk	Hospital_View
Oracle	Orcl_ITBMVA.spk	ITBM_Account_View
PostgreSQL	Post_ITBMVA.spk	ITBM_Dashboard
SQL	SQL_ITBMVA.spk	ITBM_Fact_View

### Procedure

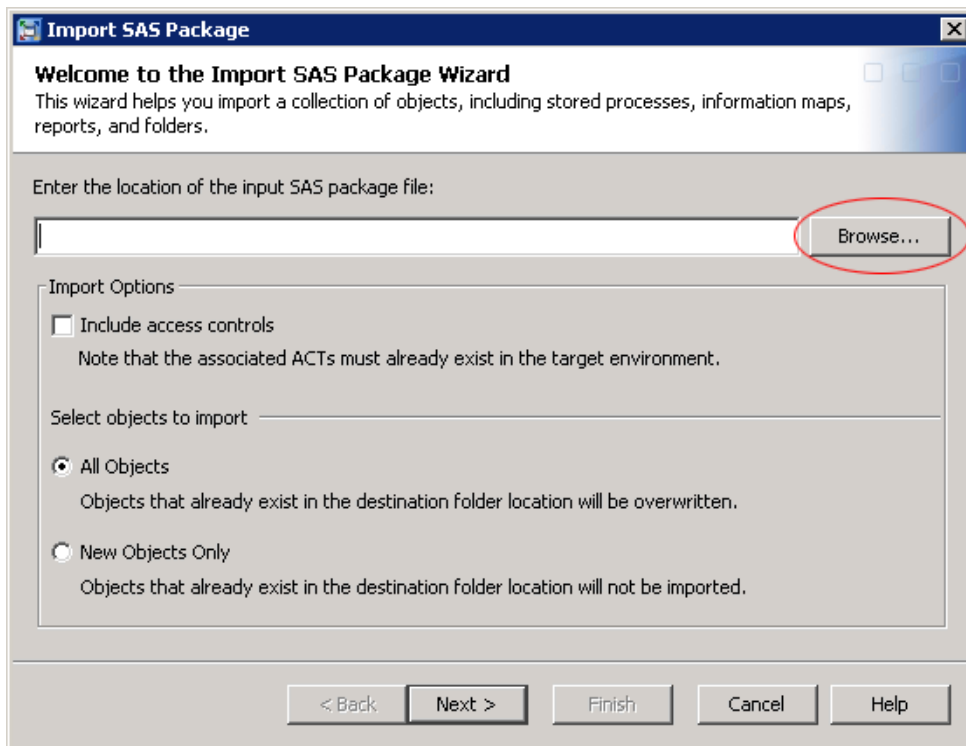
To import the sample reports, do the following.

1. Open SAS Management Console and click the **Folders** tab.
2. Select **Actions** ⇒ **Import SAS Package**.



The Import SAS Package window opens.

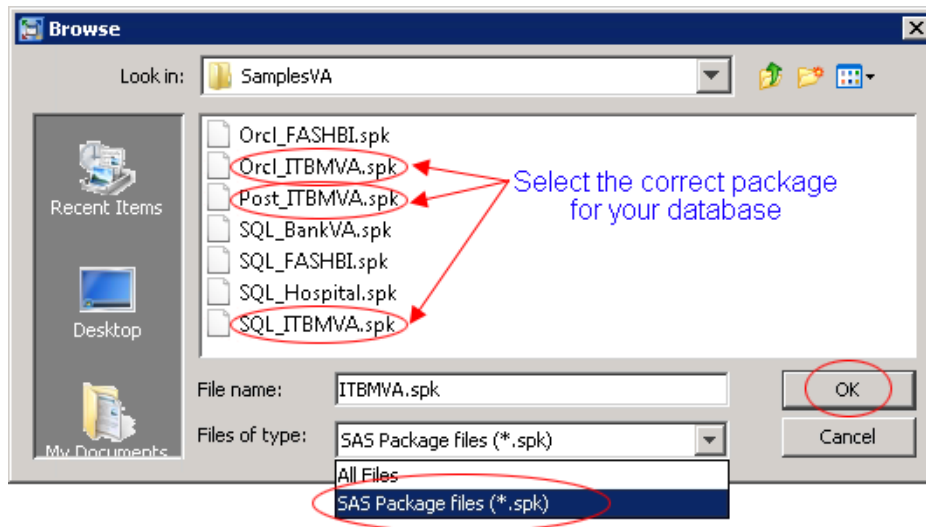
3. Click **Browse**.



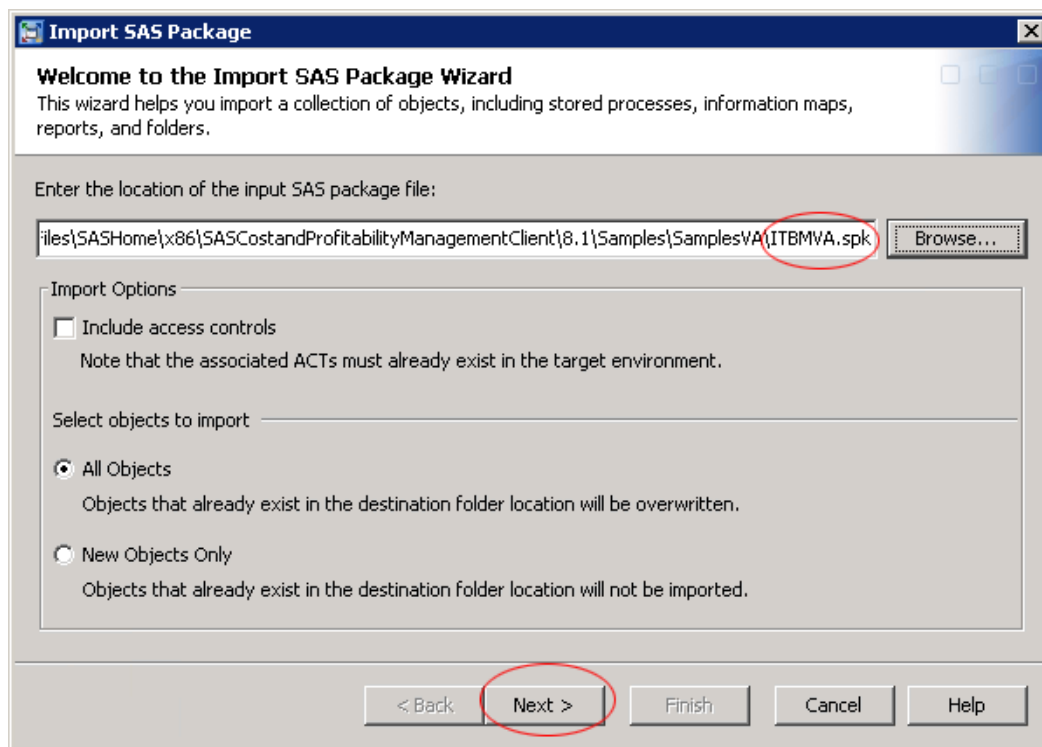
4. Navigate to the location of the sample models and reports under the **SASHome** directory, for example: **C:\Program Files\SASHome\X86\SASCostandProfitabilityManagementClient\8.1\Samples\SamplesVA**.

Change the file type to **SAS Package files (\*.spk)**.

Select **ITBMVA.spk** and click **OK**.



5. The package is ready to be imported. Click **Next**.

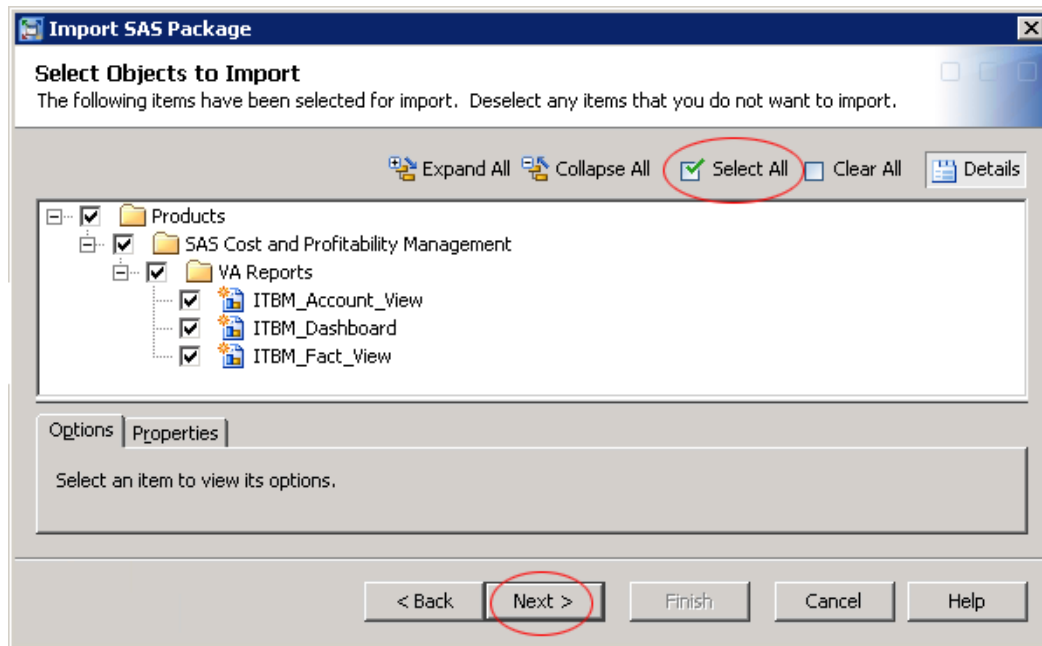


The Select Objects to Import window opens.

6. Make sure that **Select all** is selected, and then click **Next**.

The SAS Package file contains the following reports which will be placed in the VA Reports folder:

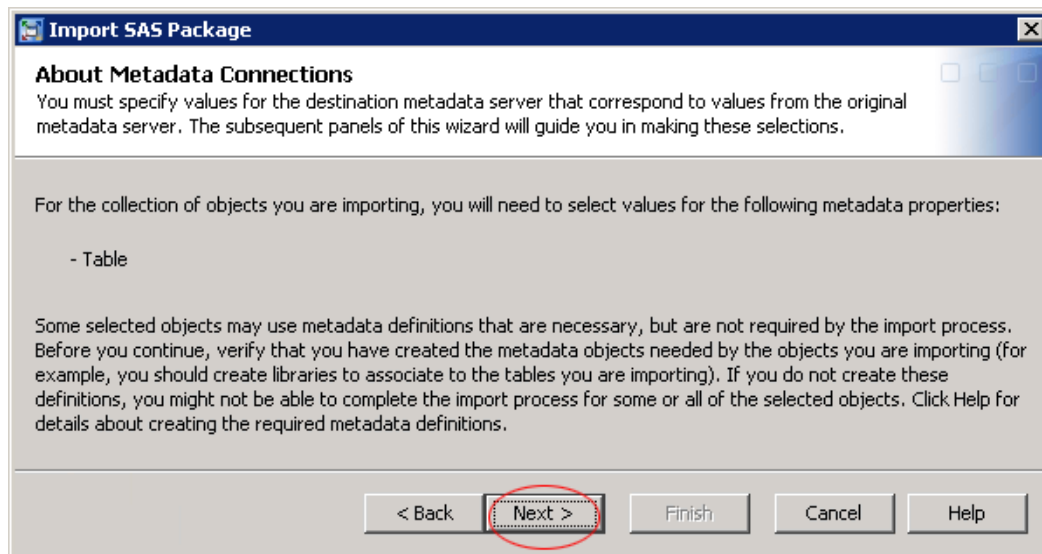
- ITBM\_Account\_View
- ITBM\_Dashboard
- ITBM\_Fact\_View




The About Metadata Connections window opens.

7. This window informs you that you must specify values for the destination metadata server (**LASR Analytic Server - cpmvm03**) that correspond to values from the original metadata server—the metadata server that was used when the sample reports were created.

Click **Next**.

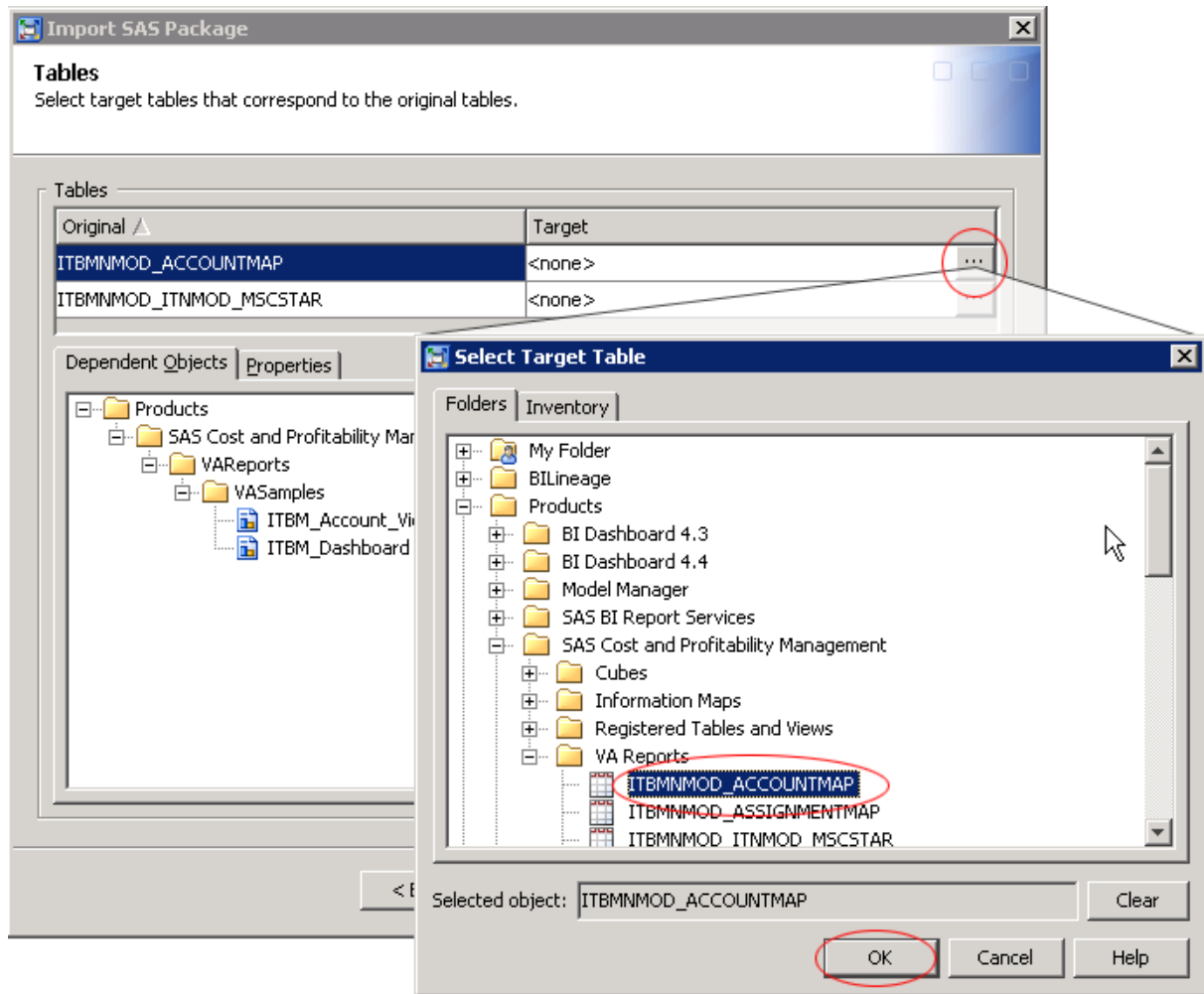


The Tables window opens.

8. Click the Browse button  and navigate to the folder, **/Products/SAS Cost and Profitability Management/VA Reports**, specified for the library **CPM LASR LIBRARY** (see “[Procedure](#)” on page 323).

Select target table names to match the original table name as shown in the following table:

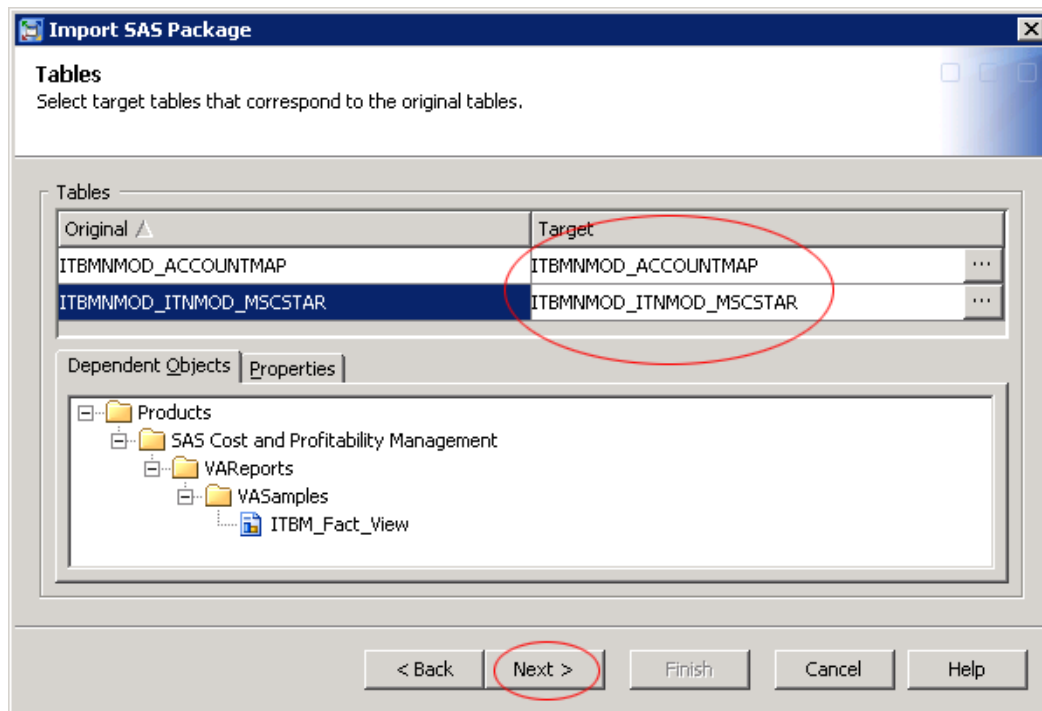
Table name in original metadata server	Table name in current metadata server
ITBMNMOD_ACCOUNTMAP	ITBMNMOD_ACCOUNTMAP
ITBMNMOD_ITNMOD_MSCSTAR	ITBMNMOD_ITNMOD_MSCSTAR



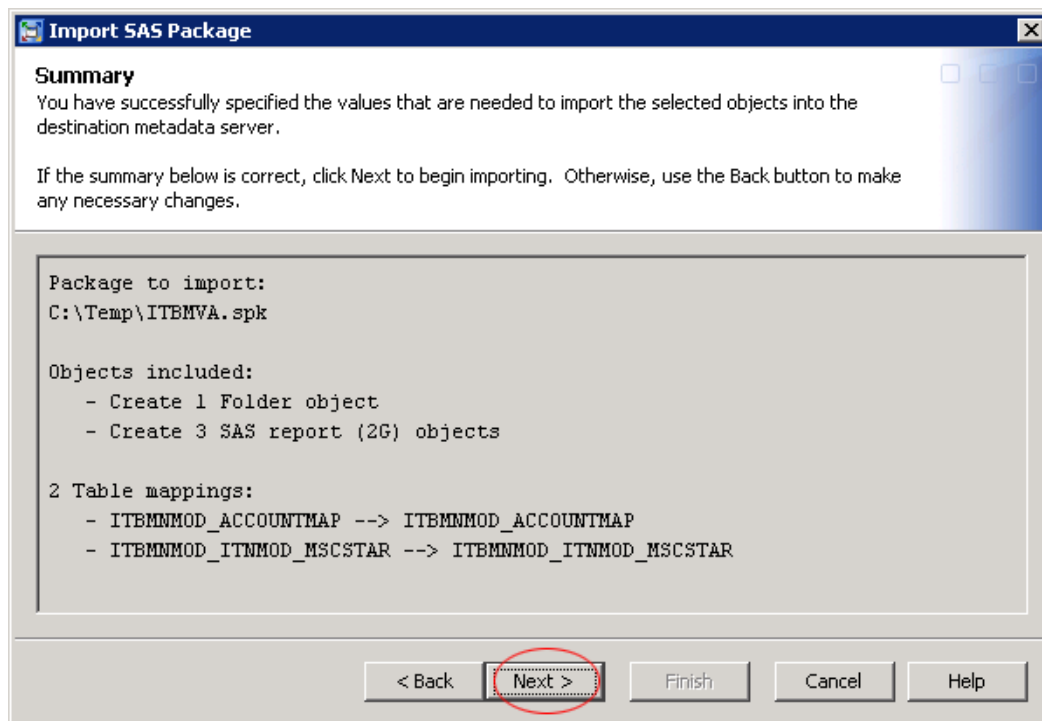
If you were to import each report individually, you would see that each report uses data from the table shown below:

Report Name	Table Name
ITBM_Account_View	ITBMNMOD_ACCOUNTMAP
ITBM_Dashboard	ITBMNMOD_ACCOUNTMAP
ITBM_Fact_View	ITBMNMOD_ITNMOD_MSCSTAR

- After you have made your selections, ensure that the window looks like the following and click **Next**.

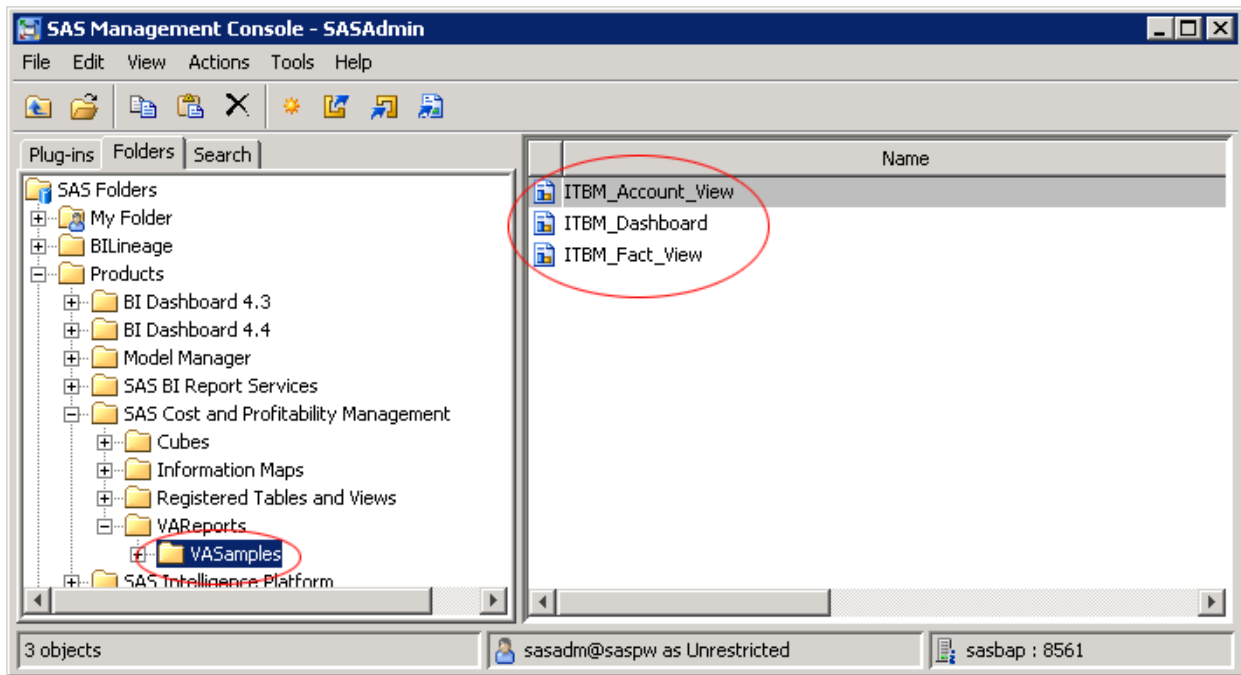


10. Review your selections, and then click **Next**.



11. The sample reports are imported as shown in the following picture.





12. Repeat the steps above for each of the SAS Package files containing sample reports. After finishing, you will have imported the following reports:

SAS Package File		Contained Reports
SQL	SQL_BankVA.spk	Bank_Account_View
		Bank_Dashboard
		Bank_Fact_View
Oracle	Orcl_FASHBI.spk	FASHBI_Acct_Analysis
SQL	SQL_FASHBI.spk	FASHBI_Dashboard
		FASHBI_Fact_Analysis
SQL	SQL_Hospital.spk	Hospital_View
Oracle	Orcl_ITBMVA.spk	ITBM_Account_View
PostgreSQL	Post_ITBMVA.spk	ITBM_Dashboard
SQL	SQL_ITBMVA.spk	ITBM_Fact_View

You can now open the SAS Visual Analytics Viewer to view the sample reports, or, even better, the SAS Visual Analytics Designer to view the reports and see how they are constructed. The following sections show the sample reports for the ITBMnMOD model in the SAS Visual Analytics Designer:

- [“View the ITBM\\_Dashboard Report” on page 344](#)
- [“View the ITBM\\_Fact\\_View Report” on page 357](#)
- [“View the ITBM\\_Account\\_View Report” on page 363](#)

## Step 7 – View the Sample Reports

### Overview

You can now view the sample reports. The following sections describe viewing sample reports for the ITBMnMOD model:

#### ITBM\_Dashboard Report

See [“View the ITBM\\_Dashboard Report” on page 344](#).

#### ITBM\_Fact\_View Report

See [“View the ITBM\\_Fact\\_View Report” on page 357](#).

#### ITBM\_Account\_View Report

See [“View the ITBM\\_Account\\_View Report” on page 363](#).

*Note:*

- Because there are minor differences in the way that the sample reports were built on the different databases supported by SAS Cost and Profitability Management, the pictures of the sample reports in this chapter might not match exactly the sample reports that you view depending on your database. For example, the sorting order for columns might be different as well as other minor differences.
- If you encounter a problem viewing the sample reports, see [“Diagnosing Potential Problems” on page 342](#).

### Diagnosing Potential Problems

The following two conditions can result in your being unable successfully to view the sample reports.

1. When you import a model and specify its name, it is important that the Name and Reference that you specify match the Name and Reference specified for the model in the cube configuration that you also import. Otherwise, the cube configuration will not be used correctly when you attempt to generate a cube for the model.

The following table shows the Name and Reference for each model that you import and the Name and Reference of corresponding cube configuration.

Cube Configuration Name	Cube Configuration Reference	Model Name	Model Reference
Bank_MSC	Bank_MSC	Banking	Banking
FASHNMSC	FASHNMSC	FASHNMod	FASHNMod
Hosp_MSC	Hosp_MSC	Hospital	Hospital
ITNMod_MSC	ITNMod_MSC	ITBMnMOD	ITBMnMOD

If the name that you assign an imported model does not match the name assigned by its cube configuration, then what happens is the following. When you attempt to generate a cube for the model, cube generation does not recognize the cube

configuration assigned to it. Then, if you attempt to remedy the problem by opening the intended cube configuration and assigning it to your incorrectly named model, the levels to which you can drill to revert to level 2 in that cube configuration—insufficient depth for the sample reports.

To remedy this problem, if you encounter it:

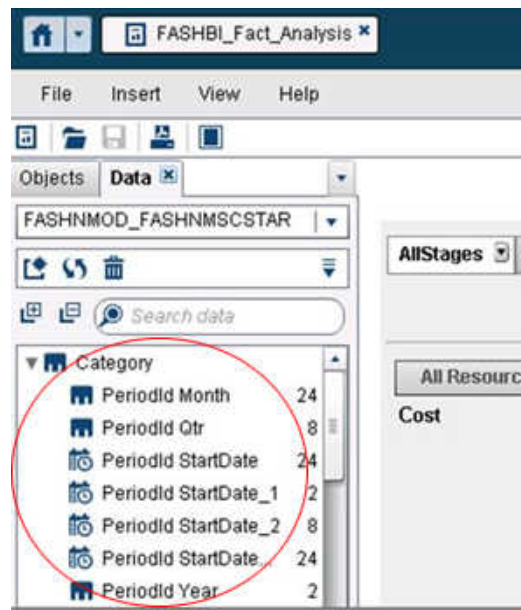
- a. Re-import the sample models and name them correctly. See [“Step 1 – Import the Models” on page 311](#).
  - b. Re-import the cube configurations. See [“Step 2 – Import the Cube Configurations” on page 316](#).
  - c. Re-calculate the models and generate cubes. See [“Step 5 – Calculate the Models and Generate Cubes” on page 327](#).
2. When you push data to LASR for the sample models, it is important that Level Names for Periods and Scenarios be the same as when the sample models were created. The sample models were created using a Period table and a Scenario table in which Level Names were the following:

	Period Table	Scenario Table
Level Names	Year	Top Scenario
	Quarter	Scenario L2
	Month	Personal Scenario

The sample reports assume that these names can be found among the data that you exported to LASR for the reports. However, Period tables and Scenario tables are shared among models. If, prior to exporting your data to LASR, you do something to change these Level Names in the Period table or Scenario table (such as importing a model that results in changing the tables) then the Level Names that are exported to LASR are not the names that the sample reports expect.

You can recognize this problem in two ways:

- In the SAS Cost and Profitability Management client, select **Tools** ⇒ **Manage Periods** and then click **Levels**. If you don’t see the Period Level Names shown above, then you can suspect that your Period table was changed after a sample model was imported and before its data was pushed to LASR.  
  
Select **Tools** ⇒ **Manage Scenarios** and click **Levels**. If you don’t see the Scenario Level Names shown above, then you can suspect that your Scenario table was changed after a sample model was imported and before its data was pushed to LASR.
- Open a sample report and look at the available categories. If they do not look like the following, then you can suspect that your Period table or Scenario table was changed after a sample model was imported and before its data was pushed to LASR.



To remedy this problem, if you encounter it:

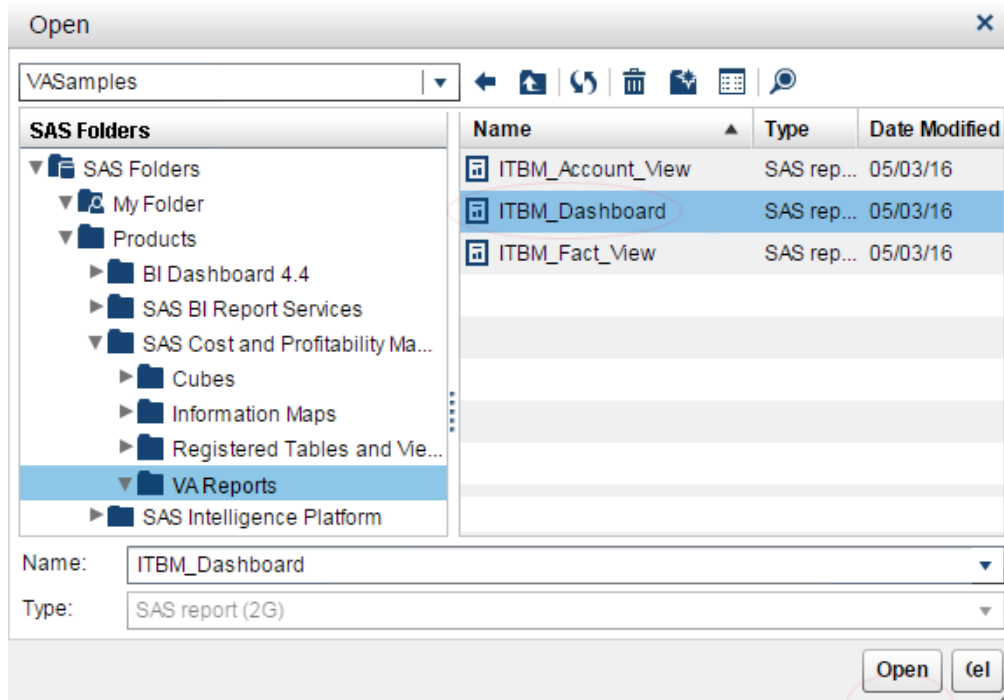
- a. Re-import the sample models. See [“Step 1 – Import the Models”](#) on page 311.
- b. Re-export and register tables. See [“Step 4 – Export and Register Tables”](#) on page 322.
- c. Re-calculate the models and generate cubes. See [“Step 5 – Calculate the Models and Generate Cubes”](#) on page 327.

---

## View the ITBM\_Dashboard Report

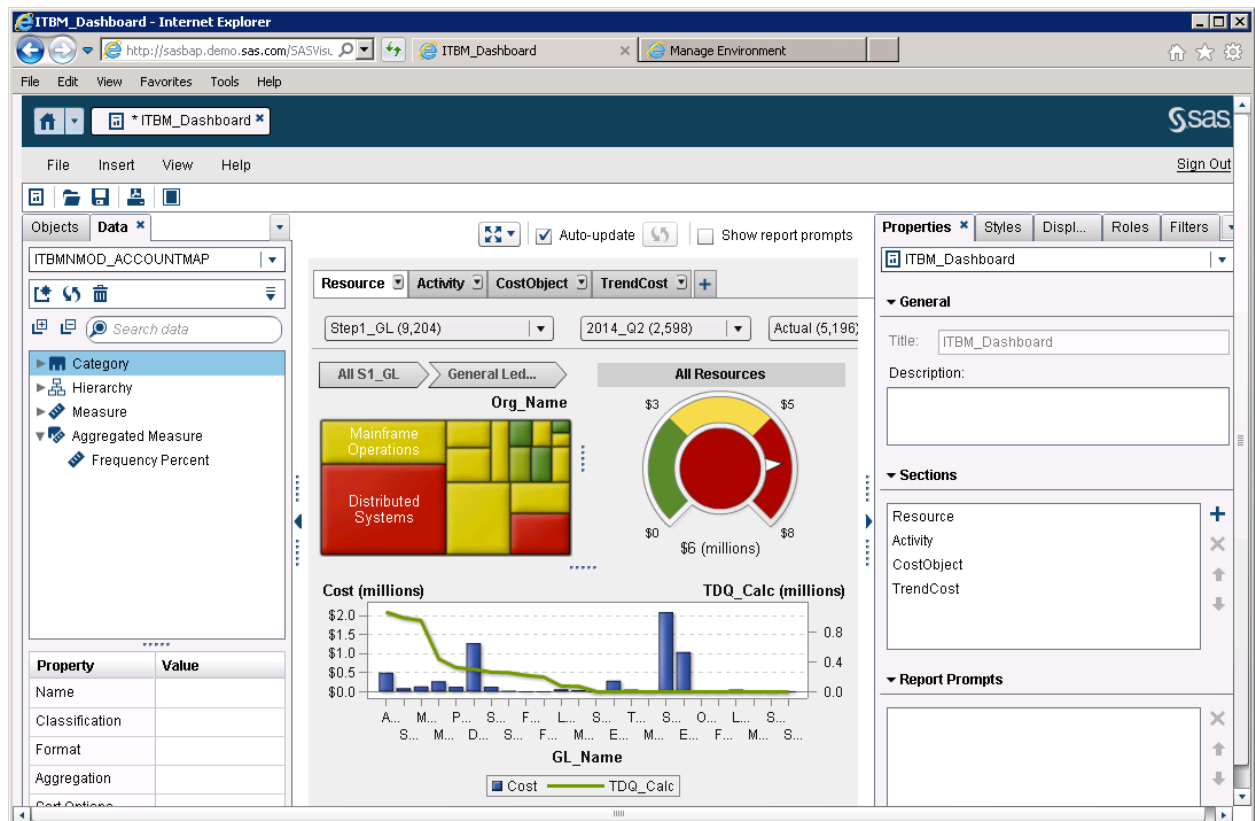
### Open the Report

1. In SAS Visual Analytics Designer, select **File** ⇒ **Open**.
2. Navigate to **SAS Folders/Products/SAS Cost and Profitability Management/VA Reports/VASamples** and select **ITBM\_Dashboard**.



3. Click **Open**.

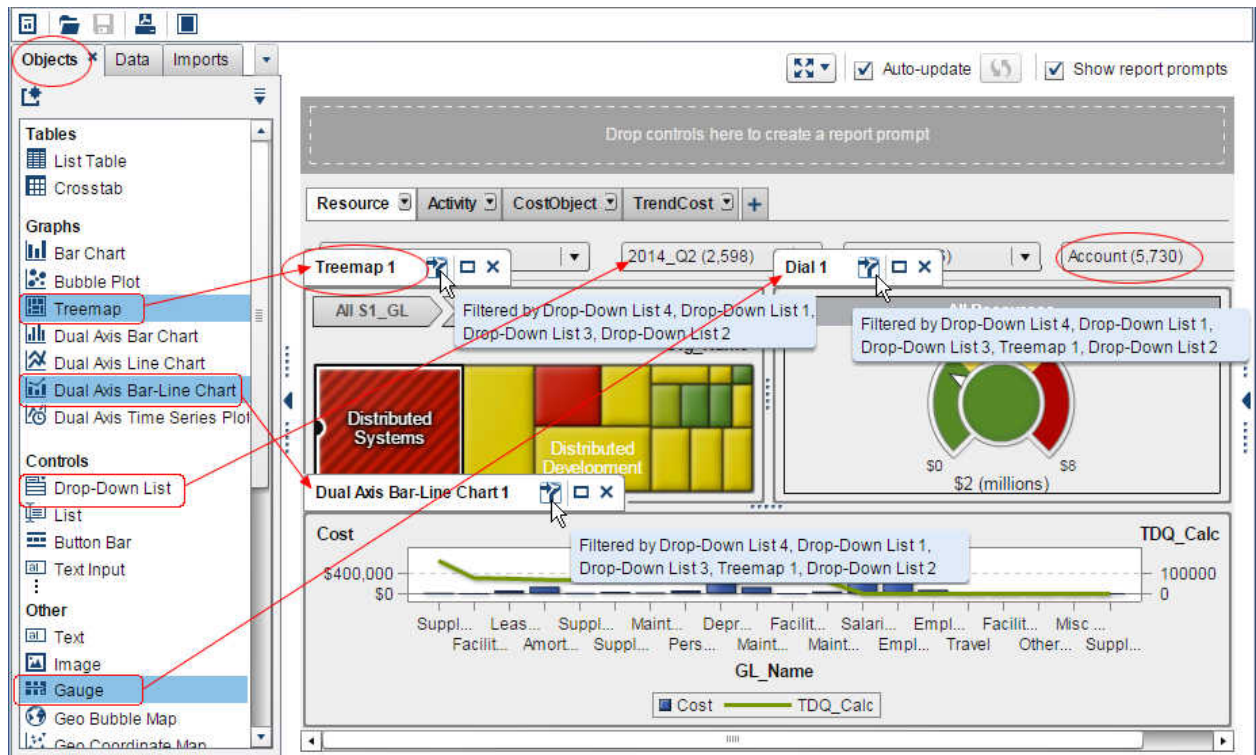
The **ITBM\_Dashboard** sample report opens in designer view.



## Objects

Click the **Objects** tab to see the things that you can add to a report. The following picture shows that the **ITBM\_Dashboard** report includes, among other objects, the following:

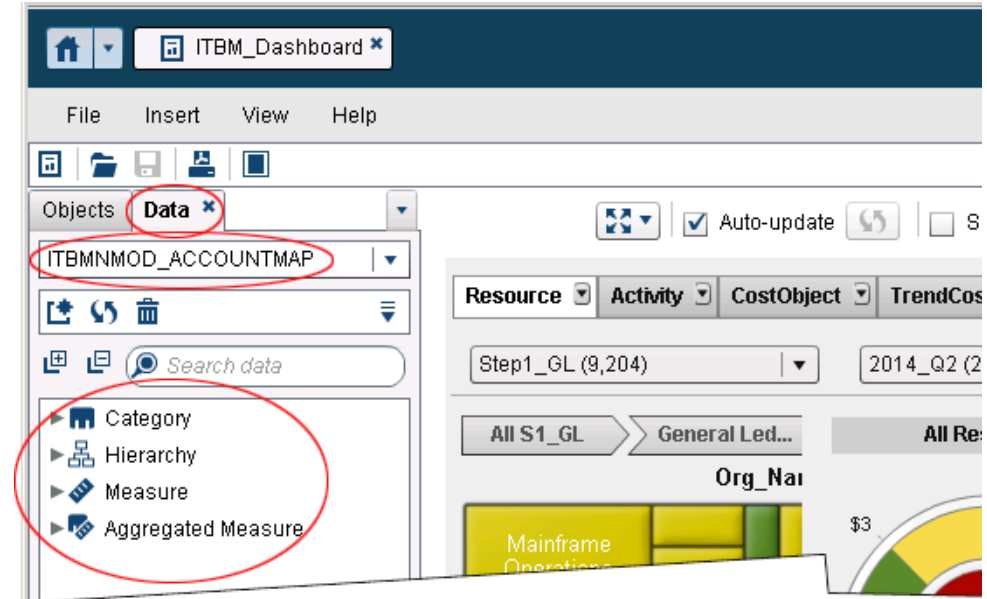
- Treemap
- Dual Axis Bar-Line Chart
- Drop-Down List
- Gauge (Dial type)



Double-click or drag an object from the **Objects** tab to the work area to add it to a report.

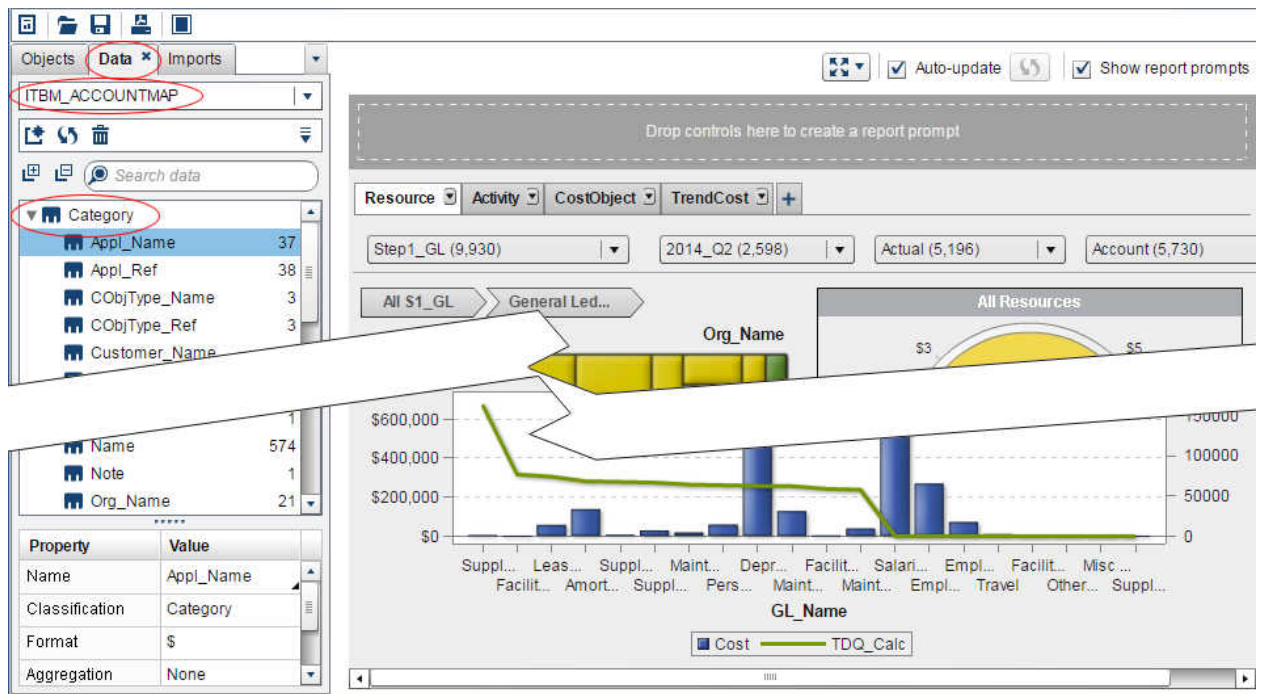
## Data

Click the **Data** tab to see the tables whose columns are available to be added to your report. The **Data** tab shows categories, hierarchies, measures, and aggregated measures.



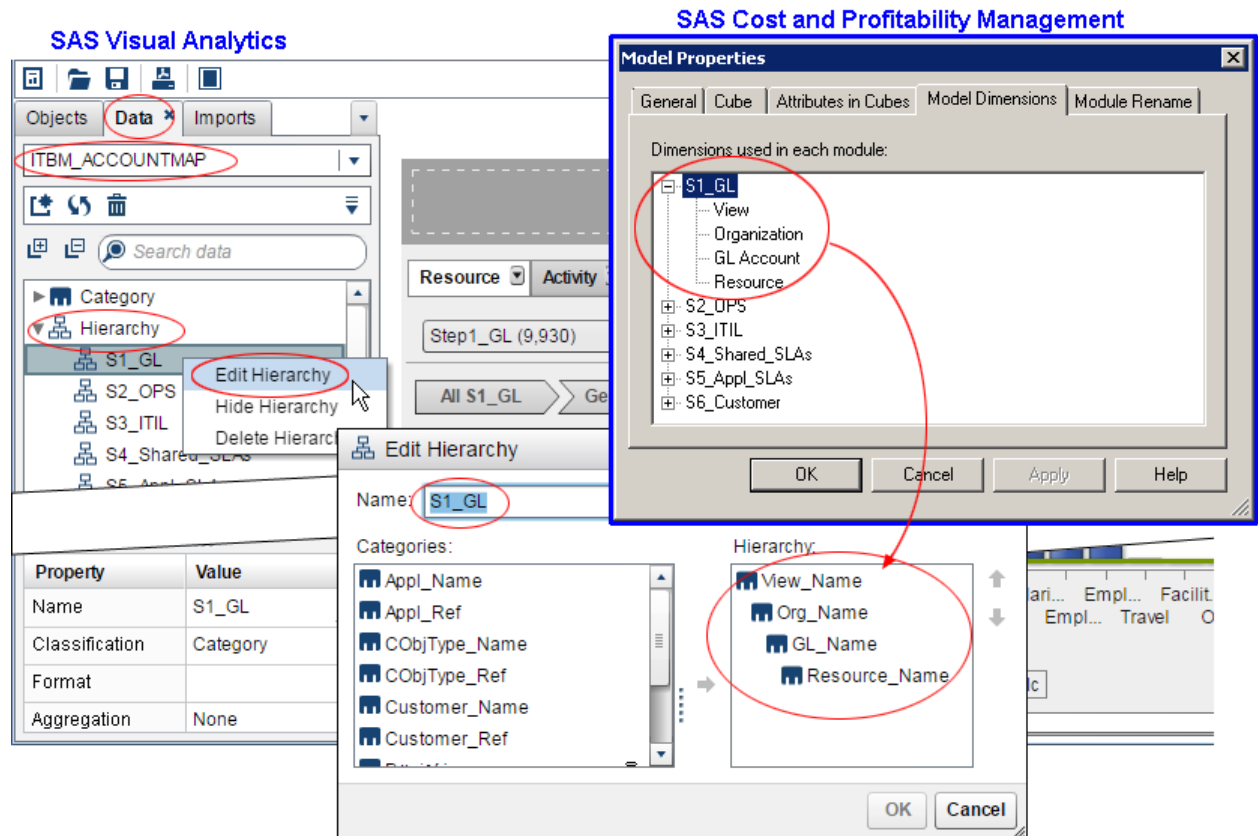
## Categories

Categories are the text fields in a table. The following picture shows some of the categories in **ITBMNMOD\_ACCOUNTMAP**.



## Hierarchies

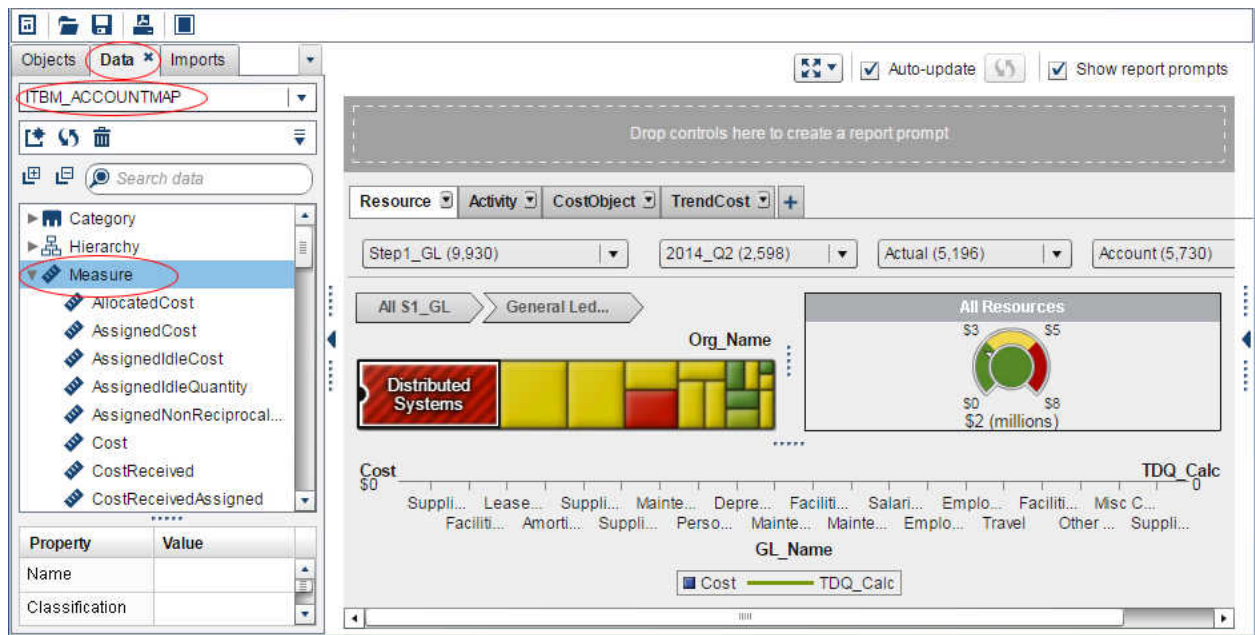
A hierarchy is an arrangement of parent-child relationships among categories. Creating hierarchies enables you to add drill-down functionality to your visualizations. You can see from the following picture that the **S1\_GL** hierarchy in the **ITBM\_Dashboard** report, mirrors the hierarchy of dimensions in the **S1\_GL** module of the **ITBMnMOD** model.



## Measures

Measures are numeric fields in a table. The following picture shows some of the measures in **ITBMNMOD\_ACCOUNTMAP**.

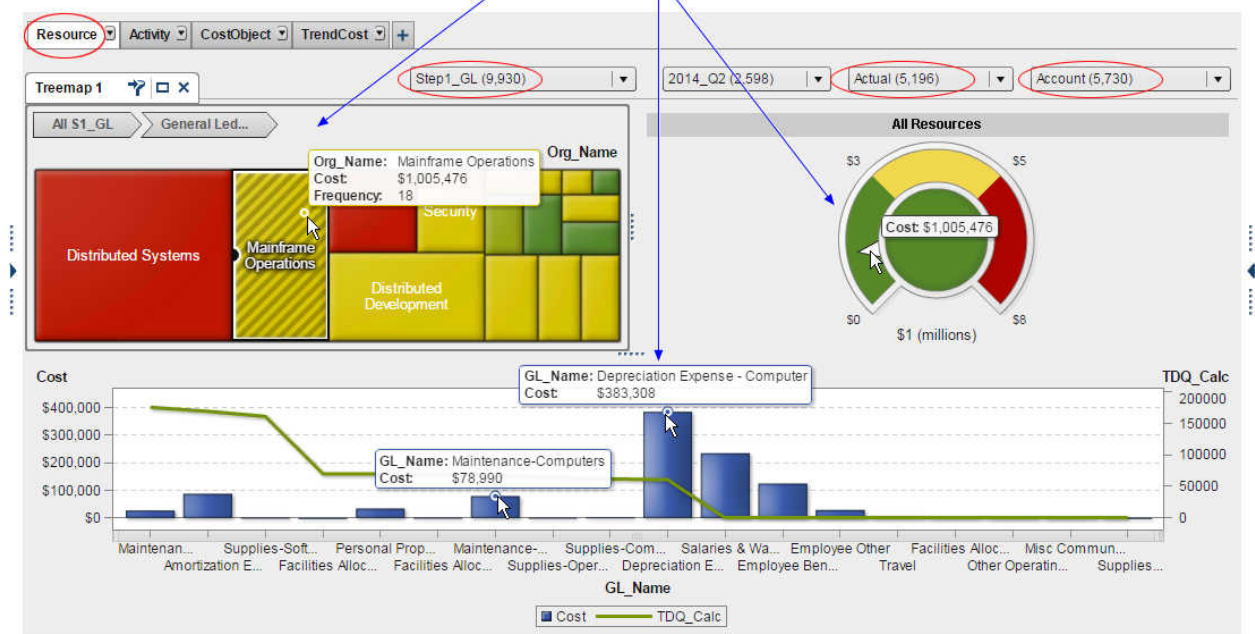




### Report Tab 1: Resource

The first tab in the **ITBM\_Dashboard** report, the **Resource** tab, shows the relative contribution of general-ledger (resource) costs in the model. Clicking on the treemap object in the top-left part of the tab changes the display of the other objects in the tab. In the following picture a user has clicked on the **Mainframe Operations** tile of the treemap which results in the dial on the top-right showing the cost of mainframe operations, and the bar-line chart in the bottom of the tab showing costs for each subdivision within mainframe operations.

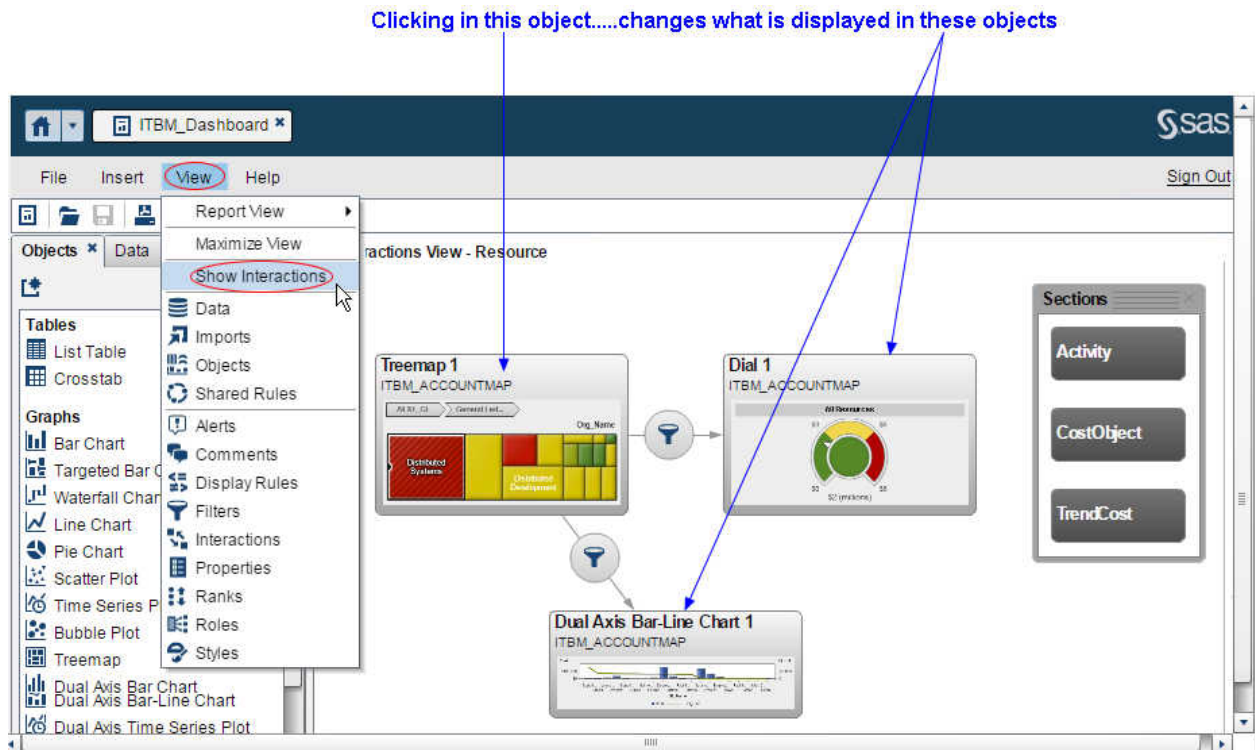
When a user clicks somewhere on this graph, these graphs change accordingly



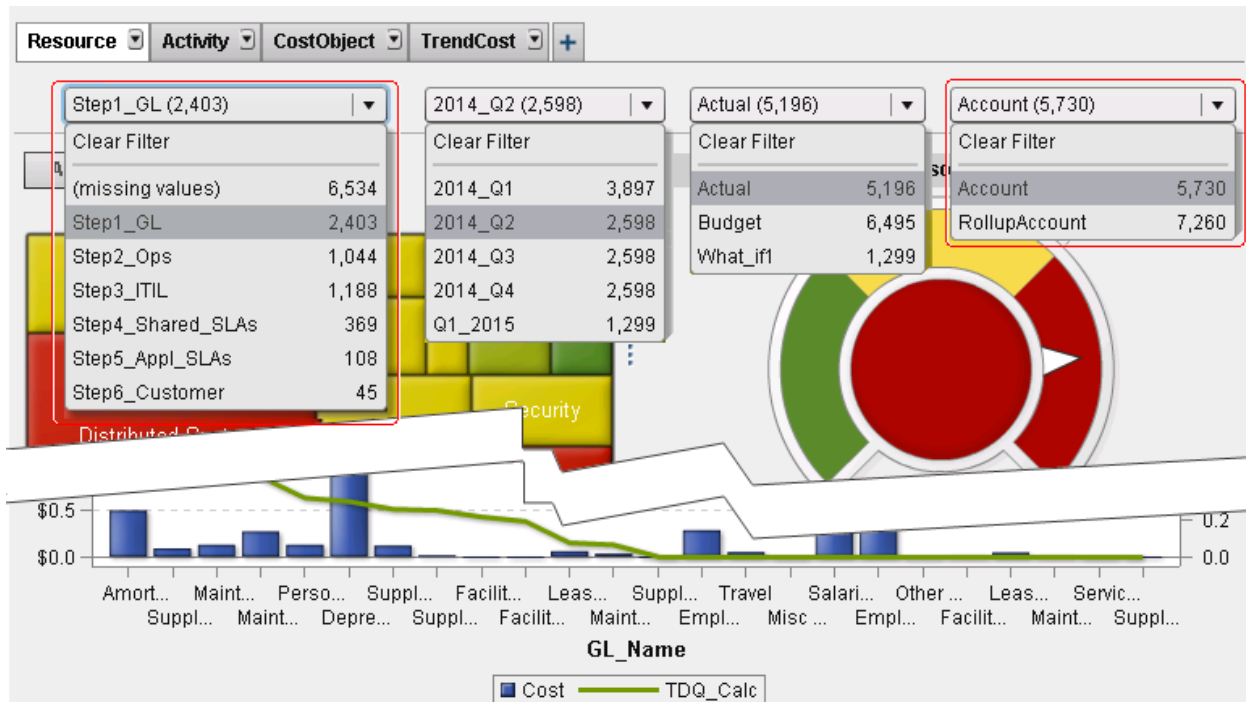
To establish similar interactions between objects in a report, select **View** ⇒ **Show Interactions** and simply draw a connection between the source and target report objects.



The following picture shows that, in the Resource tab, a connection has been drawn between the source object **Treemap 1** and two target objects: **Dial 1** and **Dual Axis Bar-Line Chart 1**.

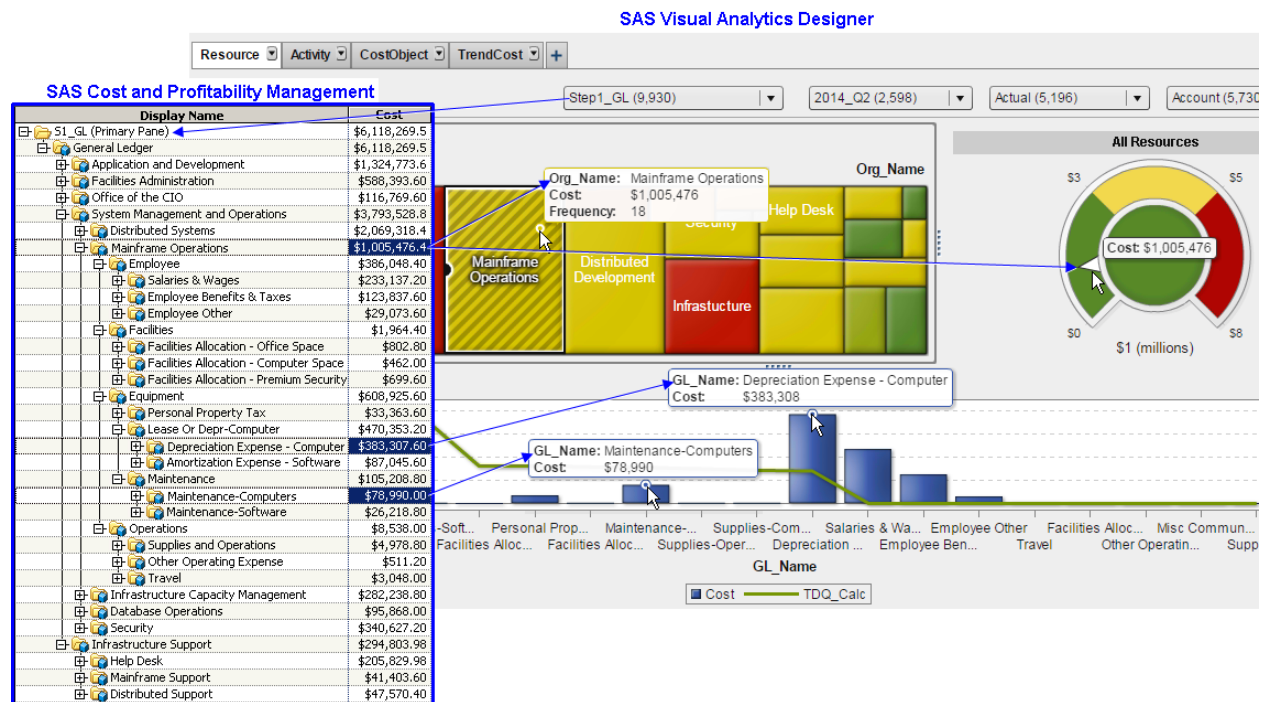


The next picture shows the options presented by the drop-down lists in the report. You can see that the first drop-down list provides the user a choice of modules (**Step1\_GL**, **Step2\_Ops**, **Step3\_ITIL**, and so on) in the model. For each module, general-ledger (resource) costs are displayed.



The next picture shows the relationship between the hierarchical module view provided by SAS Cost and Profitability Management and the graphical view provided by the **ITBM\_Dashboard** report. If you look at the part of the picture that shows the SAS Visual Analytics Designer, you can see that the treemap at the top-left of the report shows the relative cost of each of the sub-accounts under **System Management and Operations**—sub-accounts such as **Mainframe Operations** and **Infrastructure**. You can also see that **Mainframe Operations** is selected and that its cost is displayed by the dial gauge on the right of the report. And, at the bottom of the report, a bar chart shows the contributing cost of each of the sub-accounts of **Mainframe Operations**—sub-accounts such as **Depreciation Expense – Computer** and **Maintenance – Computers**.

Now, if you look at the part of the picture that shows SAS Cost and Profitability Management, you can see a module view of the **S1\_GL** module—the same module that is shown in the SAS Visual Analytics Designer (though named **Step1\_GL**). You can also see the **Mainframe Operations** account with its sub-accounts, including **Depreciation Expense - Computer** and **Maintenance-Computers**. You can also verify that the costs displayed in the module view of SAS Cost and Profitability Management coincide with the costs displayed by the SAS Visual Analytics Designer.



Instead of merely selecting a tile in the treemap, you can double-click a tile to drill down into the account that it represents. The following picture shows the results of double-clicking the **Mainframe Operations** tile. You can see from the picture that the sub-accounts of **Mainframe Operations** are displayed.



## Report Tab 2: Activity

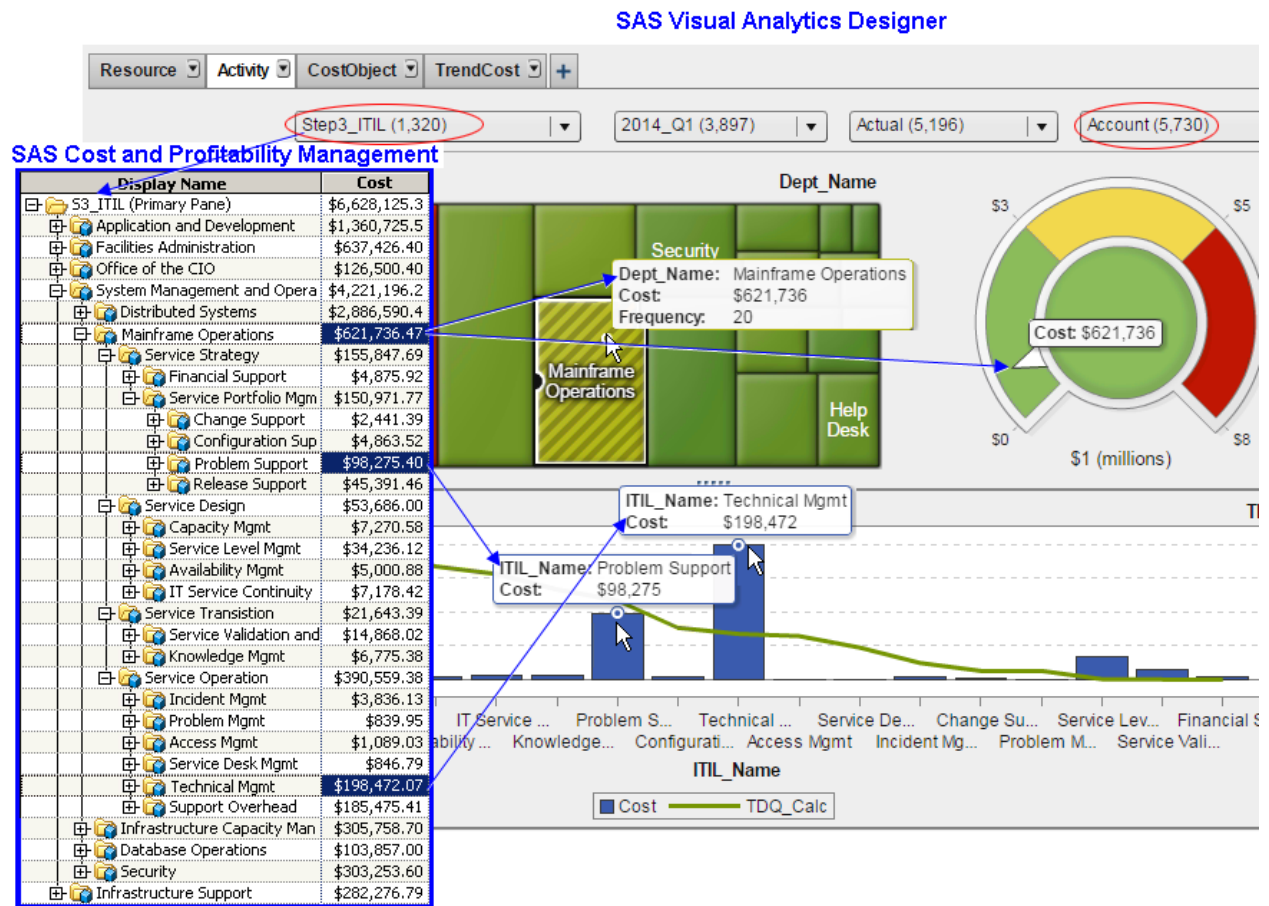
The second tab in the **ITBM\_Dashboard** report, the **Activity** tab, is very similar to the **Resource** tab. Whereas the **Resource** tab shows the relative contribution of general-ledger (resource) costs in the model, the **Activity** tab shows the relative contribution of activity costs in the model.

In the following picture if you look at the part that shows the SAS Visual Analytics Designer, you can see that the treemap at the top-left of the report shows the relative cost of each of the sub-accounts under **System Management and Operations**—sub-accounts such as **Mainframe Operations** and **Security**. You can also see that the sub-account **Mainframe Operations** is selected and that its cost is displayed by the dial gauge on the right of the report. And, at the bottom of the report, a bar chart shows the contributing cost of each of the sub-accounts of **Mainframe Operations**—sub-accounts such as **Problem Support** and **Technical Mgmt**.

Now, if you look at the part of the picture that shows SAS Cost and Profitability Management, you can see a module view of the **S3\_UTIL** module—the same module that is shown in the SAS Visual Analytics Designer (though named **Step3\_UTIL**). You can also see the **Mainframe Operations** account with its sub-accounts including **Problem Support** and **Technical Mgmt**. You can also verify that the costs displayed in the



module view of SAS Cost and Profitability Management coincide with the costs displayed by the SAS Visual Analytics Designer.

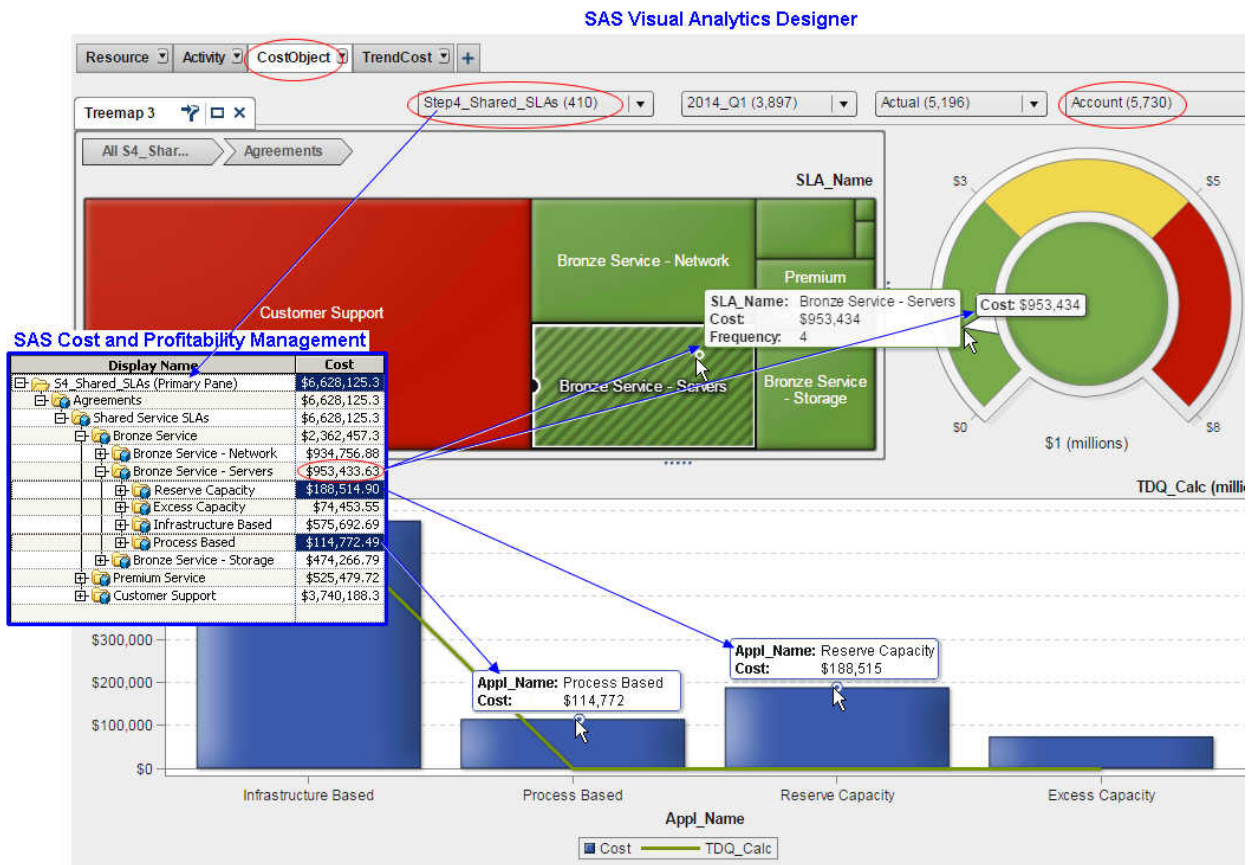


### Report Tab 3: CostObject

The third tab in the **ITBM\_Dashboard** report, the **CostObject** tab, is very similar to the previous two tabs. The **CostObject** tab shows the relative contribution of service agreement costs in the model.

In the following picture if you look at the part that shows the SAS Visual Analytics Designer, you can see that the treemap at the top-left of the report shows the relative cost of each of the sub-accounts under **Shared Service SLAs**—sub-accounts such as **Customer Support** and **Bronze Service - Servers**. You can also see that the sub-account **Bronze Service - Servers** is selected and that its cost is displayed by the dial gauge on the right of the report. And, at the bottom of the report, a bar chart shows the contributing cost of each of the sub-accounts of **Bronze Service - Servers**—sub-accounts such as **Reserve Capacity** and **Process Based**.

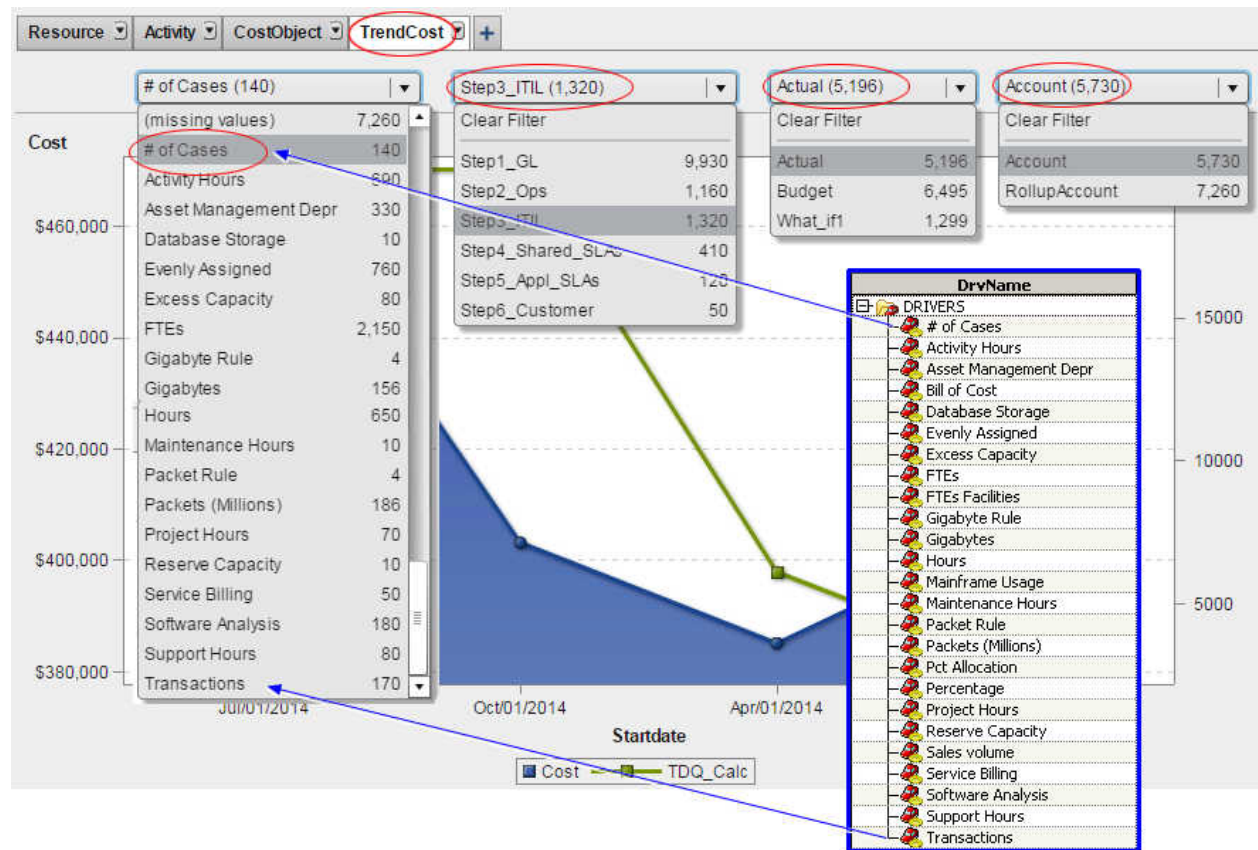
Now, if you look at the part of the picture that shows SAS Cost and Profitability Management, you can a module view of the **S4\_Shared\_SLAs** module—the same module that is shown in the SAS Visual Analytics Designer (though named **Step4\_Shared\_SLAs**). You can also see the **Bronze Service - Servers** account with its sub-accounts including **Reserve Capacity** and **Process Based**. You can also verify that the costs displayed in the module view of SAS Cost and Profitability Management coincide with the costs displayed by the SAS Visual Analytics Designer.



### Report Tab 4: TrendCost

The fourth tab in the **ITBM\_Dashboard** report, **TrendCost**, focuses on drivers. The following picture shows the following drop-down menus available in the tab:

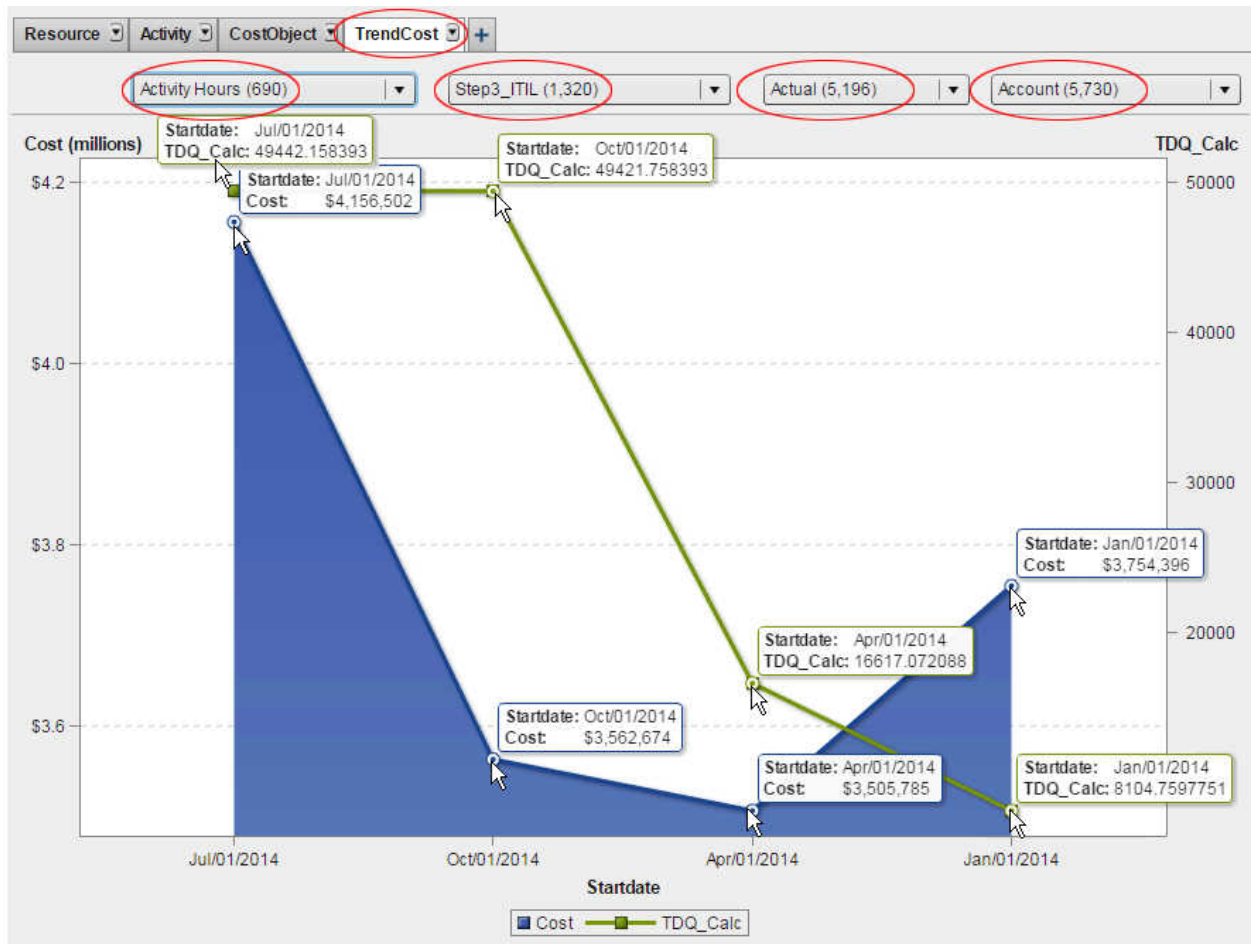
- Menu 1 Lists all the drivers in the **ITBMnMOD** model
- Menu 2 Lists all the modules in the model
- Menu 3 Lists three scenarios to choose from: **Actual**, **Budget**, **What\_if1**
- Menu 4 Lists the type of accounts that one can view: **RollupAccount**, **Account** (non-Rollup)



Whereas the previous tabs of the **ITBM\_Dashboard** report show in graphical form data that is available in tabular form inside SAS Cost and Profitability Management, the **TrendCost** tab displays data that is not available in the SAS Cost and Profitability Management.user interface. The TrendCost tab displays the sum of TDQCalc (Total Driver Quantity Calculated) for the selected driver in the selected module (a quantity that is not displayed in the SAS Cost and Profitability Management.user interface) and compares it to Cost in the same period. The following picture shows TDQCalc as compared to cost for the following selection:

Driver	<b>Activity Hours</b>
Module	<b>Step3_ITIL</b>
Scenario	<b>Actual</b>
Account Type	<b>Account (non-Rollup)</b>





## View the ITBM\_Fact\_View Report

### Overview

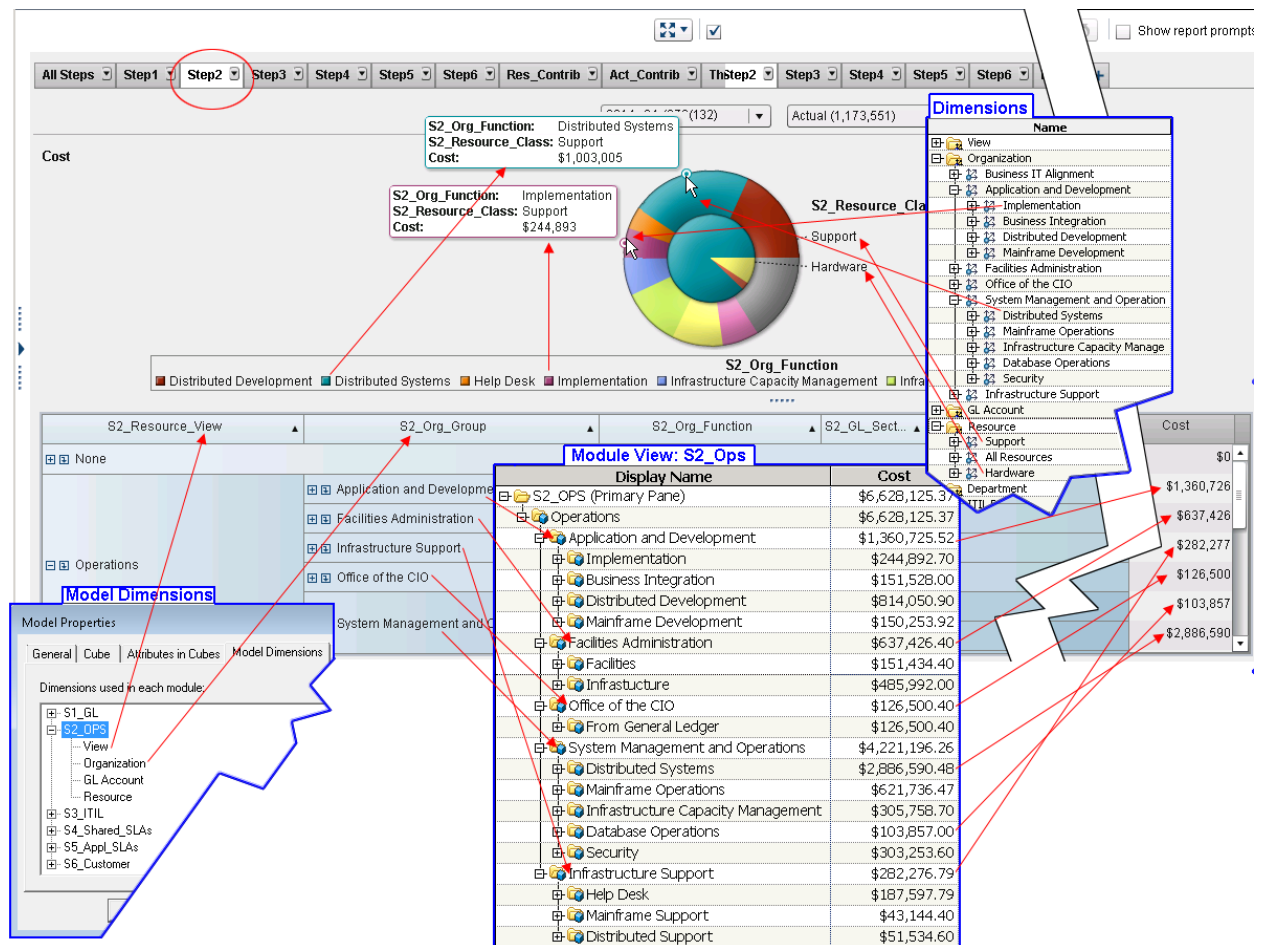
Unlike the **ITBM\_Dashboard** report which uses data from the **ITBMNMOD\_ACCOUNTMAP** table, the **ITBM\_Fact\_View** report uses data from the fact table **ITBMNMOD\_ITNMOD\_MSCSTAR**. Consequently, whereas the **ITBM\_Dashboard** report describes a static point in the model, the **ITBM\_Fact\_View** report shows cost flow across the model.

### Report Tab: Step 2

If you open the **ITBM\_Fact\_View** report and click the **Step2** tab you can see the relative contribution of different accounts to the total cost in the **S2\_OPS** module. By superimposing windows from the SAS Cost and Profitability Management client with the report, you can see how elements of the ITBMnMOD model are reflected in the report.

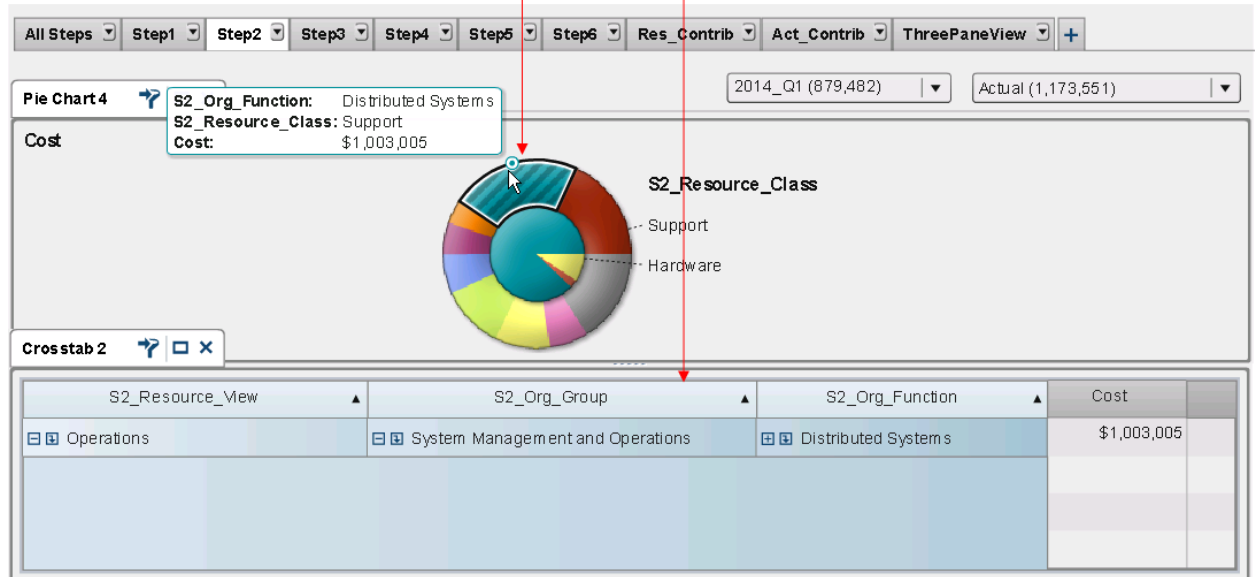
In the picture below, the outer ring of the pie chart displays costs for **Support** in the **Resource** dimension in the **S2\_Ops** module. And the inner circle displays costs for **Hardware** in the **Resource** dimension. The different segments of the pie chart display

costs for the individual organizations (the **Organization** dimension) within **Hardware** and **Software**—organizations such as **Distributed Systems** and **Implementation**.



In the following picture you can see that there is an interaction between the pie chart at the top of the report and the table at the bottom. If you select a portion of the pie chart (**Distributed Systems** in the picture), then the table at the bottom displays data for the portion selected.

If you click on the pie chart, then this display changes



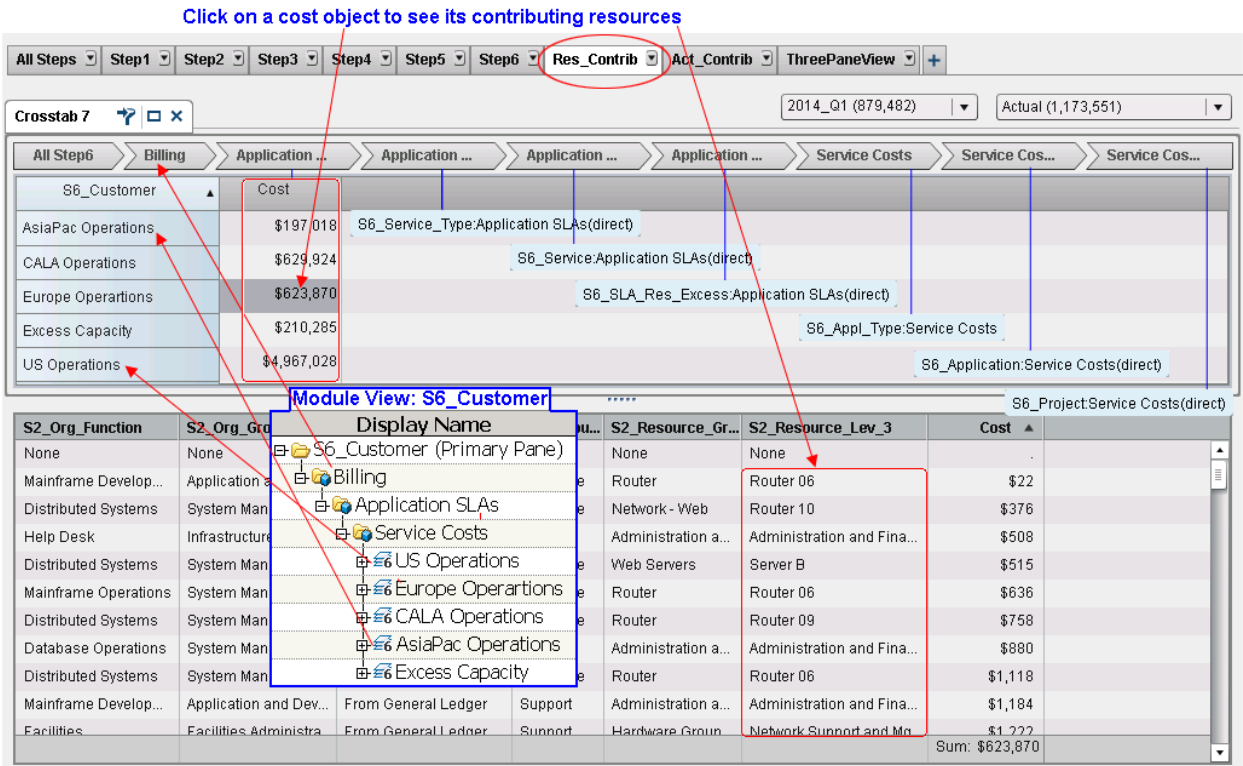
### Report Tab: Resource Contributions

If you click the **Res\_Contributions** tab of the **ITBM\_Fact\_View** report, you can see what resources accounted for costs in the **S6\_Customer** module which is a cost object module. In the following picture you can see the relative contribution of general ledger resources such as:

Router 07    Campaign DB  
Disk B       Router 09  
Router 06    and so on ...  
Server A

to the following cost objects:

US Operations  
Europe Operations  
CALA Operations  
AsiaPac Operations  
Excess Capacity



**Report Tab: Three Pane View**

If you click the **ThreePaneView** tab of the **ITBM\_Fact\_View** report, you can see a view of the model that is very similar to the three-pane view provided by SAS Cost and Profitability Management. However, there is an important difference between the two views. Whereas, as you can see in the following picture, the module view of SAS Cost and Profitability Management shows the costs in the **Application and Development** account in the **S3\_ITIL** module, the report not only shows those costs, it also shows the costs flowing into and out of the **Application and Development** account.

All Steps ▾ Step1 ▾ Step2 ▾ Step3 ▾ Step4 ▾ Step5 ▾ Step6 ▾ Res\_Contrib ▾ Act\_Contrib ▾ ThreePaneView ▾ +

Crosstab 9 2014\_Q1 (572,329) List Table 5 Actual (1,173,551)

S2_Org_Function ▲	S2_Re...	S2_Res...	Cost
Business Integration	Support	Admini...	\$60,811
Business Integration	Support	Softwa...	\$90,917
Distributed Develop...	Support	Softwa...	\$793,951
Distributed Develop...	Support	Admini...	\$20,100
Implementation	Support	Softwa...	\$189,235
Implementation	Support	Admini...	\$55,657
Mainframe Developm...	Hard...	Router	\$62,172
Mainframe Developm...	Support	Softwa...	\$75,498
Mainframe Developm...	Support	Admini...	\$12,583
None	None	None	
Sum:			\$1,360,726

Application				Cost
S3_Dept_Function	S3_Dept_Group	Application and Development	Acti...	
Business Integration				\$151,528
Service Design				\$264,281
Financial Support				\$14,067
Change Support				\$158,891
Problem Support				\$156,238
Release Support				\$195,050
Service Transistion				\$25,725
Implementation				\$244,893
Mainframe Development				\$150,254
Total				\$1,360,726

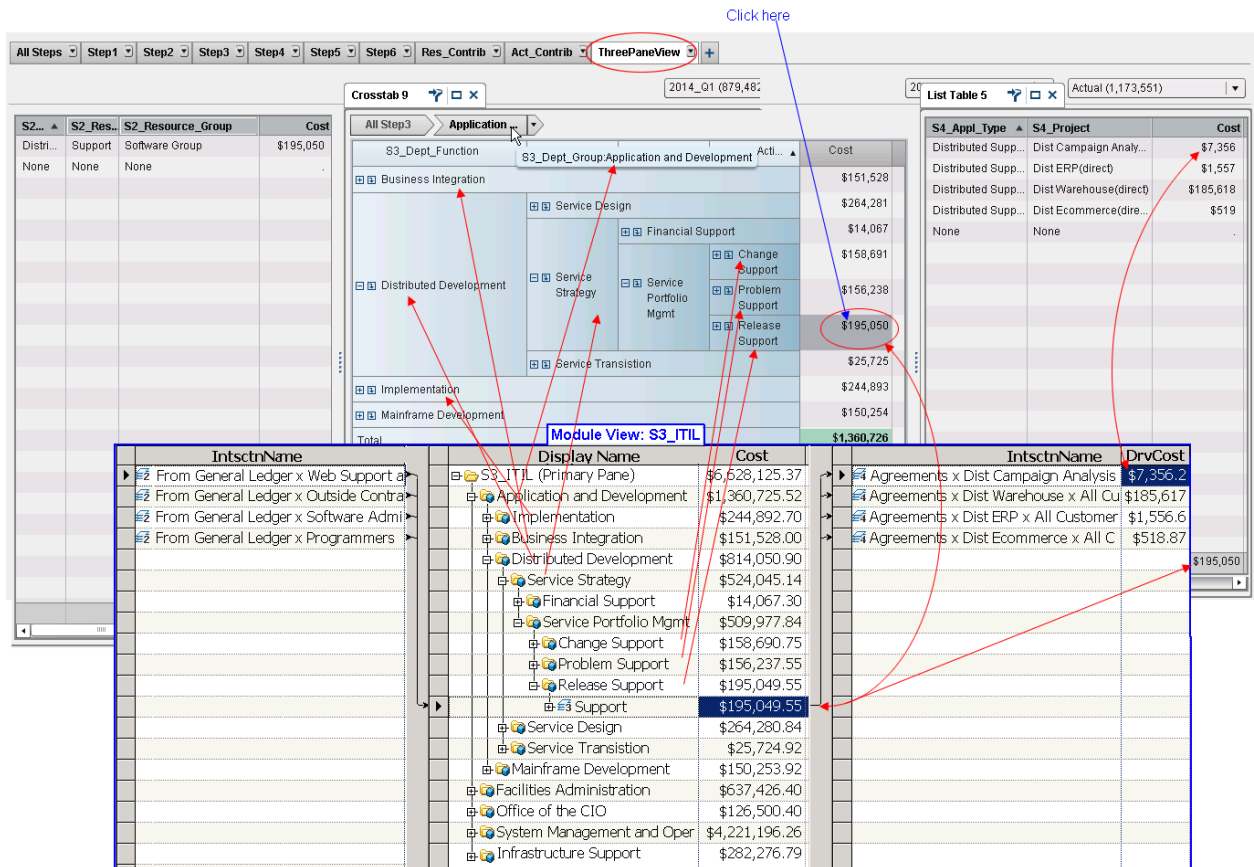
  

Module View: S3_ITIL		Cost
Display Name		
S3_ITIL (Primary Pane)		\$6,628,125.37
Application and Development		\$1,360,725.52
Implementation		\$244,892.70
Business Integration		\$151,528.00
Distributed Development		\$814,050.90
Service Strategy		\$524,045.14
Financial Support		\$14,067.30
Service Portfolio Mgmt		\$509,977.84
Change Support		\$158,690.75
Problem Support		\$156,237.55
Release Support		\$195,049.55
Service Design		\$264,280.84
Service Transistion		\$25,724.92
Mainframe Development		\$150,253.92
Facilities Administration		\$637,426.40
Office of the CIO		\$126,500.40
System Management and Operatio		\$4,221,196.26
Infrastructure Support		\$282,276.79

S4_Appi_Type ▲	S4_Project	Cost
Database Operat...	ECommerce(direct)	\$97,008
Database Operat...	Campaign Analysis(di...	\$201,136
Database Operat...	Hardware Storage(dir...	\$35,288
Database Operat...	Warehouse(direct)	\$133,025
Distributed Supp...	Dist Ecommerce(direct)	\$519
Distributed Supp...	Dist ERP(direct)	\$130,102
Distributed Supp...	Dist Campaign Analy...	\$7,356
Distributed Supp...	Dist Warehouse(direct)	\$194,330
Help Desk Direct	Help Warehouse(dire...	\$169,643
Help Desk Direct	Help ERP(direct)	\$43,454
Help Desk Direct	Help ECommerce(dir...	\$14,462
Help Desk Direct	Help Campaign(direct)	\$60,577
Infrastructure Ba...	Infrastructure Based(...	\$62,172
Mainframe Opera...	ERP(direct)	\$211,654
None	None	
Sum:		\$1,360,726

If you click the cost for the **Release Support** account on the report, you can see that the contributing costs to that account are displayed on the left, and the costs flowing out of the account are displayed on the right. And, because this is a leaf account, you can see that the data displayed in the report corresponds to the data displayed in the SAS Cost and Profitability Management three-pane view.



To reinforce the point that the SAS Visual Analytics report displays data not available in the SAS Cost and Profitability Management client, collapse the **Service Strategy** account to see the two lower-level accounts that constitute it:



- **Financial Support**
- **Service Portfolio Mgmt.**

Then click on the cost for **Service Portfolio Mgmt**, which is not a leaf account since (as we saw in the picture above) it contains three leaf accounts:

- **Change Support**
- **Problem Support**
- **Release Support**

After clicking on the cost for **Service Portfolio Mgmt** you can see, in the left-hand side of the report, the account that contributes to it, and in the right-hand side of the report you can see the accounts to which it contributes costs.

Collapse this, and click here

All Steps ▾ Step1 ▾ Step2 ▾ Step3 ▾ Step4 ▾ Step5 ▾ Step6 ▾ Res_Contrib ▾ Act_Contrib ▾ ThreePaneView ▾ +		
Crosstab 9  		
		2014_Q1 (879,482) ▾ Actual (1,173,551) ▾
<b>S2_Resource_CL...</b>	<b>S2_Resour...</b>	<b>Cost</b>
Support	Software G...	\$509,978
None	None	.
Sum: \$509,978		

All Step3 >> Application ...			
S3_Dept_Function ▲	S3_ITIL_Proce...	S3_ITIL_Sub...	Cost
Business Integration			\$151,528
Service Design			\$264,281
Financial Support			\$14,067
Service Strategy			\$509,978
Service Portfolio Mgmt			
Service Transition			\$25,725
Implementation			\$244,893
Mainframe Development			\$150,254
Total			\$1,360,726

S4_Appli_T...	S4_Project	Cost
Database Oper...	Campaign .	\$57,347
Database Oper...	Hardware S	\$44
Database Oper...	ECommerc	\$67,118
Distributed Sup...	Dist ERP(di	\$115,757
Distributed Sup...	Dist Econ..	\$519
Distributed Sup...	Dist Wareh.	\$185,618
Distributed Sup...	Dist Camp..	\$7,356
Help Desk Direct	Help ERP(d	\$2,627
Help Desk Direct	Help ECom	\$7,882
Help Desk Direct	Help Wareh	\$12,086
Help Desk Direct	Help Camp	\$19,442
Mainframe Ope	ERP(direct)	\$34,181
None	None	\$509,978

## View the ITBM\_Account\_View Report

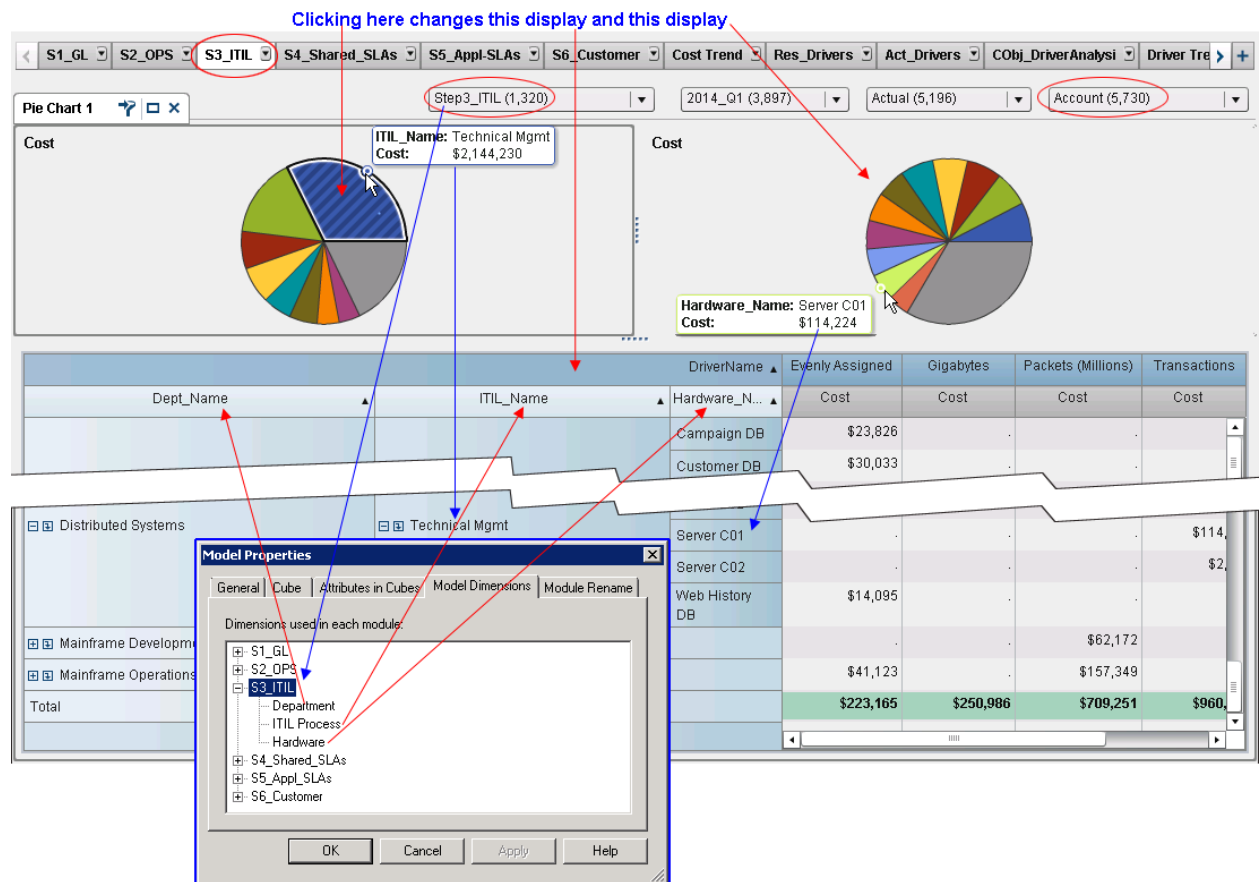
### Overview

If you open the **ITBM\_Account\_View** report you can see one tab for each stage in the model. Here we describe three of the tabs:

- “Report Tab: S3\_ITIL – Activity Module” on page 363
- “Report Tab: S6\_Customer – Cost Objects” on page 364
- “Report Tab: Act\_Drivers – Activity Module Drivers” on page 365

### Report Tab: S3\_ITIL – Activity Module

If you click the **S3\_ITIL** tab of the **ITBM\_Account\_View** report you can see the breakdown of costs in the **S3\_ITIL** module. In this report, if you click the pie chart at the top-left of the report, then the pie-chart at the top-right changes as does the table at the bottom of the report. In the following picture you can see that the **Technical Mgmt** portion of the top-left pie chart has been selected. And the top-right pie chart as well as the table at the bottom display a breakdown of costs for **Technical Mgmt**.

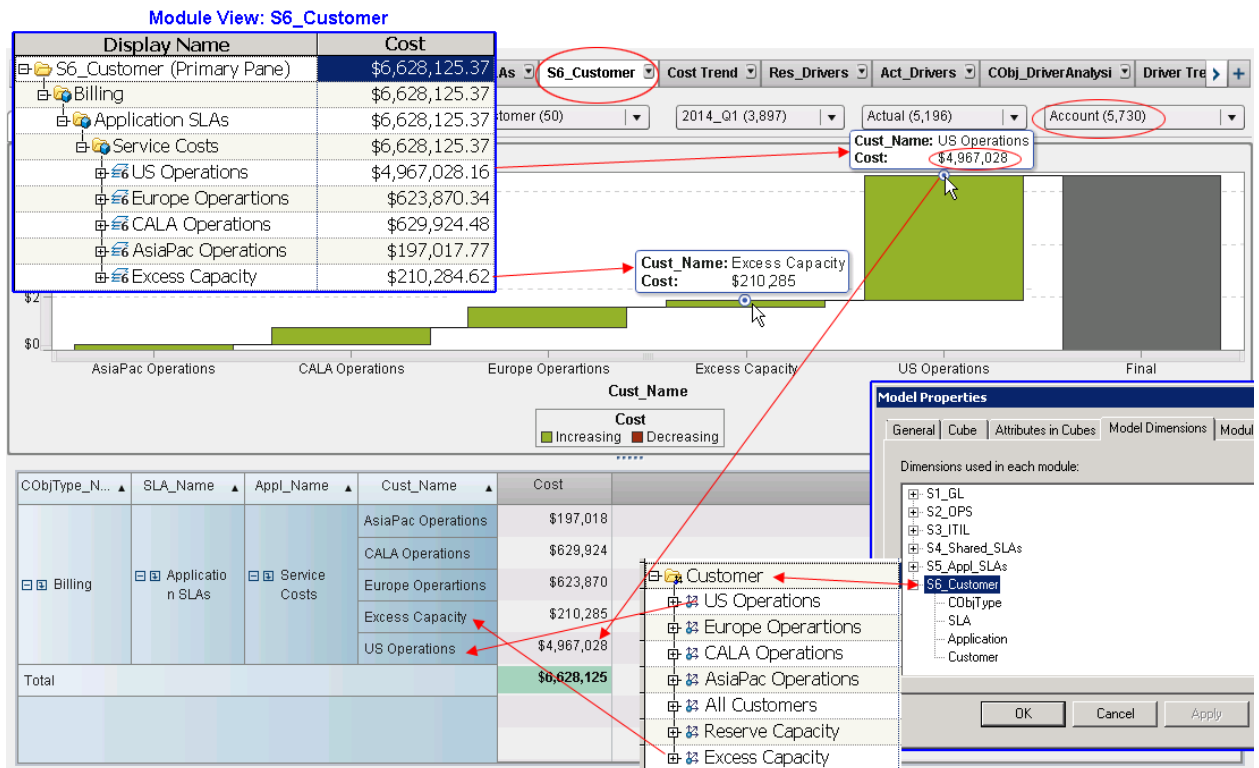


**Report Tab: S6\_Customer – Cost Objects**

If you click the **S6\_Customer** tab of the **ITBM\_Account\_View** report you can see the breakdown of costs in the **S6\_Customer** module. In the following picture you can see relative contributions of:

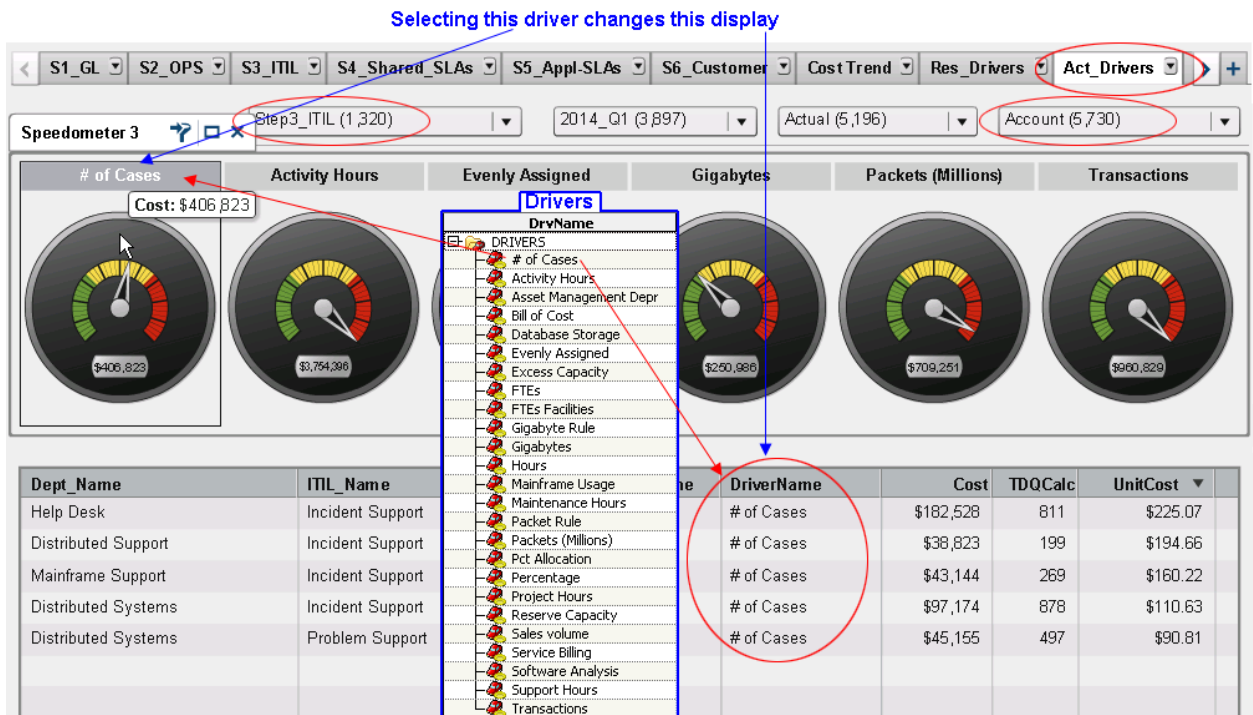
- US Operations  
Europe Operations  
CALA Operations  
AsiaPac Operations  
US Operations  
All Customers  
Reserve Capacity  
Excess Capacity





### Report Tab: Act\_Drivers – Activity Module Drivers

If you click the **Act\_Drivers** tab of the **ITBM\_Account\_View** report you can see **Cost**, **TDQCalc**, and **UnitCost** for drivers in the **S3\_ITIL** module (named **Step3\_ITIL** in the report). If you click one of the pie charts at the top of the report, then the table at the bottom of the report changes to display data for the selected pie chart (driver).





## **Part 8**

---

# SAS Profitability Management

### *Chapter 24*

**Working with SAS Profitability Management** ..... 369



## Chapter 24

# Working with SAS Profitability Management

---

<b>Specify the SAS Profitability Management Input Library</b> .....	<b>369</b>
<b>Use Account Data with SAS Profitability Management</b> .....	<b>371</b>
<b>Mark Accounts as Behaviors</b> .....	<b>373</b>
Mark an Individual Account .....	373
Mark Multiple Accounts .....	373
Search for Accounts Marked as Behaviors .....	373
<b>Publish Behaviors to SAS Profitability Management</b> .....	<b>374</b>
<b>Create a SAS Profitability Management Input Library</b> .....	<b>378</b>
<b>Map to the Behavior Table</b> .....	<b>382</b>

---

## Specify the SAS Profitability Management Input Library

The Profitability Management input library is the library on the Metadata Server where the behavior table is created when you publish behaviors. A SAS Profitability Management input library contains the model's behavior table, dimension tables, transaction tables, report hierarchies, and report layouts.

The behavior table contains the accounts that you have marked as behaviors with the `isBehavior` attribute. After publishing behaviors, you can open SAS Profitability Management and use with a Profitability Management model the behavior table that you created.

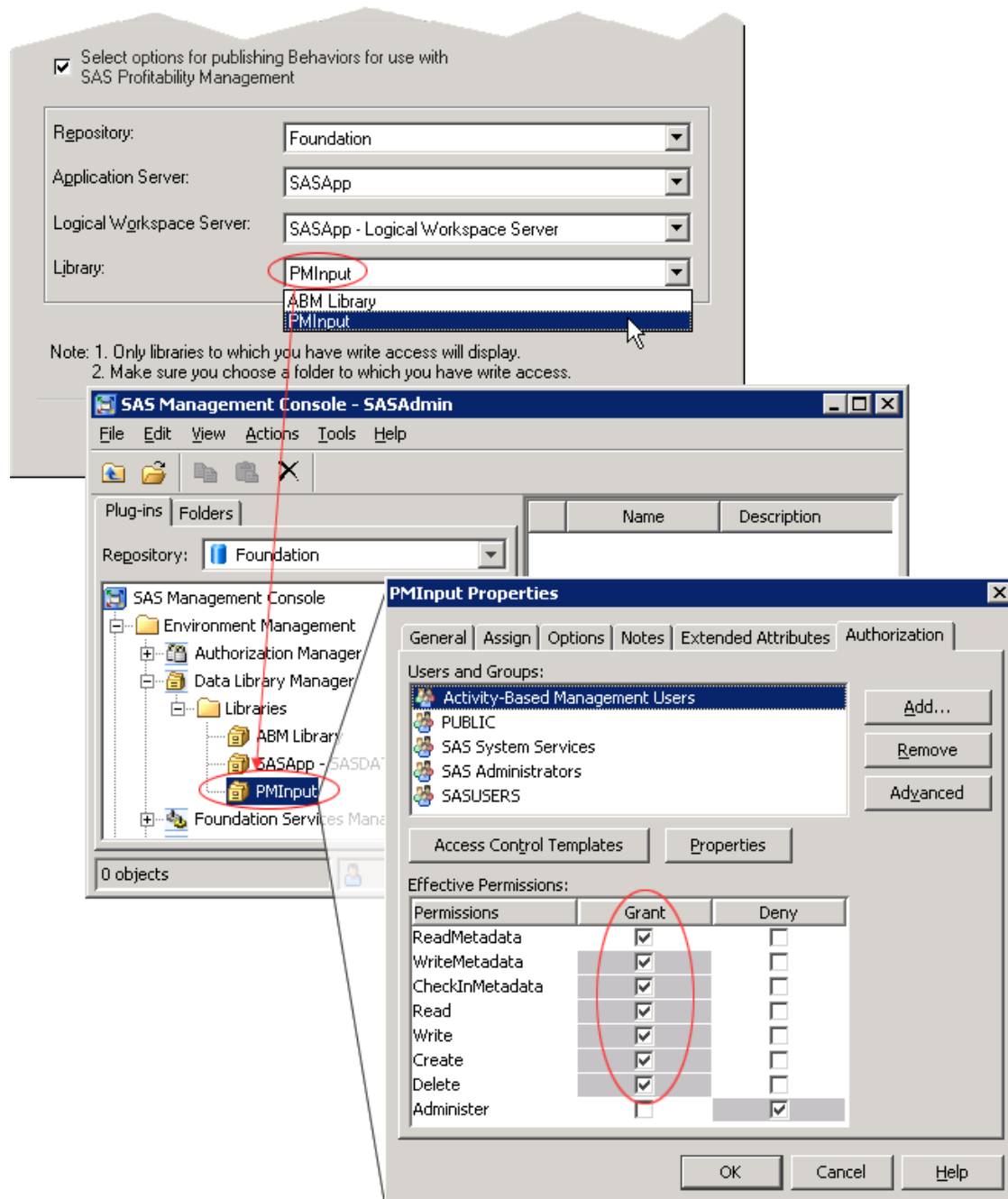
*Note:* Do not publish to the SAS Profitability Management data library. The data library contains a list of all the models in a SAS Profitability Management installation. And, it stores pointers to information residing in each model's input library.

*Note:* For more information on creating an input library, see [“Create a SAS Profitability Management Input Library” on page 378](#).

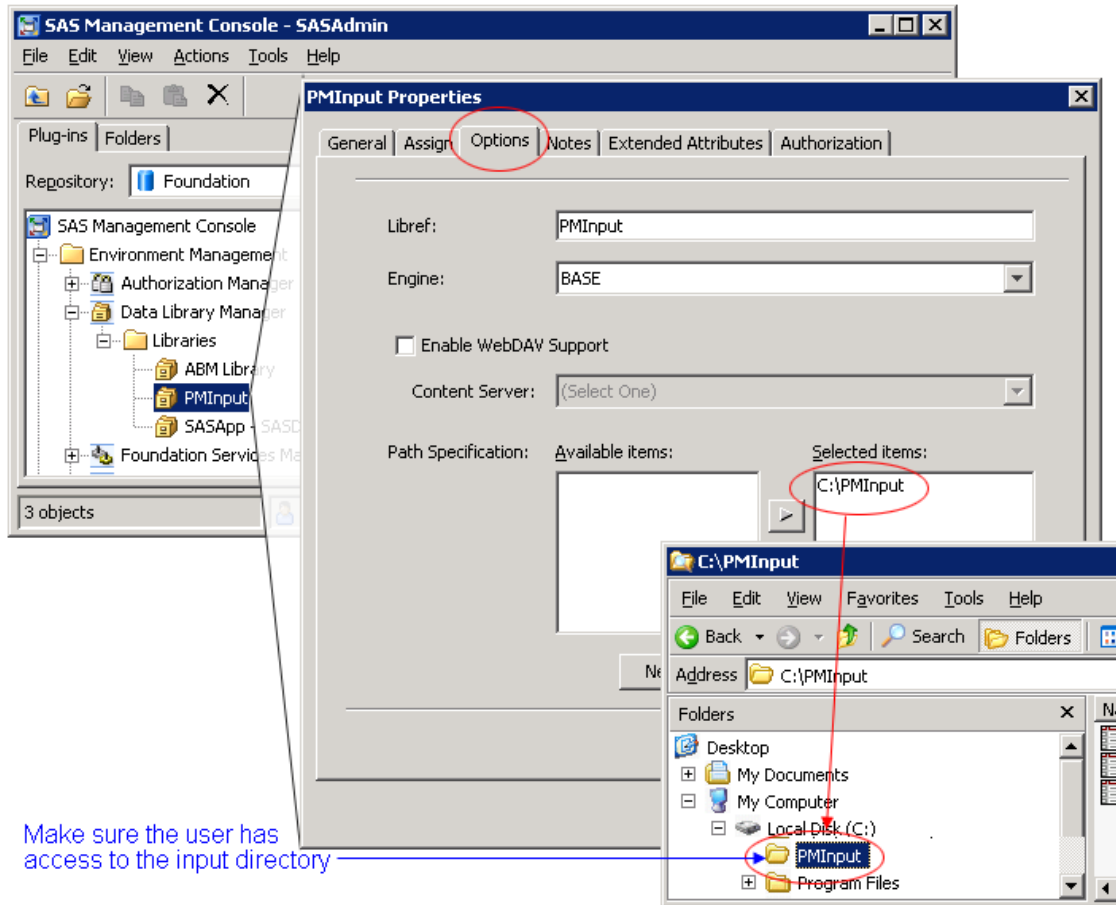
To specify the SAS Profitability Management input library:

1. Select **Tools** ⇒ **Metadata Server Options** .  
The Metadata Server Options dialog box opens.
2. Specify a Repository, Application Server, and Logical Workspace Server.
3. Select an input Library from the drop-down list of libraries.

*Note:* The SAS Cost and Profitability Management user who publishes behaviors must have write access to the SAS Profitability Management input library to which the behaviors are written. The drop-down list displays for selection only those libraries to which the user has access.



*Note:* The SAS Cost and Profitability Management user who publishes behaviors must also have write access to the library's directory on the machine where the behaviors are published. The following picture shows a SAS Profitability Management input library named PMInput whose machine directory is C:\PMInput. In this example, the SAS Cost and Profitability Management user must have write access to the C:\PMInput directory.



4. Click **OK**.

## Use Account Data with SAS Profitability Management

You can use the data in SAS Cost and Profitability Management accounts with SAS Profitability Management. The process involves two steps:

1. Mark accounts as behaviors
2. Publish the behaviors to SAS Profitability Management

Publishing behaviors means that they are written to a behavior table for use with SAS Profitability Management. A behavior table contains source items with a transaction cost. A behavior table has the following required columns:

Position	Name	Maximum Length	Description
1	Time	Char 32	Defines the period for the costs
2	ID	Char 32	The identifying reference for the behavior

Position	Name	Maximum Length	Description
3	Name	Char 32	The name of the behavior
4	Total Value	Numeric 8	The total source amount that will be spread
5	Unit Value	Numeric 8	The unit cost for each transaction with this source

- The columns must appear in the order shown.
- Each column must have the length shown.
- The name of the column is arbitrary.

*Note:* Only one of the Unit-Value and Total-Value fields may contain a non-zero value for any row of the table.

The following picture shows a sample behavior table.

	Time	ID	Name	TotalValue	UnitValue
1	2006_Q1_Actual	20002	CCT_OTP_Manage transactions	16468.0817	0
2	2006_Q1_Actual	20003	ATM_CHK_Check balance	0	1.44909029
3	2006_Q1_Actual	20004	ATM_CHK_Deposits	0	0.17800894
4	2006_Q1_Actual	20005	ATM_CHK_Fund Transfer	0	0.80104504
5	2006_Q1_Actual	20006	ATM_CHK-Withdrawals	0	3.81449376
6	2006_Q1_Actual	20007	ATM_CRC-Withdrawals	0	1.52364654
7	2006_Q1_Actual	20008	ATM_OTP_Manage transactions	212489.27	0
8	2006_Q1_Actual	20009	ATM_REC_Deposits	0	0.10640691
9	2006_Q1_Actual	20010	ATM_SAV_Check balance	0	2.12900845
10	2006_Q1_Actual	20011	ATM_SAV_Deposits	0	0.29142294
	2006_Q1_Actual	20012	ATM_SAV_Fund Transfer	0	1.17689661
2120	2006_Q4_Actual	12001	CHG_SAV_SCO	0	5.78
2121	2006_Q4_Actual	13001	Credit For Funds	0	1
2122	2006_Q4_Actual	13002	Charge For Funds	0	1
2123	2006_Q4_Actual	14001	Provision For Losses	0	1
2124	2006_Q4_Budget	10001	Credit Card interest Income	0	1
2125	2006_Q4_Budget	10002	Loan Interest Income	0	1
2126	2006_Q4_Budget	10003	Mortgages Income	0	1
2127	2006_Q4_Budget	11001	Savings Interest Payments	0	1
2128	2006_Q4_Budget	11002	Certificates of Deposit Payments	0	1
2129	2006_Q4_Budget	11003	Investment Securities Payments	0	1
2130	2006_Q4_Budget	12001	Credit Card Fees	0	1
2131	2006_Q4_Budget	12002	ATM Fees	0	1
2132	2006_Q4_Budget	12003	Investment Account Fees	0	1
2133	2006_Q4_Budget	12004	Checking Account Fees	0	1
2134	2006_Q4_Budget	13001	Credit for Funds	0	1
2135	2006_Q4_Budget	13002	Charge For Funds	0	1
2136	2006_Q4_Budget	14001	Provision For Losses	0	1

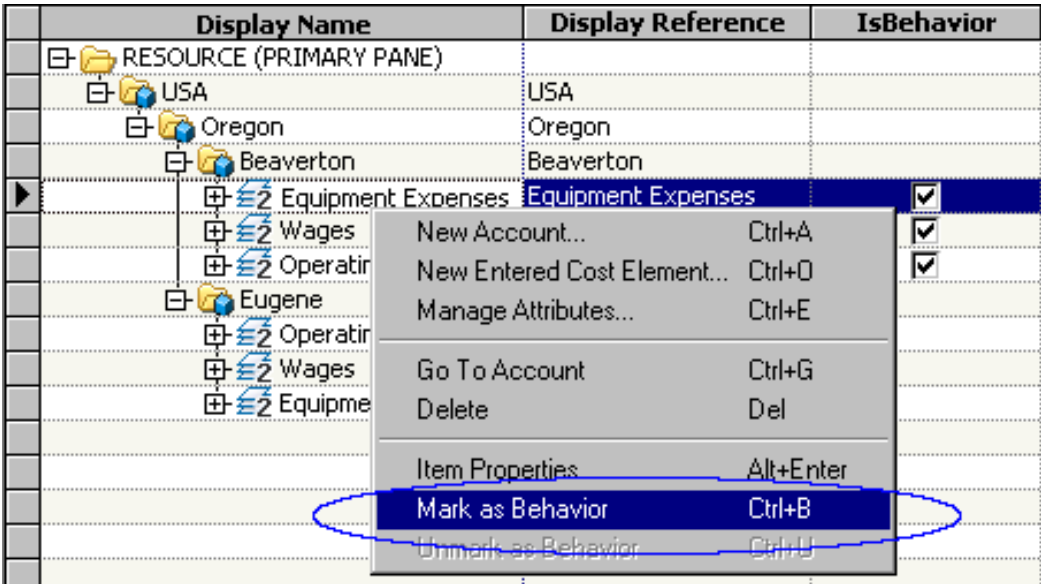


## Mark Accounts as Behaviors

Mark accounts as behaviors before publishing them to SAS Profitability Management. You can mark an individual account or multiple accounts.

### Mark an Individual Account

- 1. Select the account.
- 2. Right-click the account.
- 3. Select Mark as Behavior.



### Mark Multiple Accounts

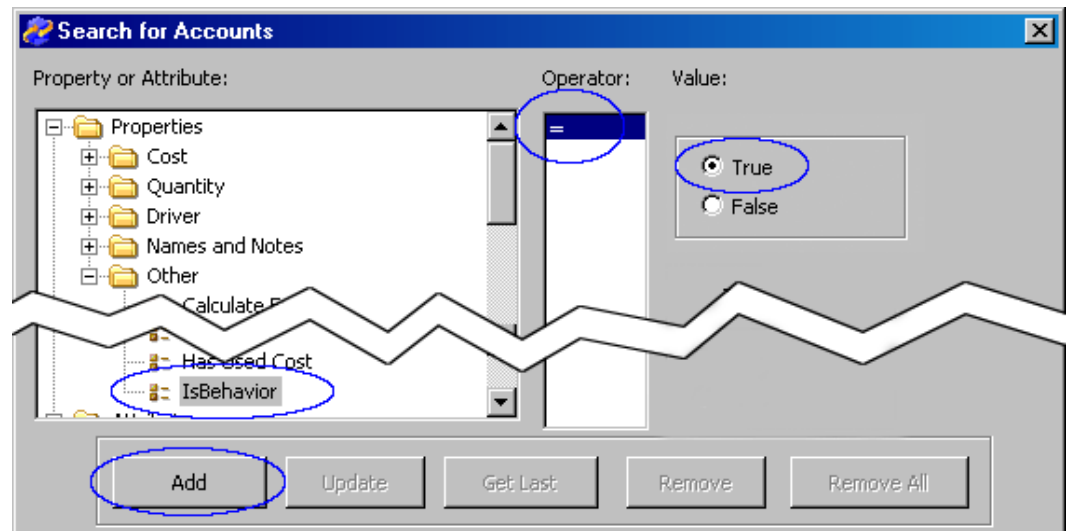
- 1. Open the Search for Accounts dialog box.
- 2. Search for the accounts that you want to mark as behaviors.
- 3. From the list of accounts that are found, select the ones to be marked.
- 4. Click Actions, and then select Mark as Behaviors.

### Search for Accounts Marked as Behaviors

- 1. Open the Search for Accounts dialog box.
- 2. Select the **IsBehavior** property (under **Properties** ⇒ **Other**).
- 3. Select the = operator.
- 4. Select **True**.
- 5. Click **Add**.

6. Click **Search**.

Accounts that have been marked as behaviors are listed.



## Publish Behaviors to SAS Profitability Management

Select **File** ⇒ **Publish** ⇒ **Behaviors**. The Publish Behaviors wizard opens.

*Note:* Before you can publish behaviors, you must specify the SAS Profitability Management Library so that SAS Cost and Profitability Management knows where to store the published behaviors. And, you must have marked some accounts as behaviors.

1. Select a model and specify the name of the behavior table to be created.

**Publish Behaviors - Location** [X] Step 1 of 5

**Location**

Select a model and one or more period/scenario associations to export.  
Specify dataset name and option for create, replace or append type of operation for behavior table.

Select a model for publishing behaviors:

(Select Model)

Select period/scenario associations for publishing behaviors:

Select All

Clear All

Behavior table name:

Select an option for behavior table:

Create Table

< Back Next > Finish Cancel Help

**Model name**

Select the model whose accounts you want to mark as behaviors.

**Period/Scenario associations**

Select the period/scenario associations for which you want to publish the data.

**Behavior table name**

Specify a name for the behavior table.

**Option**

Create table

Creates a behavior table. If a table with the same name exists, then the operation quits with an error message, and the existing table is undisturbed.

Replace table

Replaces an existing table with the same name.

Append to table

Appends records to an existing table.

2. Map properties and attributes of the accounts being published to fields in the resulting behavior table being created.

**Publish Behaviors - Map Properties and Attributes** [X]

**Map Properties and Attributes** Step 2 of 5

Map properties and attributes which will become rates.

Create mapping from properties and attributes against each field:

Id: [ ]

Name: [ ]

Period: [ ]

For below fields map for either Unit value or Total value.

Unit value: [NOT SELECTED]

Total value: [NOT SELECTED]

[ < Back ] [ Next > ] [ Finish ] [ Cancel ] [ Help ]

**Id**

is the identifying reference for the behavior

**Name**

is the name of the behavior

**Period**

defines the period for the costs

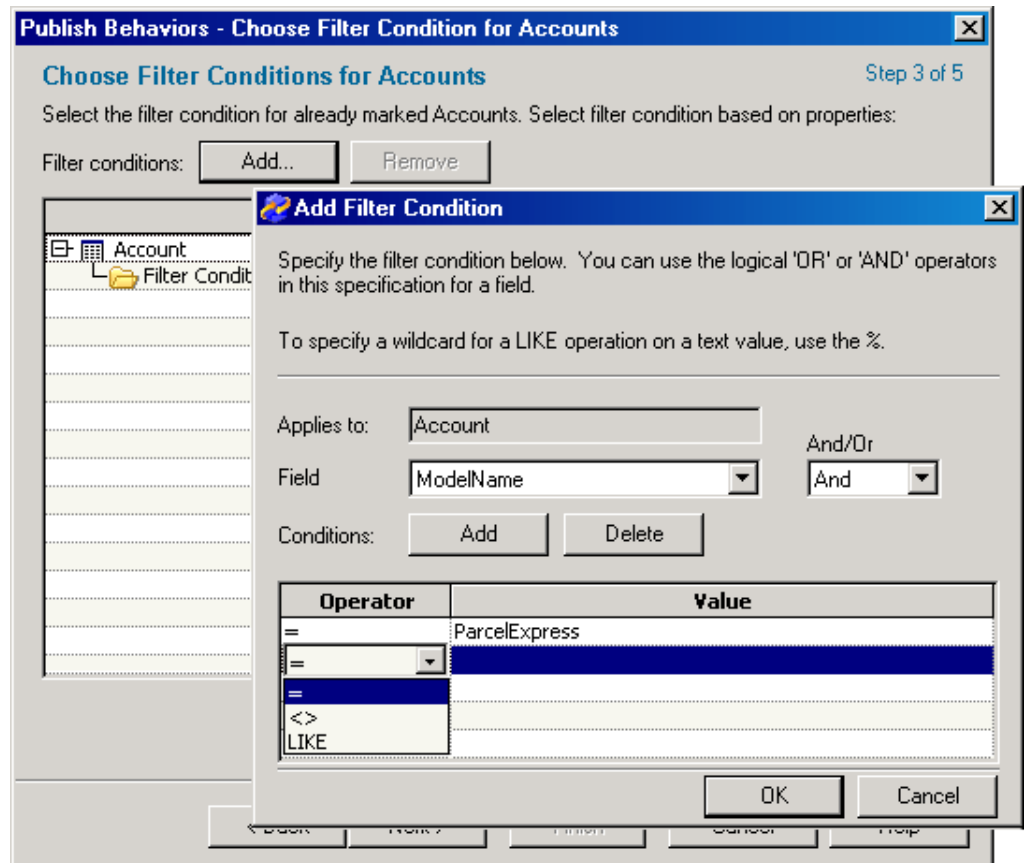
**Unit value**

is the unit cost for each transaction with this source. If you select a Unit value, then you may not select a Total value.

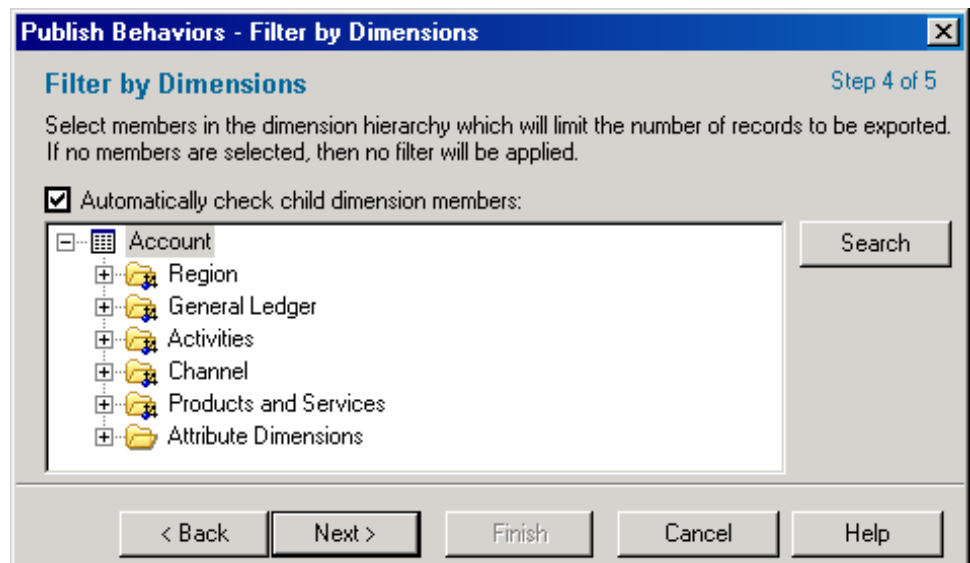
**Total value**

is the total source amount that will be spread. If you select a Total value, then you may not select a Unit value.

3. You can optionally set a condition that an account must satisfy to be published as a behavior. This allows you to select a subset of all the accounts that have been marked as behaviors.



4. You can further limit the number of accounts to be published by selecting dimensions. If you do not select any dimensions, then accounts (which are marked as behaviors) from all dimensions are published.



5. Review your selections, and then click **Finish**.

The published accounts are written to a behavior table in the Profitability Management Library.

**See Also**

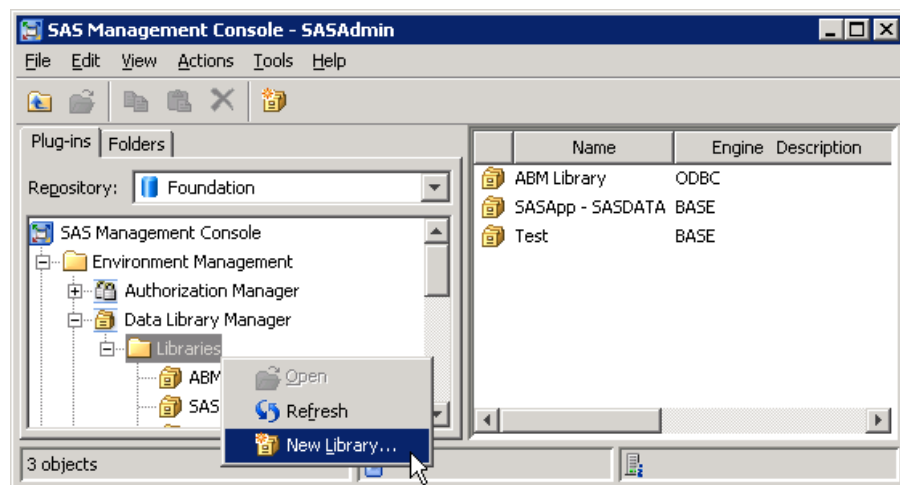
- “Mark Accounts as Behaviors” on page 373
- Chapter 24, “Working with SAS Profitability Management,” on page 369

---

## Create a SAS Profitability Management Input Library

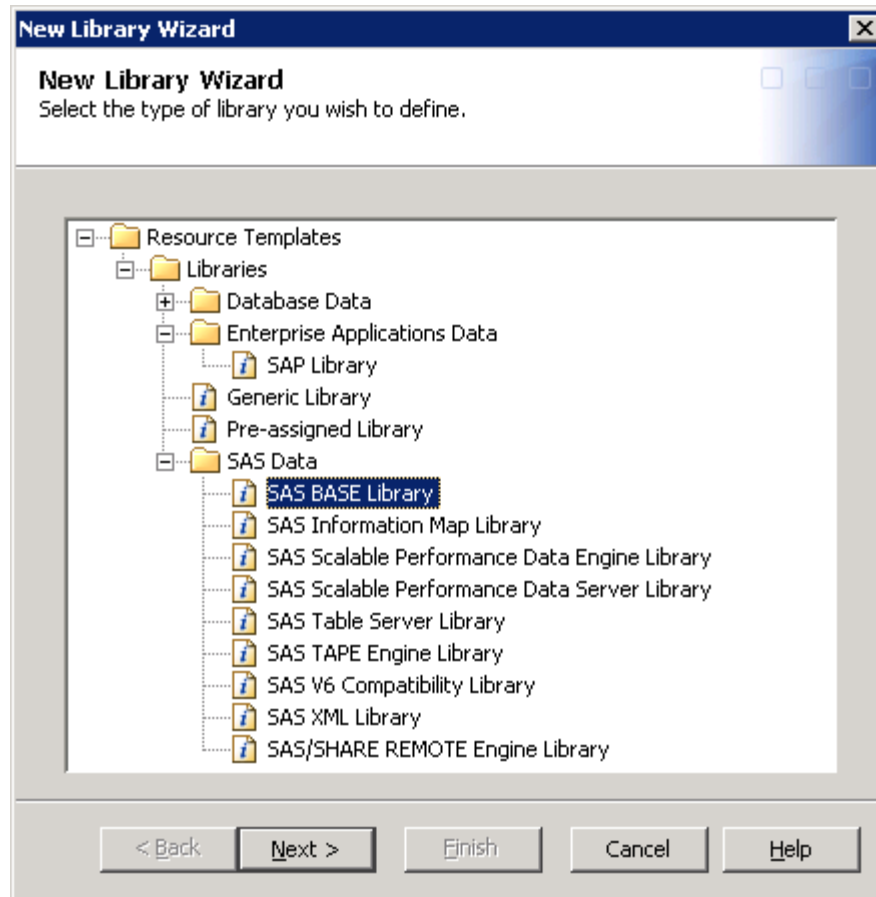
If a SAS Profitability Management input library does not already exist, follow these instructions to create one.

1. Open SAS Management Console.
2. Select the **Plug-ins** tab.
3. Right-click **Libraries** (under Data Library Manager) and select **New Library**.



4. Select the type of library, and then click **Next**.

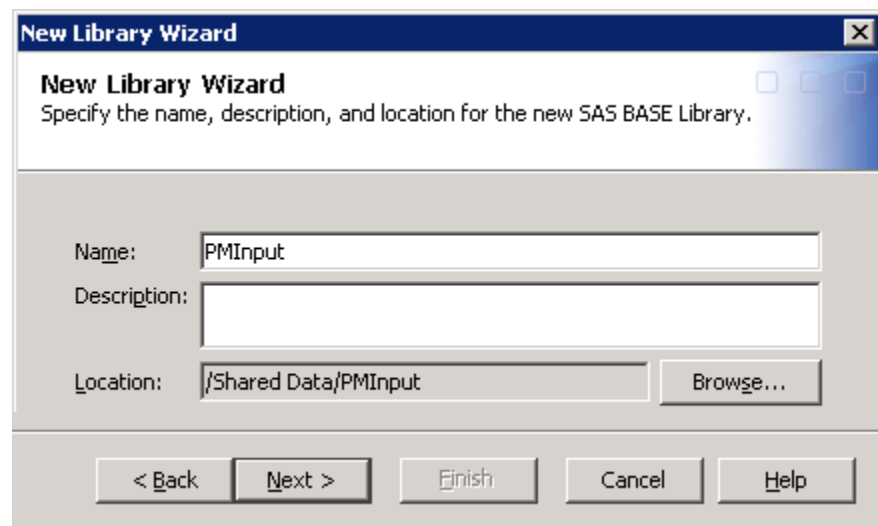
For input libraries, SAS Profitability Management supports any library type for which you can create a libref. The following picture shows selecting a library type of SAS Base Library.



- Specify the library Name and the Location of its metadata folder, and then click **Next**.

#### Name

This is the text that appears in the navigation and display areas of SAS Management Console, but is not the LIBREF for the library. The name can contain up to 60 characters. It must be unique within the folder specified in the Location field. It must also be unique on all servers where the library is assigned.

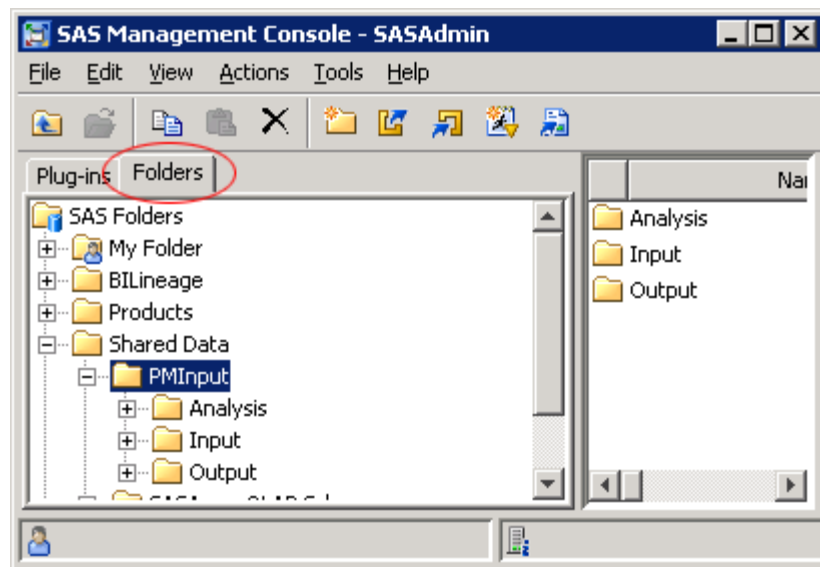


### Folder

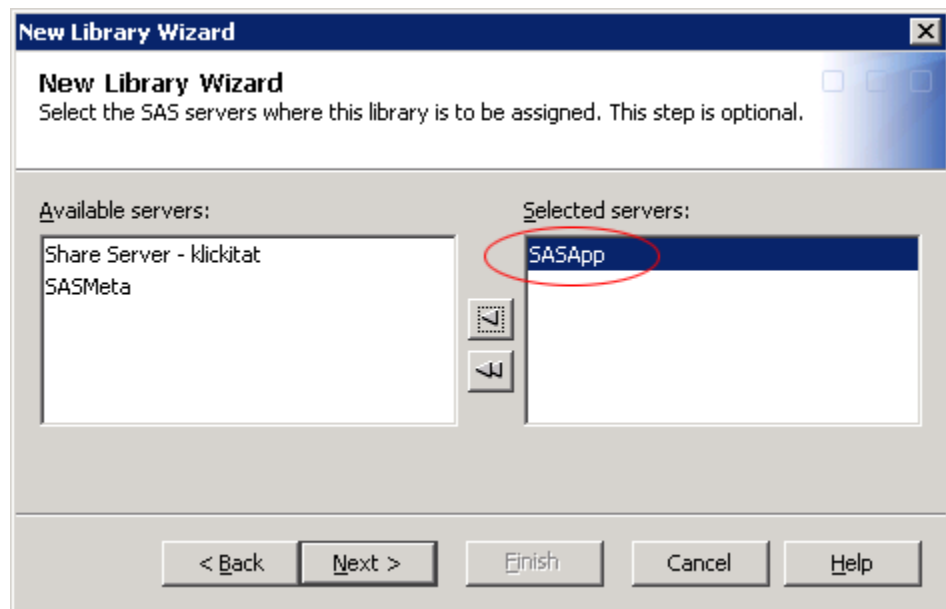
It is strongly recommended that you choose a unique folder for each library. If two libraries share the same metadata folder and both libraries contain same-named items, then the metadata folder will contain incorrect information about one of those items.

*Note:* In order to specify the Location of the library's metadata folder, you should have already created the folder. To create the library's metadata folder, do the following:

1. Click the Folders tab.
2. Right-click and select New Folder.
3. Name the folder, and then click Finish.



6. Select **SASApp** as the SAS server, and then click **Next**.



7. Specify a Libref for the library and the Path of the machine directory where the library's data is stored, and then click **Next**.



**New Library Wizard**  
Enter the following library properties.

Libref: pminput

Engine: BASE

☐ Enable WebDAV Support

Content Server: (Select One)

Path Specification: Available items: Data  
Selected items: C:\PMInput

New... Edit... Delete

Advanced Options...

< Back Next > Finish Cancel Help

8. Review your choices, and then click **Finish**.

**New Library Wizard**

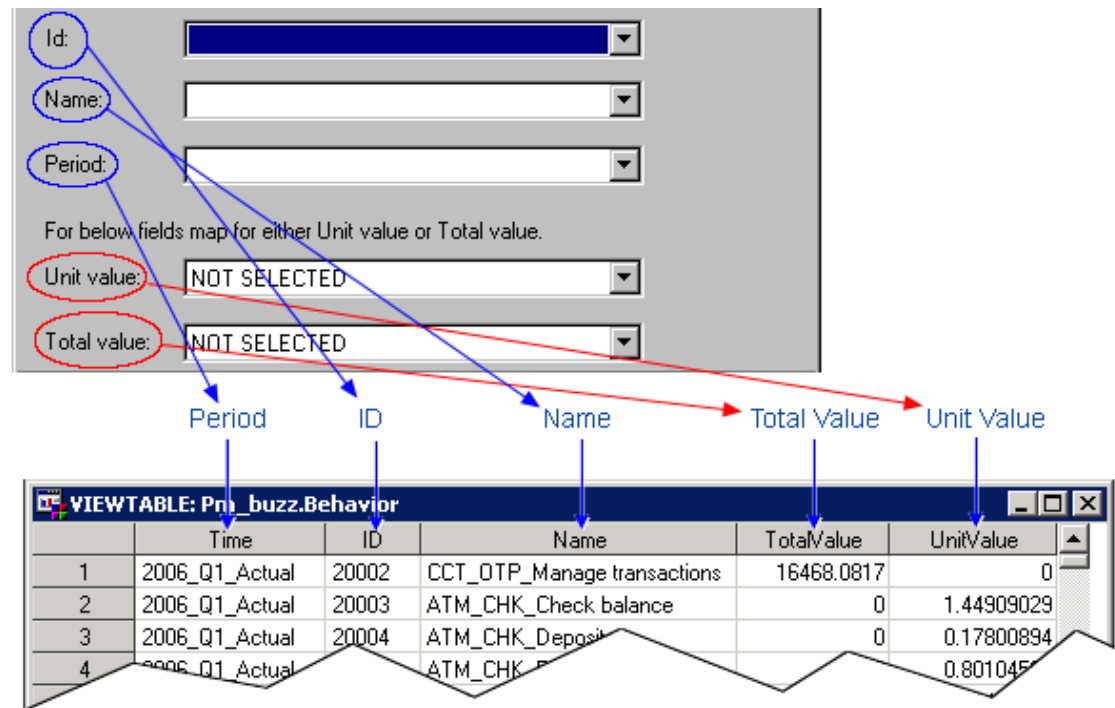
The following library will be created:

Library: PMInput  
Location: /Shared Data/PMInput  
Assigned to SAS Servers: SASApp  
Libref: pminput  
Engine: BASE  
Path Specification: C:\PMInput  
Library is pre-assigned: No

< Back Next > Finish Cancel Help

## Map to the Behavior Table

The following picture shows a sample mapping to a behavior table. Both the names of the fields in the pictured table and their order in the table are arbitrary. You identify the columns to SAS Cost and Profitability Management by the process of mapping.



A SAS Profitability Management behavior table contains source items with their transaction costs. A behavior table has the following required columns:

Position	Name	Maximum Length	Description
1	Time	Char 32	Defines the period for the costs
2	ID	Char 32	The identifying reference for the behavior
3	Name	Char 32	The name of the behavior
4	Total Value	Numeric 8	The total source amount that will be spread
5	Unit Value	Numeric 8	The unit cost for each transaction with this source

- Each column must have the specified length.
- The name of each column is arbitrary.
- The order of the columns is arbitrary.

*Note:* Only one of the Unit-Value and Total-Value fields may contain a non-zero value for any row of the table.

The following picture shows a sample behavior table.

Diagram labels pointing to the table headers:

- Period (points to Time)
- ID (points to ID)
- Name (points to Name)
- Total Value (points to TotalValue)
- Unit Value (points to UnitValue)

	Time	ID	Name	TotalValue	UnitValue
1	2006_Q1_Actual	20002	CCT_OTP_Manage transactions	16468.0817	0
2	2006_Q1_Actual	20003	ATM_CHK_Check balance	0	1.44909029
3	2006_Q1_Actual	20004	ATM_CHK_Deposits	0	0.17800894
4	2006_Q1_Actual	20005	ATM_CHK_Fund Transfer	0	0.80104504
5	2006_Q1_Actual	20006	ATM_CHK_Withdrawals	0	3.81449376
6	2006_Q1_Actual	20007	ATM_CRC_Withdrawals	0	1.52364654
7	2006_Q1_Actual	20008	ATM_OTP_Manage transactions	212489.27	0
8	2006_Q1_Actual	20009	ATM_REC_Deposits	0	0.10640691
9	2006_Q1_Actual	20010	ATM_SAV_Check balance	0	2.12900845
10	2006_Q1_Actual	20011	ATM_SAV_Deposits	0	0.29142294
11	2006_Q1_Actual	20012	ATM_SAV_Fund Transfer	0	1.17689661
2120	2006_Q1_Actual	12001	CHG_SAV_Account	0	5.78
2121	2006_Q4_Actual	13001	Credit Card Funds	0	1
2122	2006_Q4_Actual	13002	Charge For Funds	0	1
2123	2006_Q4_Actual	14001	Provision For Losses	0	1
2124	2006_Q4_Budget	10001	Credit Card interest Income	0	1
2125	2006_Q4_Budget	10002	Loan Interest Income	0	1
2126	2006_Q4_Budget	10003	Mortgages Income	0	1
2127	2006_Q4_Budget	11001	Savings Interest Payments	0	1
2128	2006_Q4_Budget	11002	Certificates of Deposit Payments	0	1
2129	2006_Q4_Budget	11003	Investment Securities Payments	0	1
2130	2006_Q4_Budget	12001	Credit Card Fees	0	1
2131	2006_Q4_Budget	12002	ATM Fees	0	1
2132	2006_Q4_Budget	12003	Investment Account Fees	0	1
2133	2006_Q4_Budget	12004	Checking Account Fees	0	1
2134	2006_Q4_Budget	13001	Credit for Funds	0	1
2135	2006_Q4_Budget	13002	Charge For Funds	0	1
2136	2006_Q4_Budget	14001	Provision For Losses	0	1

