SAS® Cost and Profitability Management 8.3: User's Guide

SAS® Documentation
May 9, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS® Cost and Profitability Management 8.3: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Cost and Profitability Management 8.3: User's Guide

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

May 2018

8.3-P1:abmug

Contents

PART 17 Calculation 523

PART 18 Generating Cubes 561

What's New in SAS Cost and Profitability Management 8.3

Overview

SAS Cost and Profitability Management 8.3 has the following new features and enhancements:

- Reports
- Performance enhancements

Reports

You can export model data for various periods and scenarios to generate the following types of reports:

- Destination Furthest
- Driver - Cost and Rate
- Multi-Level Contributions
- Resource Contribution
- Unassigned Costs
- Unit Cost
- Dimensional Attribute Cost

SAS Cost and Profitability Management contains a template for each report type. SAS Cost and Profitability Management exports the model data in a table in the database. Use the database tables to view, share, and analyze the report.

Performance Enhancements

Performance enhancements made to SAS Cost and Profitability Management 8.3 ensure a better user experience.

Part 1

What's Changed from SAS Activity-Based Management

Chapter 1

Changes from the Predecessor Product

Introduction

This section is for users of SAS Activity-Based Management. It describes differences between SAS Cost and Profitability Management and SAS Activity-Based Management so that users of the older product know what they need to do differently in SAS Cost and Profitability Management. It also describes changes to existing models that are migrated from SAS Activity-Based Management to SAS Cost and Profitability Management. Users who are new to SAS Cost and Profitability Management who are not migrating from its predecessor product, SAS Activity-Based Management, can skip this section.

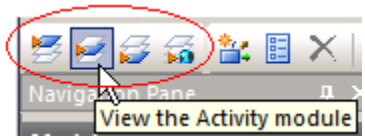
The differences are the following:

- “More Modules” on page 5
- “Consolidation of Drivers” on page 6
- “Profitability Module No Longer Exists” on page 11
- “Entered Cost Elements Now Have Two Kinds of Cost” on page 11
- “External Units Are Now Ordinary Accounts” on page 13
- “Column Layouts Are Simpler” on page 16
- “Backing Up Models Done Differently” on page 18
- “Attributes on Dimension Members Are Applied Automatically” on page 19
- “You Must Acquire a Write Lock on a Model to Edit It” on page 16
- “Some Public Views Are No Longer Generated Automatically” on page 20
- “You Can Export More Assignments and Survey Different Fields” on page 25
- “Some Properties Are Obsolete” on page 27
- “Database Connections Done Differently” on page 28
- “Database Connections Done Differently” on page 28
- “Import Catches More Errors” on page 28

More Modules

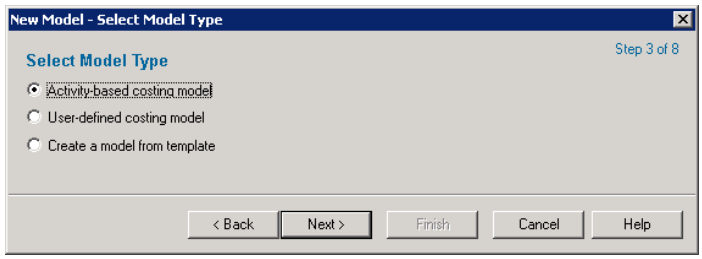
Previously:

Every model had at most four modules (External Unit, Resource, Activity, and Cost Object), each of which you could access by clicking its icon on the tool bar, as shown in the following picture:

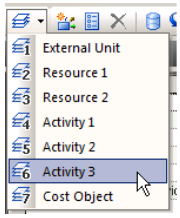


And, Now:

When you create a new model, you can create it with up to ten modules, and you can name each module whatever you like. You can also continue to create ABC models with the standard four modules. And, you can create a new model from a template that remembers your choices from a previously created model.



Because a model can now contain up to ten modules, it is no longer practical on the tool bar to dedicate a separate icon for each module. Instead, there is one **Select Module** icon from which you activate a drop-down menu to select the module that you want to open.



Consolidation of Drivers

Previously:

In previous releases, there were seven types of drivers whose properties are summarized in the following table:

Type of Driver	Properties Used	How DrvQtyCalc is Calculated
Evenly assigned	DQF	DrvQtyCalc = DQF=1 where DQF is automatically assigned a value of 1 for every assignment path of the driver. DQF is not changeable by a user for this driver.
Percentage	DQF	DrvQtyCalc = DQF
Basic	DQF DQV	DrvQtyCalc = DQF + (DQV x Dest.TDQ)
Weighted	DQF, DWF DQV, DWV	DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
BOC	DQF, DWF DQV, DWV	DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
Sales volume	SoldQty. TDQ	DrvQtyCalc = TDQ
Calculated	See Chapter 55, "Properties That Can Be In Formulas," on page 495.	DrvQtyCalc = the value of the driver formula

And, Now:

Overview

The number of drivers has been consolidated to those shown in the following table:

Type of Driver	Properties Used	Formula
Evenly assigned	DQF	DrvQtyCalc = DQF = 1 DQF is automatically assigned a value of 1 for every assignment path of the driver. DQF is not changeable by a user for this driver.
Percentage	DQF	DrvQtyCalc = DQF

Type of Driver	Properties Used	Formula
Standard	See Chapter 55, "Properties That Can Be In Formulas," on page 495.	DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ) or, a user-specified formula

Standard Driver

The Standard driver replaces the following previous drivers:

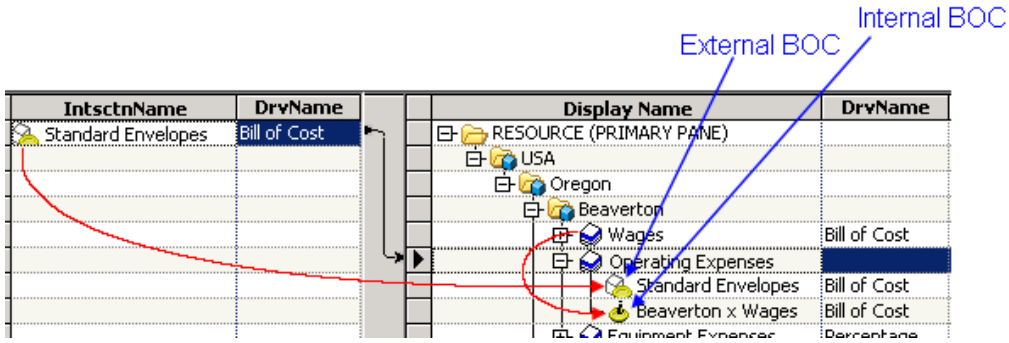
Basic
Weighted
Bill of Cost
Sales Volume
Calculated

You can do anything with a Standard driver that you could have done previously with the drivers that it replaces.

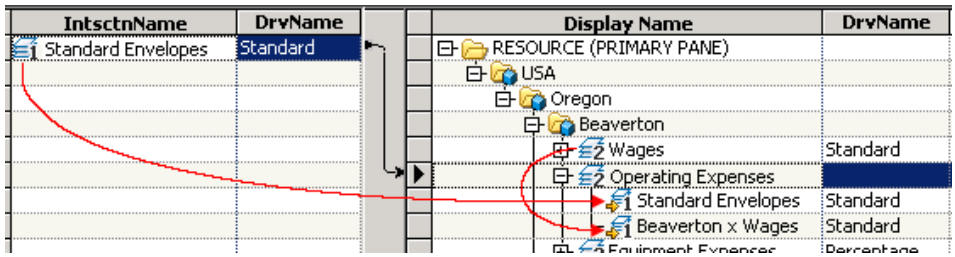
Bill of Cost Driver

Because the Bill of Cost driver no longer exists, the icons that used to represent Internal BOC Cost Elements and External BOC Cost Elements are not longer meaningful. Neither is the property HasBOC, which no longer exists.

The following picture shows how External BOC and Internal BOC used to be represented with distinct icons:

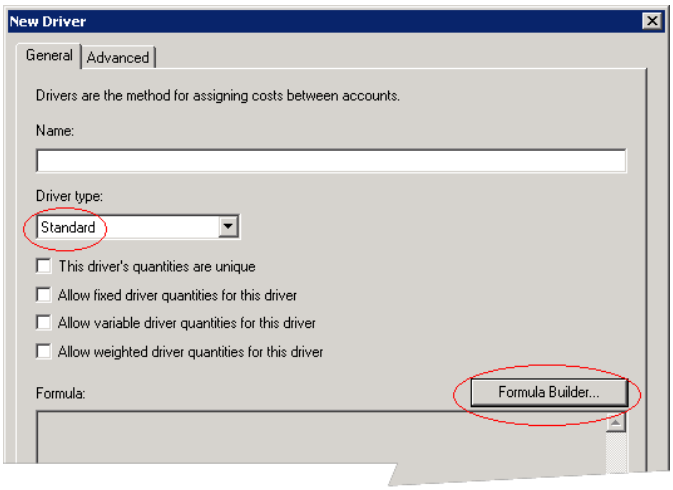


The following picture shows the same assignments with the Standard driver replacing the Bill of Cost driver. Notice that the icons for the received cost elements are the same as for received cost elements from any driver—they are not distinctive.



Calculated Driver

The Calculated driver no longer exists as a separate driver type. To create a calculated driver now, you create a standard driver with a driver formula.



See Chapter 44, "Calculated Driver," on page 393.

Sales Volume Driver

The Sales Volume driver was characterized by the following two facts:

- It was a rule-based driver—it used a system-generated rule formula to create assignments.
- It was a calculated driver. Instead of using the following standard system formula for calculating DrvQtyCalc (the number of units going to a destination account):

  ```
  DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
  ```

  it used the following formula:

  ```
  DrvQtyCalc=SoldQty
  ```

The following describes what happens to the Sales Volume driver now:

- If you import a model from a previous version of SAS Activity-Based Management, and that model contains a Sales Volume driver, then the Sales Volume driver is converted to a Standard driver with a rule formula and a driver formula. This Standard driver creates the same assignments and returns the same value for DrvQtyCalc as did the Sales Volume driver from the previous release.
- If you want to create a driver now that does exactly what a Sales Volume driver did previously, then you should use a Standard driver with a rule formula and the driver formula **TDQ**.

The following describes how the Sales Volume driver used to create assignments and how you can create the same assignments with a rule-based driver.

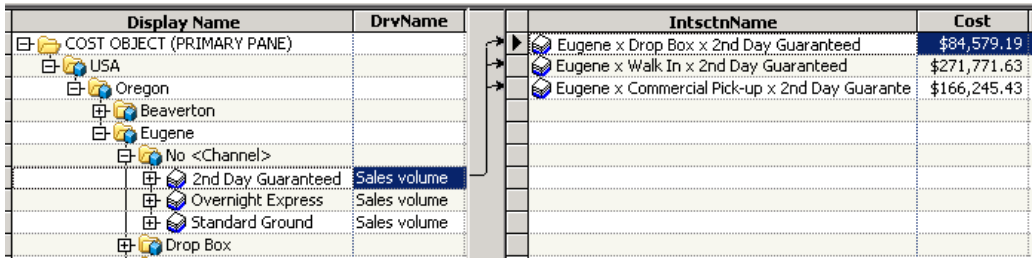
When you attached a Sales Volume driver to an account, then calculation automatically made an assignment to every possible destination account for which both of the following conditions are true:

1. SoldQuantity is non-zero in the destination account.
2. Speaking roughly, source account dimension members match destination account dimension members. Speaking precisely, for every dimension in the Profitability module:
   - either the dimension member in the source account for that dimension is the No dimension member (see "IsNone function" on page 481), or

- the dimension member in the source account for that dimension has a matching dimension member in the dimension signature of the destination account. For information on the dimension signature, see "Introduction" on page 223.

Note: In general, the dimensions in the Cost Object module are the same as those in the Profitability module. So, in general, the second condition can be restated as follows: "For every dimension in the Cost Object module..".

For example, consider the following picture:



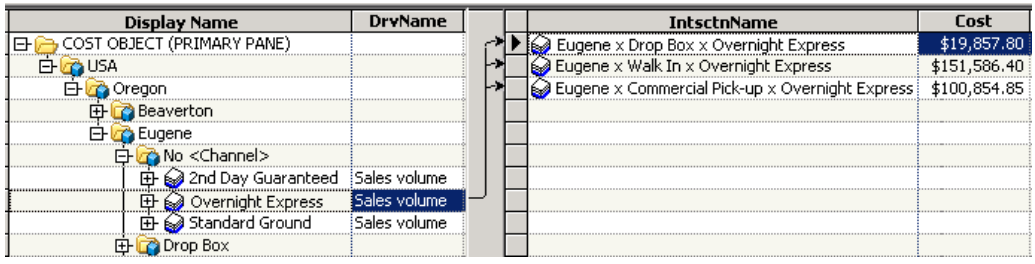
You can see that the dimension members of the source account match the dimension members of the destination account as shown in the following table:

This source account ...	matches the following destination accounts ...
Eugene x No <Channel> x 2nd Day Guaranteed	Eugene x Drop Box x 2nd Day Guaranteed Eugene x Walk In x 2nd Day Guaranteed Eugene x Commercial Pick-Up x 2nd Day Guaranteed

That is:

- For the Region dimension, Eugene in the source account matches Eugene in the destination account.
- For the Channel dimension, No <Channel> in the source account matches Drop Box, Walk In, and Commercial Pick-Up in the destination account.
- For the Product dimension, 2nd Day Guaranteed in the source account matches 2nd Day Guaranteed in the destination account.

Consider, also, the following picture:



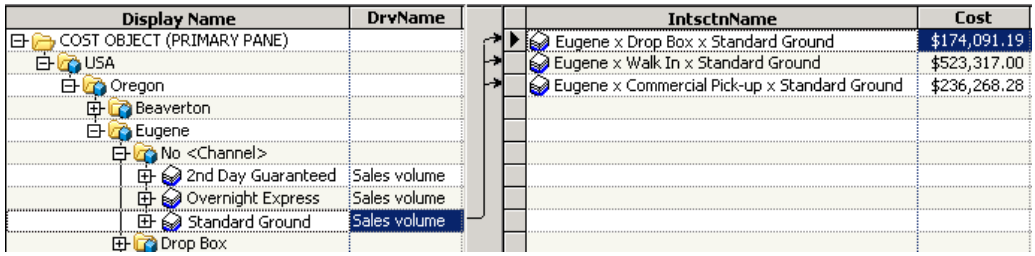
Again, you can see that the dimension members of the source account match the dimension members of the destination account, as shown in the following table:

This source account ...	matches the following destination accounts ...
Eugene x No <Channel> x Overnight Express	Eugene x Drop Box x Overnight Express Eugene x Walk In x Overnight Express Eugene x Commercial Pick-Up x Overnight Express

That is:

- For the Region dimension, Eugene in the source account matches Eugene in the destination account.
- For the Channel dimension, No <Channel> in the source account matches Drop Box, Walk In, and Commercial Pick-Up in the destination account.
- For the Product dimension, Overnight Express in the source account matches Overnight Express in the destination account.

Finally, consider the following picture:



And, again you can see that the dimension members of the source account match the dimension members of the destination account, as shown in the following table:

This source account ...	matches the following destination accounts ...
Eugene x No <Channel> x Standard Ground	Eugene x Drop Box x Standard Ground Eugene x Walk In x Standard Ground Eugene x Commercial Pick-Up x Standard Ground

That is:

- For the Region dimension, Eugene in the source account matches Eugene in the destination account.
- For the Channel dimension, No <Channel> in the source account matches Drop Box, Walk In, and Commercial Pick-Up in the destination account.
- For the Product dimension, Standard Ground in the source account matches Standard Ground in the destination account.

To make the same assignments, you can use the following rule formula:

```
IntersectionMatch(Region, Chnnl, Prod_Serv) AND SoldQuantity <> 0
```

Note: When a model from a previous release is migrated to or imported into SAS Cost and Profitability Management, any occurrence of a SalesVolume driver is automatically converted to a rule-based driver with a rule formula that uses the IntersectionMatch() function. You can use these automatically converted occurrences to familiarize yourself with the IntersectionMatch() function.

Profitability Module No Longer Exists

Previously:

The Profitability module was used for defining the dimensions for the Sales Volume driver.

And, Now:

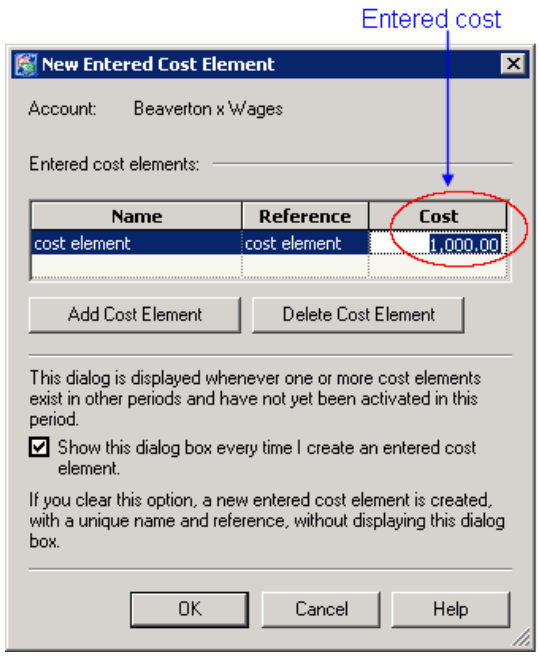
Because as part of the consolidation of drivers the Sales Volume driver no longer exists, there is no longer any need for a Profitability module. Therefore, it no longer exists. See "Sales Volume Driver" on page 8.

Entered Cost Elements Now Have Two Kinds of Cost

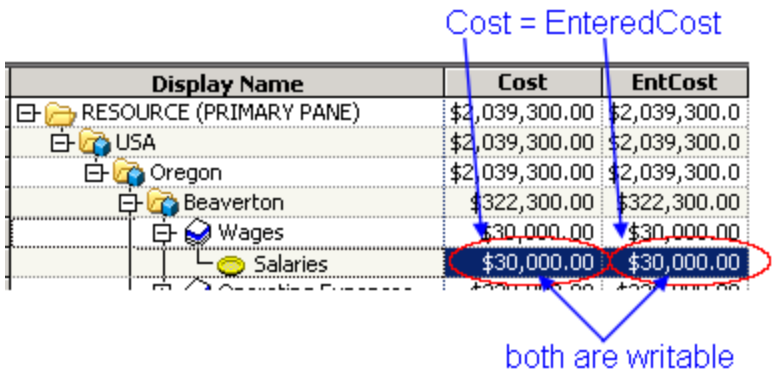
Previously:

Previously, an entered cost element:

- contained only one kind of cost—a lump-sum Entered Cost.
- could not exist in the External Unit module.



The following picture shows an entered cost element in a module view. Notice that the Cost in the cost element equals EnteredCost. And, because the two are the same, you could write into either the Cost column or the EnteredCost column of the cost element.



And, Now:

An entered cost element:

- can contain one or both of the following kinds of cost:

  EnteredCost
  : EnteredCost is a lump-cost assigned to a cost element.

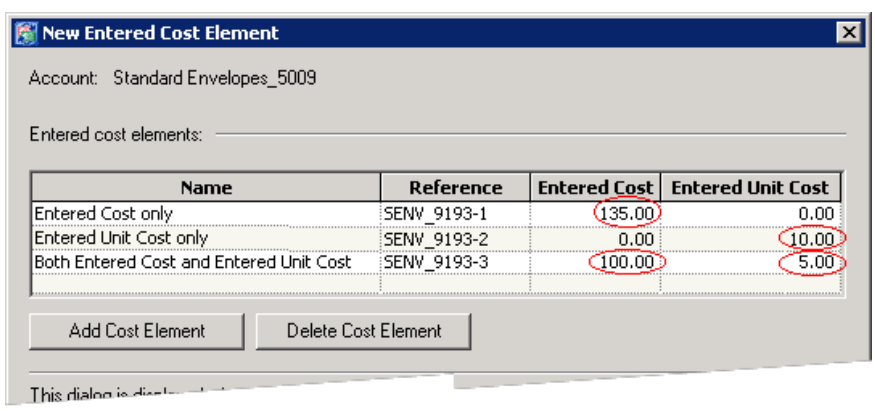
  EntUnitCost
  : EntUnitCost is what used to be referred to in the External Unit module as "UnitCost". EntUnitCost is the user-entered cost per unit.

    You can specify EntUnitCost for any cost entered element for any account in any module.

    The cost in a cost element is the sum of its EnteredCost plus its EntUnitCost times the TDQ of its parent account (the number of units output by the account).

- can belong to an account in any module.

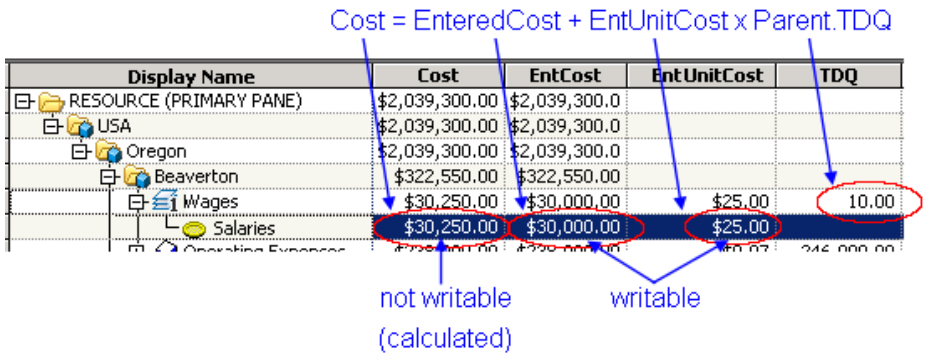
Now, all the cost in a model originates with entered cost elements.



The following picture shows an entered cost element in a module view. Notice that the cost element has both EnteredCost and EntUnitCost. The Cost in the cost element is calculated according to the following formula:

```
Cost = EnteredCost + (EntUnitCost x Parent.TDQ)
```

And, because the Cost of a cost element is now a calculated value and not a synonym for EnteredCost, you can not enter values into the Cost column of a cost element in the module view.



Display Name	Cost	EntCost	EntUnitCost	TDQ
RESOURCE (PRIMARY PANE)	$2,039,300.00	$2,039,300.0		
USA	$2,039,300.00	$2,039,300.0		
Oregon	$2,039,300.00	$2,039,300.0		
Beaverton	$322,550.00	$322,550.00		
Wages	$30,250.00	$30,000.00	$25.00	10.00
Salaries	$30,250.00	$30,000.00	$25.00	

External Units Are Now Ordinary Accounts

Previously:

In previous releases, the External Unit module differed from other modules because it contained External Units instead of ordinary accounts. External Units differed from ordinary accounts:

- a user could specify its Unit Cost, whereas the Unit Cost for ordinary accounts was calculated by the system. The name “Unit Cost”, thus, was ambiguous—sometimes denoting a user-entered property and sometimes denoting a system-calculated property.
- a user could not specify Entered Cost for an External Unit, whereas a user could specify Entered Cost for ordinary accounts.

  Note: More accurately, a user could add Entered Cost Elements to an account and the total of all Entered Cost Elements rolled-up to the account level as the account’s Entered Cost.

The following picture shows an External Unit module with three dimension members (Envelopes, Flats, Boxes) and five External Units:

Standard Envelopes
2nd Day Flat
Overnight Flat
Large Box
Small Box

Display Name	Cost	UnitCost	EntCost
EXTERNAL UNITS	$195,885.60		$0.00
Envelopes	$1,560.60		$0.00
Standard Envelopes	$1,560.60	$0.06	
Flats	$18,200.00		$0.00
2nd Day Flat	$0.00	$0.14	
Overnight Flat	$18,200.00	$0.14	
Boxes	$176,125.00		$0.00
Large Box	$118,750.00	$0.95	
Small Box	$57,375.00	$0.75	

You can see that each External Unit has a user-specified UnitCost and no EnteredCost.

And, Now:

Overview

The External Unit module differs only in name ("External Unit") from other modules. Now, an External Unit module contains ordinary accounts and is like any other module.

The following picture shows the same External Unit module with ordinary accounts instead of External Units.

Display Name	Cost	EntUnitCost	EntCost
External Unit (Primary Pane)	$195,985.60	$1.29	$57,475.00
Envelopes	$1,560.60	$0.06	$0.00
Standard Envelopes	$1,560.60	$0.06	$0.00
Standard Envelope_ce	$1,560.60	$0.06	$0.00
Flats	$18,200.00	$0.28	$0.00
2nd Day Flat	$0.00	$0.14	$0.00
2nd Day Flat_ce	$0.00	$0.14	$0.00
Overnight Flat	$18,200.00	$0.14	$0.00
Overnight Flat_ce	$18,200.00	$0.14	$0.00
Boxes	$176,225.00	$0.95	$57,475.00
Large Box	$118,850.00	$0.95	$100.00
Large Box_ce	$118,850.00	$0.95	$100.00
Small Box	$57,375.00	$0.00	$57,375.00
Small Box_ce	$57,375.00	$0.00	$57,375.00

EntUnitCost and EntCost both roll up

Only EntUnitCost

Both EntUnitCost and EntCost

Only EntCost

In this picture you can see the following:

Cost elements

An account can contain cost elements, and a cost element can contain one or both of the following:

- EntUnitCost (Entered Unit Cost)
- EntCost (Entered Cost)

Roll-up

EntUnitCost and EntCost roll up from the cost-element level to the account level:

EntUnitCost
: EntUnitCost for an account is the sum of the EntUnitCost of its cost elements.

EntCost
: EntCost for an account is the sum of the EntCost of its cost elements.

EntUnitCost vs. UnitCost

EntUnitCost is to be distinguished from UnitCost:

EntUnitCost
: In addition to being user-entered, it is a property of a cost element and rolls-up as a derived property of an account.

  Any account in any module can contain a cost element with EntUnitCost.

UnitCost
: is a property of an account and is calculated by the system with the following formula:

```
UnitCost = Cost / OutQty
```

  Every account in every module has a calculated UnitCost which may be identical to its EntUnitCost (assuming that the account contains cost elements with EntUnitCost).

Cost

The Cost in an account is calculated according to the following formula:

```
Cost = EntCost + (EntUnitCost x TDQ)
```

In other words, the cost in an account is its entered cost plus its cost per unit times the number of units of its output.

Note: This formula applies because, as mentioned before, EntUnitCost and EntCost roll up from the cost-element level to the account level. Similarly, the cost in a cost element is the sum of its EntCost plus its EntUnitCost times TDQ of its parent account.

In order to emphasize the point that the External Unit module is now like any other module, the following picture shows a Cost Object module with the same structure of accounts as the previous External Unit module. You can see that accounts in any module can have EntUnitCost.

Display Name	Cost	EntUnitCost	EntCost
Cost Object (Primary Pane)	$195,985.60	$1.29	$57,475.00
Portland County	$1,560.60	$0.06	$0.00
Portland City	$1,560.60	$0.06	$0.00
Portland City_ce	$1,560.60	$0.06	$0.00
Washington County	$18,200.00	$0.28	$0.00
Seattle City	$0.00	$0.14	$0.00
Seattle City_ce	$0.00	$0.14	$0.00
Tacoma City	$18,200.00	$0.14	$0.00
Tacoma City_ce	$18,200.00	$0.14	$0.00
Wake County	$176,225.00	$0.95	$57,475.00
Cary City_ce	$118,850.00	$0.95	$100.00
Cary City	$118,850.00	$0.95	$100.00
Raleigh City_ce	$57,375.00	$0.00	$57,375.00
Raleigh City	$57,375.00	$0.00	$57,375.00

EntUnitCost and EntCost both roll up

Only EntUnitCost

Both EntUnitCost and EntCost

Only EntCost

To Summarize

- UnitCost is now always a calculated value and you can not modify it. UnitCost is calculated according to the following forumula:

  ```
  UnitCost = Cost/OutputQty
  ```

  Previously, UnitCost in an External Unit denoted what is now referred to as EntUnitCost and you could modify it. Now, UnitCost and EntUnitCost are unrelated.

- To specify EntUnitCost you must now specify it on a cost element—not on an account. And, any account in any module can have a cost element with EntUnitCost—not just External Units in the External Unit module.

- Now when you create an activity-based costing model, it contains by default an External Unit module. However, that module differs from other modules only in name—it has the same characteristics and capabilities as any other module. If you want, you can delete the External Unit module.

  If you are creating a user-defined costing model, you can choose to dedicate one module to play the role that used to be played by the External Unit module—to contain entities that are external to the model and for which you specify the entered unit cost.

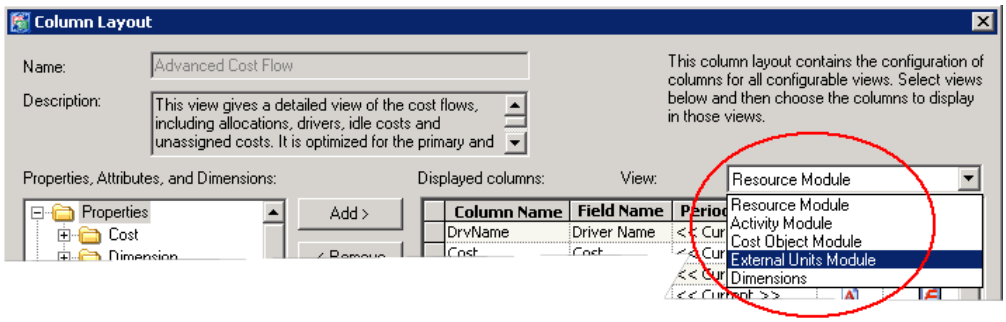
- When you import an model from a previous version for SAS Activity-Based Management, External Units in the External Unit module are converted to ordinary accounts in a module named “External Unit”. Furthermore, a cost element is automatically created for each of those accounts. Such a cost element has the EntUnitCost property to replace the UnitCost property that the External Unit used to have.

- The staging table for External Units no longer exists. EntUnitCost is now in the EnteredCostElement staging table. See “EnteredCostElement table” in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

Column Layouts Are Simpler

Previously:

A column layout contained the layout for every module in a model, plus the layout for the Dimensions view.



And, Now:

A saved column layout contains the layout for a single module and can be applied to any module or to the Dimensions view.

See Chapter 37, “Working with Column Layouts,” on page 319.

You Must Acquire a Write Lock on a Model to Edit It

Previously:

Because models were stored in a database, the database management system handled model access. Multiple users could edit a model at the same time, and multiple users could view a model at the same time as another user was editing it.

And, Now:

Overview

Because models are no longer stored in a database, there is no database management system to handle model access. If you want to edit a model, you must first acquire a write lock on it. If you want to view a model, you must acquire a read lock on it. Only one user can acquire a write lock on a model at a time and edit that model. However,

when one user has a write lock on a model, multiple users can acquire a read lock on the model to view it.

The following describes how to open and close models for editing or viewing.

Write Lock

You must acquire a write lock on a model to edit it. When you have a write lock, no one else can acquire a write lock on the model. You must release the write lock on the model so that another user can acquire a write lock on it.

Acquire

To acquire a write lock, double-click the model in the Workspace Manager. Alternatively, right-click the model and select **Open**. The model opens for editing.

Note: A modeler must have read/write access to open the model with a write lock.

Release

- To release the write lock on a model, the modeler with the lock can select **Model ⇨ Close Model and Release Lock** when the model is open. Selecting this action closes the model.

  TIP It is good practice to release the write lock on a model that you are no longer actively working on so as to free up system resources. Each write lock involves a running process that uses system memory. So, if you are not actively working on a model, it is desirable that you release its write lock to free up memory.

- The write lock that is held by a modeler is released if the modeler closes the client application. But, the write lock is not released if the modeler acquires a write lock on another model.

  Note: If you want to release a lock by closing the model or by closing the client application when an operation (such as exporting data, calculating costs, and so on) is in progress, you must cancel the operation first or wait for the operation to finish.

- An administrator or a modeler with read/write access to the model can release any user's lock on the model by selecting **Tools ⇨ Model Locking Status**. See "Model Locking Status Dialog Box" on page 99.

Read Lock

When you open a model for viewing, you automatically acquire a read lock on it. Multiple users can open a model with a read lock and view it simultaneously.

Acquire

- If you have read access to the model, double-click the model in the **Workspace Manager** to acquire the read lock. Alternatively, right-click the model and select **Open**.

- If you have read/write access to the model, right-click the model in the **Workspace Manager** and then select **Open with Read Lock** to acquire the read lock on the model.

Release

- To release the read lock on a model, the modeler with the read lock can do any of the following:

  - In the **Workspace Manager**, right-click the model and select **Release Read Lock**.

  - Select **Model ⇨ Close Model and Release Lock**.

  - Acquire a read lock on a different model.

- Close the client application.

Note: If you want to release a lock by closing the model or by closing the client application when an operation (such as exporting data, calculating costs, and so on) is in progress, you must cancel the operation first or wait for the operation to finish.

- An administrator or a modeler with read/write access to the model can release the lock on the model by selecting **Tools** ⇨ **Model Locking Status**. See "Model Locking Status Dialog Box" on page 99.

Backing Up Models Done Differently

Previously:

To back up your models, it was sufficient to back up the database in which the models were stored.

And, Now:

Because models are now stored in memory-mapped files rather than in a database, you can no longer back up the database in order to back up your models. SAS Cost and Profitability Management provides a batch utility for backing up and restoring all your models. See Chapter 17, "Backup and Restore Utility," in *SAS Cost and Profitability Management: Data Administration Guide*.

Stage Names and References Can Be Whatever You Want

Previously:

To specify the direction of cost flow between stages, you were required to follow a strict naming convention such that the alphabetic order of stage references determined the direction of cost flow between stages—for example, stage1 flows to stage2 which flows to stage3, etc.

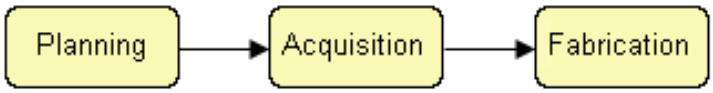
And, Now:

The direction of cost flow between stages is determined by the display order of stage attributes rather than by their reference, so you can assign any name and reference to stages that you want. For example, in the following picture you can see that the display order of three stage attributes is:

1. Planning
2. Acquisition
3. Fabrication

Name	Reference	Type
ATTRIBUTES (PRIMARY PANE)		
Stages	Stages	Dimension
Planning	Planning	Dimension Member
Acquisition	Acquisition	Dimension Member
Fabrication	Fabrication	Dimension Member

Consequently, *Planning* flows to *Acquisition* which flows to *Fabrication* because that is their display order. (Previously, *Acquisition* would have flowed to *Fabrication* which would have flowed to *Planning* because that is the alphabetic order of their references)



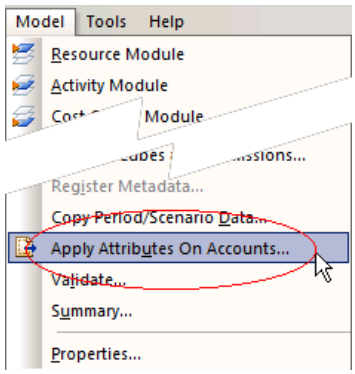
Note: When you migrate a model from a previous release, the new display order for stages is made to be the same as the alphabetic order of the stage references.

Attributes on Dimension Members Are Applied Automatically

Previously:

After attaching an attribute to a dimension member, in order to apply that attribute to the accounts whose dimension signatures included that dimension member, you had to either:

- calculate the model
- select **Model** ⇨ **Apply Attributes On Accounts**



See Chapter 34, “Attributes on Dimension Members,” on page 277.

And, Now:

To apply attributes on dimension members to accounts, it is no longer necessary either to calculate the model or to invoke a menu action. Attributes on dimension members are now automatically applied to the appropriate accounts when you attach an attribute to a dimension member. And, they are automatically removed from accounts when you remove an attribute from a dimension member.

Some Public Views Are No Longer Generated Automatically

Previously:

Public views were automatically registered in SAS Management Console to mirror the database tables that constituted a model and its cubes and fact tables.

The following model-independent views were created at the time of database creation:

PublicModel
PublicModelStatus
PublicPeriod
PublicScenario

The following model-specific views were created during model creation:

*modelRef*_PV_Assignment
*modelRef*_PV_AssignmentExt
*modelRef*_PV_AttributeValue
*modelRef*_PV_Dimension
*modelRef*_PV_DimMember
*modelRef*_PV_Driver
*modelRef*_PV_Attributes
*modelRef*_PV_EnteredCE

The following model-specific views were created during model calculation:

*modelRef*_PV_Account
*modelRef*_PD_*dimShortRef*
*modelRef*_PD_CostElement
*modelRef*_PD_Driver
*modelRef*_PD_Module
*modelRef*_PD_Period
*modelRef*_PD_Scenario
*modelRef*_PD_YesNo

The following model-specific views were created during cube/fact table generation

*modelRef*_PF_RC
*modelRef*_PF_SSC
*modelRef*_PF_*cubeConfigRef*
*modelRef*_PF_MSC

And, Now:

Because models are no longer stored in a database, database tables and the public views that are based on them no longer exist. However, you can still create the same database tables and views as in previous releases.

The following tables describe the structure of a model. You must create these tables explicitly by selecting **Model** ⇨ **Export and Register Tables**. See "How To Export and Register Tables" on page 701.

Table Name	Description
*ModelRef*_ACCOUNTMAP	A join of *ModelRef*_PV_ACCOUNT with other registered tables. Includes details for multiple periods, dimensional signature, and all numeric properties. See "modelRef_ACCOUNTMAP " in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_ASSIGNMENTMAP	A join of *ModelRef*_PV_ASSIGNMENT with other registered tables. Includes details for source and destination dimensional signatures, and all entered and calculated properties. See "modelRef_ASSIGNMENTMAP" in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ACCOUNT	Details for multiple periods, dimensional signature, and all numeric properties. See "modelRef_PV_ACCOUNT " in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ASSIGNMENT	Details for source and destination dimensional signatures, and all entered and calculated properties. See "modelRef_PV_ASSIGNMENT " in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ASSIGNMENTEXT	Details for source and destination dimensional signatures, and all entered and calculated properties. See "modelRef_PV_ASSIGNMENTEXT " in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ATTRIBUTE	Details defining the attribute, attribute types, formulas for calculations, and default values. See "modelRef_PV_ATTRIBUTE" in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

Table Name	Description
*ModelRef*_PV_ATTRIBUTEVALUE	Attribute attachment to the model accounts and the numeric value for the attribute. See “modelRef_PV_ATTRIBUTEVALUE ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_DIMENSION	Details defining the model's dimensions. See “modelRef_PV_DIMENSION ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_DRIVER	Details defining the drivers, driver types, and formulas for rules assignment and for calculations. See “modelRef_PV_DRIVER ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ENTEREDCE	Details of all entered cost elements, with their corresponding account attachment and values. See “modelRef_PV_ENTEREDCE ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

The following model-independent views are created automatically during model definition.

View Name	Description
PUBLICMODEL	Contains a list of models. See “PUBLICMODEL ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICMODELSTATUS	For each model, lists its period scenario associations and whether each one is calculated or not.. See “PUBLICMODELSTATUS” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICPERIOD	Contains a list of periods with their start and end dates. See “PUBLICPERIOD ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICSCENARIO	Contains a list of scenarios. See “PUBLICSCENARIO ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

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

View Name	Description
*ModelRef*_PD_COSTELEMENT<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> 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 “modelRef_PD_COSTELEMENT<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_*dimShortRef*<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_REGIONRC ABC_PD_REGIONSSC ABC_PD_REGIONC1001	Single dimension-member details: level by level, noting ID, Reference, and Name. See “modelRef_PD_dimShortRef<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_DRIVER<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_DRIVERRC ABC_PD_DRIVERSSC ABC_PD_DRIVERC1001	Driver ID and corresponding Driver Name. See “modelRef_PD_DRIVER<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_MODULE<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_MODULERC ABC_PD_MODULESSC ABC_PD_MODULEC1001	Details defining the types of modules. See “modelRef_PD_MODULE<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

View Name	Description
*ModelRef*_PD_PERIOD<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_PERIODRC ABC_PD_PERIODSSC ABC_PD_PERIODC1001	Details defining the periodic hierarchy. See “modelRef_PD_PERIOD<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_SCENARIO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_SCENARIORC ABC_PD_SCENARIOSSC ABC_PD_SCENARIOC1001	Details defining the types of available scenarios. See “modelRef_PD_SCENARIO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_YESNO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_YESNORC ABC_PD_YESNOSSC ABC_PD_YESNOC1001	Dimensional definition for Boolean values: Text strings. See “modelRef_PD_YESNO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

View Name	Description
*ModelRef*_PF_<**suffix**> where <**suffix**> is RC, SSC, or MSC_C<cubeID> for a multi-stage contributions cube. For example: ABC_PF_RC ABC_PF_SSC ABC_PF_MSC_C1001 If you have checked **Also load tables into a library for the SAS LASR Analytic Server**, 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: *ModelRef*_PF_<**suffix**> where <**suffix**> is RCSTAR, SSCSTAR, or MSC_C<cubeID>STAR for a multi-stage contributions cube. For example: ABC_PF_RCSTAR ABC_PF_SSCSTAR ABC_PF_MSC_C1001STAR	Fact Table: Stages and member IDs for each step through contribution. Source table for cube generation. See “modelRef_PF_<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide* .

See Also

- Chapter 15, “Working with Registered Tables,” in *SAS Cost and Profitability Management: Data Administration Guide*
- Chapter 16, “Registered Table Schemas,” in *SAS Cost and Profitability Management: Data Administration Guide*

You Can Export More Assignments and Survey Different Fields

Previously:

SAS Activity-Based Management imposed a limit of 100,000 assignments when you exported survey data.

The following table summarizes for each type of survey all the staging-table fields that the SAS Activity-Based Management Surveys application allowed to be surveyed.

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

Module	Type of survey	Fields that can be updated
External Unit	Quantity: (*account*)	Assignment.DriverQuantityFixed
	Unit Costs	ExternalUnit.UnitCostEntered
Resource	Resource Drivers	Assignment.DriverQuantityFixed
	Resource Costs	EnteredCostElement.EnteredCost
	Numeric Attribute	ValueAttributeAssociation.NumericValue
Activity	Activity Drivers	Assignment.DriverQuantityFixed
	Numeric Attributes	ValueAttributeAssociation.NumericValue
Cost Object	Cost Object Drivers	Assignment.DriverQuantityFixed
	Revenues and Sold Quantities	Account.Revenue Account.SoldQuantity
	Output Quantities	Account.OutputQuantityUE
	Numeric Attributes	ValueAttributeAssociation.NumericValue

And, Now:

SAS Cost and Profitability Management imposes no limit as to the number of assignments that you can export with your survey data.

The following table summarizes for each type of survey all the staging-table fields that the SAS Cost and Profitability Management Surveys application allows to be surveyed.

Notice that the following properties are no longer supported:

Revenue
SoldQuantity

These fields will be removed from already-existing surveys that are migrated to the latest release of SAS Cost and Profitability Management.

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 “Assignment table” in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

Type of survey	Fields that can be updated
Entered Cost Element	EnteredCostElement.EnteredUnitCost EnteredCostElement.EnteredCost See "EnteredCostElement table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.
Output Quantities	Account.OutputQuantityUE See "Account table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.
Attribute	ValueAttributeAssociation.NumericValue ValueAttributeAssociation.StringValue See "ValueAttributeAssociation table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

For full documentation on surveys, see Chapter 4, "Introduction to Surveys," in *SAS Cost and Profitability Management: Data Administration Guide*.

Some Properties Are Obsolete

Previously:

The following properties were supported:

ModuleType
HasBOC
ReceivedBOC
ReceivedAssignmentCost

And, Now:

Those properties are no longer meaningful and, consequently, are not supported in any formula.

ModuleType
is obsolete now.

HasBOC and ReceivedBOC
BOC (Bill of Cost) used to be distinguished by the fact that it originated in the External Unit module. Now that the External Unit module no longer exists as a distinctive type of module, there is no longer a significant distinction to be made about where an assignment originates.

ReceivedAssignmentCost
with the elimination of ReceivedBOC, ReceivedAssignmentCost=ReceivedCost. So, ReceivedAssignmentCost is redundant and is replaced by ReceivedCost.

Database Connections Done Differently

Previously:

To connect to a database for importing or exporting, SAS Activity-Based Management used a JDBC driver.

And, Now:

To connect to a database for importing or exporting, SAS Cost and Profitabililty Management uses a SAS server-side driver. You can also use a client-side Microsoft Office driver with Microsoft Access and Microsoft Excel.

See Chapter 73, "Connecting to a Database," on page 617.

Import Catches More Errors

Error checking during import has been strengthened to ensure that it catches the same errors as the UI. For example:

Duplicate cost element names

The UI does not allow two cost elements with the same name under one account even if the cost elements have different references. However, import previously allowed this. Now, import issues a warning in the log and adds a suffix to same-named cost elements to prevent them having the same name.

Name or Reference same as property name

The following description applies to all of the following:

- attribute name
- attribute reference
- dimension name
- dimension reference

Although the UI generally does not allow any of these to be the same as a system-defined property (such as `Cost` or `Type`) so as to prevent ambiguity, import previously allowed this. Now, in order to support existing models where an attribute name or reference, or a dimension name or reference, is already the name of a system-defined property, the following is the case in 8.1 and above:

If an attribute name or reference, or a dimension name or reference, is the name (such as `Cost`) of a system-defined property of an account, entered cost element, or assignment, then:

- The UI does not allow it.
- Import into 8.1 and above fails.
- Migration into 8.1 and above will be successful, but a new reference will be created during migration by appending a number to the existing reference (for example, `Cost` becomes `Cost_1`), and a message in the log file will signal the change.

If an attribute name or reference, or a dimension name or reference, is the name (such as **`Type`**) of a system-defined property of something other than an account, entered cost element, or assignment, then:

- The UI does not allow it.
- Import into 8.1 and above will be successful.
- Migration into 8.1 and above will be successful.

NONE dimension members

Previously, both the UI and import allowed the creation of accounts whose dimension member for every dimension was the NONE dimension member. In 8.1 and above, the creation of such accounts is not allowed. If you import a model containing such an account, import issues the following error message: `Creation of an account with only NONE member reference value for every defining dimension is not supported`, and the import will not create that account.

Functionality Not Supported in this Release

The Performance Measures functionality is not implemented in this release but will be provided in a follow-up release.

The following functionality has been discontinued indefinitely:

- Contributions Query
- Information Maps
- Integration with SAS Strategy Management
- Add-In for SAS Enterprise Guide
- PROC ABC

Part 2

Getting Started

Chapter 2

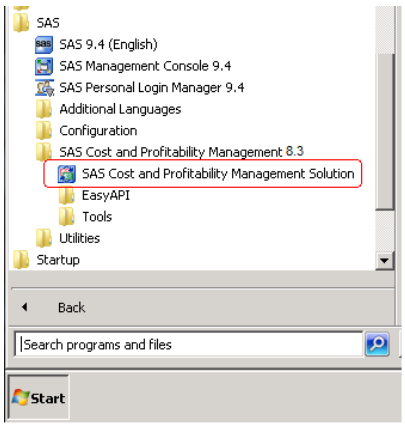
Logging On

Log On Dialog Box

Logging On

To log on to the SAS Cost and Profitability Management client:

1. Select **Start** ⇨ **SAS** ⇨ **SAS Cost and Profitability Management: 8.3** ⇨ **SAS Cost and Profitability Management Solution**

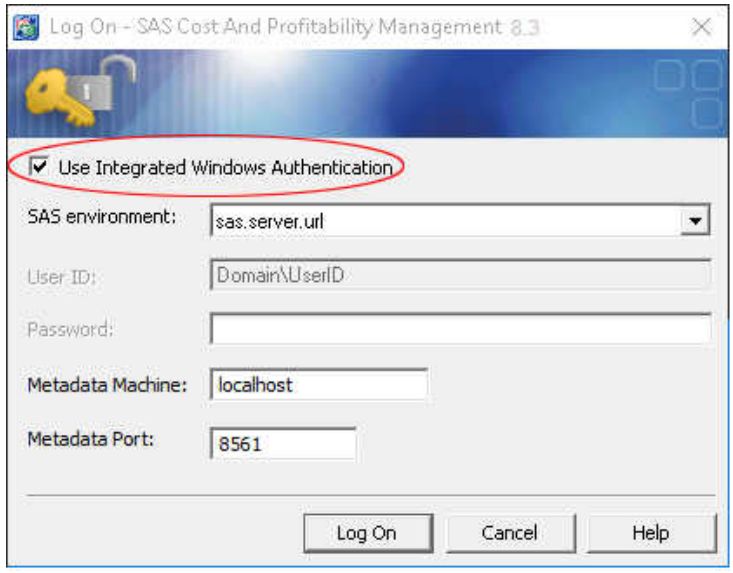


   The log on dialog box opens.

2. Select whether or not to use Integrated Windows Authentication (IWA).

   With IWA, you don't have to specify a user ID and password because you are logged onto SAS Cost and Profitability Management with the same user ID and password that you used to log onto your Windows desktop. See "Enabling Integrated Windows Authentication (IWA)" on page 36.

Use Integrated Windows Authentication



Specify the following:

SAS Environment

From the drop-down list, select a SAS Web Infrastructure Platform server. See "Connecting to a SAS Environment" on page 35.

Metadata Machine

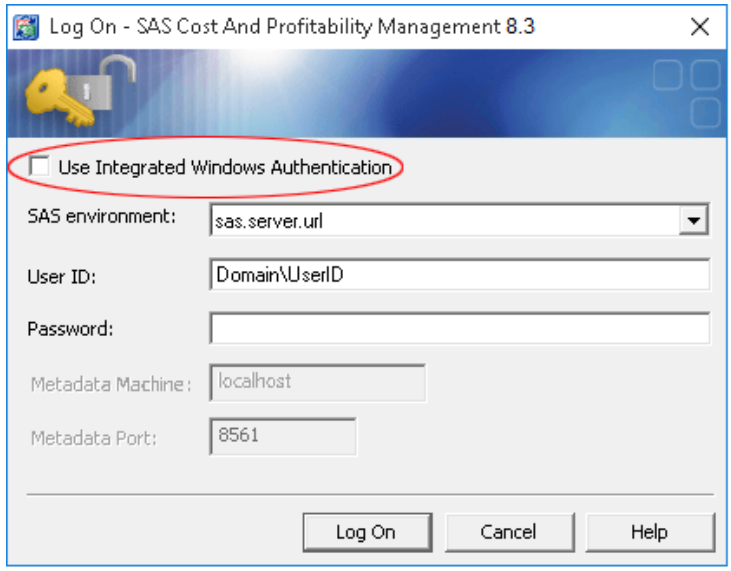
Specify the metadata server machine to use for authentication.

Note: It is not necessary that the metadata server machine be Microsoft Windows. IWA is supported in a Linux environment. The only requirement is that the machine on which the client is running is Windows and the user is logged in to that machine using valid network credentials that correspond to a user account in SAS metadata with SAS Cost and Profitability Management capability.

Metadata Port

Specify the port to use for authentication.

Don't Use Integrated Windows Authentication



Specify the following:

SAS Environment
From the drop-down list, select a SAS Web Infrastructure Platform server. See "Connecting to a SAS Environment" on page 35.

User ID
ID of user registered in SAS Management Console. See "Creating Users" in Chapter 1 of *SAS Cost and Profitability Management: Data Administration Guide*.

Your SAS Cost and Profitability Management user name and password might or might not be the same as your network user name and password.

- If they are the same, then you can use your user name and password to connect to the SAS environment. If IWA is configured, you can connect directly using the IWA option which does not require you to type a user name or password.
- If they are not the same, then you then you must supply the user name as defined in metatdata and the password.

Password
Password of the registered user.

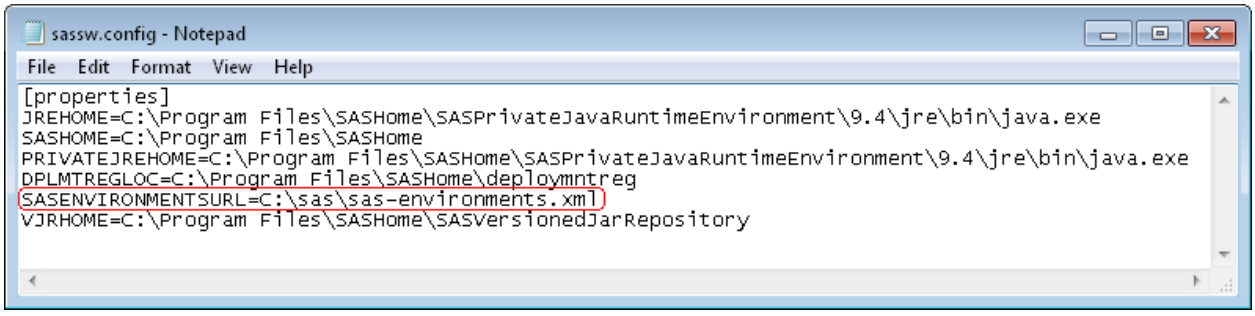
Note: If multiple languages are installed on your computer, the language that is used is determined by the location setting in Windows (select **Settings** ⇨ **Control Panel** ⇨ **Regional and Language Options**).

Connecting to a SAS Environment

Overview

When you log on to SAS Cost and Profitability Management, the file sassw.config (whose default installation location is **`C:\Program Files\SASHome`**) provides the URL of the file sas-environments.xml. The following picture shows a sample sassw.config file containing the following statement pointing to the location of the file sas-environments.xml.

```
SASENVIRONMENTSURL=C:\sas\sas-environment.xml
```



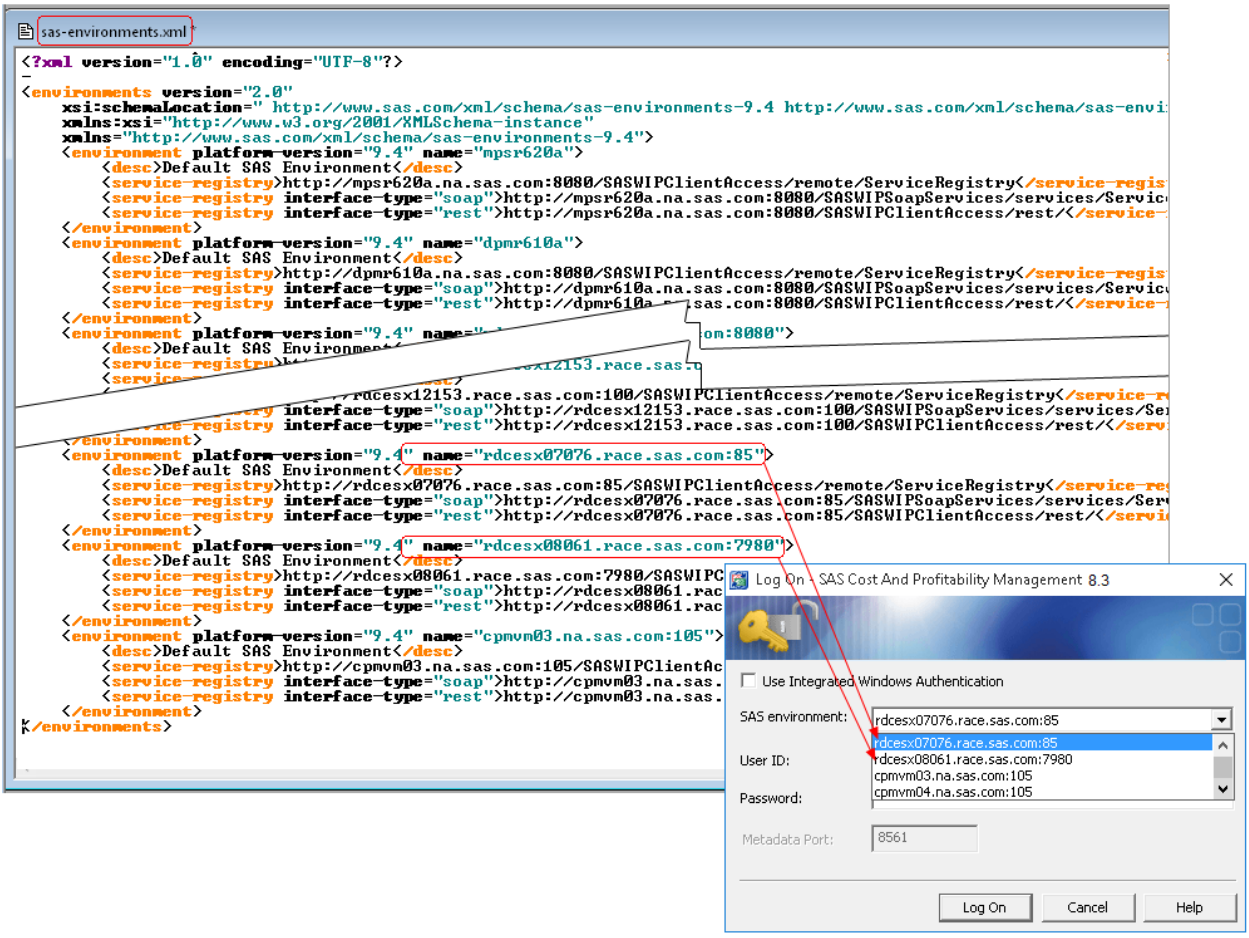
Of course the file sas-environments.xml can reside on another machine. For example:

```
SASENVIRONMENTSURL=http://rde01011.sas.com:8080/SASLogon/sas-environment.xml
```

The file, sas-environment.xml, associates each entry in the SAS environment drop-down list of the Log On dialog with the URL of a SAS WIP Server (Web Infrastructure Platform). The WIP Server authenticates your logon information and provides an interface to a SAS Cost and Profitability Management Metadata Server. Because you, as a client user, log on to a WIP Server instead of directly to a SAS Cost and Profitability

Management Metadata Server, the Metadata Server can be changed without affecting your client log-on procedure.

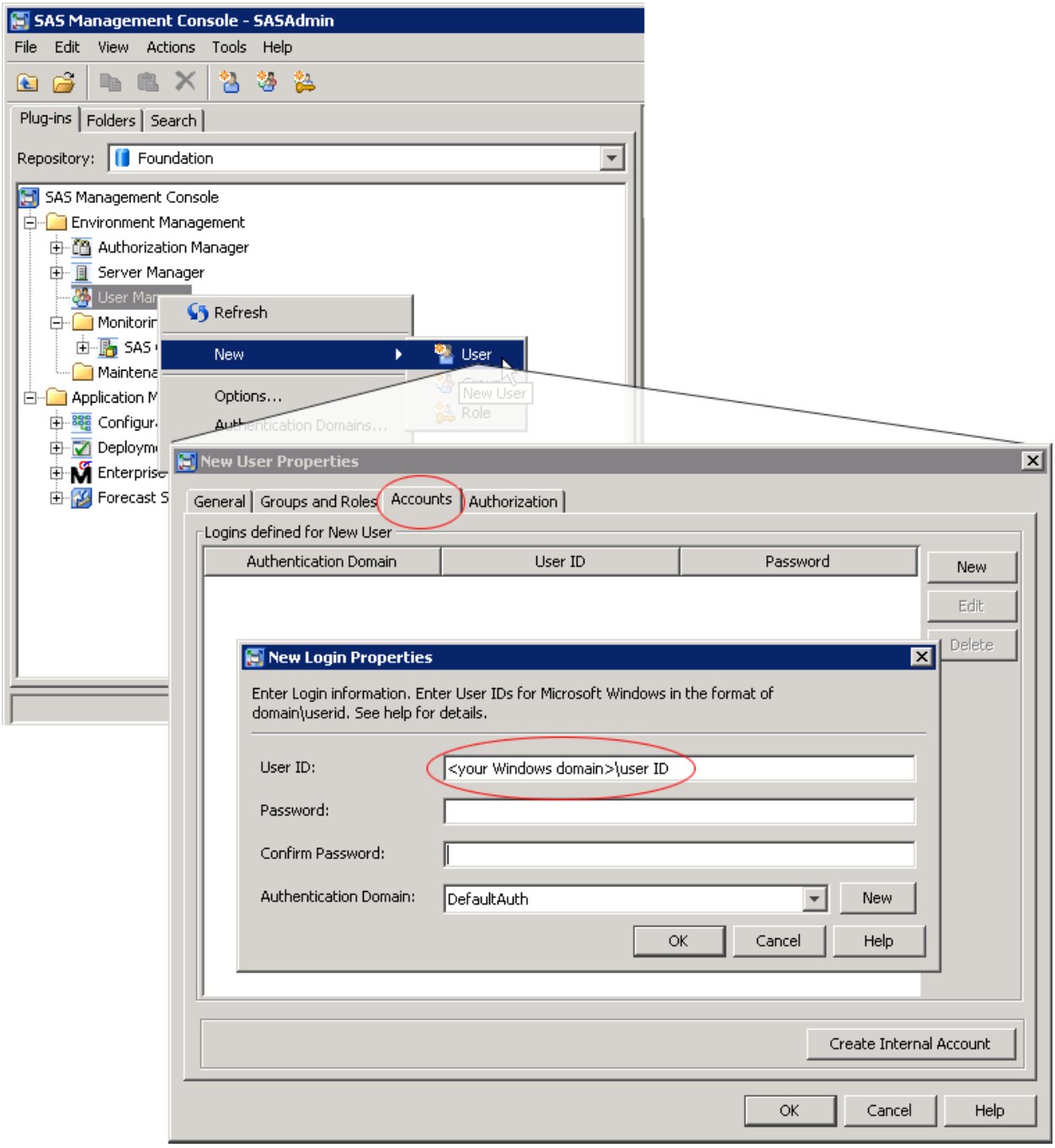
The following picture displays a sample **sas-environments.xml** file with several environments defined.



Enabling Integrated Windows Authentication (IWA)

If your metadata server and mid-tier are configured to accept IWA connections, then you can use Windows authentication for logging on to SAS Cost and Profitability Management. There are three requirements for Windows authentication:

1. You must be using a network (Active Directory) user ID.
2. You must create a user in SAS Management Console with *domain\user ID*, as shown in the following picture. You do not need to put in passwords.



There are two things to consider:

- how the user name appears to the middle tier
- how the user name appears to the object spawner

By default, Tomcat strips the Kerberos realm off the user name so that is how it is seen by the middle tier. On UNIX, the object spawner also strips the realm off the user name in the ticket. So if the server tier is on UNIX, it is sufficient to have a single account in Metadata with no domain or password. For example:

```
myUsername - DefaultAuth
```

On Windows, the object spawner uses the native SSPI API to validate the ticket, and this returns the user name in the form *<domain>\<username>*. So if the server tier is

on Windows, by default you must have two accounts for each user in Metadata. For example:

```
myUsername - DefaultAuth
myDomain\myUsername - DefaultAuth
```

3. On the Logical Workspace Server, **Server Access Security** must be set to **Negotiate**.

   Normally, this is done automatically by the SAS Deployment Wizard (SDW) during installation when you select to do an IWA innstallation. You can verify that it has been done by doing the following:

   a. Open SAS Management Console.

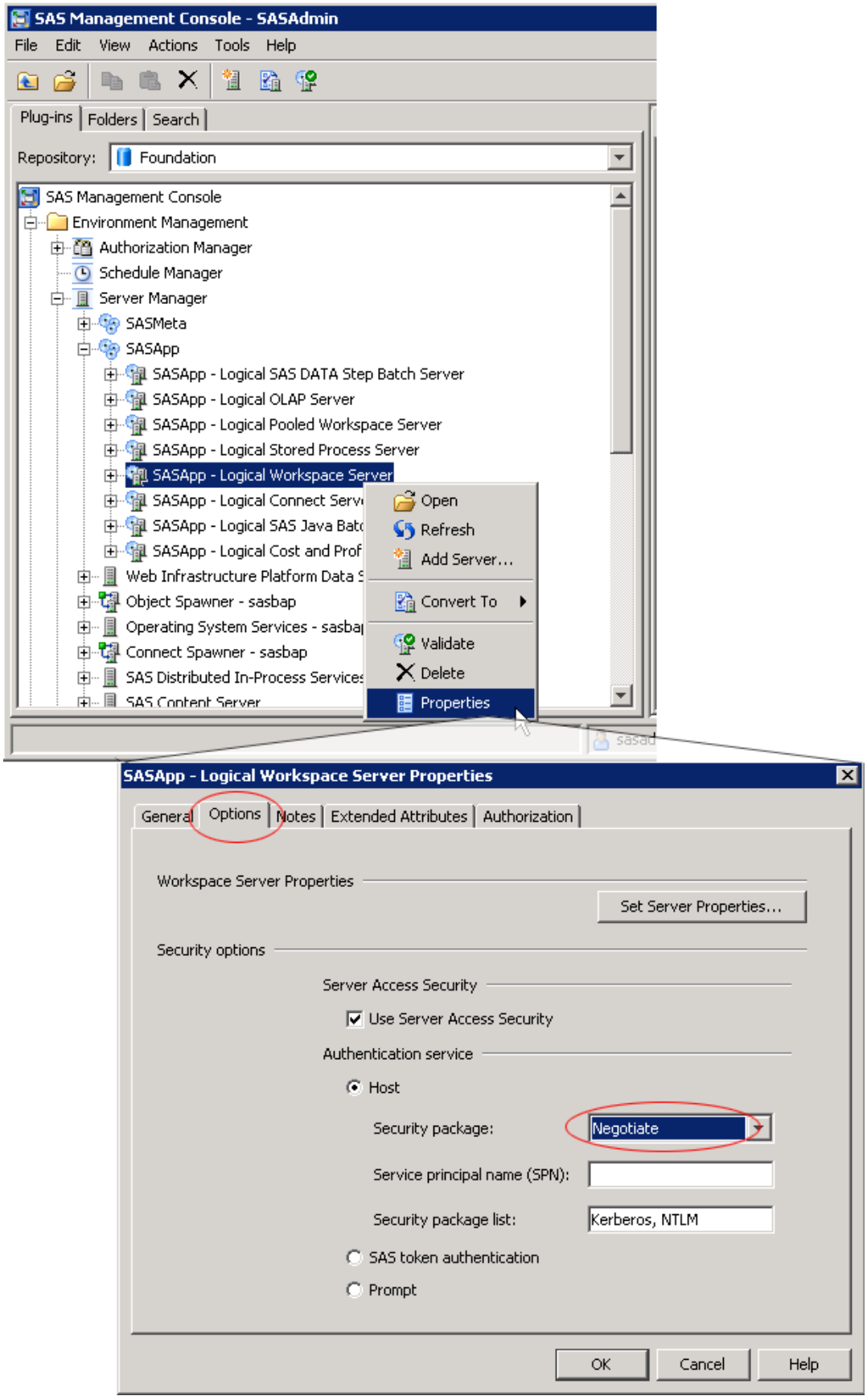
   b. From the **Plug-in** tab of the , expand **Environment Management** ⇨ **Server Manager** ⇨ **SASApp**.

   c. Right-click **SASApp – Logical Workplace Server** and select **Properties**.

   d. Click the **Options** tab.

   e. For **Server Access Security** verify that **Negotiate** is selected.

   f. Click **OK**.


SAS Management Console - SASAdmin
File Edit View Actions Tools Help
Plug-ins Folders Search
Repository: Foundation
SAS Management Console
Environment Management
Authorization Manager
Schedule Manager
Server Manager
SASMeta
SASApp
SASApp - Logical SAS DATA Step Batch Server
SASApp - Logical OLAP Server
SASApp - Logical Pooled Workspace Server
SASApp - Logical Stored Process Server
SASApp - Logical Workspace Server
Open
Refresh
Add Server...
Convert To
Validate
Delete
Properties
SASApp - Logical Workspace Server Properties
General Options Notes Extended Attributes Authorization
Workspace Server Properties
Set Server Properties...
Security options
Server Access Security
Use Server Access Security
Authentication service
Host
Security package:
Negotiate
Service principal name (SPN):
Security package list:
Kerberos, NTLM
SAS token authentication
Prompt
OK
Cancel
Help

Note: If your metada server machine accepts IWA connections (you can log in to SAS Management Console using IWA), then a user can succeed in logging on to the SAS Cost and Profitability Management client even if it was not installed to use IWA—provided that the user has the appropriate SAS Cost and Profitabililty Management capabilities (see Chapter 1, "User Capabilities and Groups," in *SAS Cost and Profitability Management: Data Administration Guide*). If you intend users to log on using IWA, it is recommended that you specify IWA when you install SAS Cost and Profitabililty Management. For more information on enabling SAS Cost and Profitability Management for IWA, see *SAS Cost and Profitability Management 8.3: Installation, Migration, and Configuration Guide*.

Chapter 3

Working in SAS Cost and Profitability Management

Introduction

By applying direct and indirect business costs to activities, SAS Cost and Profitability Management enables managers to get a true understanding of the costs and profits that are associated with a product, customer, service, or business process.

An activity-based management system identifies activities, associates resources (expenditures) with those activities, and flows the cost of activities to cost objects.

With SAS Cost and Profitability Management, you can analyze business trends, and you can make the results of your analyses available to business professionals throughout your organization. You do not need to know how to program or how to use database tools.

Parcel Express Tutorial Model

You can import an already-completed model for the Parcel Express Tutorial by doing the following:

1. Select **File** ⇨ **Import** ⇨ **Model Data**.
2. Select **XML or ZIP File** as the type of data you want to import, and click **Next**.
3. Browse to, and select the following file: **`<install directory>\x86\SASCostandProfitabilityManagementClient\8.3\Samples\Models\Native\ParcelExpressTutorial.xml`**
4. Name the tutorial model. You can name it anything you want and provide a reference value for it. Click **Next**, review your choices, then click **Finish**.

As soon as the import operation summary appears, the tutorial model is imported. You must calculate the model to view calculated data.

The General Work Flow

Overview

By storing model data in a structure that is optimized for analytical purposes, SAS Cost and Profitability Management provides fast and intuitive analysis of broad trends and relationships. SAS Cost and Profitability Management is designed specifically for analyzing data. The following design features enable SAS Cost and Profitability Management to respond rapidly to complex queries that involve large amounts of data:

- Data is stored in a multidimensional data structure.
- Some of the summary calculations are performed before you request them.

Enter Business Data

1. Plan periods and scenarios. Then, set up periods and set up scenarios for each SAS Cost and Profitability Management server.

   On each server, all the periods and scenarios are shared across all models. Your organization should determine the period hierarchy that works best for all models. Then, set up periods and scenarios.
2. Create a model or import model data.
3. Calculate costs.
4. Generate OLAP cubes.

Analyze Business Data

- Use OLAP cubes to analyze business data.

Make Business Analyses Available to Others

- Set up and save column layouts and OLAP views.
- (Optional) Export OLAP views.

Model Availability

Sometimes, you might not be able to perform certain tasks in a model. For example, if you attempt to edit a part of the model that someone else is editing, you will not be able to edit.

Other tasks affect model availability, regardless of what type of user you are. These tasks require that the model data be in a stable state for some amount of time. You do not want someone to change the costs of accounts while a cube is being generated. Likewise, when SAS Cost and Profitability Management is calculating a model's costs, you do not want someone to change an account's cost. Allowing such a change would cause the final calculated costs to be wrong.

SAS Cost and Profitability Management prevents changes from being made during the following tasks:

- calculating costs
- generating cubes
- exporting data

Model Size and Performance

The number of accounts, assignments, dimensions, and dimension members within a model determine how long the model is unavailable, as well as how long it takes to perform a task. In a model that has few of these items (a small model), tasks complete faster than in a model that has numerous items (a large model). Model complexity can also affect such tasks as calculation and cube generation. For example, models that make extensive use of driver rules or have very large systems of reciprocal assignments can take longer to calculate. Cube generation for models containing very long assignment paths can take longer than for simpler models.

Model size affects performance when you interact with SAS Cost and Profitability Management (such as when you expand a rollup account in the Resource module, the Activity module, or the Cost Object module, when you copy period/scenario association data, and when you create an account)..

Saving and Refreshing Data

Saving

SAS Cost and Profitability Management enables multiple users to interact with a model that is located on a server. If you are unfamiliar with such an application, there are several differences from a desktop application.

- Unlike a desktop application in which you must explicitly save data, data is saved automatically.

  Automatic saving enables all users to see the most current data. (The availability of data is controlled by your user type and the status of the period/scenario association.)

  Unlike a desktop application in which your changes can be canceled, changes cannot be canceled.

  Because changes are saved for multiple users, your individual changes are immediately merged with other users' changes and cannot be separated. You can cancel a dialog box or wizard, but once you click **OK**, your changes are saved in the model.

- Unlike a desktop application, which stores a copy of your data in memory so that you can undo your changes, SAS Cost and Profitability Management writes directly to the model.

  This enables all users to see the most current data immediately.

  You cannot undo your changes except by re-entering previous values. However, to preserve the state of the model before you make changes, you can export the model. You can then revert to the previous version by importing the model.

Refreshing

As you work in SAS Cost and Profitability Management, some information is updated automatically. However, it might not be updated immediately. You must click the button to make the updated information available to you.

Note: You might not see the new information. It could be on another view or in part of a hierarchy that is collapsed.

Users can see changes to global items, such as exchange rates and period/scenario associations, only after they restart SAS Cost and Profitability Management.

Sessions and Tasks

Multiple Sessions

You can create multiple sessions of SAS Cost and Profitability Management at the same time: that is, you can launch several instances of the SAS Cost and Profitability Management client application at one time and use them as though they are different applications. (The client application sessions can even talk to different SAS Cost and

Profitability Management servers.) Multiple sessions enable you to easily compare multiple models or different areas of the same model.

Canceling a Server Task

You can cancel a server task that you initiate. However, canceling a task that makes changes to model data (such as an import) might leave the model in an unusable state.

You might need to cancel a server task when you realize that the task will require too much time or when you start a server task by mistake.

Note: A system administrator can cancel any task, regardless of who started it. If a system administrator cancels a task that you started, a message will notify you.

See Also

"Manage Tasks Dialog Box" on page 65

See Also

- "Manage Tasks Dialog Box" on page 65
- "Operation Summaries Dialog Box" on page 66

Shortcut Keys

Navigation tasks

Shortcut key	Action performed
Alt+Left Arrow	Move backward
Alt+Right Arrow	Move forward
Numeric keypad plus (+)	Expand the hierarchy
Numeric keypad minus (-)	Contract the hierarchy

Basic editing tasks

Shortcut key	Action performed
Ctrl+X	Cut selected text to the Windows clipboard
Ctrl+C	Copy selected text to the Windows clipboard

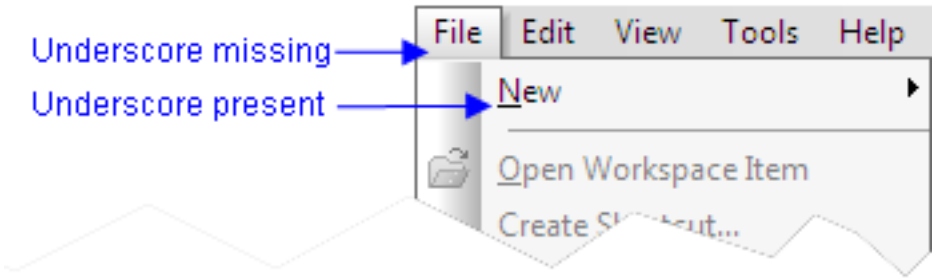
Shortcut key	Action performed
Ctrl+V	Paste text from the Windows clipboard
Ctrl+Q	Create a new folder in Workspace Manager or on the Attributes view
F5	Refresh the information

Modeling tasks

Shortcut key	Action performed
Ctrl+A	Create a new item (other than an entered cost element)
Ctrl+O	Create a new entered cost element
Ctrl+E	Manage the attributes that have been added to an account
Ctrl+G	Go to the selected account
Alt+Enter	Show the item's properties
F7	Create assignments between the selected account and all of the accounts in the left and right assignments panes
Shift+F7	Delete all incoming assignments to the selected account and all outgoing assignments from the selected account. *Note:* It is not necessary that either the incoming accounts or the outgoing accounts be visible in an Assignments pane for the assignments to be deleted. In fact, it is not even necessary that an Assignments pane be displayed.

Notes:

- The JAWS screen-reading program occasionally locks input fields of HTML pages, preventing you from entering data. If this happens while you are using JAWS, press Alt+N to re-enable screen input.
- The JAWS screen-reading program assigns the label graphics plus a random number to unlabeled graphics. You can assign you own label using the JAWS Graphics Labeler.
- Due to a Windows bug, the underscore is not displayed for top-level menu items. For example, in the following picture the underscore is missing for File. However, the ALT key still works, even for File.



Using the JAWS Graphics Labeler

If a graphic in SAS Cost and Profitability Management has not been assigned a label, JAWS assigns it the label graphic followed by a random number. You can use the graphics labeler to assign such graphics a more meaningful label. You can either label a graphic manually or you can use the auto-labeler.

To label a graphic manually:

1. Find the graphic that you want to label.
2. Press Insert + g to open the graphics labeler.
3. Type your label in the edit field.
4. Tab to the next edit field to enter the graphic label for your Braille display.

   When you press Tab again, you see three radio buttons that indicate where you want to save your labeled graphic. Choose where you want to save it, and then press Enter to activate the label.

To label a graphic manually, make sure that **All Graphics** is selected in the JAWS verbosity settings.

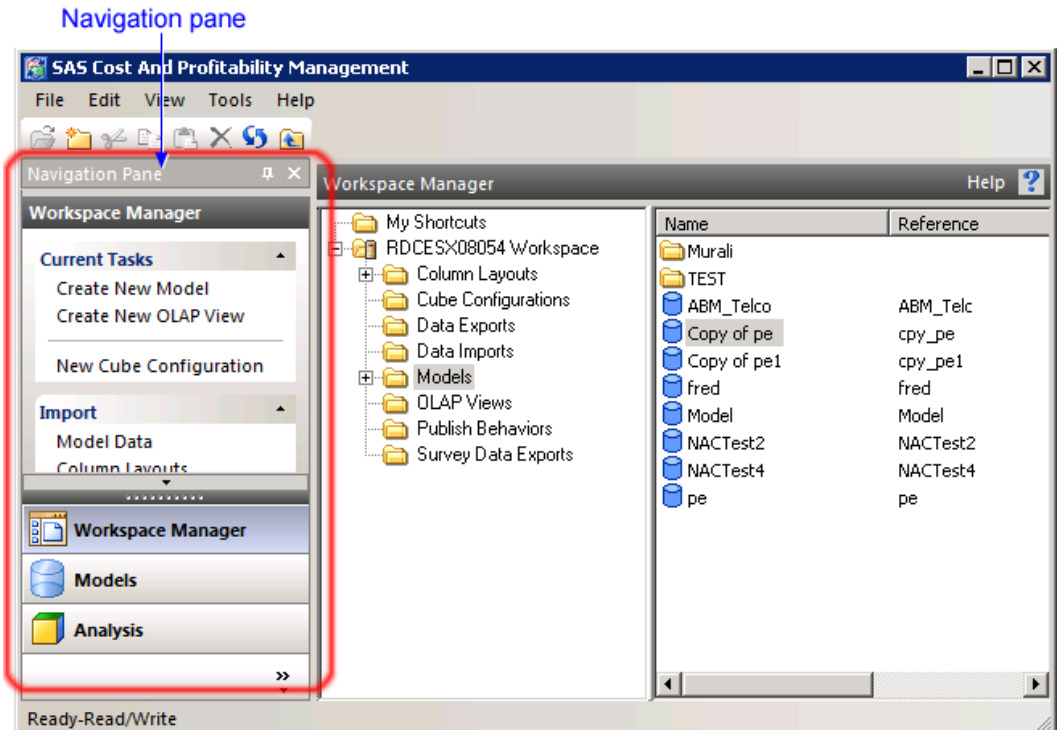
Chapter 4

Navigation Pane

What Is the Navigation Pane?

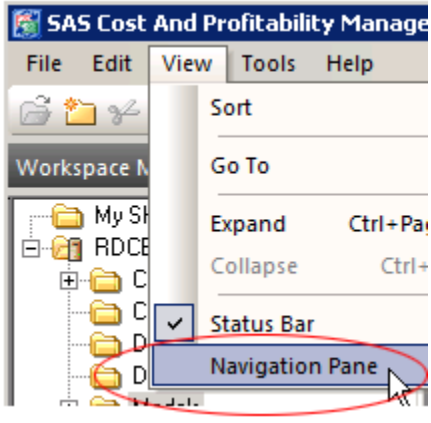
The Navigation Pane allows you to navigate among the workspaces that constitute SAS Cost and Profitability Management:

- Workspace Manager See "Workspace Manager" on page 53.
- Models See "Models Workspace" on page 92.
- Analysis See "Analysis Workspace" on page 583.



How to Access the Navigation Pane

If the Navigation Pane is not visible, then select **View** ⇨ **Navigation Pane**.

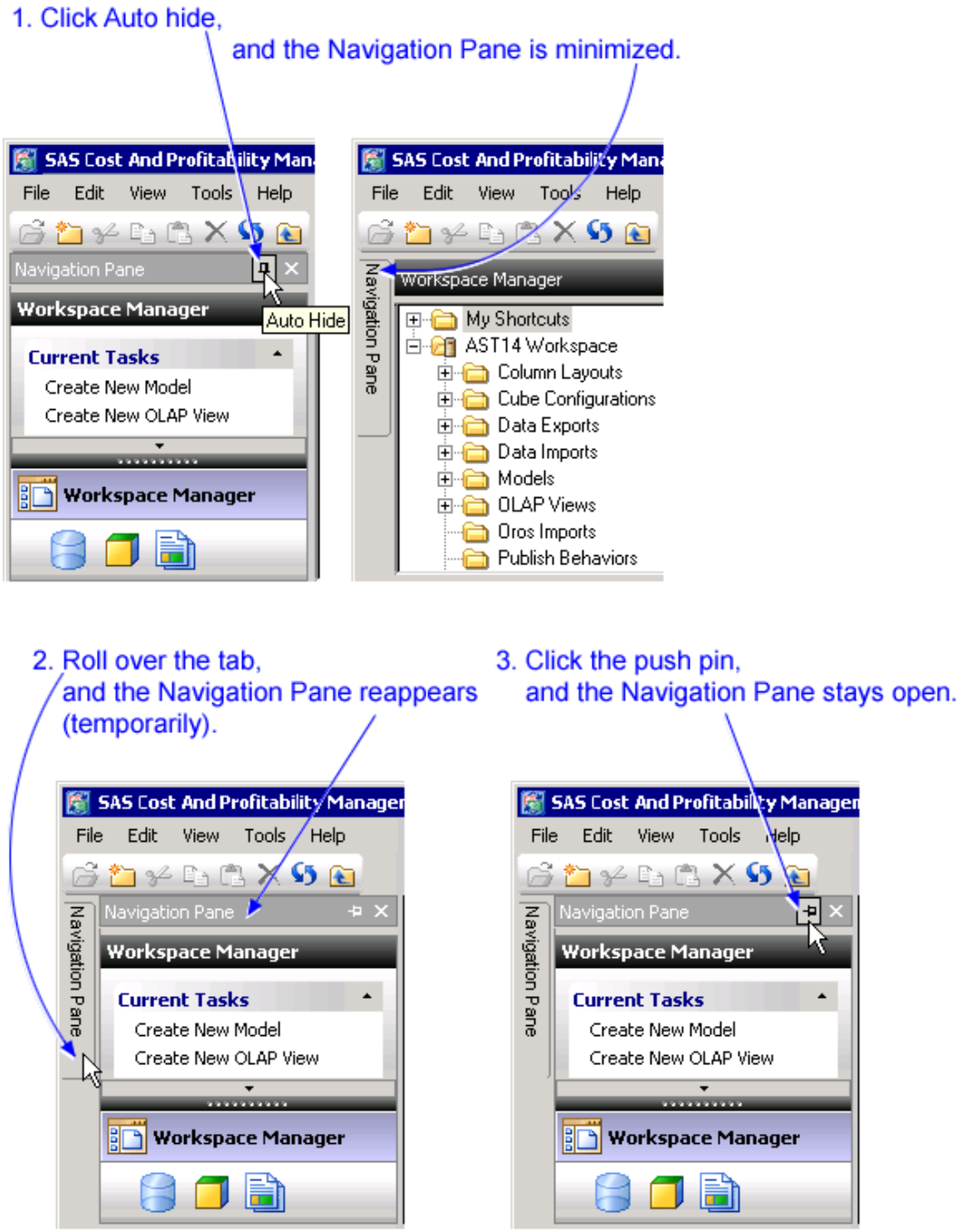


Minimize the Navigation Pane

1. Click the Auto hide icon (the push pin) to minimize the Navigation Pane.

   When you roll over the minimized tab, the Navigation Pane returns but only temporarily.

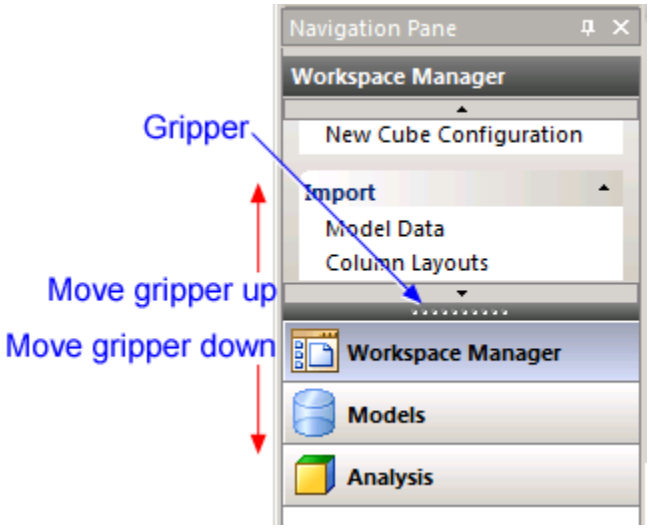
2. Click the Auto Hide icon again to keep the Navigation Pane open (or select **View** ⇨ **Navigation Pane**).



Partition the Navigation Pane

Move the gripper up or down to change the relative size of the task area and the button area in the Navigation Pane.

Note: You cannot move the gripper up further than is necessary to fully display all the buttons in the button area.



Chapter 5

Workspace Manager

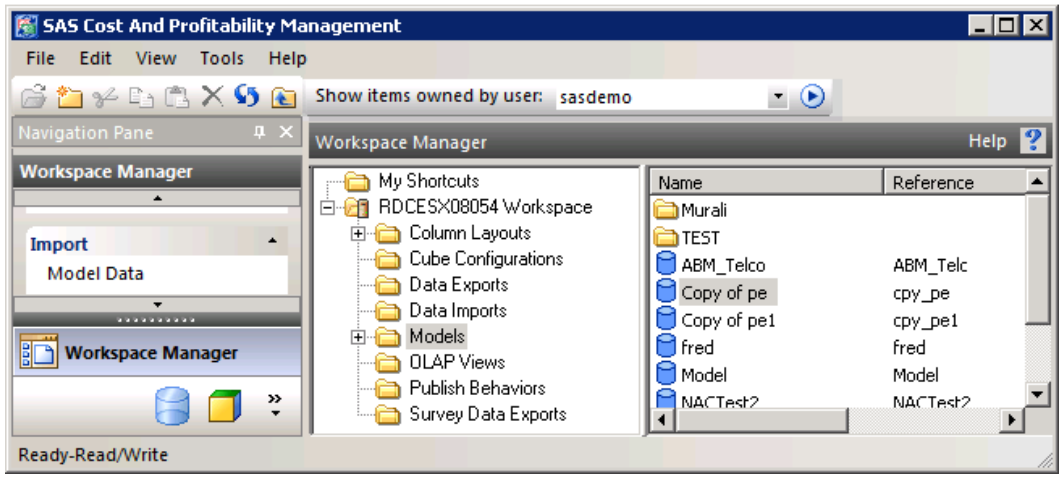
Workspace Manager

Overview

Use the Workspace Manager to start frequently used tasks. You can perform tasks from the Workspace Manager without first opening a model.

The Workspace Manager provides a treeview of all SAS Cost and Profitability Management elements—from models to cube configurations—and gives you access to them all.

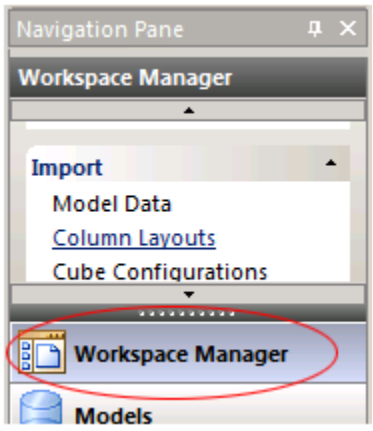
Note: The availability of these features depends on your permissions.



Note: The **Show items owned by user** drop-down list is available only to administrators.

How to Access the Workspace Manager

1. If the Navigation Pane is not visible, click **View** ⇨ **Navigation Pane**.
2. Click **Workspace Manager**.



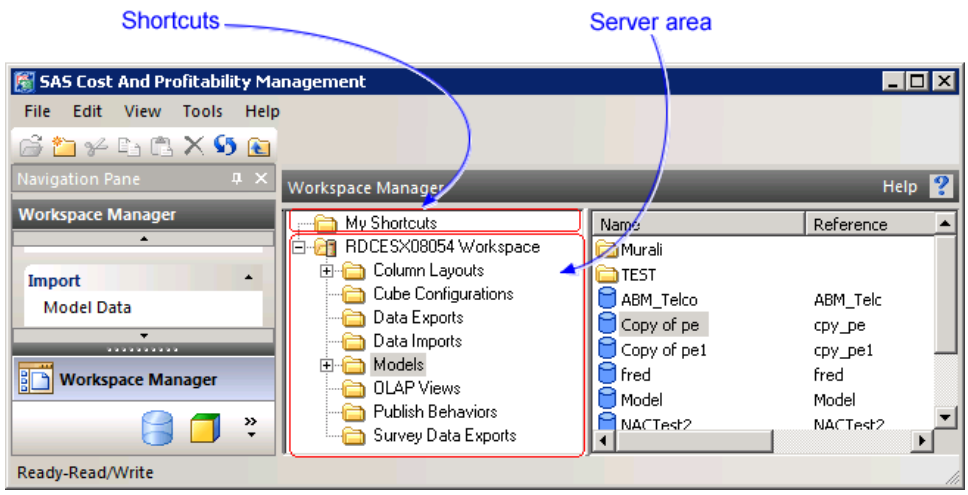
Workspace Manager Layout

About the Workspace Manager Layout

Workspace Manager looks similar to Windows Explorer, and most of the commands and techniques that you use in Windows Explorer, including drag-and-drop, function identically in Workspace Manager.

Workspace Manager displays information in two panes. When you select an item in the left pane, the contents of the selected item are shown in the right pane.

The left pane has two main parts: the server area and My Shortcuts.



Server Area

The server area shows you all the items on a SAS Cost and Profitability Management server. The name of the workspace area is the name of the server on which the items are stored, followed by the word Workspace. For example, if the server is named ABCdata, the server area is named ABCdata Workspace. You cannot change this name.

In the server area, you see only those items for which you have permission. Also, you see the folders that have been created by all users, although you do not necessarily see the contents of each folder.

If you are a SAS Cost and Profitability Manager administrator, you see every item, regardless of who owns the item, and you can interact with every item (though you might not have permission to view the data contents of some items).

Creating and Deleting Items in the Server Area

You cannot rename or delete the top-level folders in the server area. However, if you have the necessary permissions, you can create and delete subfolders within these top-level folders.

When you delete a model, workspace items that depend on that model are not deleted. As a result, when you open a workspace item, you might see an error message about the missing model.

Note: When you see a workspace item without an associated model, you can delete the item and then create an identical item that is based on another model. Alternatively, you can edit the item to base it on another model. You cannot change the model that is associated with an item. To avoid the error message, delete the workspace item, and then create an identical item that is based on another model.

My Shortcuts

My Shortcuts enables you to organize the items that you need for your work. You can create shortcuts to items that are in the server area. Then, use a shortcut to open and use an item.

Note: Even when the item that a shortcut refers to is renamed or moved, or when the folder that contains the item is renamed, the shortcut still works.

My Shortcuts holds only folders and shortcuts. Each user of SAS Cost and Profitability Management has a different shortcut area, so you see only the folders and shortcuts that you create. You can use these folders and shortcuts from any computer that is connected to the server on which you create the folders and shortcuts.

Items in the server area are arranged by type. By contrast, in My Shortcuts, you can arrange folders by task and project, and you can create shortcuts to many different types of items in those folders.

Ownership and Permissions for Server Area Workspace Items

A workspace stores the items, such as column layouts and models, that are created in SAS Cost and Profitability Management. A workspace is shared by all users on the same server, and it enables your organization to define standard items once and to apply them to different models as needed.

When you create a server area item, SAS Cost and Profitability Management assigns ownership to you. You can modify the item, rename it, or move it.

To each group established with SAS Management Console, you can assign the permission to read or to read and write to any item that you own. If you do not assign permissions to a group, users who are members of that group cannot see the item.

Create a shortcut to a workspace item

1. In Workspace Manager, select an item in the server area.
2. Select **File** ⇨ **Create Shortcut**.

   The Create Shortcut dialog box appears.

Review or change the properties of a workspace item

1. In Workspace Manager, select an item.
2. Select **Edit** ⇨ **Item Properties**.

   The Item Properties dialog box appears.

Create a folder

1. In Workspace Manager, select a folder under which to create the new folder.
2. Select **Edit** ⇨ **New Folder**.

   The New Folder dialog box appears.

Refresh Workspace Manager

Select **View** ⇨ **Refresh**.

Show items that are owned by a specific user

Note: This feature is available only to administrators.

From the **Show items by user** menu, select a user.

See Also

- "Workspace Manager" on page 53
- Chapter 6, "User Capabilities and Groups," on page 61

Item Properties Dialog Box

About the Item Properties Dialog Box

In the Item Properties dialog box, you can review or change information about a workspace item and you can set permissions for the workspace item.

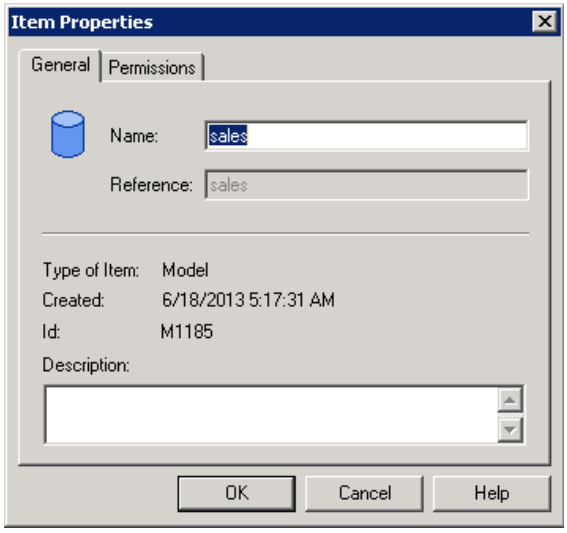
Note: The availability of these features depends on your permissions.

How to Access the Item Properties Dialog Box

In Workspace Manager, select an item and select **Edit** ⇨ **Item Properties**. The Item Properties dialog box appears.

Specify Information

1. Click the **General** tab.



2. Type the Name.

   The name must follow the naming guidelines. See "Naming Conventions" on page 69.

   TIP When you rename an item, notify other users. Otherwise, they might look for an item name that no longer exists.

3. Type the Description.

Set Permissions

Note: Permissions are not applicable to shortcuts.

1. Click the **Permissions** tab.
2. Select an Owner.
3. In the **Privileges** list, select or clear the check boxes that are next to each listed group.

Note: To assign permissions to a group, you must be a member of that group or you must obtain administrative capabilities.

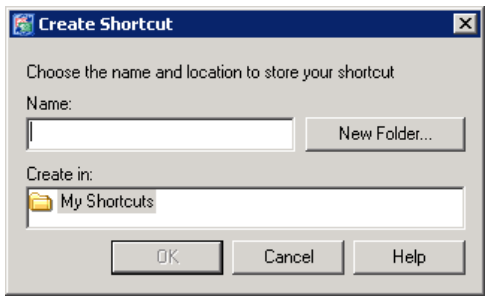
See Also

- Chapter 6, "User Capabilities and Groups," on page 61
- Chapter 1, "User Capabilities and Groups," in *SAS Cost and Profitability Management: Data Administration Guide*

Create Shortcut Dialog Box

About the Create Shortcut Dialog Box

In the Create Shortcut dialog box, you can name a shortcut and select its location in My Shortcuts.



How to Access the Create Shortcut Dialog Box

In Workspace Manager, select a workspace item in the server area and select **File** ⇨ **Create Shortcut**. The Create Shortcut dialog box appears.

Specify information

1. Type the Name of the shortcut.

   The name must follow the naming guidelines.
2. In the **Create in** list, select the folder in which you want to place the shortcut.

3. To create a new folder in which to place the shortcut, click **New Folder**.

   The New Folder dialog box appears.

TIP You can create a shortcut by dragging an object from the server area into My Shortcuts.

New Folder Dialog Box

About the New Folder Dialog Box

In the New Folder dialog box, you can name a new folder in Workspace Manager.

How to Access the New Folder Dialog Box

Do one of the following:

- In Workspace Manager, select **Edit** ⇨ **New Folder**.
- In the Create Shortcut dialog box, click **New Folder**.

Specify Information

Type the folder Name.

Chapter 6

User Capabilities and Groups

Overview

The abilities that you possess as a SAS Cost and Profitability Management user depend on:

- the capabilities that you inherit from the groups to which you belong. See "Capabilities" on page 61.
- the permissions that are granted to the groups to which you belong.

  The owner of an item, such as a model, can grant Read or Read/Write access to the model to all the members of a group. See "Giving Read or Read/Write Permission to Members of a Group" on page 62.

For complete information on users, groups, capabilities, and roles, see Chapter 1, "User Capabilities and Groups," in *SAS Cost and Profitability Management: Data Administration Guide*. Also see Chapter 2, "How to Manage Permissions," in *SAS Cost and Profitability Management: Data Administration Guide*.

Capabilities

When a SAS Cost and Profitability Management administrator created your SAS Cost and Profitability Management account, you inherited one or more of the following capabilities by virtue of belonging to a group with those capabilities.

Note: A group has capabilities because of the roles that the group is a member of.

- Create Models
- View Models
- Create Cubes
- View Cubes
- Audit
- Administer
- Create Surveys
- Take Surveys
- Create What-If Analysis
- View What-If Analysis

Quite likely, most users will have inherited either Create Models capability or View Models capability. The Create Models capability provides the abilities of a Modeler in previous releases of SAS Cost and Profitability Management. It 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.

Similarly the View Models capability provides the abilities of a Business User in previous releases of SAS Cost and Profitability Management.

If you do not know what capabilities you have inherited, ask your SAS Cost and Profitability Management administrator.

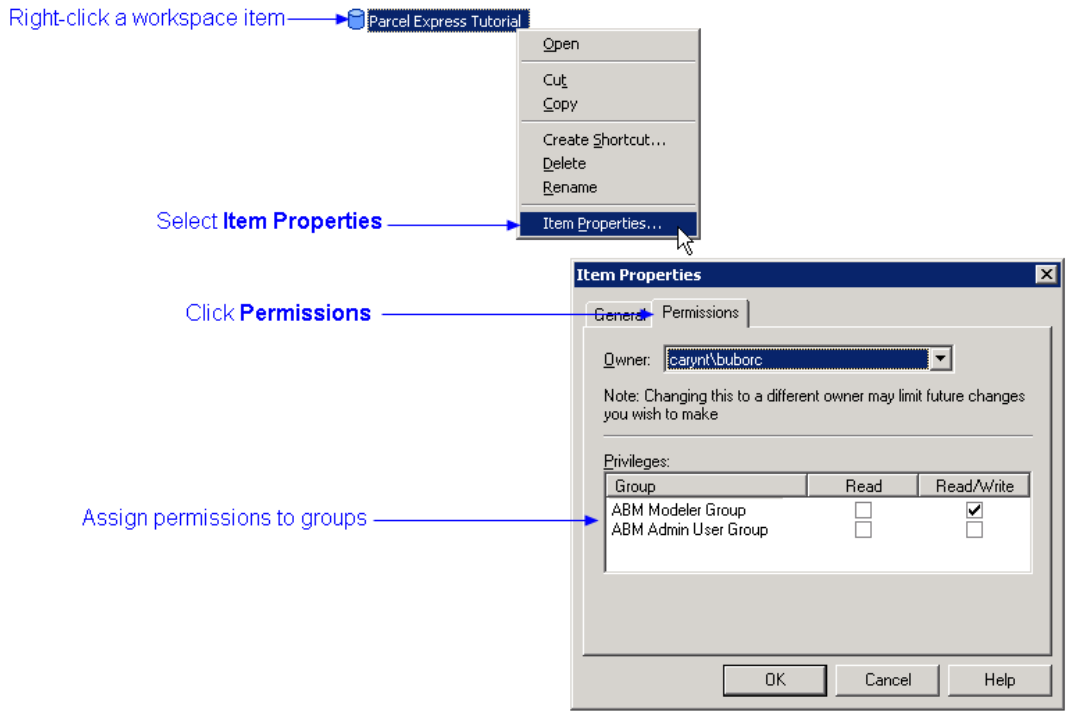
Group Permissions

The administrator creates groups for your organization and assigns you to one or more groups. Your abilities depend on the Read and Read/Write permissions granted to the groups to which you belong.

Giving Read or Read/Write Permission to Members of a Group

The creator of a model or other workspace item is, by default, its owner. Of course the owner has Read/Write access to the item. The owner can grant Read or Read/Write access to the item to users in a group. If you are a member of a group that has been granted access to a model or other workspace item, then you have that access as a member of the group.

The owner of a model or other item (or an administrator) grants access to the item to members of a group by selecting the item in the Workspace Manager and selecting **Edit** ⇨ **Item Properties**. Then, by checking either **Read** or **Read/Write**, the owner grants that access to members of the group.

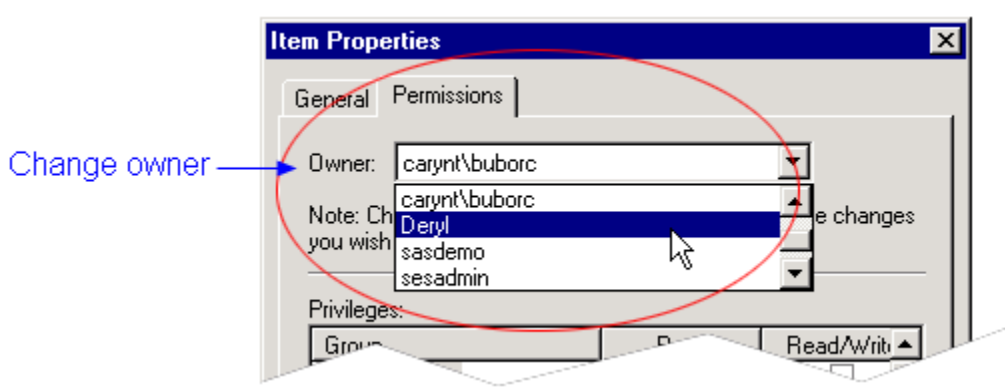


Note: Only the owner of an item (or an administrator) can grant access to that item. Non-administrator members of a group that have been granted Read/Write access to an item cannot grant access to other groups.

If you do not know to what groups you belong, ask your SAS Cost and Profitability Management administrator.

Changing Owners

The creator of a model or other workspace item is, by default, its owner. The owner (or an administrator) can transfer ownership to another user. To transfer ownership, the owner (or administrator) selects the item in the Workspace Manager and selects **Edit** ⇨ **Item Properties**. Then the owner (or administrator) selects a new owner from the drop-down list of users.



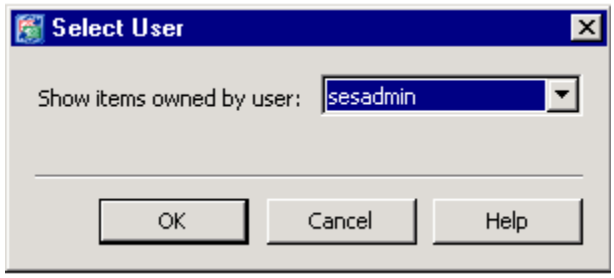
Note: The drop-down list of potential owners includes those users who have Create capability for the item. For example, you can transfer owership of a model only to those users who have Create Models capability.

Select User Dialog Box

About the Select User Dialog Box

Use this dialog box to display items owned by a particular user.

Note: This dialog box is available only to users with Administrator capability.



How to Access the Select User Dialog Box

Select **Tools** ⇨ **Select User.**

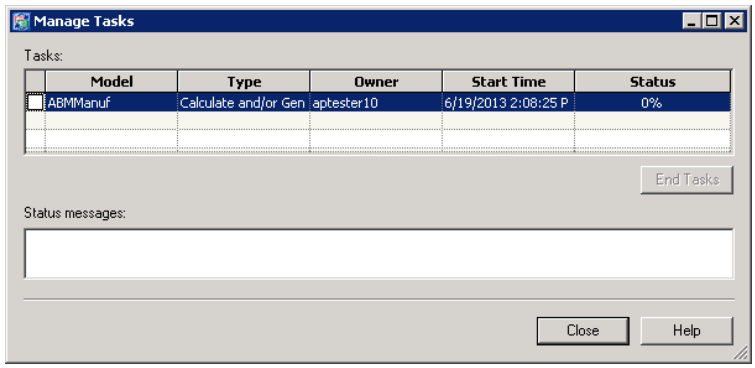
Chapter 7

General Application Tools

Manage Tasks Dialog Box

About the Manage Tasks Dialog Box

In the Manage Tasks dialog box, you can monitor the status of tasks that are being performed on a SAS Cost and Profitability Management server.



Note: You can perform the following tasks without first opening a model.

How to Access the Manage Tasks Dialog Box

Do one of the following:

- Click **Tasks** in the status bar.

  The **Tasks** button is visible only during an operation that you can cancel, and only when the status bar is not hidden.

- Select **Tools** ⇨ **Manage Tasks**.

Cancel a Task

1. From the list of **Tasks**, select a task.

   You can select more than one task. A check mark is displayed next to the selected tasks.

2. Click **End Tasks**.

   There might be a delay before the operation is canceled.

Operation Summaries Dialog Box

About the Operation Summaries Dialog Box

In the Operation Summaries dialog box, you can see a list of all of the operations that you have performed on a SAS Cost and Profitability Management server. For example, an operation entry is added to this list whenever you calculate costs or generate a cube.

Note: Except for administrators, the Operation Summaries dialog box displays summaries for a specific user account in a specific domain, such as LOCALMACHINE/Alex. If Alex also has an account in the domain COMPANY, and if he logs in as COMPANY/Alex, he will not see the summaries for the account LOCALMACHINE/Alex.

Note: You cannot change this information.

How to Access the Operation Summaries Dialog Box

Select **Tools** ⇨ **Operation Summaries**.

Delete a Summary

1. Select the check box next to a summary.
2. Click **Delete**.

View Details about a Server Operation

1. Select the check box next to a summary.
2. Click **View**.

   The Operation Summary window appears.
3. Click **Show Details**.

Export the Operation Summaries

1. Click **Export**.

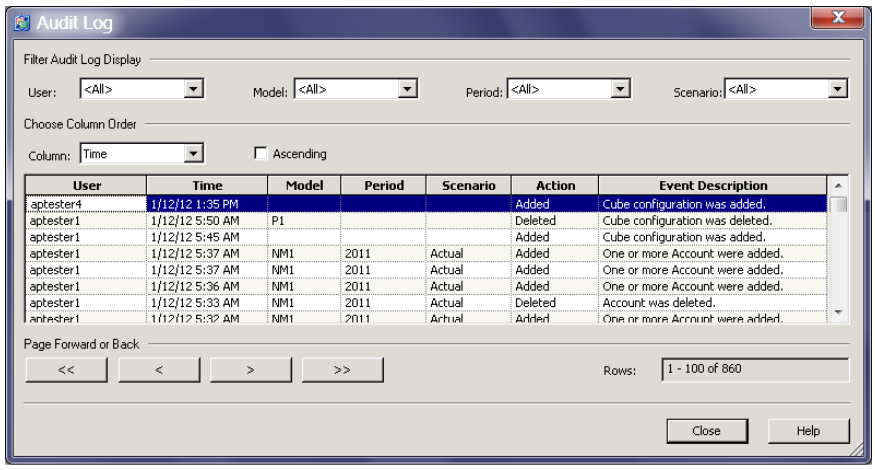
   The Save As dialog box appears.
2. Navigate to a location.
3. Type the File name.

Audit Log Window

About the Audit Log Window

The availability of this feature depends on your permissions.

In the Audit Log window, you can see a history of the actions that have been performed on a SAS Cost and Profitability Management server. The history includes information such as what action was performed, the user who performed the action, and the item that was acted upon.



How to Access the Audit Log Window

Select **Tools** ⇨ **Audit Log**.

Sort the Information

Under **Choose column order**, select a column to sort on from the **Column** drop-down list. Then select **Ascending** to sort in ascending order on the contents of the selected column. Or, deselect **Ascending** to sort in descending order on the contents of that column.

Limit the Information

From the **User**, **Name**, **Period**, and **Scenario** menus, select values to use for limiting the information.

Chapter 8

Naming Conventions

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.

For additional considerations concerning stage attributes, see "Stage Attributes" on page 275.

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.

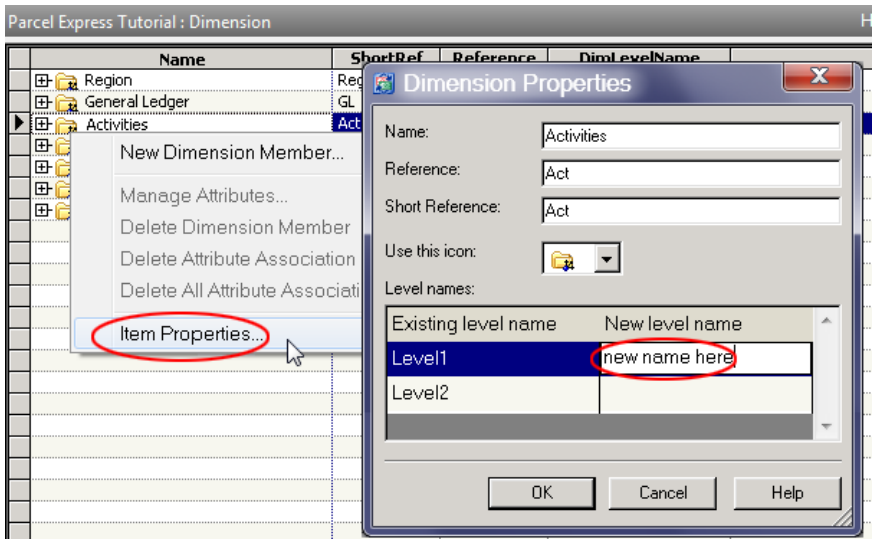
- Note: Because of the mechanism used by SAS Cost and Profitability Management to store dimension level names, some user-specified names will cause conflicts with the underlying database (regardless of whether Microsoft SQL Server or Oracle is used). These conflicts will appear as obscure error messages 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 256 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 256 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.

For additional considerations concerning stage attributes, see “Stage Attributes” on page 275.

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.

Part 3

Models

Chapter 9

Model Concepts

Elements of a Model

Periodic and Structural Elements

A *model* is the basic container for information in SAS Cost and Profitability Management. A meaningful model reflects the organization that it is modeling and uses terms that are familiar to the people who work at the organization.

A model contains two types of elements:

Periodic elements

Periodic elements can exist in one period/scenario and not in another. Periodic elements are stored separately for each period/scenario in a model. The following elements are periodic:

- Accounts
- Entered Cost Elements
- Assignments
- Attributes
- Currency Rates

Structural elements

Structural elements are independent of any period/scenario. A structural element exists in every period/scenario in a model. The following elements are structural:

- Modules
- Dimensions
- Dimension Members
- Drivers

Periods and Scenarios

You define periods and scenarios independently of any model. Any period and scenario can be shared by all the models on a server.

A *period* is any range of dates such as a year, month, week, or quarter. You can define a hierarchy of periods such as months within quarters within years.

A *scenario* is a label that you associate with a period so that you can distinguish the data for that period with that label from data for the same period with a different label. Two common labels are “Actual” and “Budget”. Using those labels (scenarios) allows you to distinguish actual costs and budgeted costs in a period. You can also define a hierarchy of scenarios such as Budget–>WhatIf1, Budget–>WhatIf2.

See “Step 2: Specify Initial Period and Scenario” on page 105.

Period and Scenario Associations

A *period and scenario association* (also referred to as a period/scenario association, or just period/scenario) is an ordered pair consisting of one period and one scenario. You associate a period with a scenario when you specify the initial period and scenario for a model. Until you create additional period/scenario associations, all periodic model data belongs to the initial period/scenario association.

See “Create a Period/Scenario Association” on page 169.

Modules

A *module* contains a set of accounts. The set of modules forms an ordered set of sets of accounts. The set of modules is an ordered set because it determines the direction of cost flow in a model:

- An account in one module can flow costs to any other account (including itself) in the same module.

- An account in one module can flow costs to any other account in any module after its own module in the direction determined by the module order.

Modules are structural elements. The modules that you define for a model exist in every period/scenario in the model.

See Chapter 21, "Working with Modules," on page 179.

Dimensions

Dimensions are the top-level categories that lay out the types of accounts that exist in your model. Examples of dimensions are: Region, Channel, Product, Customer, Resource, General Ledger, Materials, Department, and Activity.

Dimensions are structural elements. The dimensions that you define for a model exist in every period/scenario in the model.

See Chapter 24, "What Are Dimensions?," on page 195.

Dimension Members

Dimension members form a tree structure under their respective dimensions and further refine the types of accounts that exist in your model. For example, the tree structure under the region dimension might be:

- Americas
  - US
    - New York
    - Washington, D.C.
    - Cary
  - Canada
  - Brazil
- Europe
  - Germany
  - England
  - France
- Asia
  - China
  - Korea
  - Japan

Dimension members are structural elements. The dimension members that you define for a model exist in every period/scenario in the model.

See "Dimension Members" on page 196.

Accounts

An *account* is a point where costs accumulate in a model and is defined by an intersection of dimension members—one dimension member from each dimension in a

module. For example, if a model contains the dimensions Region, Activity, and Product, one account might be the following:

```
London x Shipping x Hat
```

Where London is a dimension member of Region, Shipping is a dimension member of Activity, and Hat is a dimension member of Product.

Accounts are periodic elements. The accounts in one period/scenario may exist in another period/scenario. And, when an account (same dimension signature) exists in multiple period/scenarios, its cost in one period/scenario is independent of its cost in another period/scenario.

See Chapter 27, "What are Accounts?," on page 223.

Cost Elements

The cost in an account is the sum of its entered cost elements and its received cost elements:

Entered Cost Element

An *entered cost element* is one that you create and to which you assign costs directly by giving it one or both of:

Entered Cost	A lump sum that you give to the cost element
Entered Unit Cost	If you specify Entered Unit Cost, then the cost for the cost element is determined by multiplying that unit cost times the number of units of the parent account that are assigned to destination accounts.

Note: All of the cost in a model originates with entered cost elements.

Received Cost Element

A *received cost element* is one that you create indirectly by creating an assignment from one account to another. The cost that the destination account receives from that source account is a received cost element.

Cost elements are periodic elements. The cost elements in one period/scenario may exist in another period/scenario. And, when a cost element (with the same name and reference in the same module) exists in multiple periods/scenarios, its cost in one period/scenario is independent of its cost in another period/scenario.

See Chapter 30, "What Are Cost Elements?," on page 259.

Attributes

Attributes are items that you define to associate categories or values to accounts. Attributes are either:

Value attributes

A *value attribute* has a value that is true of an account. Value attributes can be of the following kind:

- Numeric
- Text

Tag attributes

A *tag attribute* (also known as Boolean) does not have a value. You can apply a tag attribute to one or more accounts, and you can use a tag attribute in a formula to identify accounts that have that tag attribute.

Dimension attributes

Dimension attributes are like ordinary (structural) dimensions because they contain members (dimension member attributes) that you can use to group or categorize the types of accounts that exist in your model. For example, the dimension attribute FixedVariable can contain the dimension member attributes Fixed and Variable. By applying either the dimension member attribute Fixed or the dimension member attribute Variable to accounts, you can distinguish fixed costs from variable costs in your model. Stage attributes are a special case of dimension attributes used for a specific purpose.

Note: Like numeric attributes, dimension attributes can appear in the OLAP cubes that are generated from your model. Text and tag attributes do not appear in OLAP cubes.

Attributes are periodic elements. An attribute that is attached to an account in one period/scenario may or may not be attached to the same account in another period/scenario. And, when a value attribute is attached to the same account in multiple periods/scenarios, its value in one period/scenario is independent of its value in another period/scenario.

See Chapter 33, "Types of Attributes," on page 271.

Properties

Properties have values that are true of the elements of a model. Properties are either:

System-defined

System-defined properties are those that are available for every model. You provide either the values for such properties or the data that is used to calculate the values. For example, Cost is a system-defined property of an account. You might assign one account $100 of cost, whereas you might assign another account $0 of cost.

All model elements have system-defined properties.

System-defined properties are structural elements. They exist in every period/scenario in a model. However, the value of a system-defined property may be either structural or periodic depending on the particular property. For example, DriverName, which is a system-defined property, has the same value for a particular driver in every period/scenario, whereas DrivenQuantity, which is also a system-defined property of drivers, can have different values for the same driver in different period/scenarios.

See:

- Chapter 90, "Alphabetic List of Properties," on page 745
- Chapter 91, "Property Relationship Diagrams," on page 803
- Chapter 60, "Properties in Calculation," on page 527

Drivers

A *driver* distributes costs from a source account to one or more destination accounts. There are four types of drivers:

Evenly assigned

Distributes the cost evenly to each of the destination accounts.

Percentage

Distributes the cost by percentages that you specify to each of the destination accounts.

Standard
: Uses a system-defined formula for distributing the cost to each of the destination accounts. The formula is system-defined, but you determine the values of the properties used in the formula.

Calculated
: Uses a user-defined formula for distributing the cost to each of the destination accounts. You define the formula, and you determine the values of the properties and attributes in the formula.

Drivers are structural elements. They exist in every period/scenario in a model. And, if a driver has a formula or a rule formula, then those formulas are the same in every period/scenario. However, the evaluation of a formula can result in different values in different period/scenarios. And, a particular driver that is attached to an account in one period/scenario may not be attached to the same account in a different period/scenario.

See Chapter 40, “Introducing Drivers,” on page 355.

Assignments

An *assignment* is a directional connection or association between a single source account and one or more destination accounts. It defines the distribution of costs between accounts using the source account’s driver. The amount of cost that flows is controlled by the driver. A single source account can make assignments to any number of destination accounts, but each source account can have only one driver.

Assignments are periodic elements. An assignment between two accounts in one period/scenario may or may not exist between the same accounts in another period/scenario. And, when an assignment exists between the same accounts in multiple periods/scenarios, its values in one period/scenario are independent of its values in another period/scenario.

See “What Is an Assignment?” on page 431.

Column Layouts

A *column layout* is the set of properties and attributes that display when you view the modules in your model or when you view the dimension members in your model. You can customize column layouts so that the properties and attributes that are displayed are the ones that you want to see. You can create any number of column layouts.

Column layouts are structural elements. Once a column layout has been defined in association with a model and saved, it exists in every period/scenario in a model. However, a column layout can contain properties and attributes that are specific to a particular period/scenario. By default, the values displayed in a column layout for attributes and properties are their values for the current period/scenario. Column layouts created for one model can also be loaded into other models. If a column layout is displayed that contains properties and attributes that are specific to a particular period/scenario and there is no data for that period/scenario in the model, then those properties and attributes are displayed as blank.

See “What is a Column Layout?” on page 319.

Currencies and Exchange Rate Tables

When you create a model, you must specify its base currency which determines how costs are displayed. You can also create exchange rate tables for converting costs to other

currencies. An exchange rate table applies to a single period/scenario association in a model and can contain any number of currencies.

See Chapter 12, “Working with Currencies,” on page 129.

See Also

“Create a Model” on page 104

Guidelines for Creating the Model Structure

Anticipate the Needs of Users

Before you build a model, define the goals of the model. Each organization has different goals that affect how a model is designed. There is no single correct way to design a model.

Before you build the model, consider the following questions:

- What are the required outputs, such as cubes?

  The people in your organization might need to answer questions such as, What does this product or service cost? and Why does it cost that?

  The type and detail of analysis that users need to perform determines the model structure that you define.

- How well do you know your organization?

  Consider your organization's systems, products or services, and customers before starting to build a model.

- What types of information do the people in your organization need?

  Ensure that the model accommodates the needs and goals of the organization. Include only that information that supports the needs and goals.

- What level of detail is important to your organization?

  If you include too much detail, you might complicate the analysis without adding useful information. A detailed model requires more maintenance in the future. If you include too little detail, you might fail to reveal opportunities for improvement.

Building a model is an iterative process. Few people build a model with all of the necessary information the first time. A model often requires several iterations to determine the best balance between too much information and not enough information.

Communicate often with those people who will use the information provided by the model. Are you including the information that meets their needs? Is the amount of detail enough? Do these people need to analyze the data in ways you have not accommodated in the model?

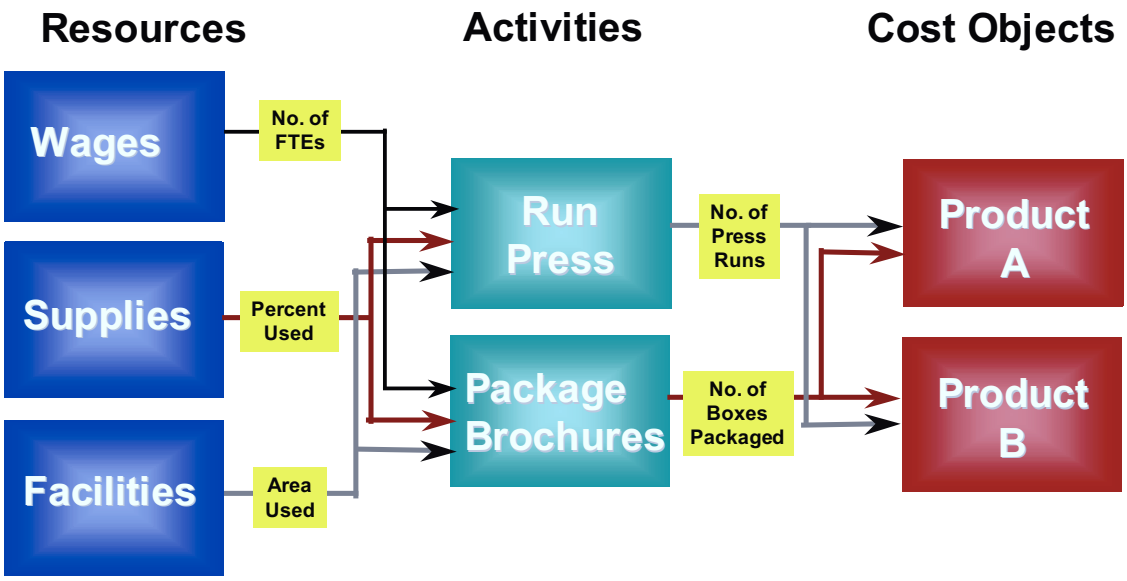
The best method to design a model is to create a paper model before attempting to build the model in SAS Cost and Profitability Management.

Create a Paper Model

A paper model helps you focus on the model structure before entering data into the model. Sketch the model structure on paper, and note the important levels in the

organizational hierarchy. Then, review the paper model with other people in your organization to determine whether you are approaching their needs correctly.

Your paper model might look something like the following:



After you determine that the paper model meets the needs of the people in your organization, you should start collecting the data that is needed by the model.

Collect Data

A review of the paper model will identify the data that you need to collect. The quality and the integrity of the model depend on identifying the data and collecting the data.

There are many different ways to collect data. Most organizations use a combination of techniques and methods, such as the following:

- conducting interviews with people
- distributing questionnaires to people
- analyzing historical records
- gathering a panel of experts or focus groups
- observing people and work flow

Guidelines for Designing a Model

Each organization has different goals that affect how a model is designed. There is no single correct way to design a model.

Here are some guidelines for creating a model that runs efficiently and that provides useful information:

- Define the goals.

  What are the goals and the scope?

- Decide what types of information are needed.

What types of information do managers need? Make sure that the model accommodates the needs and goals of the organization.

- Design the model structure before entering data into the model.

  Note the important levels in the organizational hierarchy.

- Decide how much detail to include.

  What level of detail is important to your organization? If you include too much detail, you might complicate the analysis without adding useful information. A detailed model requires more maintenance in the future. If you include too little detail, you might fail to reveal opportunities for improvement.

- Combine similar accounts.

  Are particular costs incurred together, or are multiple costs caused by the same factor? If so, you might want to combine such items. For example, if your organization's General Ledger lists the details of travel expenses, such as airfare, hotel, or entertainment, you might want to combine these accounts into one travel expense account. Also, if some accounts have small costs, you might want to combine the small accounts into accounts that represent more general categories.

- Group related accounts.

  Group accounts into rollup accounts (folders) if the accounts have similar functionality or if the accounts are linked to similar activities.

- Gather costs.

  Consider the relationship between the time period that you use for gathering costs and the time period that you use for gathering resource values and activity driver values.

- Collect driver data.

  When you collect driver data to assign, for example, activity costs to cost objects, ensure that the data is current, available, and accurate. Ask individuals in your organization to verify the data's accuracy.

- Minimize the use of unique driver quantities.

  Unique driver quantities can use a lot of memory. They increase your processing time, and they do not provide a calculation advantage.

Guidelines for Sharing Models, Configurations, and Data

Overview

After you create a model and analyze it, you will have data that helps you and others in your organization make business decisions. You will want to share your results so that co-workers can use the business data that you have prepared. However, because much data is stored on a single server, you should establish naming conventions and other guidelines so that all users can easily find the correct information.

Workspace Manager lists items that are available for all SAS Cost and Profitability Management users who are using the same server.

Saving Items

Establish guidelines for when to save and when not to save column layouts and OLAP views.

Naming Items

Overview

To manage the lists of items that are on a SAS Cost and Profitability Management server, establish a naming convention. Decide whether a name should include the owner's initials or a departmental or functional prefix or suffix. Names of items affect the sort order within folders in Workspace Manager.

Naming Import Configurations and Models

For model names and import configurations, you might want to include the following information:

- the goal for the model
- the users that can work with the model
- a short way of referring to the types of structures, dimensions, attributes, or measures that are selected when data is imported
- the period/scenario associations
- the date on which the model was imported

Naming Column Layouts

For column layout names, you might want to include the following information:

- the users or teams who use the column layout
- the models or types of models that use the column layout
- the validation or purpose of the layout

Naming OLAP Views

By default, each description on the Analysis workspace contains the following information:

- the description that was entered when the OLAP view was saved
- the date on which the OLAP view was saved
- the model name
- the type of predefined cube
- the network login of the person who saved the OLAP view

Removing Items

Your organization should encourage users to remove items that were automatically saved or that they saved and no longer need. You can establish guidelines for when to remove items.

Sharing Data with Others

For people who do not use SAS Cost and Profitability Management, you can export cubes to a Microsoft Excel spreadsheet.

Chapter 10

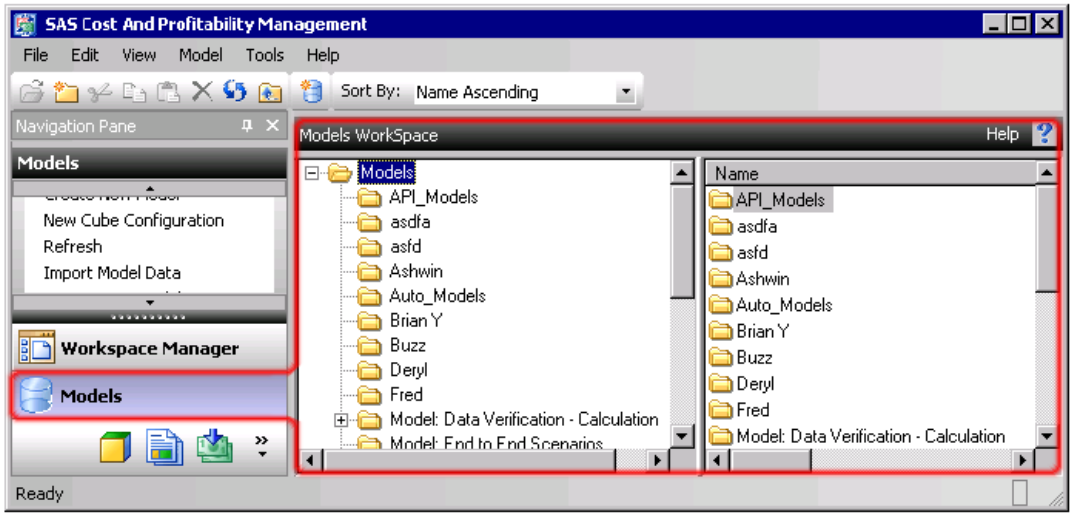
Windows for Models

Models Workspace

About the Models Workspace

Note: The availability of these features depends on your permissions.

From the Models workspace, you can open a model.



The list of Folders and the list of Models correspond to the Models branch of the server area in Workspace Manager.

How to Access the Models Workspace

Do one of the following:

- If no model is open, click **Models** in the Navigation Pane.
- Select **Model** ⇨ **Change Model or Context** to open a different model.

Sort information in the Models Workspace

1. Click the **Sort By** link.

   A menu appears.

2. Select an option. The options contain the following criteria:

Name	The name of the model
Date and Time	The date and time which the model was created

New Model Dialog Box

About the New Model Dialog Box

Use the New Model dialog box to create a new model. Creating a model entails specifying:

- model name, reference, and base currency.
- initial period/scenario association.
- the number of modules, their names, and their order.
- the dimensions belonging to each module.

See Also

- "Create a Model" on page 104
- Chapter 9, "Model Concepts," on page 79

How to Access the New Model Dialog Box

Do one of the following:

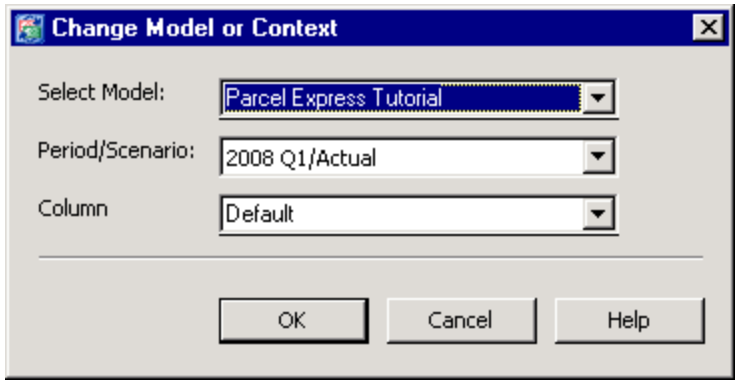
- Select **File** ⇨ **New** ⇨ **Model** from the menu bar.
- Select **Create New Model** from the Navigation Pane (see Chapter 4, "Navigation Pane," on page 49).

Change Model or Context Dialog Box

About the Change Model or Context Dialog Box

Use this dialog to do any of the following:

- Open a different model, specifying its period/scenario association and column layout
- Change the period/scenario association for the currently open model
- Change the column layout of the currently open model



How to Access the Change Model or Context Dialog Box

Select **Model** ⇨ **Change Model or Context**.

Note: You must be in the Models Workspace for this menu item to be available.

See Also

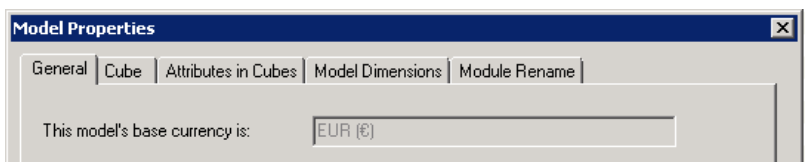
"Models Workspace" on page 92

Model Properties Dialog Box

About the Model Properties Dialog Box

The availability of these features depends on your permissions.

In the Model Properties dialog box, you can review or change information about a model.



The Model Properties dialog contains the following tabs:

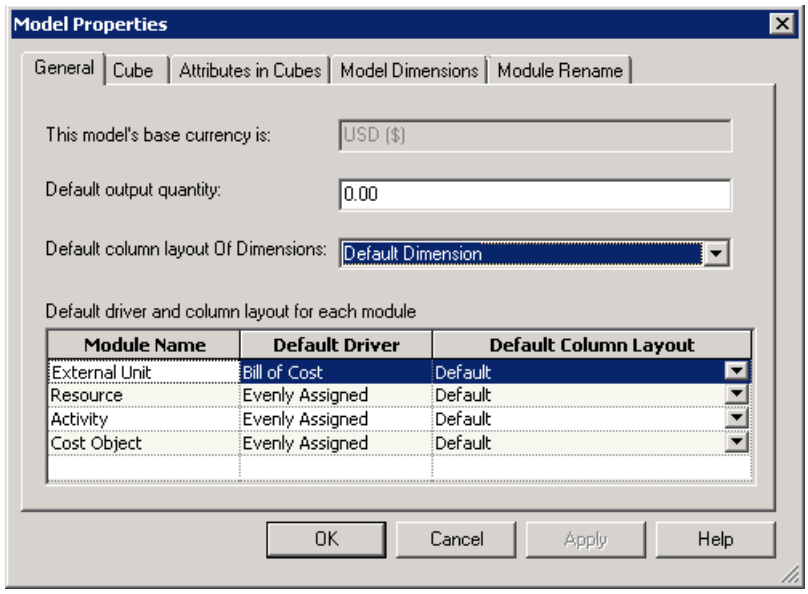
Tab	Function
General (p. 95)	Specify general options
Cube (p. 96)	Specify **In** or **Out** for each module of a Muti-stage Contributions cube
Attributes in Cubes (p. 97)	Specify the numeric attributes to be included in cubes
Model Dimensions (p. 97)	View the dimensions that each module contains

Tab	Function
Module Rename (p. 97)	Rename any module

How to Access the Model Properties Dialog Box

Open a model and select **Model** ⇨ **Properties**.

General Tab



Click the **General** tab, and choose among the following options:

Model's base currency

A model's base currency is the currency that was selected when the model was created. You cannot change the model's base currency.

Default Output Quantity

This property is intended for demonstration purposes only and not for working models. The value that you specify for **Default Output Quantity** is substituted for Driver Quantity Calculated under certain conditions when the value of Driver Quantity Calculated is zero. This allows those accounts to drive costs for demonstration purposes that they wouldn't otherwise drive.

Default column layout for dimensions

This is the column layout that is displayed by default for the Dimensions view. Whenever you are in the Dimensions view you can activate a different layout.

Default driver and column layout for each module

- The default driver is the driver that is associated with an assignment when you first create the assignment. You can change the driver subsequently after making the assignment.
- The default column layout for a module is the layout that is displayed by default whenever you view that module. Whenever you are viewing that module you can

activate a different layout. However, the next time you open the model and view that module, the its default column layout is displayed again.

Cube Tab

The options that you select here are automatically selected as defaults when you create a new cube configuration for the model. You can, however, override the defaults when you create the cube configuration.

Note: Changing these options after a cube has been generated does not affect the generated cube. It only affects the defaults for the creation of new cube configurations.

1. Click the **Cube** tab.
2. Choose one of the following options:
   - **Define stages by each module**. Each module defines a separate stage (one stage per module) in the order in which you defined the modules, or in the following order for a traditional activity-based costing model:
     1. External Unit
     2. Resource
     3. Activity
     4. Cost Object
   - **Define stages by a dimension attribute named Stages**.

     Each stage is defined by a dimension member attribute in a dimension attribute named Stages. For more information, see "Stage Attributes" on page 275 and "Add Stage Attributes to Accounts" on page 313.
3. In the **Use the cost as it flows in or out of each stage** list, select the **Cost Flow (In or Out)** for each module or stage.

   If you select **In**, then only costs flowing into accounts in the module or stage are included in the cube. If you select **Out**, then only costs flowing out of accounts in the module or stage are included in the cube.

   Note: The **In**/**Out** option only makes a difference in the case of accounts that make assignments to other accounts in the *same* module or stage (or in an earlier stage). If such is the case, then not all cost flowing into that account also flow out of it from the *same stage or module*. If such is the case, then you must decide whether you want the cube to show cost flowing into the account in that stage or module, or cost flowing out of the account from that stage or module. If, on the other hand, there are no accounts that make assignments to other accounts in the same module or stage (or in an earlier stage), then the **In**/**Out** option makes no difference. All cost flowing into an account in one stage or module also flows out of it from that module or stage (or, at least is accounted for as None if it remains in the account).

Attributes in Cubes Tab

Overview

Use the **Attributes in Cubes** tab of the Model Properties dialog box to work with attributes. However, the function of this tab is different for a single-stage contribution cube than for a resource contribution cube or a multi-stage contribution cube.

Single-Stage Contribution Cube

To include numeric attributes in a single-stage contribution cube:

1. Select the **Attributes in Cubes** tab.
2. Select the numeric attributes to be included in the cube.

   If you generate a single-stage contribution cube, then the numeric attributes that you select are included in the cube.

Resource Contribution Cube and Multi-Stage Contribution Cube

The numeric attributes that you select on the **Attributes in Cubes** tab are automatically checked for inclusion when you create a new cube configuration for the model. You can, however, decide not to include the properties when you create the cube configuration, and you can include different attributes. The attributes that you select here are only checked by default. You can reverse the decision for any particular cube configuration.

1. Click the **Attributes in Cubes** tab.
2. Select the numeric attributes to be checked by default in a new cube configuration.

See Also

See also: "Include Numeric Attributes in a Cube" on page 577.

Model Dimensions Tab

1. Click the **Model Dimensions** tab.
2. In the Dimensions used in each module list, expand a module.

   The dimensions that are used in the module are displayed.

Module Rename Tab

This tab allows you to rename any module in a model.

Model Summary Window

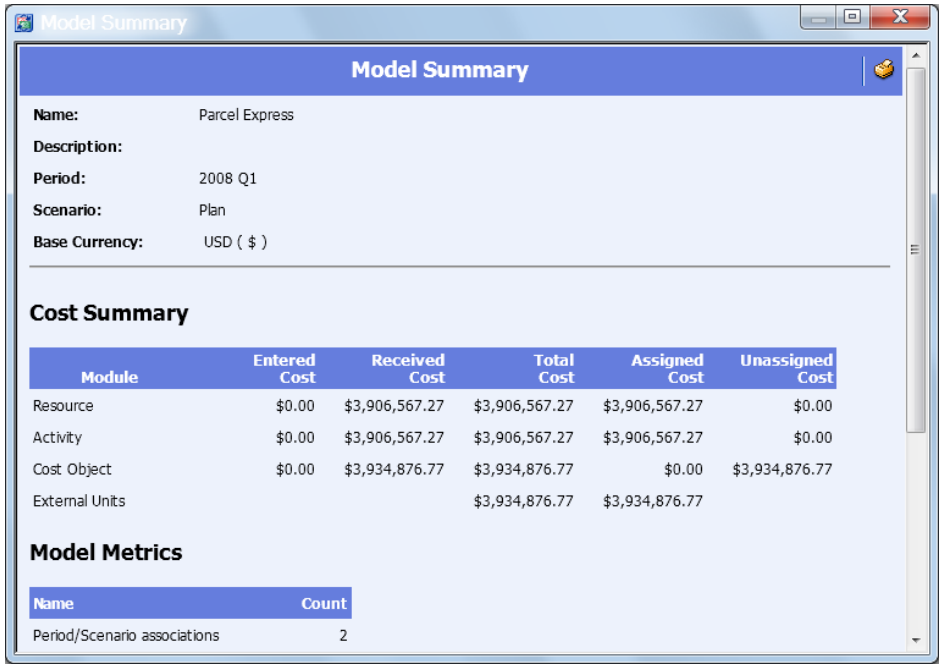
About the Model Summary Window

In the Model Summary window, you can see summary information about various aspects of a model for the current period/scenario association, such as total costs for each module and the number of cycles (reciprocal cost assignments) in the model.

Note: You cannot change this information.

How to Access the Model Summary Window

Open a model and select **Model** ⇨ **Summary**.



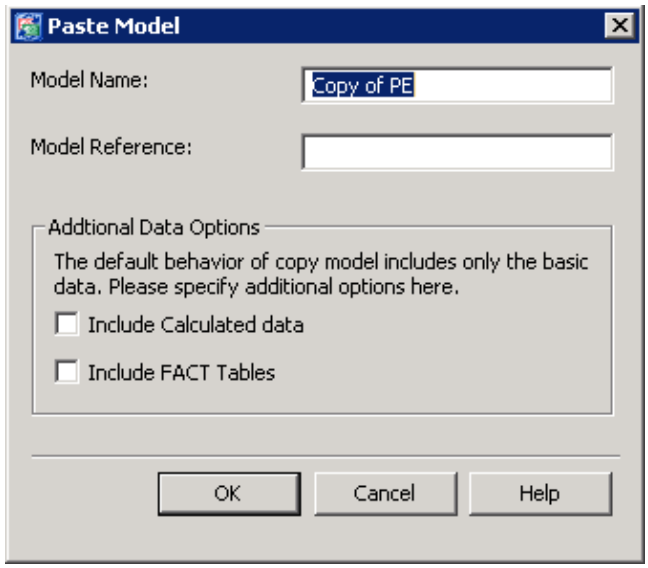
Copy a Model

Before making major changes to a model, you can make a copy of the model that you can refer to, or revert to, later on.

To copy a model:

1. Go to the Workspace Manager.
2. Select the model that you want to copy, and then select **Edit** ⇨ **Copy** (or, right-click the model and select **Copy**).

3. Select the directory in which you want to place the model. Then select **Edit** ⇨ **Paste** (or right-click the directory and select **Paste**). The Paste Model dialog box opens.



4. If the model that you want to copy was previously calculated and fact tables were generated, then you can include the calculated data and fact tables along with the model. If you choose to copy the fact table, then the calculated data is automatically included. Choose none, one, or both of the following options:

   - Include Calculated data
   - Include FACT Tables

   Whereas models that you export and re-import must be recalculated and their fact tables regenerated, models that you copy do not require those steps.

Model Locking Status Dialog Box

Overview

Use the Model Locking Status dialog box to:

- see what users have a lock on a particular model
- terminate the locks on a particular model

How to Access the Model Locking Status

To access the Model Locking Status dialog box:

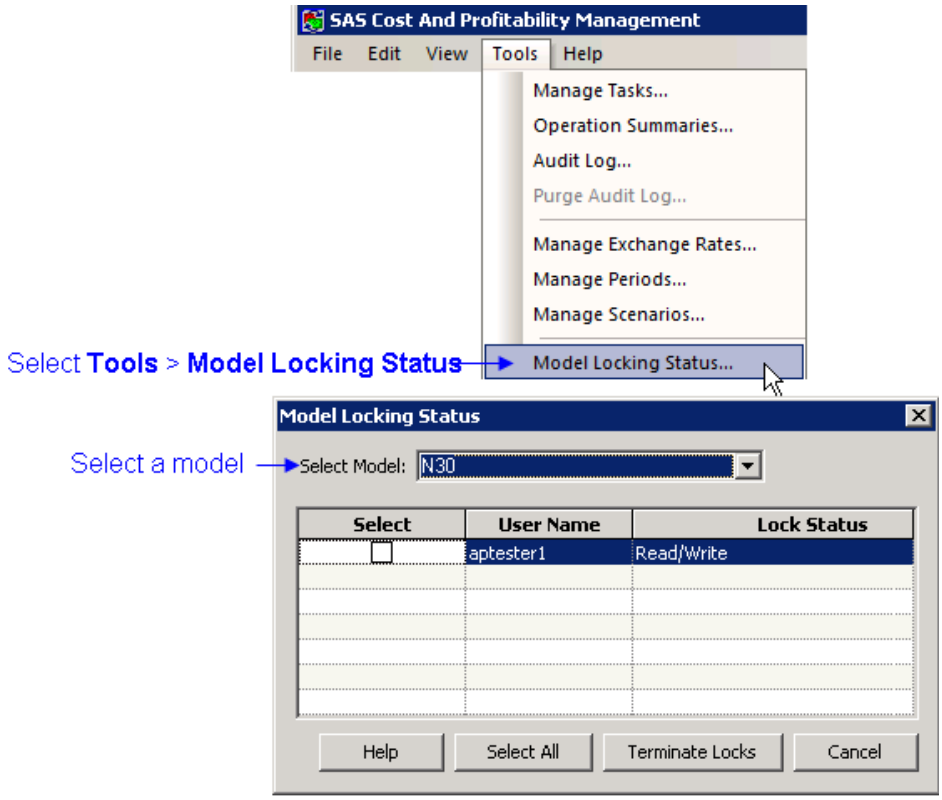
- Select **Tools** ⇨ **Model Locking Status**.

See What Users Have a Lock on a Particular Model

To see what users have a lock on a particular model:

1. Select **Tools** ⇨ **Model Locking Status**.
2. Select a model from the drop-down list.

   The Model Locking Status dialog box shows the users that have a lock on that model and the type of lock.



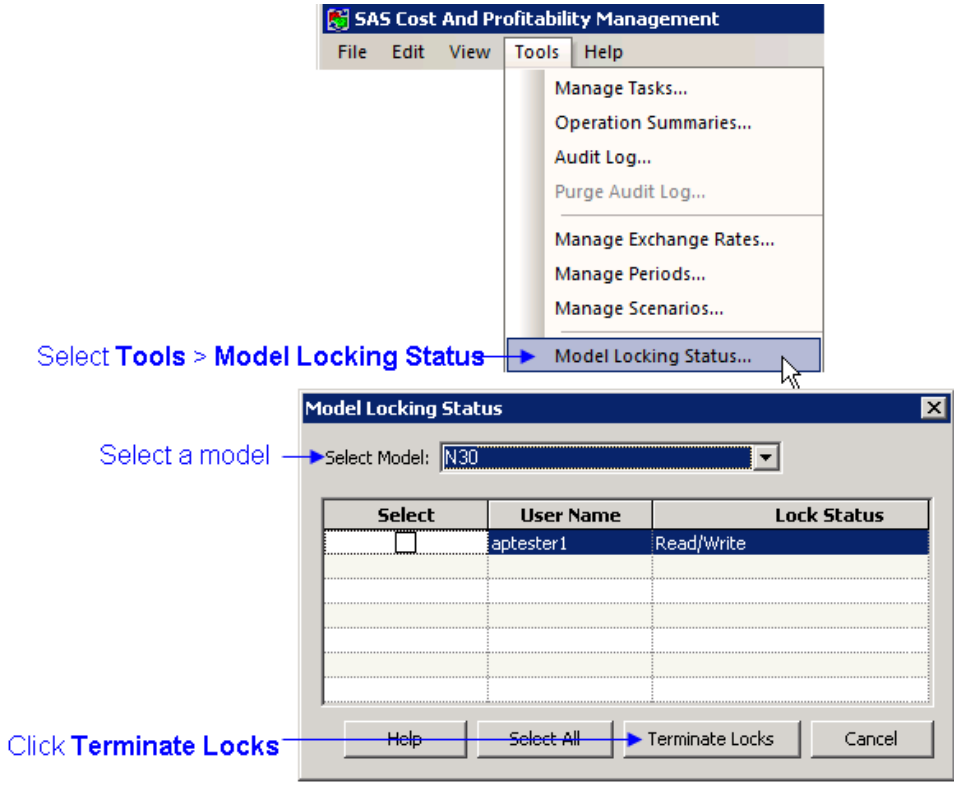
Terminate a Lock on a Particular Model

An administrator and a modeler with Read/Write access to the model can terminate the lock of another user. A viewer can terminate the viewer's own Read lock—but not the lock of another user.

To terminate a lock on a particular model:

1. Select **Tools** ⇨ **Model Locking Status**.
2. Select a model from the drop-down list.

   The Model Locking Status dialog box shows the users that have a lock on that model and the type of lock.
3. Select the lock(s) to terminate.
4. Click **Terminate Locks**.



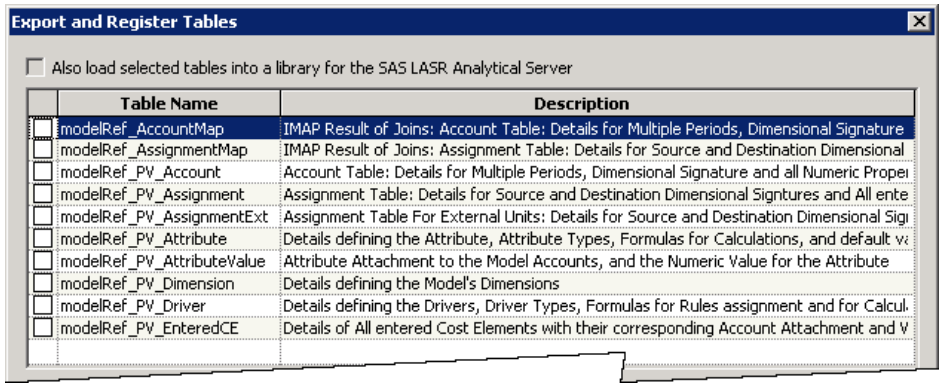
5. A warning message appears, indicating that errors could occur if an administrator or modeler with read/write access to the model terminates the write lock on a model when the modeler is working on the model. Click **Yes** to terminate the lock.

Export and Register Tables Dialog Box

Overview

Use the Export and Register Tables dialog box to export database tables that are registered in SAS Management Console for use with other SAS or non-SAS programs. The tables are exported to your default database—the one into which FACT tables are written.

Note: **Also load selected tables into a library for the SAS LASR Analytic Server** is enabled only if you have access to a library that is enabled for the SAS LASR Analytic Server. See "Metadata Server Options" on page 504.



See Also

- Chapter 82, “Export Registered Tables,” on page 695
- “Some Public Views Are No Longer Generated Automatically” on page 20

How to Access the Export and Register Tables dialog box

To access the Export and Register Tables dialog box:

- With a model open, select **Model** ⇨ **Export and Register Tables**.

Chapter 11

How To: Models

Create a Model

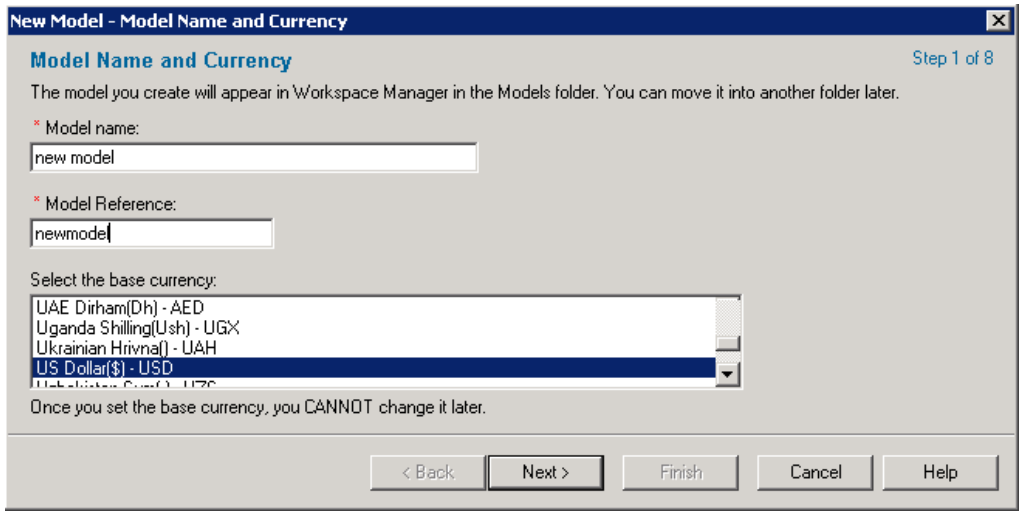
Open the New Model dialog box

To create a model, do one of the following:

- Select **File** ⇨ **New** ⇨ **Model** from the menu bar.
- Select **Create New Model** from the Navigation Pane (see Chapter 4, "Navigation Pane," on page 49).

The New Model dialog box opens. See "New Model Dialog Box" on page 93.

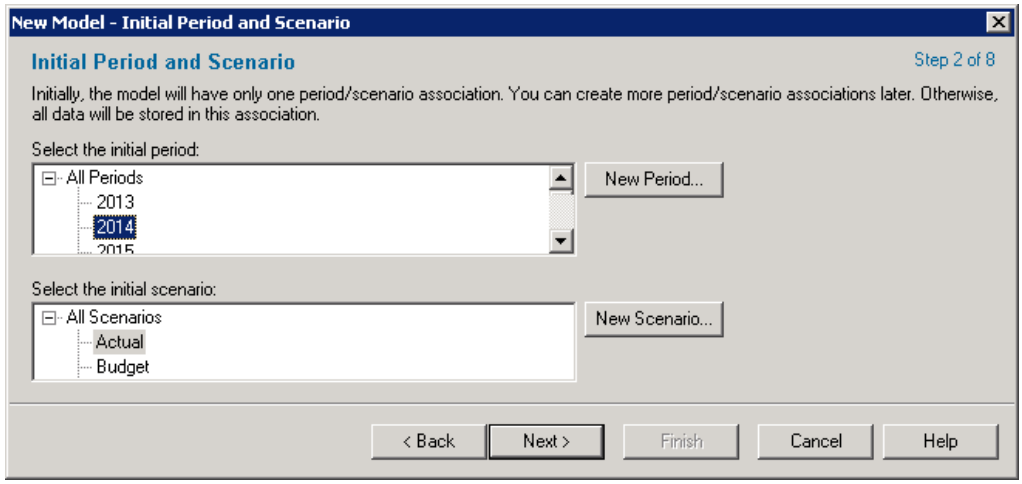
Step 1: Specify Model Name, Reference, and Currency



1. In the **Model Name** box, type the name. See "Naming Conventions" on page 69.
2. In the **Reference** field, type an 8-byte string. This short reference is used in public views. (See Chapter 16, "Registered Table Schemas," in *SAS Cost and Profitability Management: Data Administration Guide*.) Also see "Reference Conventions" on page 74.
3. From the **Select the base currency** list, select a currency.

   Notice that once you select a base currency, you cannot change it later.
4. Click **Next**.

Step 2: Specify Initial Period and Scenario



When you create a new model, you must specify an initial period and an initial scenario for the model. The pair that you select is the initial period/scenario association for the model. Until you add other period/scenario associations to your model, all your data will reside in this initial period/scenario association.

1. From the **Select the initial period** list, select a period.

   You can do either of the following:

   - Select from the periods listed.
   - Create a new period to select by first clicking **New Period**. This opens the Manage Periods dialog box, which you can use to create a new period. See "Create a Period" on page 153.

2. From the **Select the initial scenario** list, select a scenario.

   You can do either of the following:

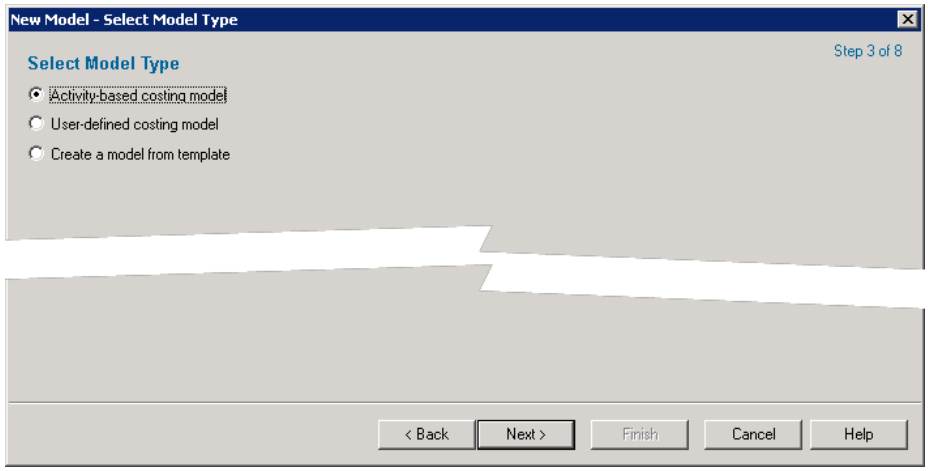
   - Select from the scenarios listed.
   - Create a new scenario to select by first clicking **New Scenario**. This opens the Manage Scenarios dialog box, which you can use to create a new scenario. See "Create a Scenario" on page 156.

3. Click **Next**.

See Also

Chapter 15, "Working with Periods and Scenarios," on page 141

Step 3: Select the Model Type



Select the type of model to create.

Activity-based costing model

By default, with this type of model there are four modules: External Units, Resource, Activity, and Cost Object. You can, however, change the number of modules. See "Type 1: Create an Activity-Based Costing Model" on page 106.

User-defined costing model

A user-defined costing model can have from 1 to 10 modules. For each module, you can specify its name, reference, and its structural dimensions. See "Type 2: Create a User-Defined Costing Model" on page 110.

Create a model from template

When you create a model from a template, you provide information contained in the template to the new model. A template includes the following information:

- the model's modules, including their order, name, and reference
- the dimensions included in each module

You can override the template information when you create a new model. See "Type 3: Create a Model from Template" on page 113.

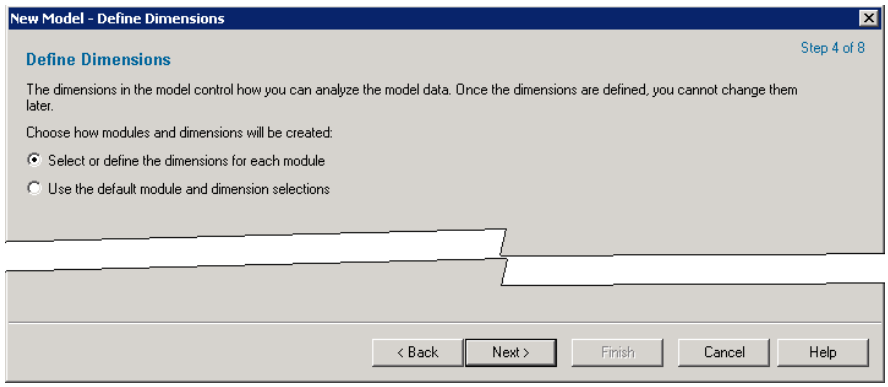
Type 1: Create an Activity-Based Costing Model

Overview

With an Activity-Based Costing Model, there are four modules by default: External Units, Resource, Activity, and Cost Object. You can, however, change the number of modules and rename them.

For the first three steps of creating any new model, see "Create a Model" on page 104.

Step 4: Choose Default or Custom Dimensions



Choose one of the following options:

Select or define the dimensions for each module

With this option, you can:

- choose how many modules to include in the model
- rename modules
- choose what dimensions to give to each module

After clicking **Next**, proceed to "Step 5: Rename the Modules and Add or Remove Modules" on page 108.

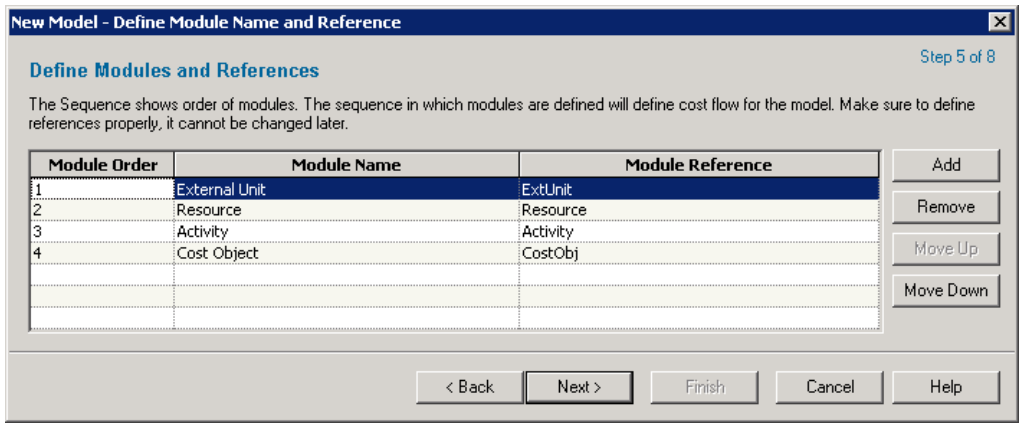
Use the default module and dimension selections

With this option, your model will contain the following modules and dimensions. You can not add or remove modules, rename them, or change their dimensions.

Module	Dimensions
External Unit	Materials
Resource	Organization General Ledger
Activity	Organization Activities
Cost Object	Customers Products and Services

Note: If you choose this option, then you have finished defining your model.

Step 5: Rename the Modules and Add or Remove Modules



You can modify the default modules by doing the following:

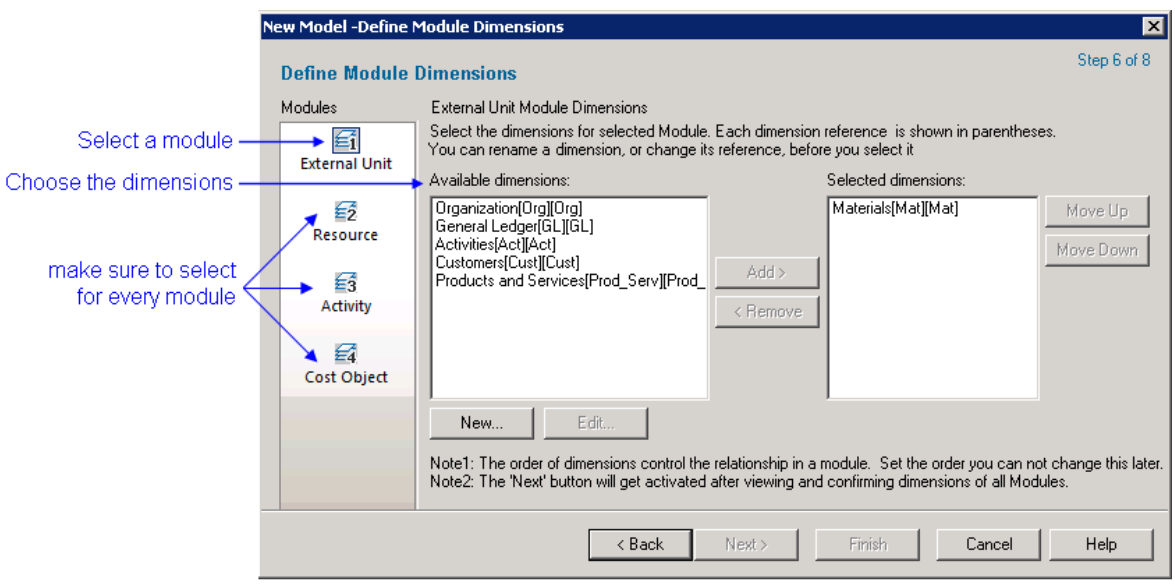
- Change a module name or reference by typing over it. See "Module naming conventions" on page 72.
- Click **Add** to add additional modules.
- Click **Remove** to remove a module.
- Click **Move Up** or **Move Down** to change the order of modules. The module order determines the direction of cost flow. See "Assignment Paths" on page 432.

Click **Next**.

Step 6: Specify Dimensions for Each Module

In the left-hand pane, click each module in turn and select its dimensions.

Note: The **Next** button is enabled only after you have selected the dimensions for every module.



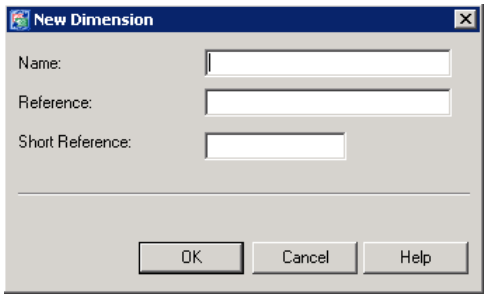
1. From the list of **Available dimensions**, select a dimension, and click Add >.

   You can select multiple dimensions but only one dimension at a time.

2. To create a new dimension, do the following:

a. Click **New**.

The New Dimension dialog box appears. See "New Dimension Dialog Box" on page 206.



b. Type the **Name**.

The name must follow the naming conventions. See "Naming Conventions" on page 69.

c. Type the **Reference**.

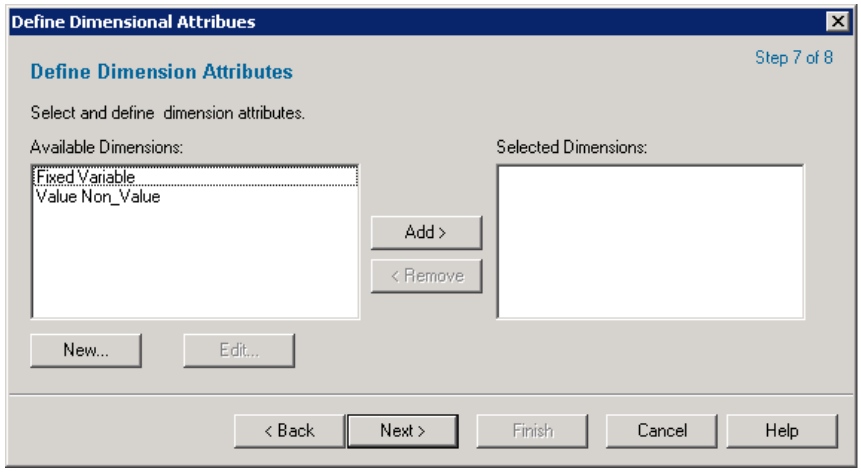
A default reference is created from the dimension's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.

d. Type a **Short Reference**. The short reference is used in public views. See Chapter 16, "Registered Table Schemas," in *SAS Cost and Profitability Management: Data Administration Guide*.

3. Click **Next**.

See Chapter 24, "What Are Dimensions?," on page 195.

Step 7: Define Dimension Attributes

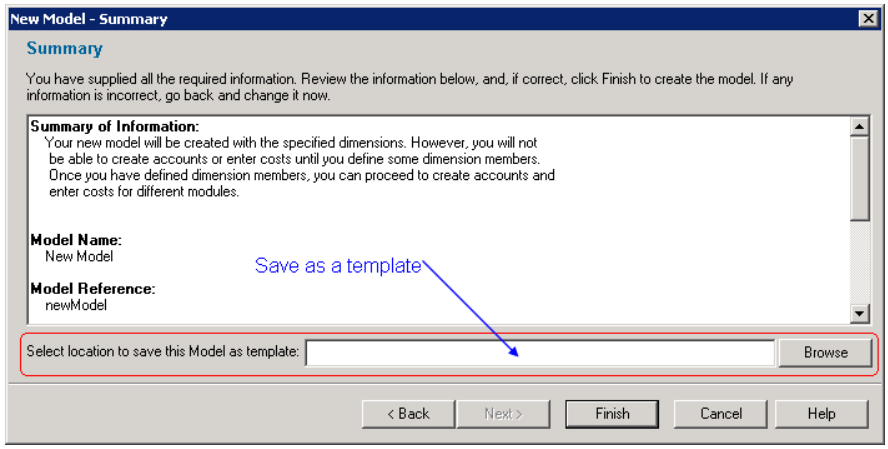


This step is optional. In addition to defining the structural dimensions whose intersections constitute the accounts in a module, you can also define dimension attributes for a model. The values of a dimension attribute (its dimension members) can be applied to the accounts in a module. A dimension attribute is available in a cube as another dimension just like the structural dimensions. A common example of a dimension attribute is Stages whose values are used to subdivide a model into temporal phases.

See also:

- "Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes" on page 272
- "Stage Attributes" on page 275

Step 8: Review Choices and Finish



1. Review the summary information.
2. If you need to change any information, click **Back**. All the information that you have specified is saved. Click **Next** to advance through the wizard.
3. Click **Browse** if you want to save the model as a template. A template includes the following information:
   - the model's modules, including their order, name, and reference
   - the dimensions included in each module

   A template is an XML file that can be stored on the machine of your choice and shared with other modelers for creating new models. See "Type 3: Create a Model from Template" on page 113.
4. Click **Finish**.

   The model is created and the Dimensions page appears for you to define dimension members.

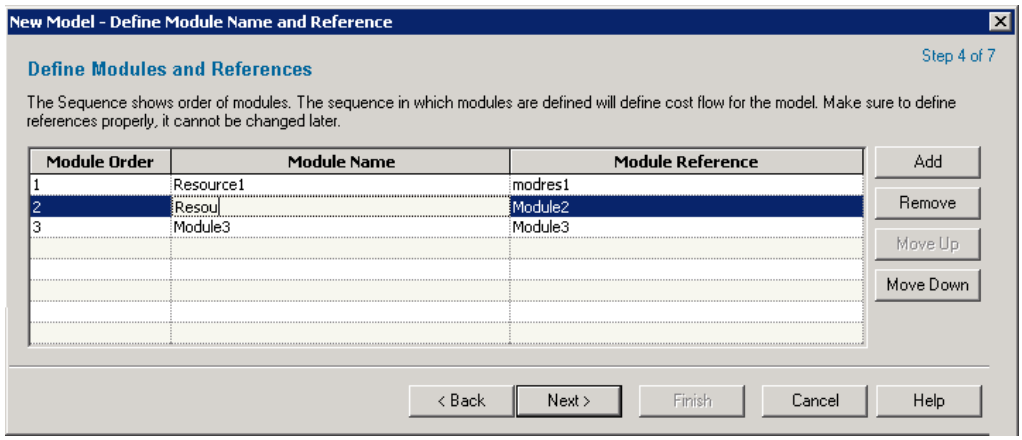
Type 2: Create a User-Defined Costing Model

Overview

A user-defined costing model can have from 1 to 10 modules.

For the first three steps of creating any new model, see "Create a Model" on page 104.

Step 4: Name the Modules and Add or Remove Modules



You can modify the default modules by doing the following:

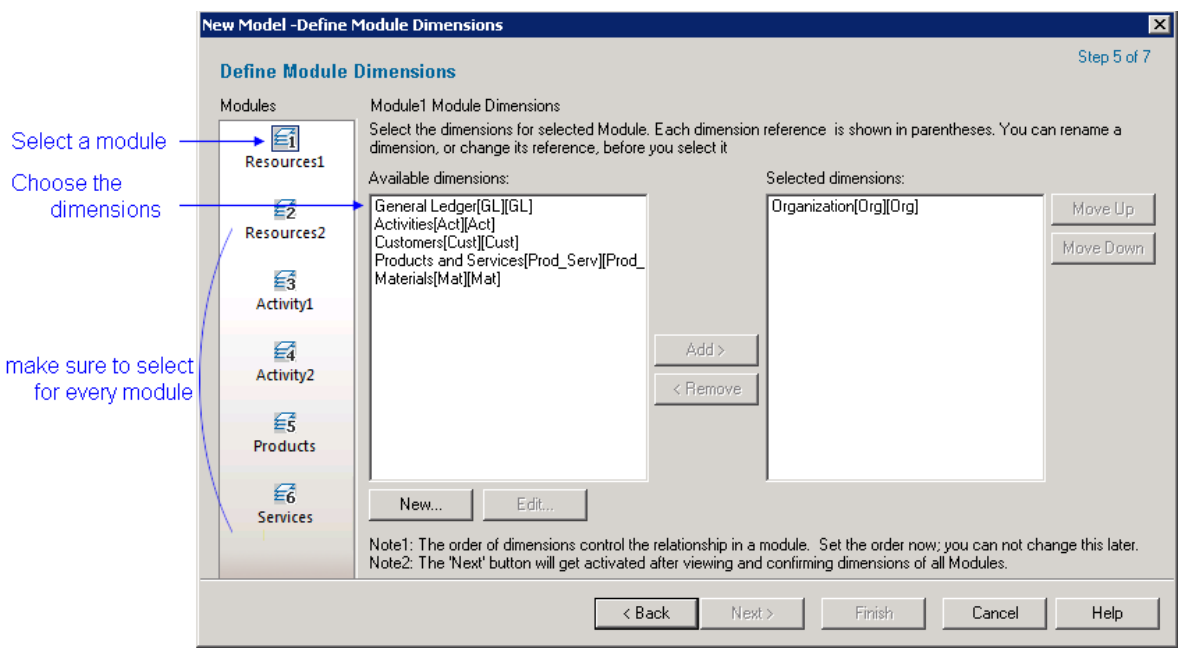
- Change a module name or reference by typing over it. See "Module naming conventions" on page 72.
- Click **Add** to add additional modules.
- Click **Remove** to remove a module.
- Click **Move Up** or **Move Down** to change the order of modules. The module order determines the direction of cost flow. See "Assignment Paths" on page 432.

Click **Next**.

Step 5: Specify Dimensions for Each Module

In the left-hand pane, click each module in turn and select its dimensions.

Note: The **Next** button is enabled only after you have selected the dimensions for every module.



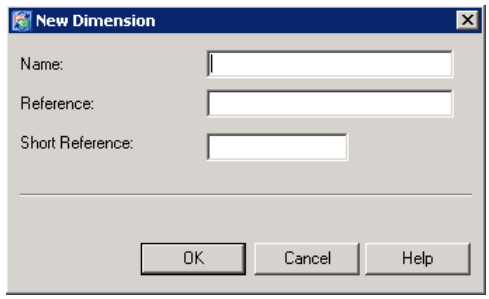
1. From the list of **Available dimensions**, select a dimension, and click Add >.

You can select multiple dimensions but only one dimension at a time.

2. To create a new dimension, do the following:

   a. Click **New**.

      The New Dimension dialog box appears. See "New Dimension Dialog Box" on page 206.



   b. Type the **Name**.

      The name must follow the naming conventions. See "Naming Conventions" on page 69.

   c. Type the **Reference**.

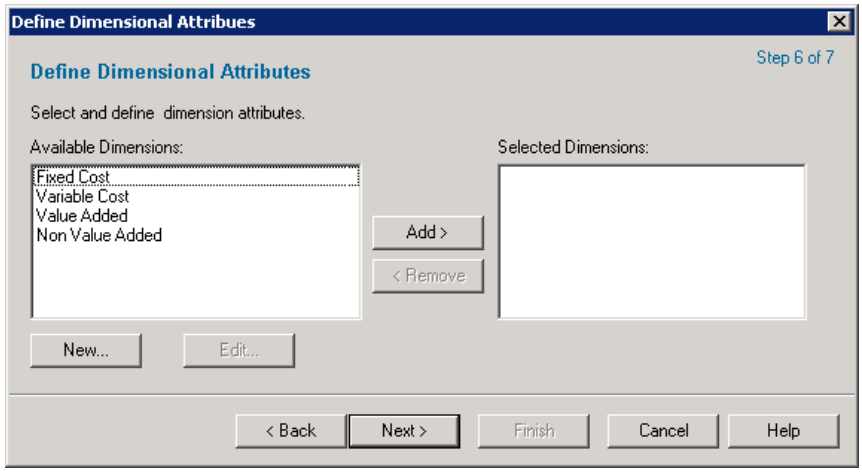
      A default reference is created from the dimension's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.

   d. Type a **Short Reference**. The short reference is used in public views. See Chapter 16, "Registered Table Schemas," in *SAS Cost and Profitability Management: Data Administration Guide*.

3. Click **Next**.

See Chapter 24, "What Are Dimensions?," on page 195.

Step 6: Define Dimension Attributes



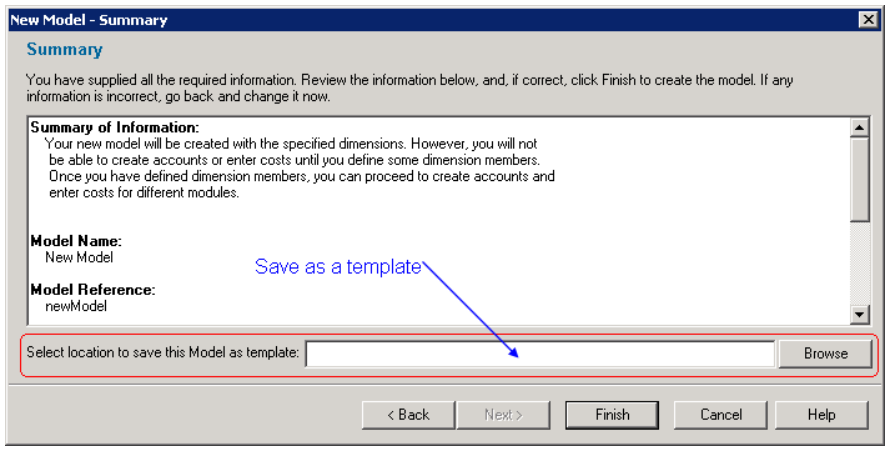
This step is optional. In addition to defining the structural dimensions whose intersections constitute the accounts in a module, you can also define dimension attributes for a model. The values of a dimension attribute (its dimension members) can be applied to the accounts in a module. A dimension attribute is available in a cube as another dimension just like the structural dimensions. A common example of a

dimension attribute is Stages whose values are used to subdivide a model into temporal phases.

See also:

- "Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes" on page 272
- "Stage Attributes" on page 275

Step 7: Review Choices and Finish



1. Review the summary information.
2. If you need to change any information, click **Back**. All the information that you have specified is saved. Click **Next** to advance through the wizard.
3. Click **Browse** if you want to save the model as a template. A template includes the following information:
   - the model's modules, including their order, name, and reference
   - the dimensions included in each module

   A template is an XML file that can be stored on the machine of your choice and shared with other modelers for creating new models. See "Type 3: Create a Model from Template" on page 113.
4. Click **Finish**.

   The model is created and the Dimensions page appears for you to define dimension members.

Type 3: Create a Model from Template

Overview

When you create a model from a template, you provide information contained in the template to the new model. A template includes the following information:

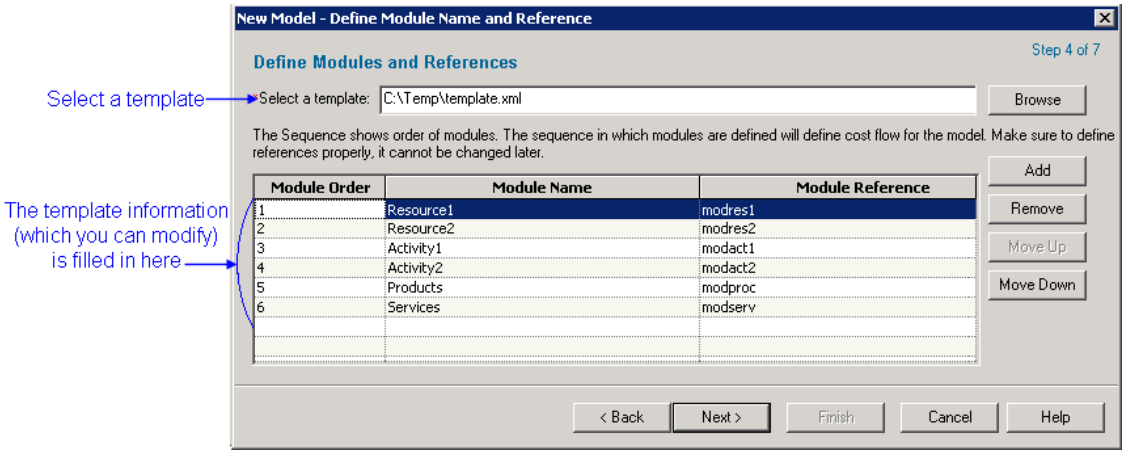
- the model's modules, including their order, name, and reference
- the dimensions included in each module

You can override the template information when you create a new model.

For the first three steps of creating any new model, see "Create a Model" on page 104.

Step 4: Specify a Template and Modify the Modules

1. Specify a template.



   When you create a model from a template, you provide information contained in the template to the new model. A template includes the following information:

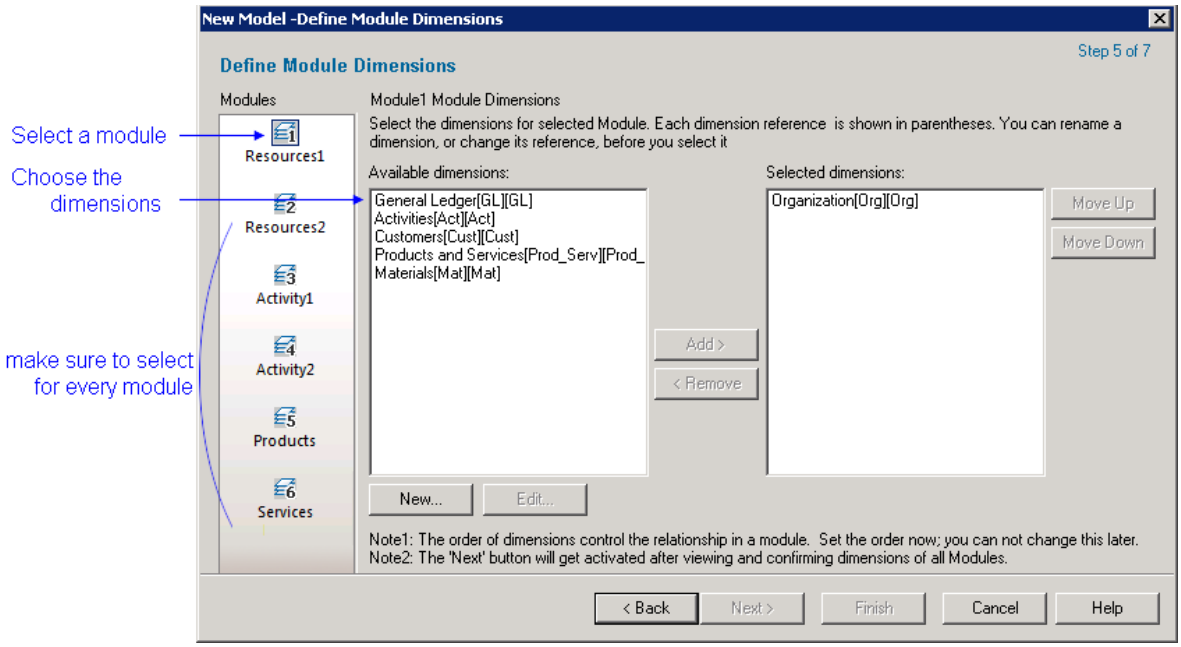
   - the model's modules, including their order, name, and reference
   - the dimensions included in each module

2. Optionally, rename the modules and add or remove modules.

Step 5: Specify Dimensions for Each Module

In the left-hand pane, click each module in turn and select its dimensions.

Note: The **Next** button is enabled only after you have selected the dimensions for every module.



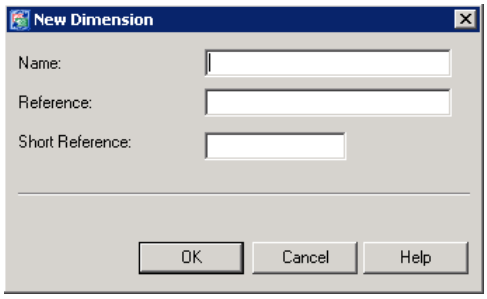
1. From the list of **Available dimensions**, select a dimension, and click Add >.

   You can select multiple dimensions but only one dimension at a time.

2. To create a new dimension, do the following:

   a. Click **New**.

      The New Dimension dialog box appears. See "New Dimension Dialog Box" on page 206.



   b. Type the **Name**.

      The name must follow the naming conventions. See "Naming Conventions" on page 69.

   c. Type the **Reference**.

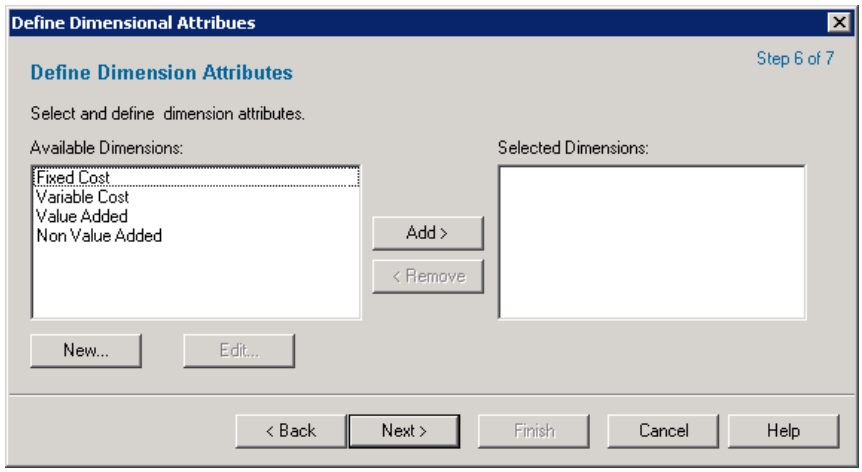
      A default reference is created from the dimension's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.

   d. Type a **Short Reference**. The short reference is used in public views. See Chapter 16, "Registered Table Schemas," in *SAS Cost and Profitability Management: Data Administration Guide*.

3. Click **Next**.

See Chapter 24, "What Are Dimensions?," on page 195.

Step 6: Define Dimension Attributes

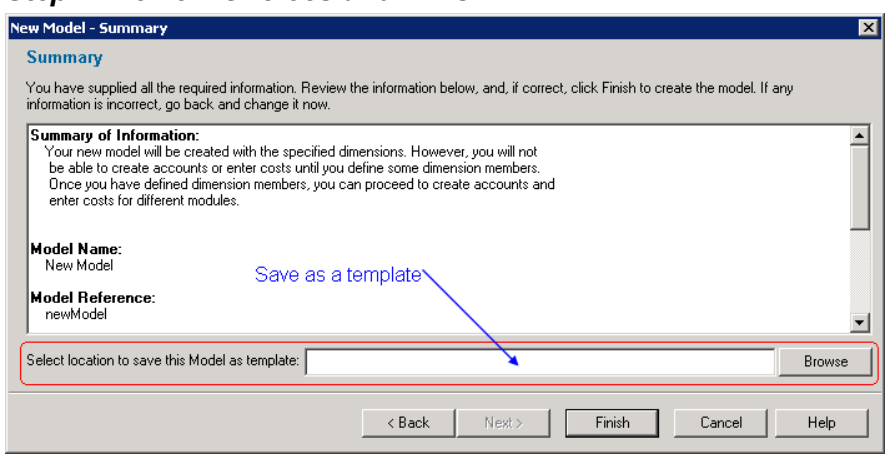


This step is optional. In addition to defining the structural dimensions whose intersections constitute the accounts in a module, you can also define dimension attributes for a model. The values of a dimension attribute (its dimension members) can be applied to the accounts in a module. A dimension attribute is available in a cube as another dimension just like the structural dimensions. A common example of a dimension attribute is Stages whose values are used to subdivide a model into temporal phases.

See also:

- "Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes" on page 272
- "Stage Attributes" on page 275

Step 7: Review Choices and Finish



1. Review the summary information.
2. If you need to change any information, click **Back**. All the information that you have specified is saved. Click **Next** to advance through the wizard.
3. Click **Browse** if you want to save the model as a template. A template includes the following information:
   - the model's modules, including their order, name, and reference
   - the dimensions included in each module

A template is an XML file that can be stored on the machine of your choice and shared with other modelers for creating new models. See "Type 3: Create a Model from Template" on page 113.

4. Click **Finish**.

   The model is created and the Dimensions page appears for you to define dimension members.

Open a Model for Editing

You must acquire a write lock on the model to edit it. When you open a model for editing, no one else can edit it. However, other users can open the model for viewing. You must have a read/write access to the model to open it for editing.

To open a model for editing, in the **Models** tab of the Navigation pane, double-click the model. The model opens for editing with a write lock.

Note: If the model is already open in read mode, then you must close the model before you can acquire a write lock on it. To close the model, select **Model** ⇨ **Close Model and Release Lock** from the open model.

Open a Model for Viewing

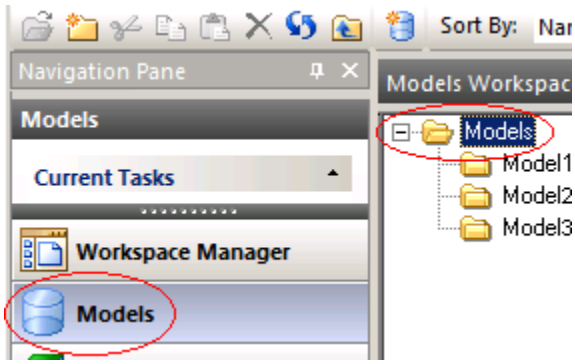
How you open a model depends on whether no model is open or whether a model is already open and you want to open a different one.

Open a Model for Viewing with No Model Open

Even though a user (with a write lock) is editing a model, multiple users can open the model simultaneously for viewing.

If no model is open:

1. Click the **Models** tab in the Navigation Pane. (See Chapter 4, "Navigation Pane," on page 49.)



Note: You can also click the **Workspace Manager** tab and then select the **Models** folder.

2. If you have only read access to the model, double-click the model in the list of models.

   If you have read/write access to the model, right-click the model in the list of models and select **Open with Read Lock**.

   Note: To close the model and thereby release the read lock, you can do any of the following:

   - In the **Workspace Manager**, right-click the model and select **Release Read Lock**.
   - Select **Model** ⇨ **Close Model and Release Lock**.
   - Acquire read lock on a different model.
   - Close the client application.

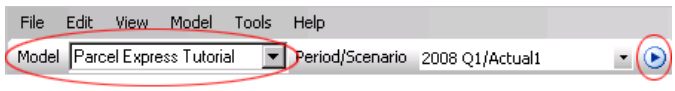
   See"Acquire or Release a Lock on a Model" on page 119.

Open a Model with a Model Already Open

Even though a user (with a write lock) is editing a model, multiple users can open the model simultaneously in read mode for viewing.

If a model is already open and you want to open a different model, do one of the following:

- Select a model from the **Model** drop-down list, and select a period and scenario from the **Period/Scenario** drop-down list, and then click the arrow icon ▶.



  If you do not see the model that you want to use, you probably do not have permission to view it.

- Click the **Go to Models Workspace** icon on the toolbar, and then select a model to open.

- Select **Model** ⇨ **Change Model or Context** to open a different model

Note: To close the model and thereby release the read lock, you can do any of the following:

- In the **Workspace Manager**, right-click the model and select **Release Read Lock**.
- Select **Model** ⇨ **Close Model and Release Lock**.
- Acquire read lock on a different model.
- Close the client application.

See"Acquire or Release a Lock on a Model" on page 119.

See Also

"Change Model or Context Dialog Box" on page 93

Acquire or Release a Lock on a Model

Overview

Because models are no longer stored in a database, there is no database management system to handle model access. If you want to edit a model, you must first acquire a write lock on it. If you want to view a model, you must acquire a read lock on it. Only one user can acquire a write lock on a model at a time and edit that model. However, when one user has a write lock on a model, multiple users can acquire a read lock on the model to view it.

The following describes how to open and close models for editing or viewing.

Write Lock

You must acquire a write lock on a model to edit it. When you have a write lock, no one else can acquire a write lock on the model. You must release the write lock on the model so that another user can acquire a write lock on it.

Acquire

To acquire a write lock, double-click the model in the Workspace Manager. Alternatively, right-click the model and select **Open**. The model opens for editing.

Note: A modeler must have read/write access to open the model with a write lock.

Release

- To release the write lock on a model, the modeler with the lock can select **Model ⇨ Close Model and Release Lock** when the model is open. Selecting this action closes the model.

  TIP It is good practice to release the write lock on a model that you are no longer actively working on so as to free up system resources. Each write lock involves a running process that uses system memory. So, if you are not actively working on a model, it is desirable that you release its write lock to free up memory.

- The write lock that is held by a modeler is released if the modeler closes the client application. But, the write lock is not released if the modeler acquires a write lock on another model.

  Note: If you want to release a lock by closing the model or by closing the client application when an operation (such as exporting data, calculating costs, and so on) is in progress, you must cancel the operation first or wait for the operation to finish.

- An administrator or a modeler with read/write access to the model can release any user's lock on the model by selecting **Tools ⇨ Model Locking Status**. See "Model Locking Status Dialog Box" on page 99.

Read Lock

When you open a model for viewing, you automatically acquire a read lock on it. Multiple users can open a model with a read lock and view it simultaneously.

Acquire

- If you have read access to the model, double-click the model in the **Workspace Manager** to acquire the read lock. Alternatively, right-click the model and select **Open**.
- If you have read/write access to the model, right-click the model in the **Workspace Manager** and then select **Open with Read Lock** to acquire the read lock on the model.

Release

- To release the read lock on a model, the modeler with the read lock can do any of the following:
  - In the **Workspace Manager**, right-click the model and select **Release Read Lock**.
  - Select **Model** ⇨ **Close Model and Release Lock**.
  - Acquire a read lock on a different model.
  - Close the client application.

  Note: If you want to release a lock by closing the model or by closing the client application when an operation (such as exporting data, calculating costs, and so on) is in progress, you must cancel the operation first or wait for the operation to finish.
- An administrator or a modeler with read/write access to the model can release the lock on the model by selecting **Tools** ⇨ **Model Locking Status**. See "Model Locking Status Dialog Box" on page 99.

See Also

"Model Locking Status Dialog Box" on page 99

Delete a Model

1. In the Models Workspace, right-click a model and select **Delete**.

   Note: Depending on your permissions, **Delete** might not be available.
2. In the confirmation dialog box, click **Yes**.
3. In the Delete Model dialog box, select the following check boxes:
   - **Delete the cube (if any) for the selected model** Select this check box to delete the cubes that are generated for the selected model.
   - **Delete the database tables (if any) that contained model data for reporting** Select this check box to delete the database tables that are created to export the model data for reporting. If the model data is exported to the default database, SAS Cost and Profitability Management deletes those default database tables. If the model data is exported to any database other than the default database, SAS Cost and Profitability Management does not delete those database tables.
4. Click **OK**.

Calculate Costs for a Model

Select **Model** ⇨ **Calculate Costs**.

The Calculate Costs dialog box appears.

See Also

- "Calculate Costs Dialog Box" on page 559
- Chapter 59, "Calculating Costs," on page 525

Change the Properties of a Model

Select **Model** ⇨ **Properties**.

The Model Properties dialog box appears.

See Also

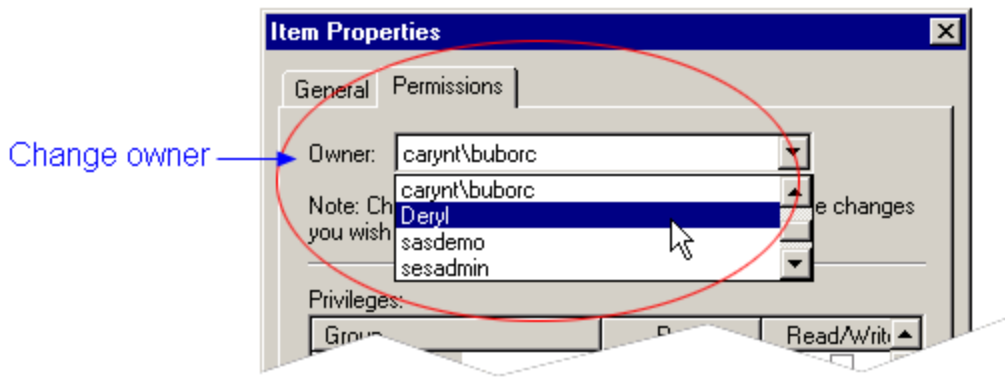
"Model Properties Dialog Box" on page 94

Change Model Owner or Give Access to a Model

Change Model Owner

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.

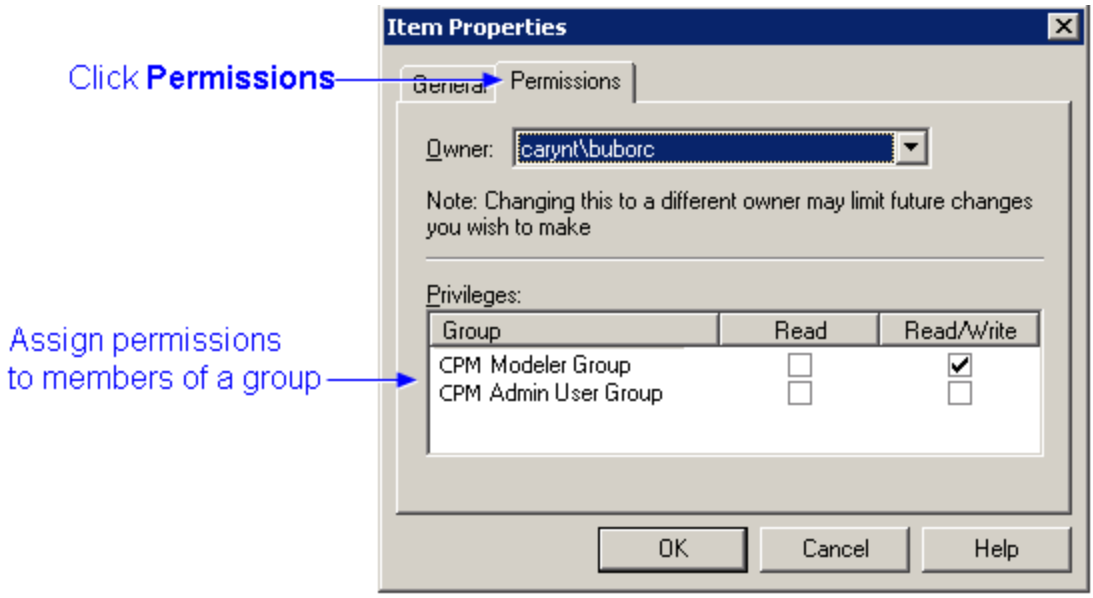
1. Select the item Workspace Manager and select **Edit** ⇨ **Item Properties**.
2. Select a new owner from the drop-down list of users.



Give Access to a Model

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:

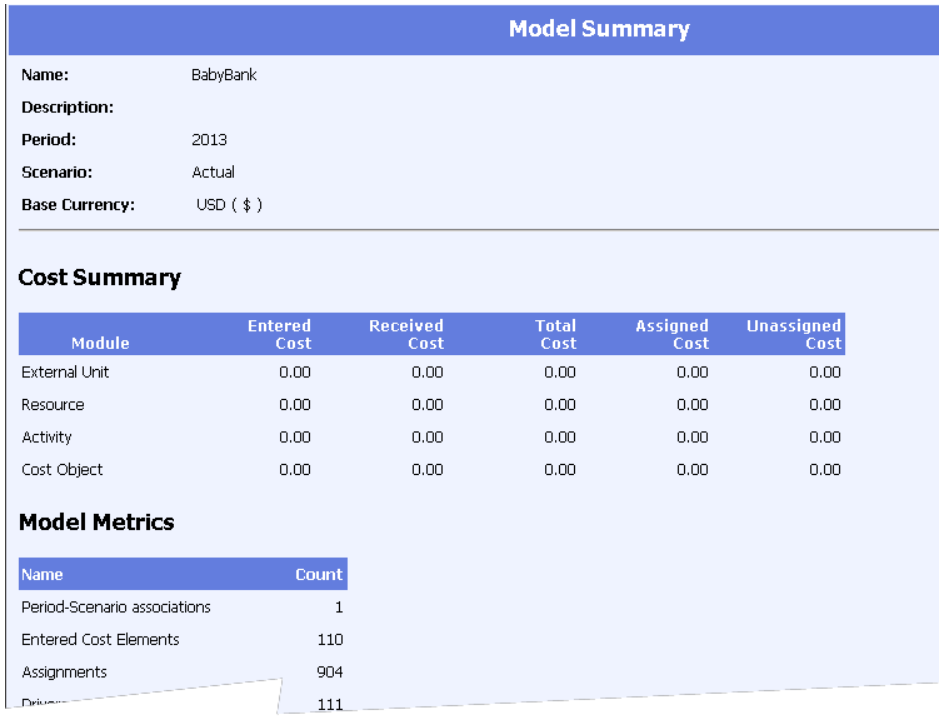
1. Select the workspace item.
2. Right-click and select **Item Properties** (or select **Edit** ⇨ **Item Properties**).
3. Click the **Permissions** tab.
4. Grant Read or Read/Write access to the desired groups. (Only groups are listed–not roles.)



Review the Summary of a Model

Select **Model** ⇨ **Summary**.

The Model Summary window appears with information concerning the model such as the number of its drivers and assignments.



Model Summary

Name: BabyBank
Description:
Period: 2013
Scenario: Actual
Base Currency: USD ($)

Cost Summary

Module	Entered Cost	Received Cost	Total Cost	Assigned Cost	Unassigned Cost
External Unit	0.00	0.00	0.00	0.00	0.00
Resource	0.00	0.00	0.00	0.00	0.00
Activity	0.00	0.00	0.00	0.00	0.00
Cost Object	0.00	0.00	0.00	0.00	0.00

Model Metrics

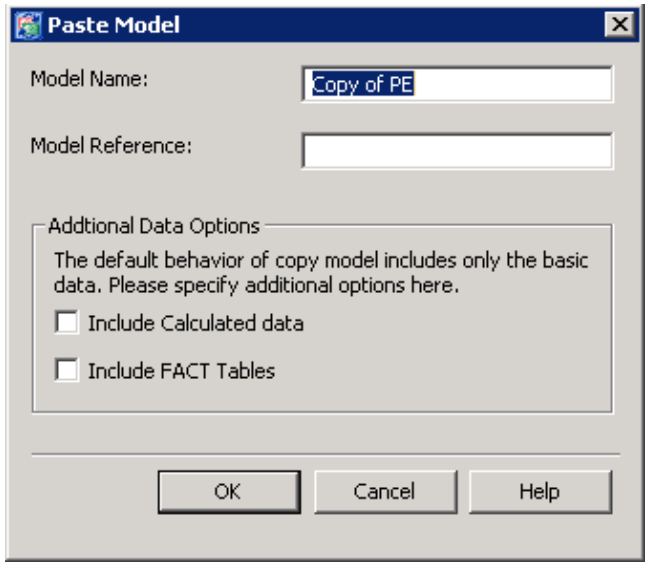
Name	Count
Period-Scenario associations	1
Entered Cost Elements	110
Assignments	904
Drivers	111

Copy a Model

Before making major changes to a model, you can make a copy of the model that you can refer to, or revert to, later on.

To copy a model:

1. Go to the Workspace Manager.
2. Select the model that you want to copy, and then select **Edit** ⇨ **Copy** (or, right-click the model and select **Copy**).
3. Select the directory in which you want to place the model. Then select **Edit** ⇨ **Paste** (or right-click the directory and select **Paste**). The Paste Model dialog box opens.



4. If the model that you want to copy was previously calculated and fact tables were generated, then you can include the calculated data and fact tables along with the model. If you choose to copy the fact table, then the calculated data is automatically included. Choose none, one, or both of the following options:

   - Include Calculated data
   - Include FACT Tables

   Whereas models that you export and re-import must be recalculated and their fact tables regenerated, models that you copy do not require those steps.

Create a Period/Scenario Association

See "Create a Period/Scenario Association" on page 169.

Change the Current Period/Scenario Association

See "Change the Current Period/Scenario Association" on page 171.

Copy Period/Scenario Data

See "Copy Model Data from One Period/Scenario to Another" on page 171.

Publish a Period/Scenario Association for a Model

Publish a period/scenario association so that it can no longer be modified and can be viewed by anyone with View Model capability. See "Publish a Period/Scenario Association for a Model" on page 174.

Note: Any model that is to be used for what-if analysis must be in the published state. See "Create a What-If Analysis" in Chapter 2 of *SAS Cost and Profitability Management: What-If Analysis*.

Export and Register Tables Containing Model Data

To export database tables that describe a model's structure and which are registered in the SAS Management Console for use with other SAS or non-SAS programs, select **Model ⇨ Export and Register Tables**.

See Also

- "Export and Register Tables Dialog Box" on page 101
- Chapter 23, "Working with SAS LASR Analytic Server," in *SAS Cost and Profitability Management: Data Administration Guide*

Part 4

Currencies

Chapter 12

Working with Currencies

Overview

A currency represents a unique monetary system that is identified by a name and by a three-letter code (currency code).

You can display a models costs in more than one currency simultaneously. However, you must first define the currencies and their exchange rates. SAS Cost and Profitability Management can display a model's costs in whatever currency you select, using the exchange rate that is associated with each currency.

When you create a model, you can select the base currency that you want to use for calculating the model costs. SAS Cost and Profitability Management saves the base currency with the model.

Note: Once you set a model's base currency, you cannot change it later.

See Also

Chapter 14, “How To: Currencies,” on page 135

Windows Currency Settings

The Windows Regional Options for your computer let you set your locale, which affects only the display of currency on your computer. A locale is a group of language settings that you want to use on your computer. You can accept the default values for the selected locale's number of digits after the decimal, or you can select a different value from a list for that locale.

Note: The base currency determines the currency symbol. Windows Regional Options do not affect the currency symbol.

Exchange Rates

An exchange rate is a multiplier that is used to convert values from one currency to another. You enter exchange rates in an exchange rates table. Because exchange rates tables are shared by all the models on the same SAS Cost and Profitability Management server, you can compare costs between models.

When you enter an exchange rate in an exchange rates table, the exchange rate's corresponding multiplier is automatically entered. After entering an exchange rate, you can change it.

You can easily add all the euro exchange rates at once. After adding the default euro exchange rates, you can change them. If a period is before 1999, the euro currency is not available.

Exchange rates can vary from one period/scenario association to another period/scenario association. To add exchange rates quickly, you can copy an exchange rates table from one period/scenario association to another period/scenario association.

The display precision that you see in an exchange rates table is determined by the Rates value in the Options dialog box.

See Also

- "CurrencyRate table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*
- "Currency Codes" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*

Selecting Currencies for an Exchange Rates Table

A currency is not the same as a locale. A locale can support multiple currencies. For example, you can use Windows to set the locale to United Kingdom. In SAS Cost and Profitability Management, you can format a unit cost column for Pound Sterling with a currency of Pound Sterling (GBP) and add a column and format it with a currency of euro.

All currencies are available for you to specify in an exchange rates table. You add the currencies that you want to use.

See Also

- "CurrencyRate table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*
- "Currency Codes" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*

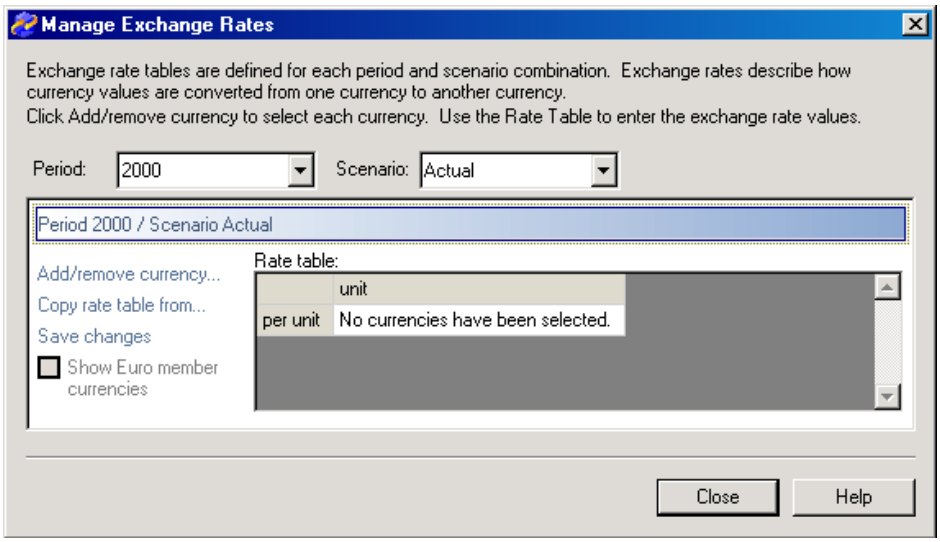
Chapter 13

Windows for Currencies

Manage Exchange Rates Dialog Box

About the Manage Exchange Rates Dialog Box

In the Manage Exchange Rates dialog box, you can set up exchange rates between currencies.



Note: You can open the Manage Exchange Rates dialog box without first opening a model.

The availability of this feature depends on your permissions.

How to Access the Manage Exchange Rates Dialog Box

Select **Tools** ⇨ **Manage Exchange Rates**.

See Also

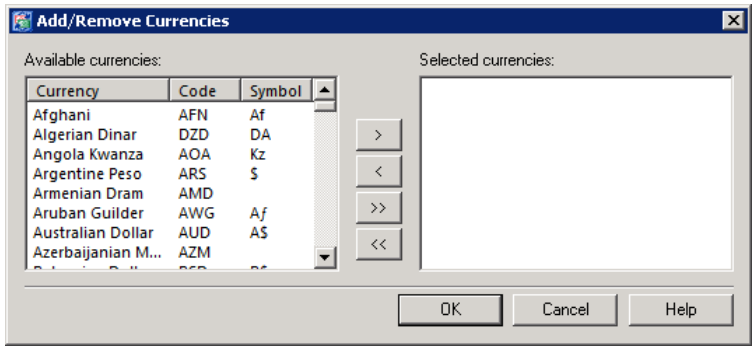
Chapter 14, "How To: Currencies," on page 135

Add/Remove Currencies Dialog Box

About the Add/Remove Currencies Dialog Box

The availability of these features depends on your permissions.

In the Add/Remove Currencies dialog box, you can manage currencies before you define exchange rates.



How to Access the Add/Remove Currencies Dialog Box

In the Manage Exchange Rates dialog box, click the **Add/remove currency** link.

See Also

"Manage Exchange Rates Dialog Box" on page 131

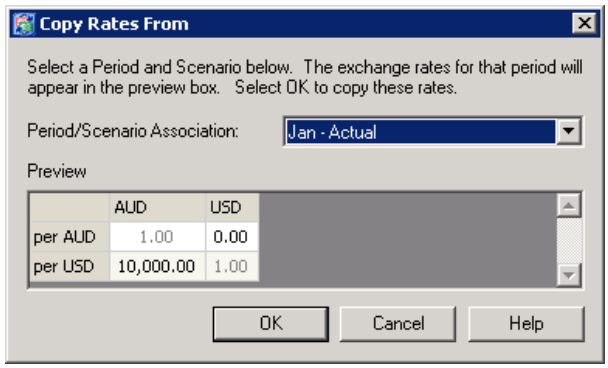
See Also

Chapter 14, "How To: Currencies," on page 135

Copy Rates From Dialog Box

About the Copy Rates From Dialog Box

In the Copy Rates From dialog box, you can copy an exchange rates table from one period/scenario association to another period/scenario association.



The availability of this feature depends on your permissions.

How to Access the Copy Rates From Dialog Box

In the Manage Exchange Rates dialog box, select **Copy rate table from**.

See Also

"Manage Exchange Rates Dialog Box" on page 131

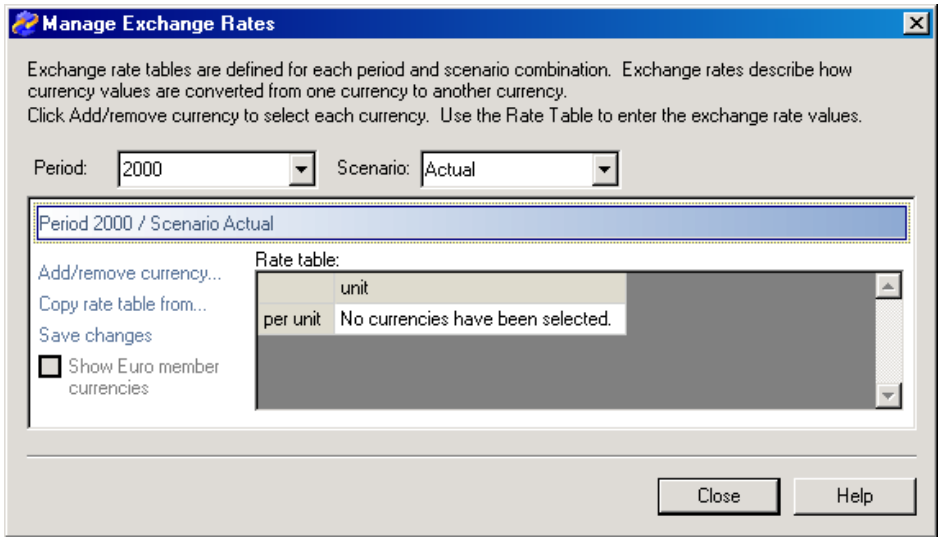
Chapter 14

How To: Currencies

Add a Currency

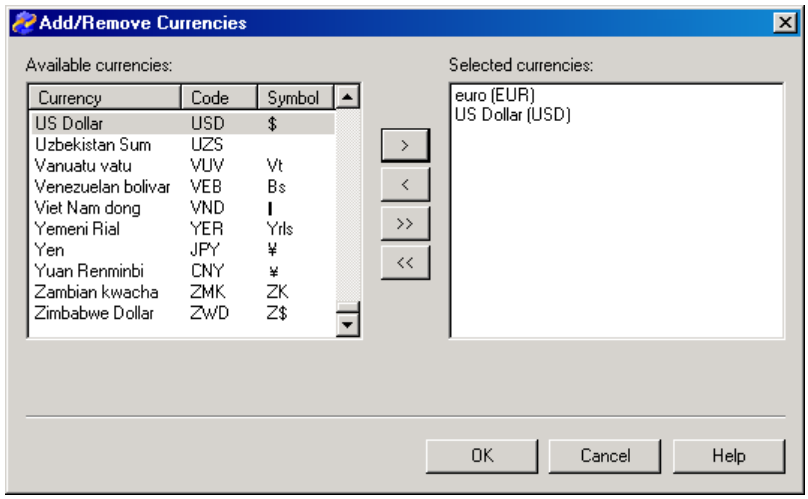
1. Select **Tools** ⇨ **Manage Exchange Rates**.

   The Manage Exchange Rates dialog box appears.



2. Click the **Add/remove currency** link.

   The Add/Remove Currencies dialog box appears.



3. In the **Available currencies** list, select one or more currencies.

   To select more than one currency, use standard Microsoft Windows selection techniques.

4. Click >.

   The currency is added to the **Selected currencies** list.

   Note: To add all currencies, click >>.

Set up Multiple Currencies

You can follow these steps to set up currencies for all the models on a SAS Cost and Profitability Management server.

1. Add currencies using the Manage Exchange Rates dialog box. See "Add a Currency" on page 135.
2. Set up exchange rates in an exchange rates table. See "Set up an Exchange Rates Table" on page 137.
3. (Optional) Add columns and change the column formats to one of the added currencies. See "Currency Formatting" on page 345.

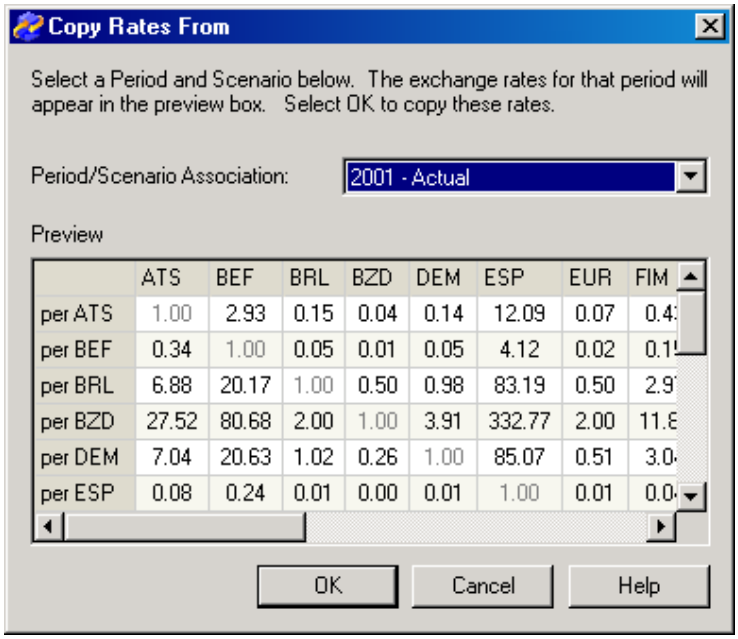
Copy an Exchange Rates Table

1. Select **Tools** ⇨ **Manage Exchange Rates**.

   The Manage Exchange Rates dialog box appears.

2. Click the **Copy rate table from** link.

   The Copy Rates From dialog box appears.

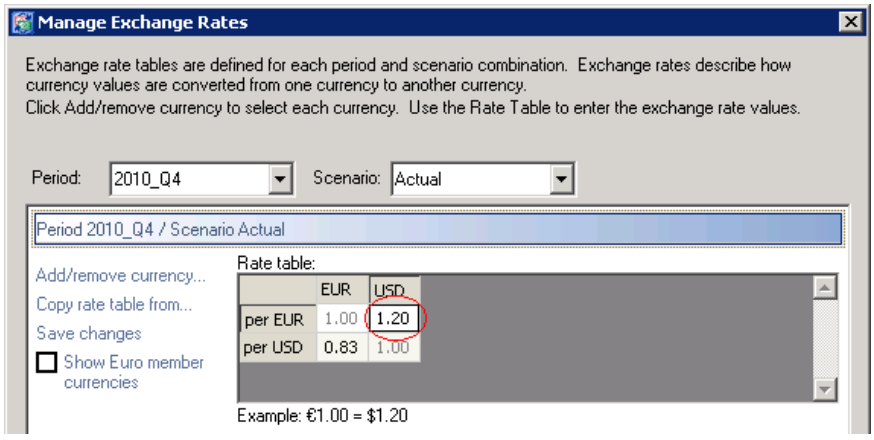


3. Select the **Period/Scenario** association that has the exchange rates table that you want to copy.

Set up an Exchange Rates Table

1. In the Manage Exchange Rates dialog, Select a Period and Scenario.
2. (Optional) Click the **Add/remove currency** link.

   The Add/Remove Currencies dialog box appears.
3. In the Rate table, click on the intersection between two currencies, and type an exchange rate.



4. Repeat the previous step for each exchange rate.
5. (Optional) Add euro exchange rates.

See Also

- "Add/Remove Currencies Dialog Box" on page 132
- "Add Euro Exchange Rates" on page 138

Add Euro Exchange Rates

1. In the Manage Exchange Rates dialog box, click **Save changes**.

   Before you can add euro exchange rates, you must save any changes to the exchange rates table.

2. Add the currency named euro:

   a. Click **Add/remove currency**.

      The Add/Remove Currencies dialog box appears.

   b. Select the currency named euro.

   c. Select the **Show Euro member currencies** option.

See Also

- "Manage Exchange Rates Dialog Box" on page 131
- "Add/Remove Currencies Dialog Box" on page 132

Copy an Exchange Rates Table

In the Manage Exchange Rates dialog box, click **Copy rate table from**

The Copy Rates From dialog box appears.

See Also

- "Manage Exchange Rates Dialog Box" on page 131
- "Copy Rates From Dialog Box" on page 133

Part 5

Periods and Scenarios

Chapter 15

Working with Periods and Scenarios

About Periods and Scenarios

SAS Cost and Profitability Management stores information for specific amounts of time (periods) and for specific sets of data (scenarios).

Note: The structure of a model is based on periods. Therefore, if you change the period/scenario association for a model, the model structure might change. In fact, the entire model structure might disappear if the period/scenario association contains no structure data.

Periods

A period can represent any number of days: a month, a quarter, a year, and so on. For example, if your organization chooses to enter data each month, then the marketing payroll cost is the amount of payroll for one month.

Period Levels

You can create a hierarchy of periods, such as FY2003 > Q1 > January. By default, each level is given a name such as Period L1. However, these names are not descriptive when you generate cubes. So, you can name a period level.

Scenarios

Scenarios are used for managing different variations of data within a period. A scenario can be any set of data: actual data, budget data, aggressive plan data, conservative plan data, and so on. The default scenarios are Actual and Budget.

Some scenarios can be agreed upon and can be set up before people start to use SAS Cost and Profitability Management. Other scenarios can be created when they are needed. That is, not all scenarios can be anticipated by an organization. Your organization might want to develop guidelines for naming and organizing scenarios in a hierarchy so that the list of scenarios on a server does not become unmanageable.

Scenario Levels

You can create a hierarchy of scenarios, such as Budget > Aggressive. By default, each level is given a name such as Scenario L1. However, these names are not descriptive when you generate cubes. So, you can name a scenario level.

Setting up Periods and Scenarios

Periods and scenarios are shared by all the models on the same server. Therefore, before people start to use SAS Cost and Profitability Management, your organization might want to set up a period hierarchy, and you might want to publish guidelines for defining scenarios.

On the OLAP Analyzer view, SAS Cost and Profitability Management automatically rolls up values for each period and scenario that is in the hierarchy above the hierarchy level that contains data.

See Also

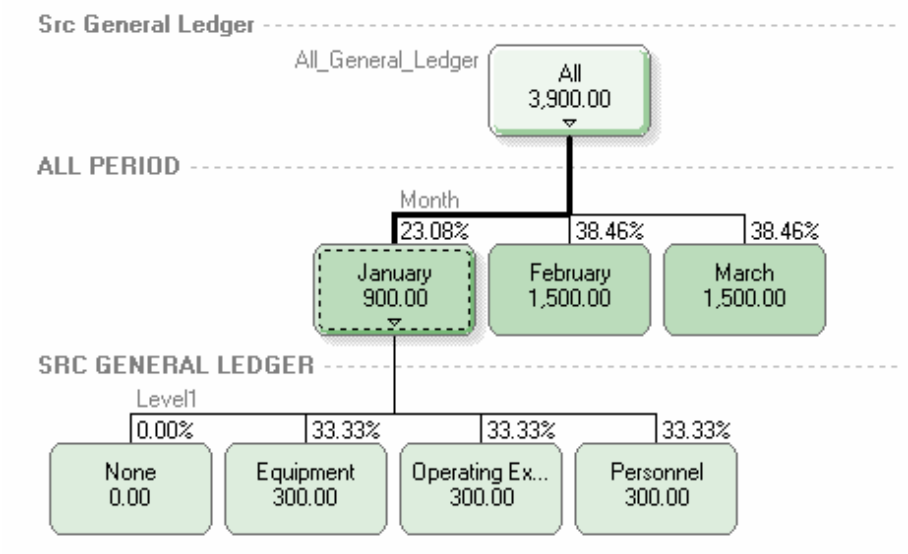
Chapter 17, “How To: Periods and Scenarios,” on page 153

Periods and Scenarios in OLAP Analysis

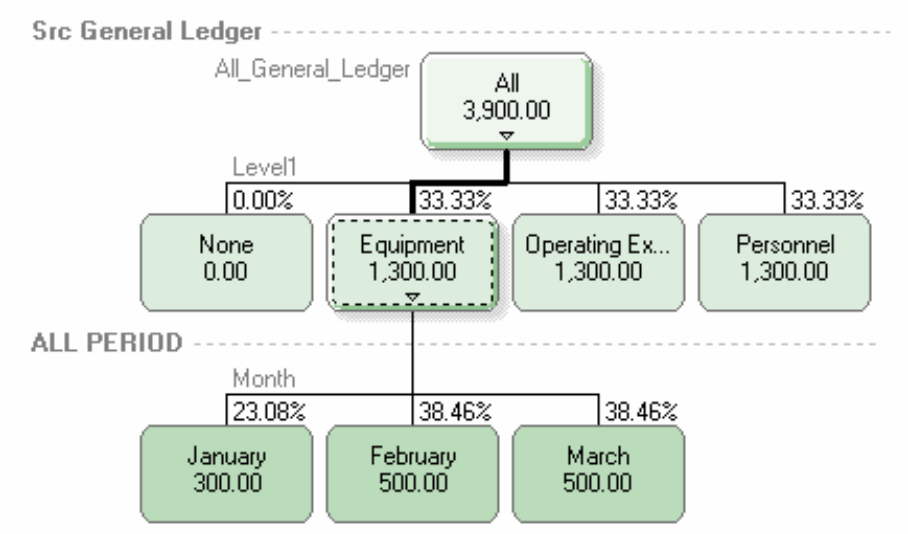
Periods and scenarios are dimensions and can be used by business users for OLAP analysis. Periods and scenarios can aggregate data or separate data.

For example, suppose that a model contains actual cost data for the first three months of 2005 in the following period/scenario associations: Jan 2005/ACTUAL, Feb 2005/ACTUAL, and Mar 2005/ACTUAL.

When the OLAP cubes are generated and all period/scenario associations are included in an OLAP cube, business users can analyze the data in several ways on the OLAP view. For example, business users can aggregate the costs by month and then by category, as shown in the following figure of the Cube Explorer View:



Or, business users can aggregate the costs by category and then by month, as shown in the following figure:



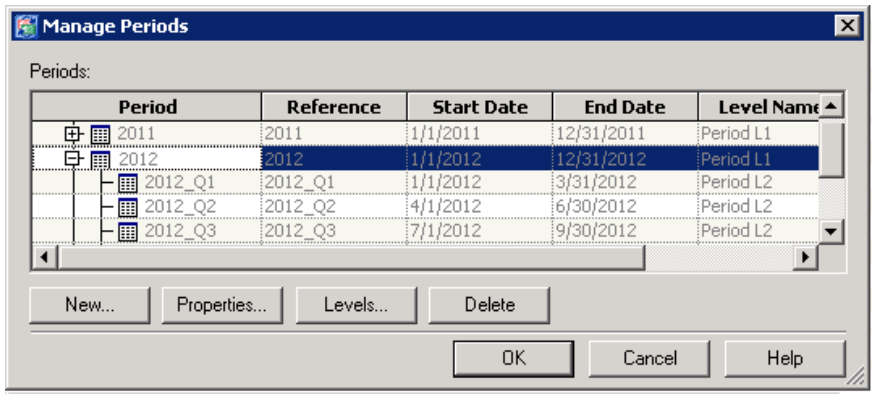
Using combinations of periods and scenarios, business users can achieve many of their analysis goals.

Chapter 16

Windows for Periods and Scenarios

Manage Periods Dialog Box

About the Manage Periods Dialog Box



In the Manage Periods dialog box, you can:

- "Create a Period" on page 153
- "Delete a Period" on page 155
- "Manage Period Level Names" on page 155

The availability of this feature depends on your permissions.

Note: You can use the Manage Periods dialog box without first opening a model.

TIP If a row is highlighted, the dates for the period are outside the date range for the parent period. This situation can occur when data is imported. Correct the dates for the highlighted period.

TIP You can widen the columns by dragging the column separators.

Use the Manage Periods dialog box to perform the following tasks:

How to Access the Manage Periods Dialog Box

Do one of the following:

- Select **Tools** ⇨ **Manage Periods** .
- In the Associate Periods and Scenarios dialog box, click **New Period**.

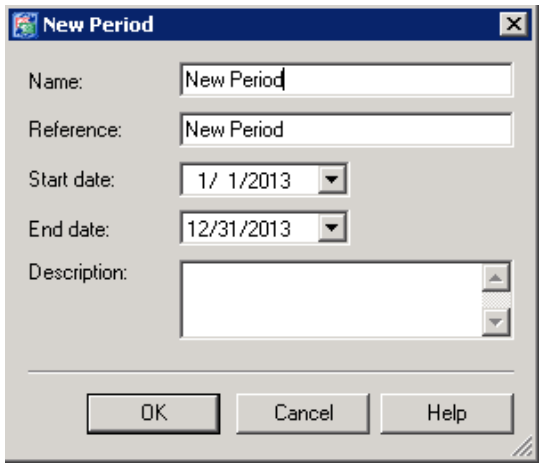
See Also

- Chapter 15, "Working with Periods and Scenarios," on page 141
- "Create a Period/Scenario Association" on page 169
- "Change the Current Period/Scenario Association" on page 171

New Period Dialog Box

About the New Period Dialog Box

In the New Period dialog box, you can name a new period and you can specify other information about the period.



Note: The availability of these features depends on your permissions.

How to Access the New Period Dialog Box

Do one of the following:

- In the Manage Periods dialog box, click **New**.
- In the Associate Periods and Scenarios dialog box, click **New Period**.

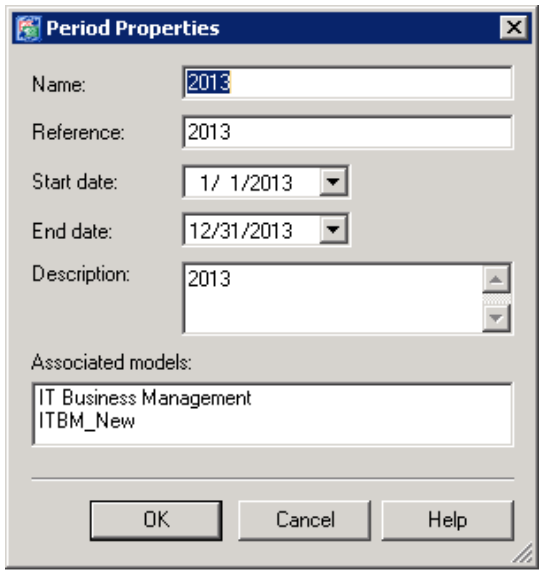
See Also

- Chapter 15, “Working with Periods and Scenarios,” on page 141
- “Create a Period” on page 153
- “Associate Periods and Scenarios Dialog Box” on page 164

Period Properties Dialog Box

About the Period Properties Dialog Box

In the Period Properties dialog box, you can review or change a period's properties.



Note: The availability of these features depends on your permissions.

How to Access the Period Properties Dialog Box

In the Manage Periods dialog box, select a period and click **Properties**.

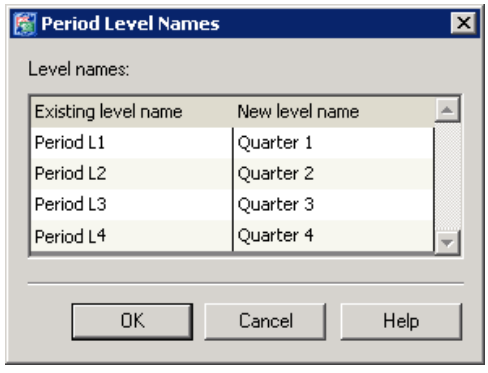
See Also

Chapter 15, "Working with Periods and Scenarios," on page 141

Period Level Names Dialog Box

About the Period Level Names Dialog Box

In the Period Level Names dialog box, you can name period levels.



Note: The availability of these features depends on your permissions.

How to Access the Period Level Names Dialog Box

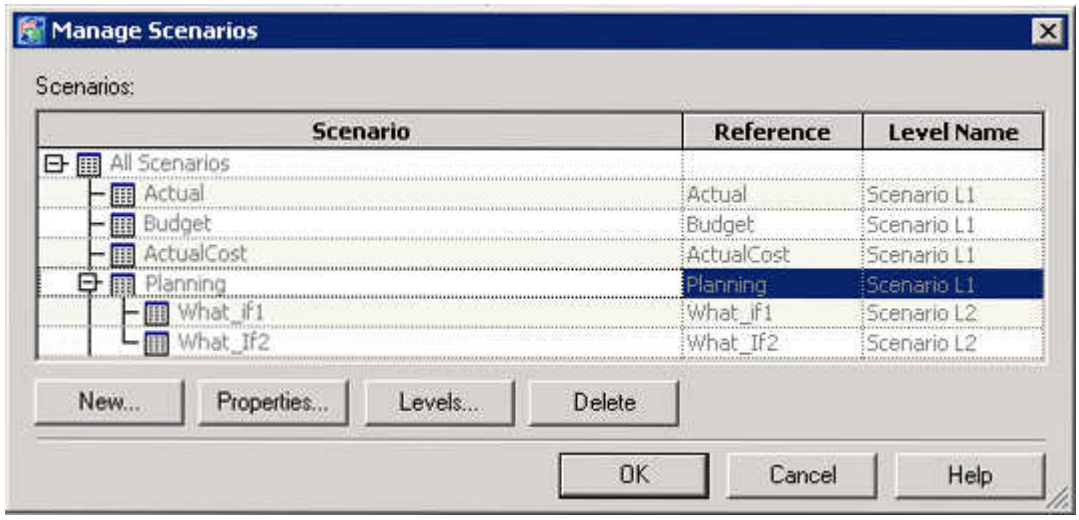
In the Manage Periods dialog box, select a period and click **Levels**.

See Also

"Manage Period Level Names" on page 155

Manage Scenarios Dialog Box

About the Manage Scenarios Dialog Box



In the Manage Scenarios dialog box, you can:

- "Create a Scenario" on page 156
- "Delete a Scenario" on page 157
- "Manage Scenario Level Names" on page 157

Note: You can perform the following tasks without first opening a model.

Note: The availability of these features depends on your permissions.

How to Access the Manage Scenarios Dialog Box

Do one of the following:

- Select **Tools** ⇨ **Manage Scenarios**.
- In the Associate Periods and Scenarios dialog box, click **New Scenario**.

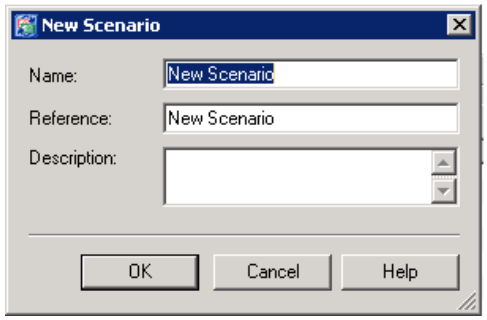
See Also

- Chapter 15, “Working with Periods and Scenarios,” on page 141
- “Create a Period/Scenario Association” on page 169
- “Change the Current Period/Scenario Association” on page 171

New Scenario Dialog Box

About the New Scenario Dialog Box

In the New Scenario dialog box, you can name a new scenario and you can specify other information about the scenario.



Note: The availability of these features depends on your permissions.

How to Access the New Scenario Dialog Box

Do one of the following:

- In the Manage Scenarios dialog box, click **New**.
- In the Associate Periods and Scenarios dialog box, click **New**.

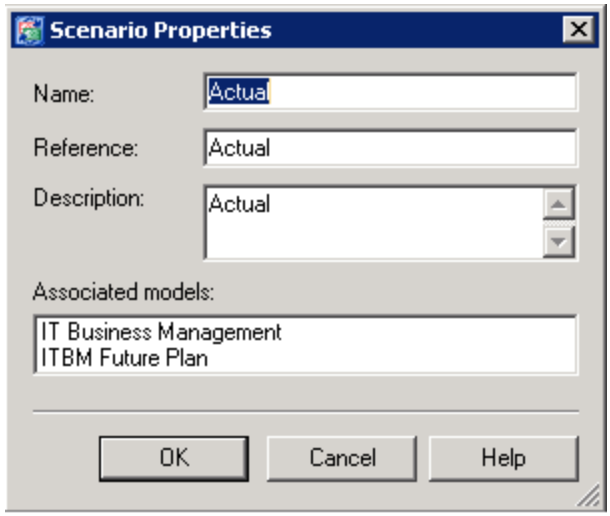
See Also

- “Create a Scenario” on page 156
- Chapter 15, “Working with Periods and Scenarios,” on page 141

Scenario Properties Dialog Box

About the Scenario Properties Dialog Box

In the Scenario Properties dialog box, you can review or change a scenario's properties.



Note: The availability of these features depends on your permissions.

How to Access the Scenario Properties Dialog Box

In the Manage Scenarios dialog box, select a scenario and click **Properties**.

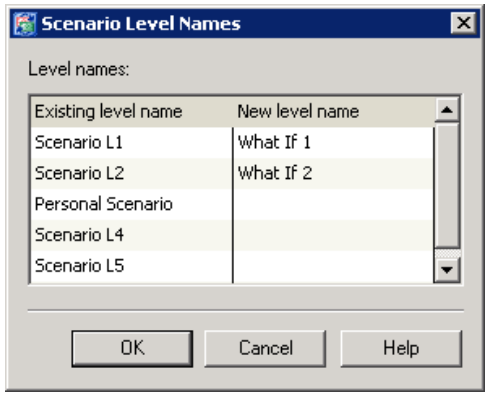
See Also

- Chapter 15, "Working with Periods and Scenarios," on page 141
- Chapter 17, "How To: Periods and Scenarios," on page 153

Scenario Level Names Dialog Box

About the Scenario Level Names Dialog Box

In the Scenario Level Names dialog box, you can name scenario levels.



The availability of these features depends on your permissions.

How to Access the Scenario Level Names Dialog Box

From the Manage Scenarios dialog box, click **Levels**.

See Also

"Manage Scenario Level Names" on page 157

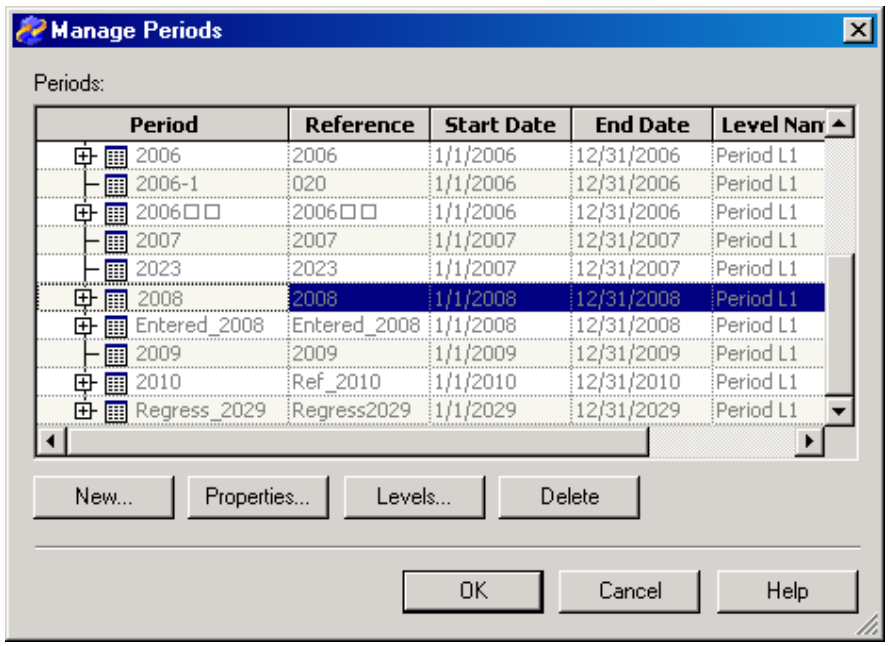
Chapter 17

How To: Periods and Scenarios

Create a Period

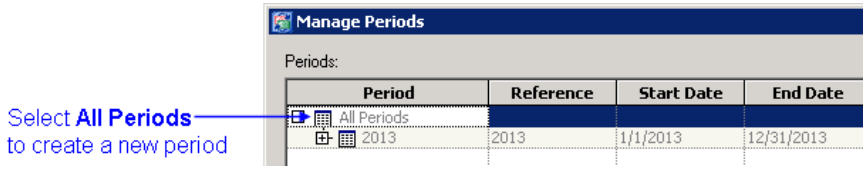
1. Select **Tools** ⇨ **Manage Periods**.

   The Manage Periods dialog box appears.

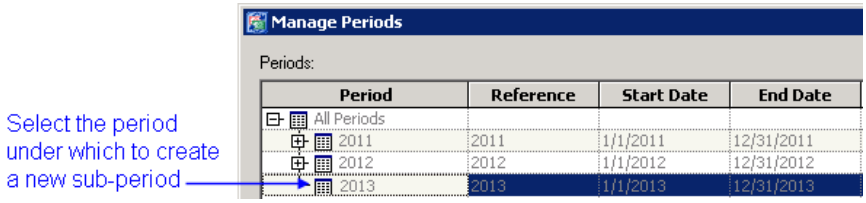


2. Do one of the following:

   - To create a brand new period, select **All Periods**.

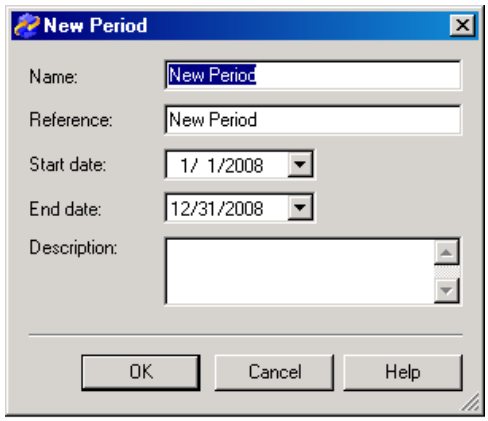


- To create a sub-period under an existing period, select the period under which to create the new period.



3. Click **New**.

   The New Period dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

5. Type the **Reference**.

   A default reference is created from the period's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74,

6. Type the **Start date** and the **End date**, or select dates from the drop-down calendar.

   When you define a period, you specify its start date and its end date. These dates are useful to people in your organization and do not affect the data.

   - The duration of the period must be at least one day.
   - The start date can be no earlier than January 1, 1980.
   - The end date can be no later than December 31, 2029.

   A period must meet these criteria:

   Periods that are at the same level in the hierarchy must meet these criteria:

   - The periods must be sequential, with no overlapping dates.
   - Each period must be within the date range of the period that is at the next higher level.

7. Type the **Description**.

See Also

"New Period Dialog Box" on page 147

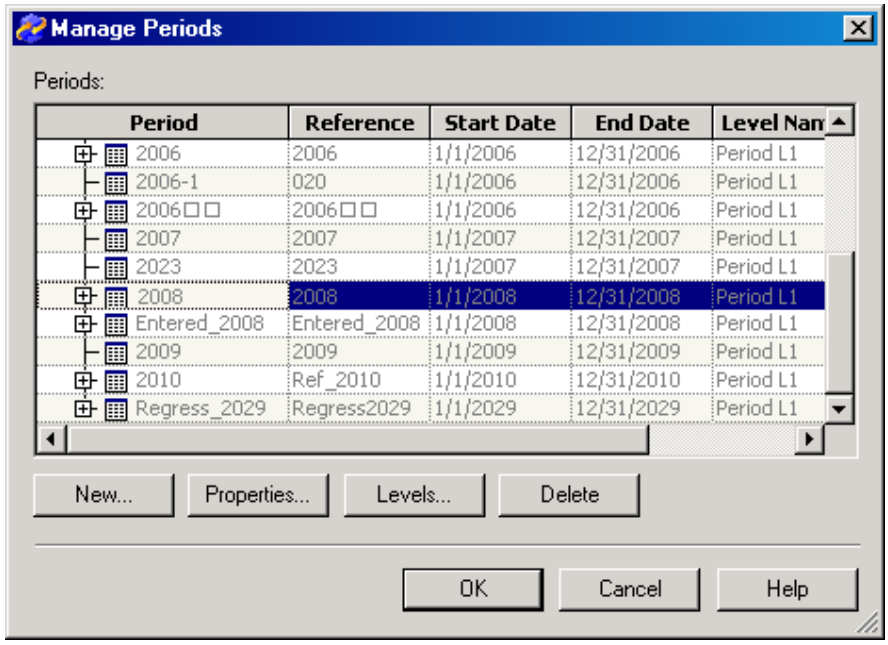
Delete a Period

1. If the period is part of a period/scenario association in a model, delete the period/scenario association from that model.
2. Select the period from the **Period** column of the Manage Periods dialog box..
3. Click **Delete**.

Manage Period Level Names

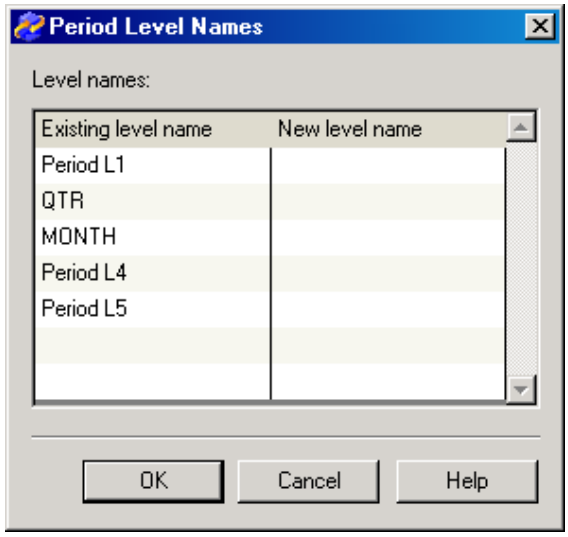
1. Select **Tools** ⇨ **Manage Periods**.

   The Manage Periods dialog box appears.



2. From the **Period** column, select a period.
3. Click **Levels**.

   The Period Level Names dialog box appears.



4. In the **New level name** column, double-click in a row and type a name.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

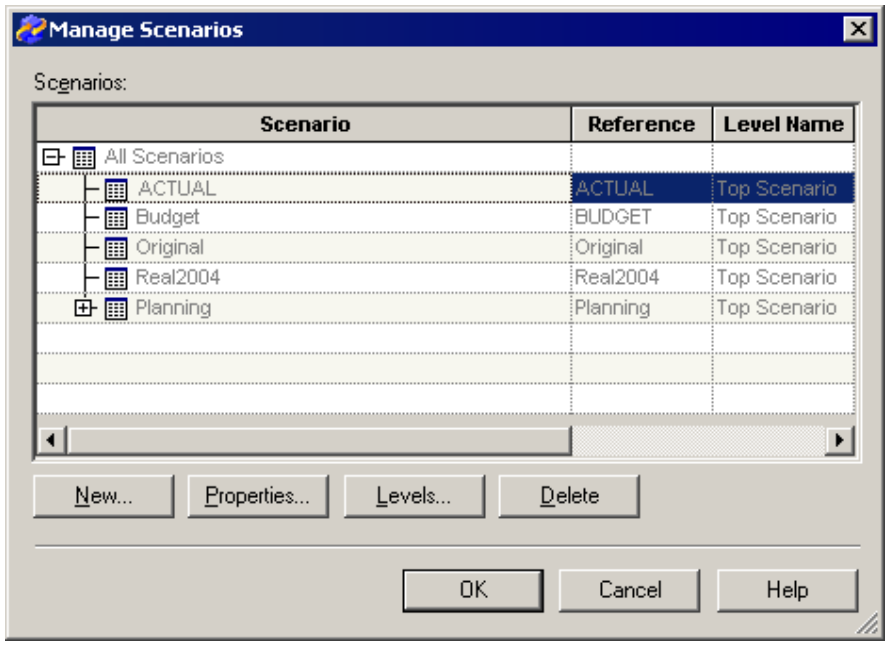
See Also

"Period Level Names Dialog Box" on page 148

Create a Scenario

1. Select **Tools** ⇨ **Manage Scenarios**.

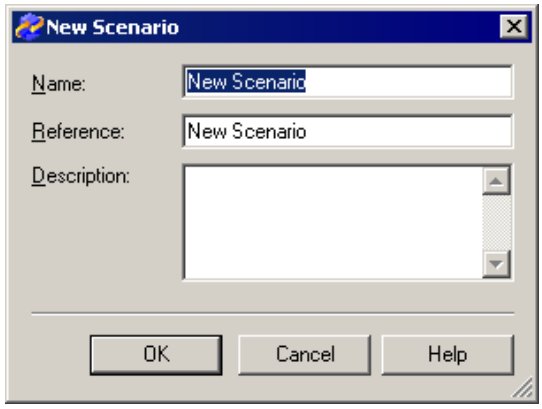
   The Manage Scenarios dialog box appears.



2. From the **Scenario** column, select a scenario under which to create the new scenario.

3. Click **New**.

   The New Scenario dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. See "Naming Conventions" on page 69..

5. Type the **Reference**.

   A default reference is created from the scenario's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74,

6. Type the **Description**.

See Also

- Chapter 15, "Working with Periods and Scenarios," on page 141
- "New Scenario Dialog Box" on page 150

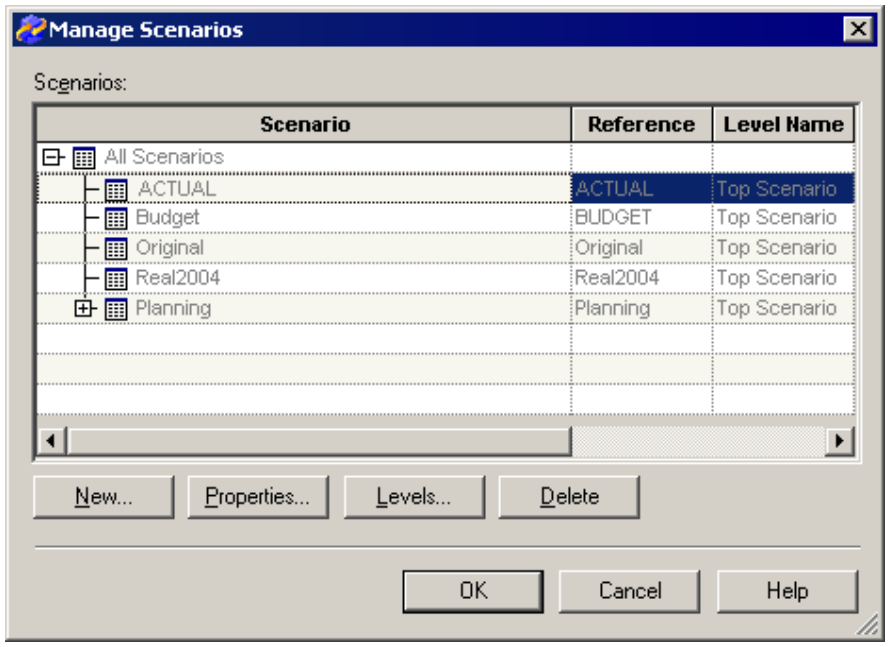
Delete a Scenario

1. If the scenario is part of a period/scenario association in a model, delete the period/scenario association from that model.
2. Select the scenario from the **Scenario** column of the Manage Scenarios dialog box..
3. Click **Delete**.

Manage Scenario Level Names

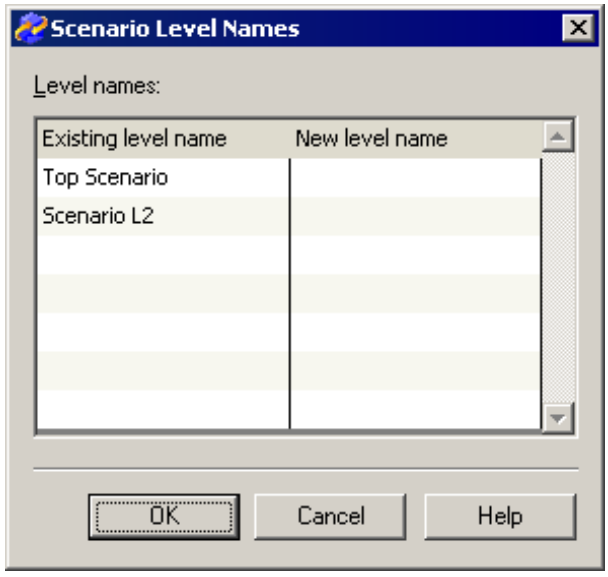
1. Select **Tools** ⇨ **Manage Scenarios**.

   The Manage Scenarios dialog box appears.



2. From the **Scenario** column, select a scenario.
3. Click **Levels**.

   The Scenario Level Names dialog box appears.



4. In the **New level name** column, double-click in a row and type a name.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

See Also

- "Scenario Level Names Dialog Box" on page 151
- "Scenario Level Names Dialog Box" on page 151

Part 6

Period and Scenario Associations

Chapter 18

Working with Period and Scenario Associations

About Period/Scenario Associations

A period/scenario association identifies a specific period, such as Q1, and a specific scenario, such as Budget. All model data must reside in a period and must pertain to a scenario. An association represents a period-scenario pair.

Depending on your permissions, the Period/Scenario menu lists in SAS Cost and Profitability Management may show only associations that have been published.

Calculation Status of Associations

The calculation status of a period/scenario association indicates whether the data in the association is complete and accurate. A calculation status can be either Not calculated or Calculated.

Status of Associations

A period/scenario association can be either Not published or Published.

By default, a period/scenario association is Not published. The data is not ready for the general viewer to see because the data is in the process of being entered.

When the data for a period/scenario association is entered and calculated, the association is published. This enables the general user to see and analyze the data in that period/scenario association.

Note: As long as a period/scenario association has a status of Published, the data within the association cannot be edited.

Typically, an association is published at the end of a period, such as at the end of a fiscal quarter.

Copying Data from One Association to Another

You can copy data from one period/scenario association to another period/scenario association. The ability to copy data enables you to propagate association data to similar periods and/or to similar period/scenario associations.

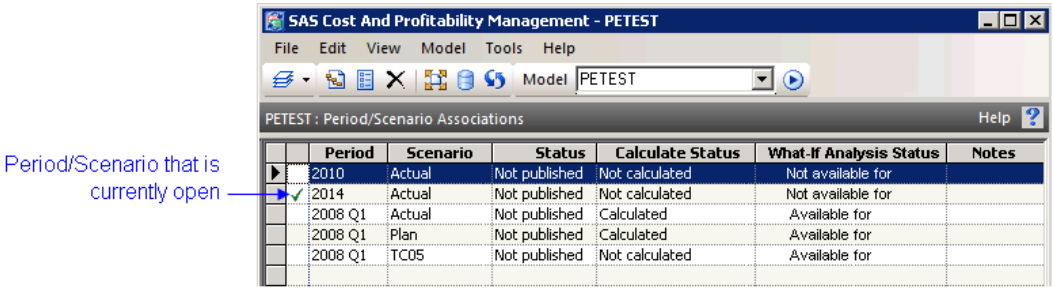
See "Copy Model Data from One Period/Scenario to Another" on page 171.

Chapter 19

Windows for Period and Scenario Associations

Period and Scenario Associations View

About the Period and Scenario Associations View



The availability of this feature depends on your permissions.

On the Period and Scenario Associations view, you can create new period/scenario associations, publish period/scenario associations, and generally manage the period/scenario associations for a model.

The period/scenario association that is open is indicated by a check in the first column:

Status
Published or Not published. See "Publish a Period/Scenario Association for a Model" on page 174.

Calculate Status
Calculated or Not calculated.

What-If Analysis Status
A model's period/scenario is available for What-If Analysis if and only if:

- the model is published
- the model is calculated
- the model has elements marked for What-If Analysis. See Chapter 71, "Marking Items for What If Analysis," on page 605.

How to Access the Period and Scenario Associations View

Open a model and select **Model ⇨ Period and Scenario Associations**.

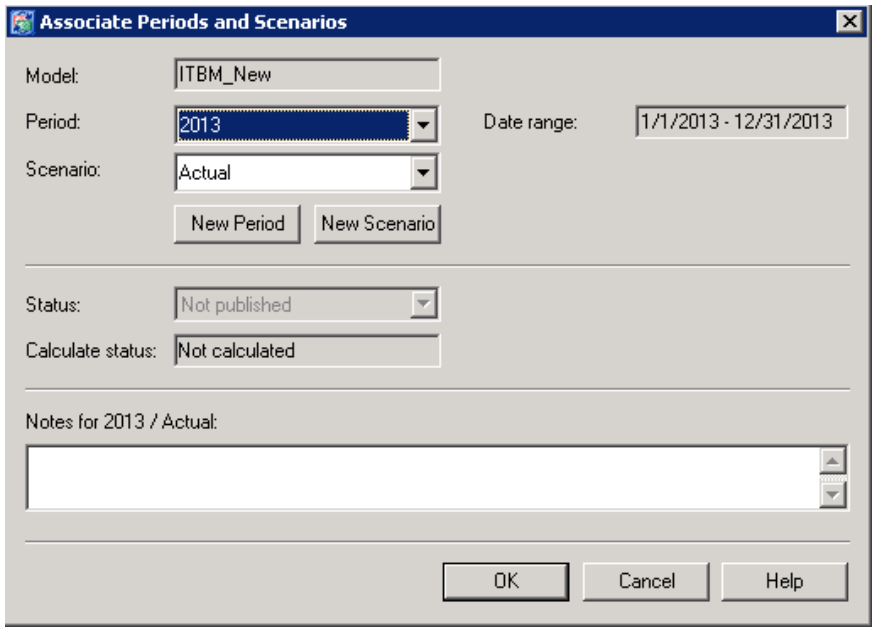
See Also

- Chapter 15, "Working with Periods and Scenarios," on page 141
- "Associate Periods and Scenarios Dialog Box" on page 164
- "Create a Period/Scenario Association" on page 169
- "Change the Current Period/Scenario Association" on page 171
- "Copy Model Data from One Period/Scenario to Another" on page 171
- "Publish a Period/Scenario Association for a Model" on page 174

Associate Periods and Scenarios Dialog Box

About the Associate Periods and Scenarios Dialog Box

In the Associate Periods and Scenarios dialog box, you can associate periods and scenarios.

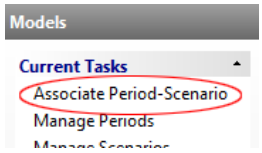


Some features may not be available depending on your permissions.

How to Access the Associate Periods and Scenarios Dialog Box

Do either of the following:

- From the task list of the Period/Scenario Associations view, click **Associate Period–Scenario** .



- On the Period/Scenario Associations view, select a period/scenario association, and then select **Edit** ⇨ **New Association**.

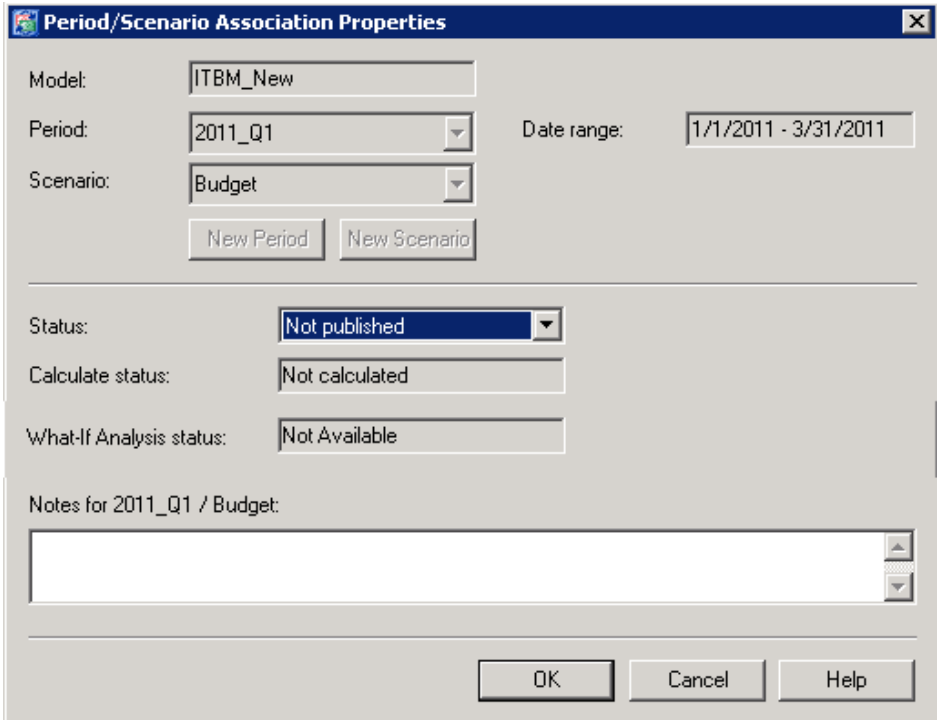
See Also

- Chapter 15, “Working with Periods and Scenarios,” on page 141
- “Create a Period/Scenario Association” on page 169
- “Change the Current Period/Scenario Association” on page 171

Period/Scenario Association Properties Dialog Box

About the Period/Scenario Association Properties Dialog Box

In the Period/Scenario Association Properties dialog box, you mark a period/scenario association as published or not published, and you can add notes to the association..



Note: The availability of these features depends on your permissions.

How to Access the Period/Scenario Association Properties Dialog Box

On the Period and scenario associations view, select an association and select **Edit ⇨ Item Properties.**

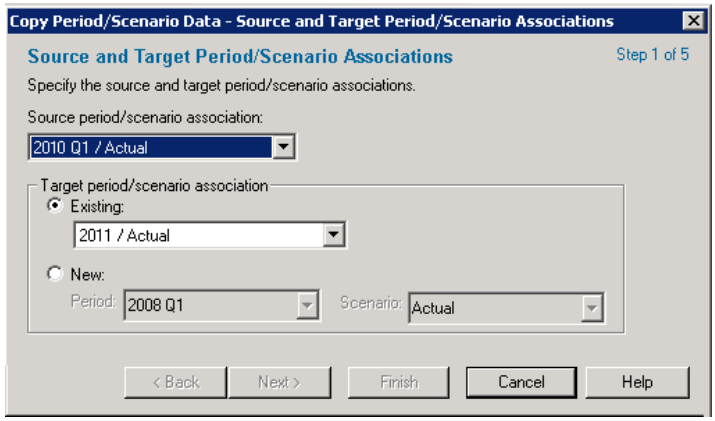
See Also

"Period and Scenario Associations View" on page 163

Copy Period/Scenario Data Dialog Box

About the Copy Period/Scenario Data Dialog Box

Use the Copy Period/Scenario Data dialog box to copy model data from one Period/Scenario association to another.



For information, see "Copy Model Data from One Period/Scenario to Another" on page 171.

How to Access the Copy Period/Scenario Data Dialog Box

1. Acquire a write lock on a model and open the model.
2. Select **Model** ⇨ **Copy Period/Scenario Data**.

   The Copy Period/Scenario Data dialog box appears.

See Also

- "Copy Model Data from One Period/Scenario to Another" on page 171
- "Create a Period/Scenario Association" on page 169
- "Change the Current Period/Scenario Association" on page 171

Chapter 20

How To: Period and Scenario Associations

Create a Period/Scenario Association

Introduction

You can create a period/scenario association in two ways:

- while creating a new model
- while editing an existing model

See Also

- Chapter 15, “Working with Periods and Scenarios,” on page 141
- “Associate Periods and Scenarios Dialog Box” on page 164
- “Change the Current Period/Scenario Association” on page 171

While Creating a New Model

When you create a new model, you must specify an initial period and an initial scenario for the model. The pair that you select is the initial period/scenario association for the model. See “Step 2: Specify Initial Period and Scenario” on page 105.

While Editing an Existing Model

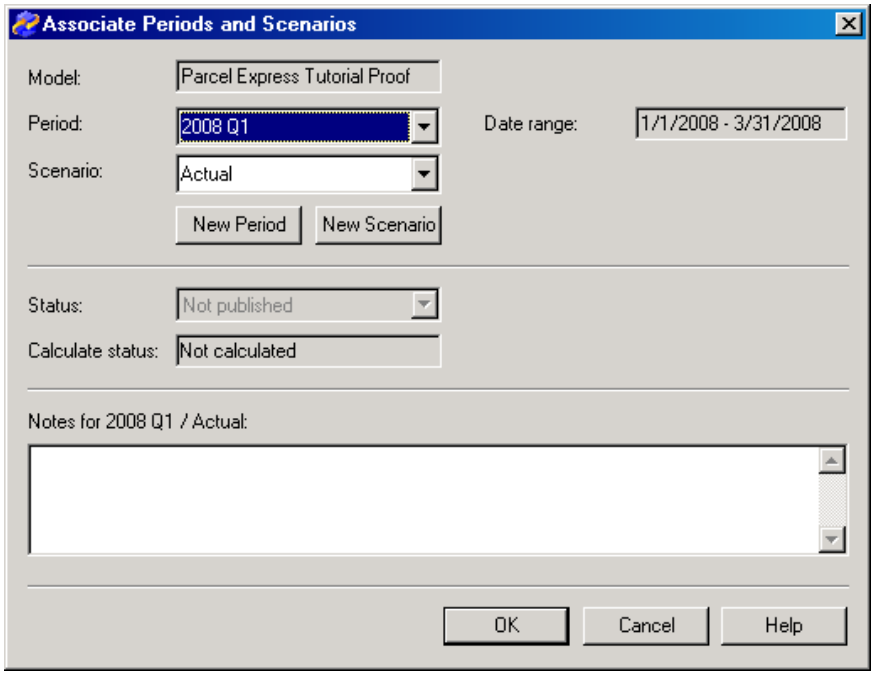
Do the following to create a new period/scenario association for the model that is currently open:

1. Select **Tools** ⇨ **Period and Scenario Associations**.

   The Period and scenario associations page appears.

2. Select **Edit** ⇨ **New Association**.

   The Associate Periods and Scenarios dialog box appears.



3. Select the **Period**. To create a new period, do the following:

   a. Click **New Period**.

      The Manage Periods dialog box appears. Create a new period as described in the Create a Period section.

   b. Select the **Period**.

4. Select the **Scenario**. To create a new scenario, do the following:

   a. Click **New Scenario**.

      The Manage Scenarios dialog box appears. Create the scenario as described in the Create a Scenario section.

   b. Select the **Scenario**.

5. Type **Notes** for the association.

Change the Current Period/Scenario Association

The current period/scenario association is the period and scenario to which costs are assigned when you add costs to a model.

To change the current period/scenario association, select from the period/scenario drop-down list on the Resource, Activity, Cost Object, or External Object view of an open model and click the arrow button.

The current period is indicated by a check mark on the Period and Scenario Association view.

To open the Period and Scenario Association view, open a model and select **Model ⇨ Period and Scenario Associations view**.

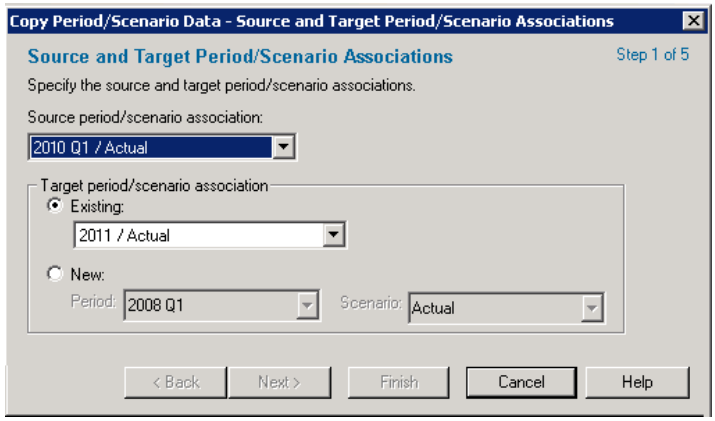
See Also

- Chapter 15, “Working with Periods and Scenarios,” on page 141
- “Associate Periods and Scenarios Dialog Box” on page 164
- “Period and Scenario Associations View” on page 163

Copy Model Data from One Period/Scenario to Another

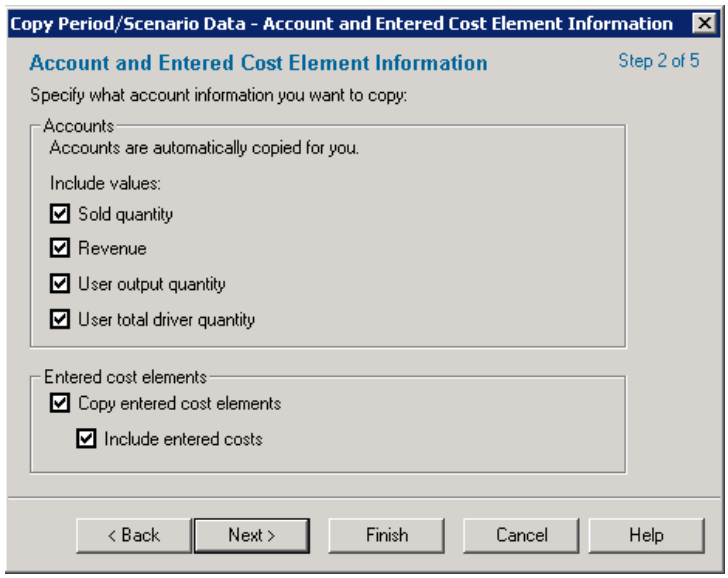
1. Acquire a write lock on a model and open the model.
2. Select **Model ⇨ Copy Period/Scenario Data**.

   The Copy Period/Scenario Data dialog box appears.



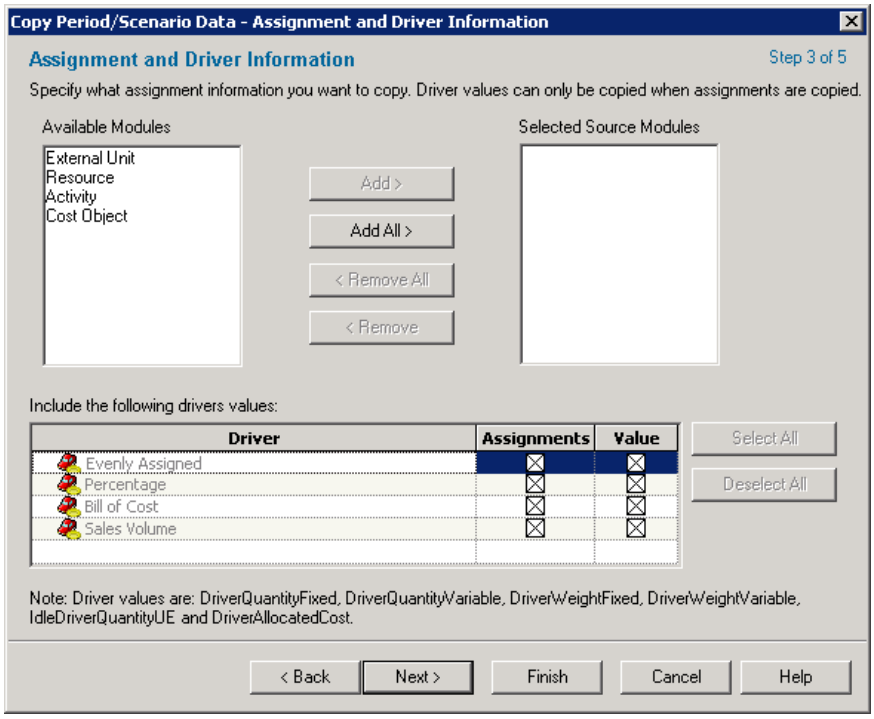
3. Select a source **Period/scenario** association.
4. For the target period/scenario association, do one of the following:
   a. Select the **Existing period/scenario association** option, and then select a period/scenario association from the drop-down list of period/scenario associations already belonging to the model.
   b. Select the **New period/scenario association** option, and then select a target period/scenario association to be added to the model.

   Then, click **Next**.
5. Specify what values are to be copied for accounts and whether the values for entered cost element are to be copied.



   Then, click **Next**.
6. Specify what assignment information you want to copy.

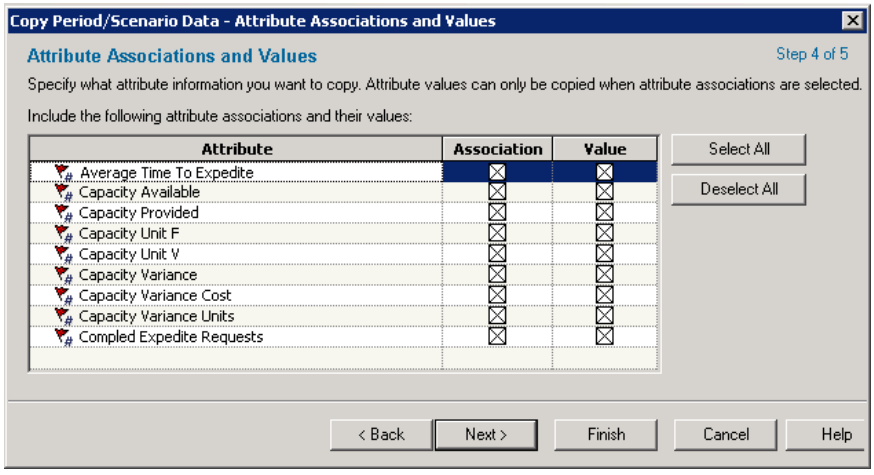
   You can select for which modules (as source modules) assignments are copied. And, for selected modules, you can select for which drivers information is to be copied.



Then, click **Next**.

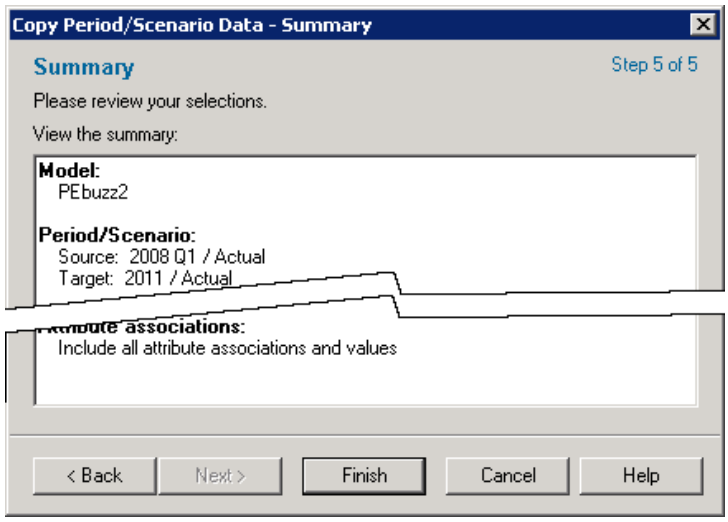
7. Specify what attribute information you want to copy.

   You can select which attribute associations are copied along with their attribute values.



Then, click **Next**.

8. Review your selections, and then click **Finish**.



See Also

- "Copy Period/Scenario Data Dialog Box" on page 167
- "Create a Period/Scenario Association" on page 169
- "Change the Current Period/Scenario Association" on page 171

Publish a Period/Scenario Association for a Model

Overview

Publishing a period/scenario association:

- makes it available for viewing by anyone with View Model capability
- is necessary for the period/scenario association to be used for what-if analysis. See "Create a What-If Analysis" in Chapter 2 of *SAS Cost and Profitability Management: What-If Analysis*.

If a period/scenario association is published, then you can not:

- modify any periodic element belonging to that period/scenario association.

If a period/scenario association is published, then you can:

- modify periodic elements belonging to other period/scenario associations in the model
- modify any structural element belonging to the model.

See "Periodic and Structural Elements" on page 79.

How to Publish

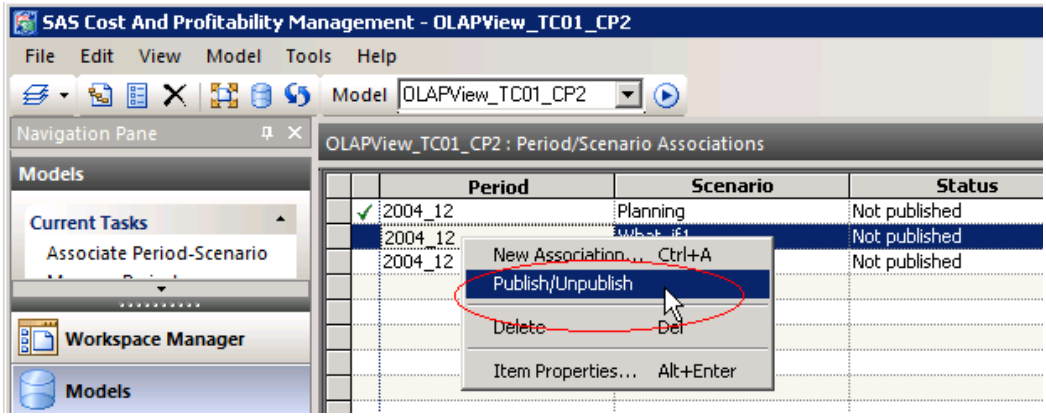
To publish a period/scenario association:

1. Select **Model** ⇨ **Period and Scenario Associations**.

The Period and scenario associations page appears.

2. Select a period/scenario association.
3. Select **Edit** ⇨ **Publish/Unpublish**.

   Or, right-click a period/scenario association, and select **Publish/Unpublish**.



See Also

"Period and Scenario Associations View" on page 163

Change the Properties of a Period/Scenario Association

1. Select a period/scenario association.
2. Select **Edit** ⇨ **Item Properties**.

   The Period/Scenario Association Properties dialog box appears.

Delete a Period/Scenario Association

1. Select a period/scenario association.
2. Select **Edit** ⇨ **Delete**.

Part 7

Modules

Chapter 21

Working with Modules

Introduction to Modules

What Is a Module?

Each module contains a set of accounts. The set of modules forms an ordered set—a set of sets of accounts. The set of modules is an ordered set because it determines the direction of cost flow in a model:

- An account in one module can flow costs to any other account (including itself) in the same module.
- An account in one module can flow costs to any other account in any module after its own module in the direction determined by the ordered set of modules.
- For more information on the direction of cost flow, see "Assignment Paths" on page 432.

Each module is displayed in its own view.

Model Parcel Express Tutorial Period/Scenario 2008 Q1PPT/Actual1 Column Layout Default

Parcel Express Tutorial Activity Module

Display Name	Display Reference	Cost
ACTIVITY (PRIMARY PANE)		$3,647,900.00
USA	USA	$3,647,900.00
Oregon	Oregon	$3,647,900.00
Eugene	Eugene	$1,717,000.00
Local Processing	Local Processing	$743,326.79
Personnel Intensive Activitie	Personnel Intensive Activiti	$187,783.21
Local Collection	Local Collection	$351,422.14

If a model is not a complete Cost and Profitability costing implementation, you are not required to use all of the available modules when building the model. For example, costs can be entered directly into activities in the Activity module, assignments can be made from those activities to products in the Cost Object module, and then costs can be calculated. In this example, you do not need the Resource module.

Modules are structural elements. The modules that you define for a model exist in every period/scenario in the model.

Column Layouts

SAS Cost and Profitability Management displays information in each module in user-defined columns in a grid. You can add or remove columns from the grid and specify how information in the columns appears. When you are satisfied with the appearance, you can save the display configuration (called a saved column layout). At a later time, you can display the saved column layout and the columns will be displayed like they were when you saved the column layout.

Note: The **Display Name** column, which is the left-most column in the column layout, is required, so you cannot remove it, change it, or reorder it.

See Also

"What is a Column Layout?" on page 319

References

Within each module, a reference is a unique identifier for an item, such as an account, a dimension, or a cost element.

References are similar to the account numbers or account codes in a chart of accounts and in a General Ledger; account numbers uniquely identify line items. Generally, the references for accounts and cost elements in the Resource module match the account numbers in an organization's chart of accounts.

See Also

"Reference Conventions" on page 74

Guidelines for Creating Modules

Before you create the modules, consider the following:

- Start by creating the Cost Object module.

Identify the expected result and work toward it. If your organization is in the service industry, the cost objects might not be known. In this situation, start by creating the Activity module.

- Identify the departments (or branches) of your organization.

  The structural dimensions for the Activity module and the Resource module depend on this knowledge. (More information is presented in the sections that address the Activity module and the Resource module.)

- Create the modules without calculating costs or generating cubes, but validate the model as you progress.

  Not calculating costs or generating cubes enables you to create a model quickly and helps you to identify flaws in the model design before you spend the time calculating costs and generating cubes.

Activity-Based Costing Modules

Introduction

Activity-based costing is the basic tool of activity-based management. Two critical limitations of traditional cost accounting systems are the following:

- the inability to analyze individual product, customer, service, or process costs with a reasonable level of accuracy
- the inability to provide useful feedback to management for the purpose of operational control

When managers of complex organizations make important decisions using traditional cost accounting systems, inaccurate and/or inappropriate cost information and profitability information cause incorrect decisions. These decisions often affect such areas as pricing, product mix, resource allocations, and budgeting.

Activity-based costing more accurately tracks costs than traditional methods because activity-based costing assumes the following:

- Activities cause expenditure of resources.
- Cost objects (the results of activities or products and services produced) create the demand for activities.

Using SAS Cost and Profitability Management, one or more models can be built to apply direct and indirect organizational costs to specific activities and processes. As a result, managers are able to see actual cost assignments and their bottom-line impacts from an operational perspective. Managers get a true understanding of the cause-and-effect relationships that link resources and processes to outputs. Therefore, business planners can easily forecast resource requirements, create budgets, and optimize capacity usage.

The Resource Module

Overview

The Resource module contains information about resources, such as salaries, materials, and depreciation. Resources are the costs that are consumed by activities such as planning, introducing new line items, advertising, or promoting products. To understand

and manage resources, you should focus on activities and on how they consume resources.

Guidelines for Creating the Structural Dimensions

The typical structural dimensions in the Resource module are General Ledger and Organization. These dimensions are denoted as General Ledger x Organization.

Before you create the structural dimensions, consider the following typical dimensions, suggested dimension level names, and example dimension members.

General Ledger Dimension

The General Ledger dimension typically contains information about expense categories and individual expenses. The information for this dimension can be found in your organization's general ledger or other transactional financial system.

The General Ledger categorizes expenses into a hierarchy, an example of which is shown in the following example:

```
1xx Wages, Salaries, and Benefits
    10x Salaries and Wages
        101 Wages - Hourly
        102 Wages - Salary Non-Exempt
        103 Wages - Salary
        104 Overtime Premium
        105 Shift Differential
        106 Management Bonus
```

The lowest level of detail shown (items 101-106) is generally captured in the General Ledger for such purposes as taxes, expense analysis, reconciliation, and internal control. Although this low level of detail is necessary for financial accounting, it is usually too detailed for a SAS Cost and Profitability Management model. This level of detail is cost classification, whereas the level of detail for a model should be cost behavior.

To obtain the level of detail of cost behavior, the lowest level of detail in the General Ledger that should be used in a SAS Cost and Profitability Management model is the next higher level in the example (10x Salaries and Wages). For example:

Dimension Level Name	Example
GL Account Group	Wages, Salaries, and Benefits
GL Account	Salaries and Wages

If the most detailed level of the General Ledger must be in a model, you can use entered cost elements to store these costs.

Organization Dimension

The Organization dimension that is created in the Activity module can be reused here. See "Guidelines for Creating the Structural Dimensions" on page 183.

See Also

- Chapter 24, "What Are Dimensions?," on page 195
- "New Account Dialog Box" on page 235

The Activity Module

Overview

The Activity module contains information about activities. Activities are tasks performed within an organization, such as entering the details of a customer order, operating a machine, or loading a pallet.

Activities can receive costs from resource accounts or from other activity accounts.

Guidelines for Creating Activities

When you create activities, consider these guidelines:

- An activity is a significant portion of an organizations cost (at least 5 percent of the total cost).
- Activities have the same type of process.
- An activity has one cost driver that is used for assigning product costs.
- An activity is a cost category that management is already concerned about.
- An activity involves and action or actions; it is not just an arbitrary accounting bucket.
- Ensure that all required activities are defined.
- To determine your organizations activities, conduct surveys of people in the organization.
- Create an activity only if management needs to know details about the activity to make decisions.

Methods of Organizing Activities

Use one of the following methods to organize the Activity module:

- Hierarchically by departments

  This is usually the best method.

- By process

  If you organize activities by their process, then organize the items within a rollup account either by their order in the process or by the order in which you want to report them.

To determine which method to use, consider how you want to assign costs to activities. Decide how you want to assign activity costs to cost objects. Keep in mind that you create assignments between accounts, not between rollup accounts.

Rollup accounts enable you to view the total cost of groups of related activities. For example, you can create a rollup account that groups activities in a process or that groups activities that are performed by a single department.

Guidelines for Creating the Structural Dimensions

The typical structural dimensions in the Activity module are Activities and Organization. These dimensions are denoted as Activities x Organization.

Before you create the structural dimensions, consider the following suggested dimension level names and example dimension members:

Activities Dimension

The Activities dimension typically contains information about business processes and individual activities.

A business process is a group of activities with a common outcome or output. Because activities are the building blocks of business processes, activities and business processes can be incorporated into a single structural dimension. The business processes can be created as the upper (more summary) dimension members, and the activities can be created as the lower (more detailed) dimension members. For example:

Dimension Level Name	Example
Macro Process	Receive Packages
Process	Collect by Region
Activity	Unload Trucks

Organization Dimension

The Organization dimension typically contains information about the organization's structure. The information for this dimension can be found in your organization's Organizational Chart.

Each department manager is responsible for the expenses incurred by his/her department. The activity-based management model that is most useful to business users preserves the departments. Preserving each department enables each department manager to examine the activities performed in the department, the costs of those activities, and how the resources for which each manager is responsible affect the activities that he/she manages.

For example:

Dimension Level Name	Example
Facility	North Carolina
Function	Manufacturing
Department	Inspection

See Also

Chapter 24, “What Are Dimensions?,” on page 195

Cost Object Module

Overview

The Cost Object module contains information about products and services. Additionally, products and services can be organized by customers, channels, regions, and so on.

Cost objects can receive costs from resources, activities, other cost objects, or any combination of these three.

Guidelines for Creating the Structural Dimensions

The typical structural dimensions in the Cost Object module are Products, Customers, and Channels. These dimensions are denoted as Product x Customer x Channel. (For

public organizations, the typical structural dimensions are Service and Region.) If your organization is complex, you might need to create other dimensions.

Before you create the structural dimensions, consider the following suggested dimension level names (so that business users can examine data at various levels).

Products Dimension

The Products dimension typically contains information about product categories and individual products. For example:

Dimension Level Name	Example
Product Line	Shoes
Product Group	Summer Sandals
Product	Leather Weave
SKU (if needed)	Tan576830

Customers Dimension

The Customers dimension typically contains information about customer categories and individual customers. For example:

Dimension Level Name	Example
Customer Group	Retail
Customer Category	Discount
Specific Customer	Big Mart
Location (if needed)	Phoenix

Channels Dimension

The Channels dimension typically contains information about distribution channel categories and individual channels. For example:

Dimension Level Name	Example
Channel Group	Traditional
Channel Category	Wholesale
Channel	Catalog
Channel Details (if needed)	Targeted Mail

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- “New Account Dialog Box” on page 235

External Unit Module

The External Unit module contains information about external unit cost elements. An external unit is an item, such as a part purchased from a supplier, whose cost is maintained outside of a SAS Cost and Profitability Management model, but needs to be accounted for in the model.

Like the other modules, the External Unit module can contain multiple dimensions. For example, if you obtain parts from several suppliers, you can create the dimensions

Supplier and Part. Then, based on the availability and cost of a part, you can enter cost information for the supplier from whom you bought the part.

Suppose that your organization is a municipality that supplies garbage-collection services to residents. You can create an external unit named Tons of Garbage that has a unit cost that is the cost per ton your organization is charged to dispose of the garbage. You can enter the number of tons of garbage collected to calculate the total charge. Also, if your organization supplies bins or carts in which residents place their garbage, you can create an external unit for the cost of each bin or cart.

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- “New Account Dialog Box” on page 235

Chapter 22

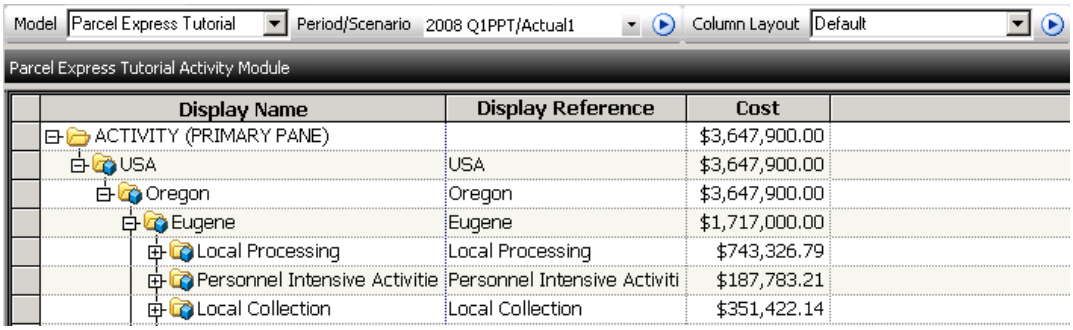
Windows for Modules

Module View

About the Module View

The availability of this feature depends on your permissions.

Use the module view to view the accounts in a module and to make assignments between accounts.



How to Access the Module View

Open a model and select **Model** ⇨ **Modules** ⇨ **<name of module>**. For example: **Model** ⇨ **Modules** ⇨ **Cost Object** .

See Also

Chapter 23, “How To: Modules,” on page 189

Chapter 23

How To: Modules

Create a Module

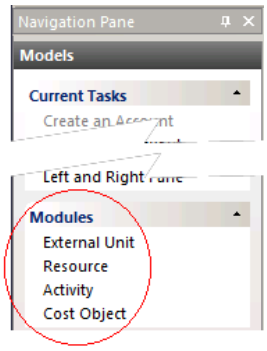
When you create a model, you create modules by specifying the number of modules desired, giving each one a name, and specifying their order.. See "Create a Model" on page 104.

Once you have created a model, you can not change the number of modules in the model or change their order.

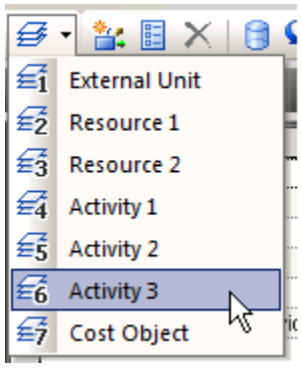
View a Module

To open a module view with the selected module as the primary pane, do one of the following:

- Select the module from the Navigation Pane.



- Select the module from the drop-down list that is available from the Select Module icon on the tool bar.



Working in a Module View

Show Assignments Panes

See "Show Assignments Panes" on page 453.

See Also

"Assignments Panes" on page 433

Make an Assignment

See "Make an Assignment" on page 450.

Show Assignments To and From an Account

See "Show Assignments To and From an Account" on page 453.

Delete Assignments

See "Delete Assignments" on page 454.

Create an account

See "Create Accounts" on page 241.

Delete an Item

1. Select an item.
2. Select **Edit** ⇨ **Delete**.

Create an Entered Cost Element

See "On an Existing Account in a Module Page" on page 266.

Go to an Account in Another Module

See "Go to an Account" on page 254.

Manage the Attributes of an Account

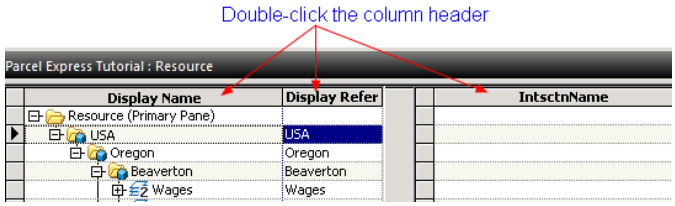
See "Specify an Attribute's Value" on page 312.

Edit the Column Layout

1. Select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

   The Column Layoutdialog box appears.

   TIP Alternatively, you can double-click a column heading.



2. To save the column layout, select **Model** ⇨ **Column Layout** ⇨ **Save As**.

   The Save Column Layout As dialog box appears.

See Also

- "Column Layout Dialog Box" on page 335
- "Save Column Layout As Dialog Box" on page 338

Change the Properties of an Item

1. Select an item.
2. Select **Edit** ⇨ **Item Properties**.

   The Item Properties dialog box appears.

See Also

“ Item (Account) Properties and Attributes Dialog Box” on page 239

Part 8

Dimensions

Chapter 24

What Are Dimensions?

Dimensions

A dimension is a category by which data can be analyzed. For example, you might want to categorize sales figures by region, by customer, and by product. Each of these categories represents a single dimension. Common dimensions are products, time, geography, customers, promotions, and sales channels.

To further illustrate, suppose that a manager says, I need to see the data items x, y, and z grouped by a, by b, and by c. In this statement, x, y, and z represent measures, and a, b, and c represent dimensions. For example, perhaps the manager needs to see revenue, cost, and profit grouped by region, by customer, and by product.

Note: For information about specifications for dimensions in Microsoft Analysis Services, see the Microsoft documentation.

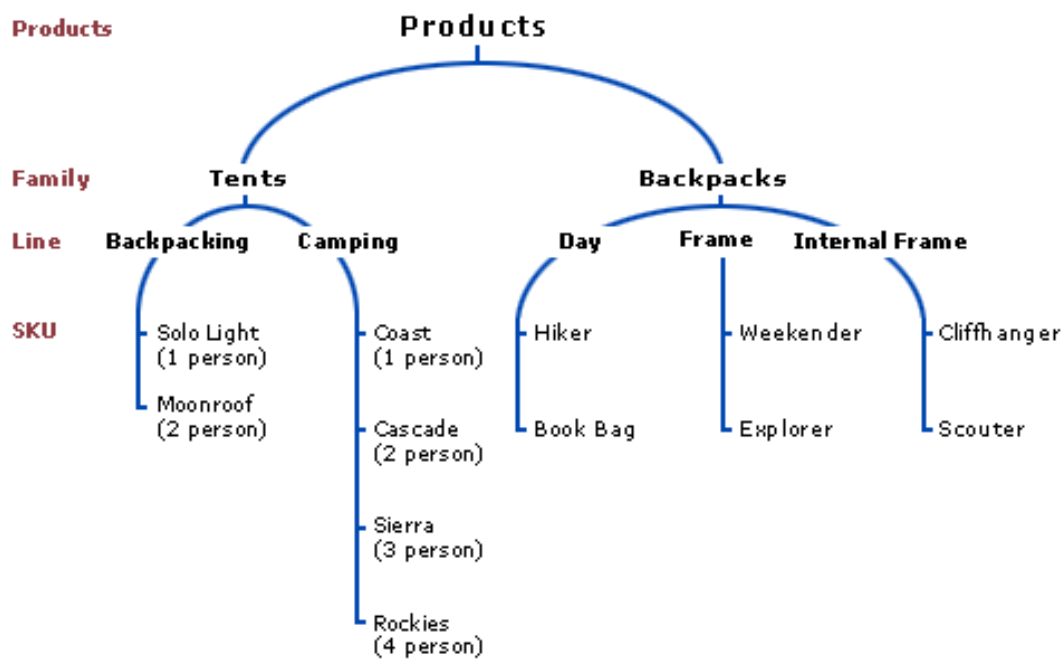
Dimensions are structural elements. The dimensions that you define for a model exist in every period/scenario in the model.

Dimension Levels

To present data in a manageable or interesting form, you can group items within a dimension. Each item is then at a specific level in a hierarchy.

By default, each dimension level is given a name, such as Level1. To make this name more meaningful during OLAP analysis, you can rename dimension levels. There is more information about dimension level naming conventions.

For example, in the Products dimension, you can start with product families (the dimension level named Family). You can divide product families first into product lines (the dimension level named Line), and then into stock keeping units (the dimension level named SKU).



In this example, the Products dimension has three levels below it. The levels enable you to show the measures for each level of information.

Dimension levels are a powerful modeling tool because they enable you to ask general questions and expand a dimension in order to reveal more detail. For example, you might first ask to see product costs for the past three fiscal years. You might notice that the costs for 2001 are higher than the costs for other years. You might explore levels of the Products dimension to see whether costs were high for a particular product family, product line, or SKU. This type of exploration is known as drill-down.

When you create a model, you select the dimension levels.

Dimension Members

Overview

Dimension members are the unique elements in the dimension levels. In the previous example, Backpacking, Frame, and Hiker are examples of dimension members.

Dimension members are structural elements. The dimension members that you define for a model exist in every period/scenario in the model.

All, No <Dimension Name>, None, and (Data) on the Module views

To understand what a Model view is showing, add a column for the Intersection Name property. The intersection name lists the dimension members in the dimension order from left to right.

In the following picture, the COST OBJECT module rollup cost represents all product costs and all customer costs. In this case, the dimension order is the Product dimension by the Customer dimension, which is often written as Product X Customer. In an intersection name, All represents all dimension members in that position in the dimension order. By looking at the intersection name (IntsctnName column), you see that the Solo Light rollup account is the intersection of the Solo Light dimension member of the Product dimension and all the dimension members of the Customer dimension (Solo Light x All). So, the Solo Light rollup account represents all of the product costs of Solo Light, which includes the costs for all customers and all other costs.

As you drill down into the COST OBJECT module rollup, you see more details about its costs. When you expand Solo Light to see its accounts, you see a No <Customer> account, which indicates costs (which are $248,253.44) that affect the total cost of the Solo Light product, but that are not directly assigned to any specific customer (such as raw materials, manufacturing, and production activities). When you look at the No <Customer> intersection name, you see that No <Customer> is the intersection of the dimension member and no customer dimension members (Solo Light x No <Customer>).

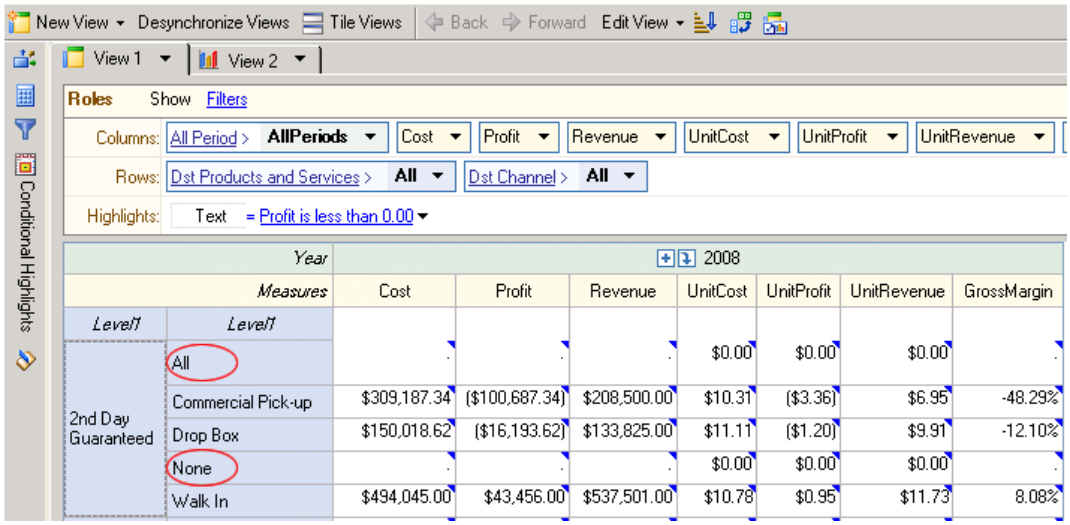
No <dimension> represents costs that are outside the identified intersections. For example, suppose that you create a model that represents your department. When you import costs from the General Ledger (which contains the costs for all departments), you can ensure that the costs of the other departments do not affect your department by putting the costs for the other departments in No <dimension>.

The other Solo Light account is LLCorn, a customer. This account's intersection name indicates that the account is the intersection of the Solo Light product and the LLCorn customer (Solo Light X LLCorn). The account's cost represents the product costs of Solo Light and the customer costs of LLCorn.

Display Name	IntsctnName	Cost
COST OBJECT (PRIMARY PANE)		$1,751,735.000
Solo Light	Solo Light x All	$256,308.859
No <Customer>	Solo Light x No <Customer>	$248,253.447
LLCorn	Solo Light x LLCorn	$256,308.859
Solo Light x No <Customer>	Solo Light x No <Customer>	$248,253.447
No <Product> x LLCorn	No <Product> x LLCorn	$8,055.412
Moonroof	Moonroof x All	$173,873.182
Entryway	Entryway x All	$226,634.782
Waterproof case	Waterproof case x All	$292,481.457
pack book	pack book x All	$181,986.542
Pack Week	Pack Week x All	$389,447.014
Pack Expl	Pack Expl x All	$47,052.759
Pack Clif	Pack Clif x All	$183,950.406
No <Product>	No <Product> x All	$122,349.124
LLCorn	No <Product> x LLCorn	$80,554.116
Preliminary RFQ	Preliminary RFQ	$4,634.809
Sales Call	Sales Call	$12,122.997
Take Order	Take Order	$8,738.214
Process Returns	Process Returns	$5,871.718
Trade Show Activities	Trade Show Activities	$35,343.803
Create BOM	Create BOM	$6,397.260
Collecting Late Bills	Collecting Late Bills	$1,341.667
Normal Recieving	Normal Recieving	$3,226.935
Expedite Order	Expedite Order	$972.603
Reviewing BOMs	Reviewing BOMs	$1,904.110
Eddy Bear	No <Product> x Eddy Bear	$14,863.730
Waters End	No <Product> x Waters End	$26,931.278

All, No <Dimension Name>, None, and (Data) in Grid View

In OLAP grid view, every dimension has the same value for the total value in the entire cube, which is shown in the **All** field. The All member of the dimension is not associated with any values, and the None value is used to balance the All value with values that are associated with other members in the dimension.



Structural Dimensions

Structural dimensions are the building blocks of the modules in a model. For example, the typical structural dimensions of the Resource module are region, organization, and General Ledger; the Activity module might be structured according to the region or organization dimension, along with an activity dimension. The combination of dimension members that uniquely identifies an account is its dimension signature.

Structural dimensions are basically "buckets" to model the flow of costs through your organization. However, this perspective of the costs is generally too detailed for other people in your organization. These people require a higher level of detail, such as details that are provided by attribute dimensions and OLAP analysis. Be aware that how you create the structural dimensions can greatly help users manipulate the model data into a form that suits their needs.

Note: Structural dimensions are created when you first create a model. You cannot delete structural dimensions later, and you cannot create new structural dimensions (but you can create dimension attributes). However, you can add and delete dimension members within each structural dimension.

Attribute Dimensions

Attribute dimensions are dimensions that are created whenever dimension attributes are created. You cannot explicitly create an attribute dimension. The SAS Cost and Profitability Management OLAP tool makes no distinction between attribute dimensions and structural dimensions.

See "Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes" on page 272.

Guidelines for Creating Dimensions

Before you create dimensions, consider the following points:

- There is a limit of 126 total user-defined dimensions in a model—including both structural dimensions and attribute dimensions.
- Think in terms of dimensions when you begin to plan the model, such as Who do I work for? At the highest level, it's your entire organization. This can be further refined by a geographic office (world area, country, and/or state or province), a business area (Sales or Finance), and a department (Public Sales or Accounts Payable).

  A dimension signature uniquely identifies an account in the model; the dimension signature is the intersection of two or more dimensions. So, think of what dimensions are needed to uniquely identify an account.
- Dimensions help users pull apart (deconstruct) information in a model. Consider the needs of the users.

  Identify a dimension by the fact that users need to have the information appear during OLAP analysis.
- Decide what required data translates into model structure or what data translates into attributes.
- After you have defined the structural dimensions, consider how to enhance the structure with attributes to support different aggregations of costs for different users.

  Attribute dimensions are for users needs for analysis. Do not put these needs into the models structural dimensions because doing so creates a burden when entering model data or when revising a model.
- For better performance and easier data entry, minimize the number of dimensions.

  Although a dimension intersection can include an unlimited number of dimensions, a smaller number (five or fewer) is more manageable.
- Ensure that you have a justifiable business need for every dimension or dimension level.

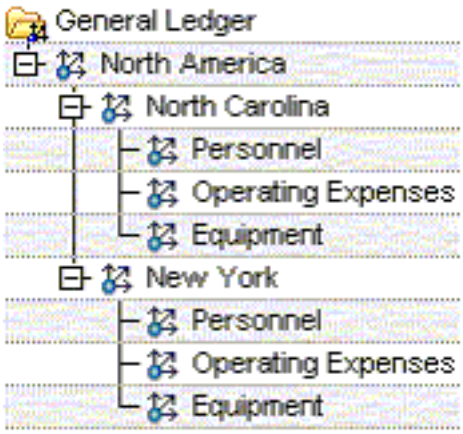
  If you create unnecessary dimensions and dimension levels, the resulting OLAP cubes provide too much information and not enough understanding.
- To help distinguish a dimension from a dimension member, consider the following points:
- If part of a model is repeated in a single dimension (such as the General Ledger accounts of Rent-North Carolina, Rent-New York, and Rent-Kansas), then that part is probably a dimension. In this example, you provide more flexibility for OLAP analysis by creating two dimensions: one dimension for the office location and one dimension for the General Ledger expenses.
- A dimension member can be identified by whether it gives or receives costs (it is part of an assignment).

Example: Multiple Dimensions for Better Analysis

Suppose that your organization wants to model the North America region, specifically the offices in the states of North Carolina and New York. The model must store the

resource costs of personnel, operating expenses, and equipment for each of these locations.

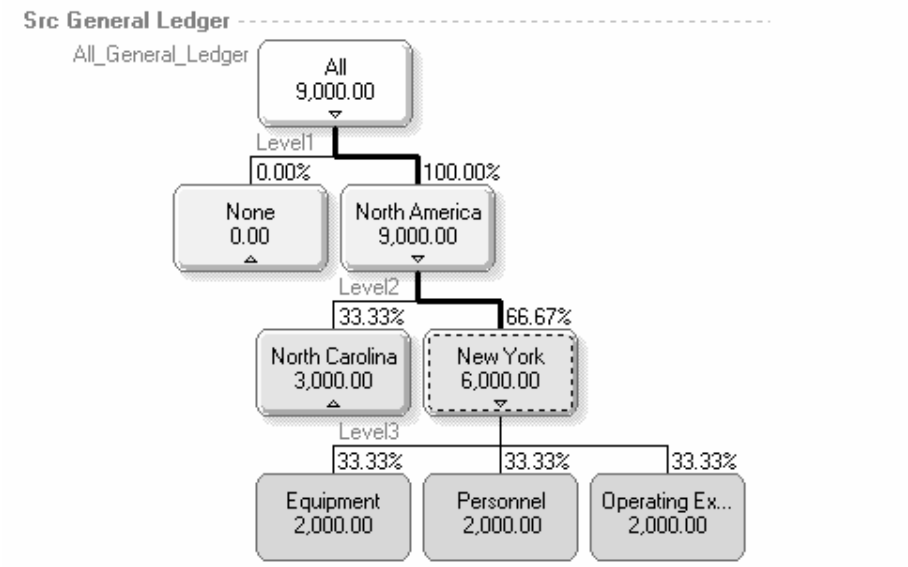
From your perspective as model builder, you can create a single, structural dimension on the Dimensions view as shown in the following figure:



In the Resource module, the structure looks like the following figure (with example values entered):

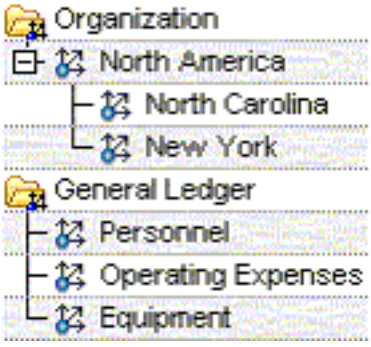
Display Name	Cost
RESOURCE (PRIMARY PANE)	$9,000.00
North America	$9,000.00
North Carolina	$3,000.00
Personnel	$1,000.00
Personnel_1017-1	$1,000.00
Operating Expenses	$1,000.00
Operating Expenses_	$1,000.00
Equipment	$1,000.00
Equipment_1019-1	$1,000.00
New York	$6,000.00
Personnel	$2,000.00
Personnel_1026-1	$2,000.00
Operating Expenses	$2,000.00
Operating Expenses_	$2,000.00
Equipment	$2,000.00
Equipment_1028-1	$2,000.00

Using this structural dimension, after you build the entire model, the calculated costs are correct. Users can analyze the cost data in the OLAP Analyzer, as shown in the following figure of the Cube Explorer View:

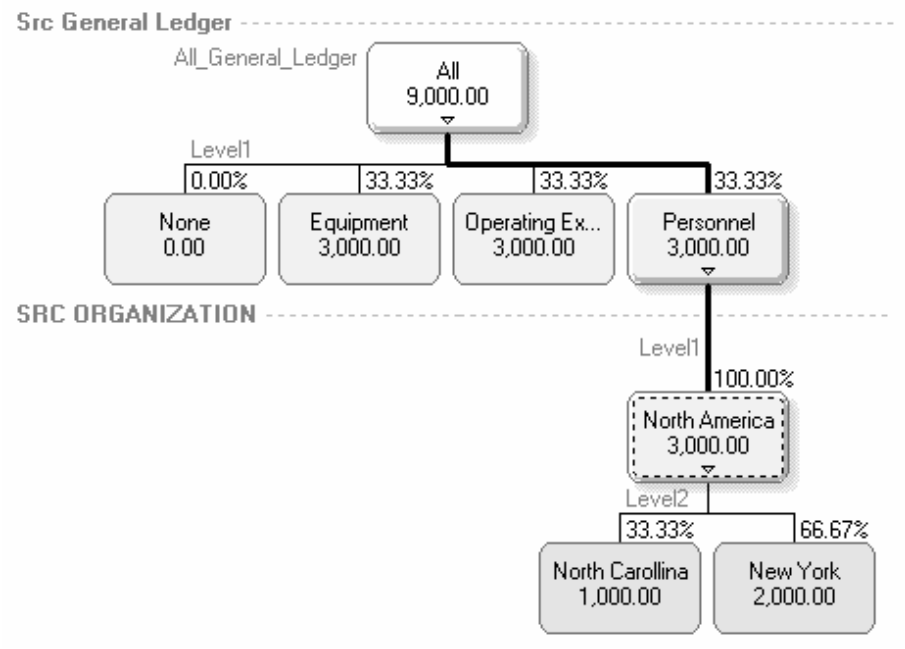


Users can examine the costs associated with New York or North Carolina (which is not expanded in the figure). However, they cannot combine the costs of individual accounts for both North Carolina and New York. For example, users cannot examine the total cost of personnel for both North Carolina and New York.

Now, suppose that you want to allow users more flexibility during OLAP analysis, so that they can combine accounts from different office locations. You can change the previous single, structural dimension to become two dimensions, as shown in the following figure of the Dimensions view:



In the Resources module, the structure looks identical to the previous example. However, in addition to the OLAP analysis shown in the previous example, users can display the model data so that it reveals the total personnel costs for both North Carolina and New York, as shown in the following figure:


Src General Ledger
All_General_Ledger
All
9,000.00
Level1
0.00%
33.33%
33.33%
33.33%
None
0.00
Equipment
3,000.00
Operating Ex...
3,000.00
Personnel
3,000.00
SRC ORGANIZATION
Level1
100.00%
North America
3,000.00
Level2
33.33%
66.67%
North Carollina
1,000.00
New York
2,000.00

Chapter 25

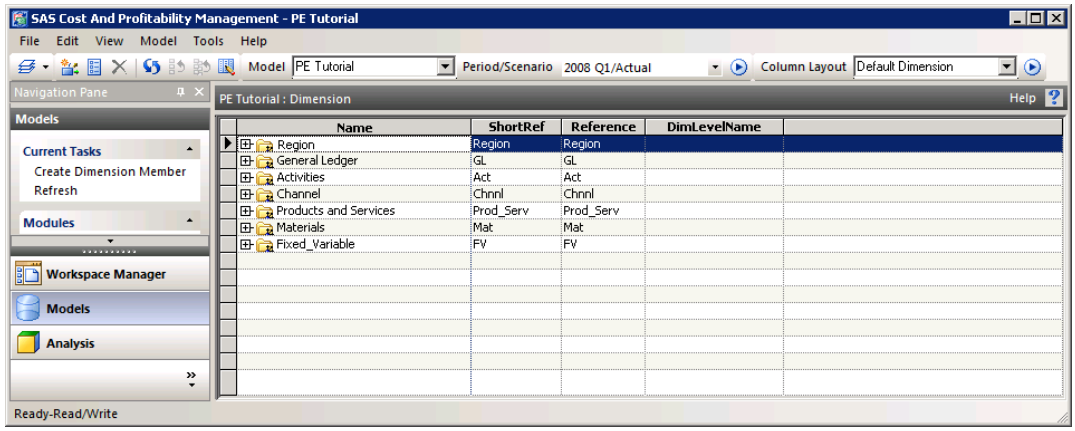
Windows for Dimensions

Dimensions View

About the Dimensions View

The availability of this feature depends on your permissions.

On the Dimensions view, you can manage a model's dimensions and the dimension member names.



Note: You cannot directly edit the information on the Dimensions view.

How to Access the Dimensions View

Open a model and select **Model** ⇨ **Dimensions**.

See Also

- Chapter 24, "What Are Dimensions?," on page 195
- "Create a Dimension Member" on page 212

New Dimension Dialog Box

About the New Dimension Dialog Box

In the New Dimension dialog box, you can create a new dimension if you are creating a new model.

Note: The availability of these features depends on your permissions.

1. Click in the **Name** column and type the name of the new dimension.

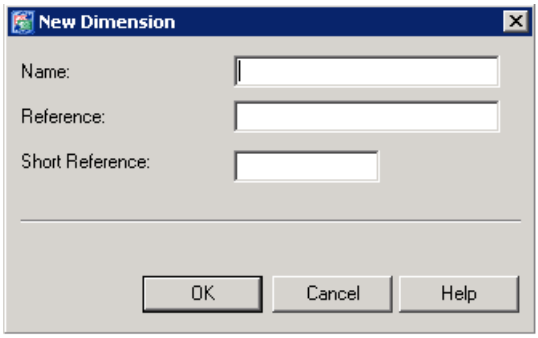
   The name must follow the naming conventions. See "Naming Conventions" on page 69.

2. Click in the **Reference** column and type the reference.

   A default reference is created from the dimension's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.

How to Access the New Dimension Dialog Box

In the New Model Wizard, when you are specifying the model's dimensions, click **New**.



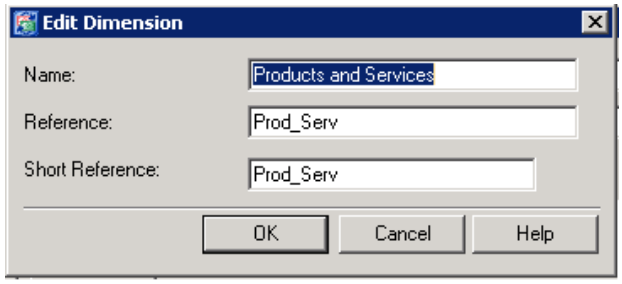
See Also

- "New Model Dialog Box" on page 93
- Chapter 24, "What Are Dimensions?," on page 195

Edit Dimension Dialog Box

About the Edit Dimension Dialog Box

In the Edit Dimension dialog box, you can edit information about a dimension if you are creating a new model.



The availability of this feature depends on your permissions.

How to Access the Edit Dimension Dialog Box

In the New Model Wizard, when you are specifying the dimensions in the model, select a dimension and click **Edit**.

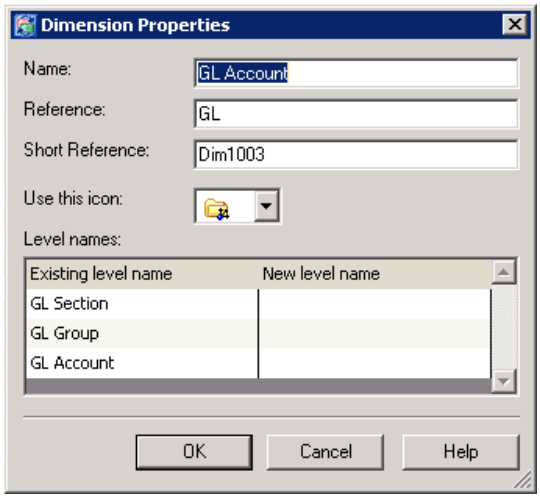
See Also

- Chapter 24, "What Are Dimensions?," on page 195
- "Dimension Properties Dialog Box" on page 208

Dimension Properties Dialog Box

About the Dimension Properties Dialog Box

In the Dimension Properties dialog box, you can review or change information about a dimension.



How to Access the Dimension Properties Dialog Box

On the Dimensions view, select a dimension and select **Edit** ⇨ **Item Properties**.

See Also

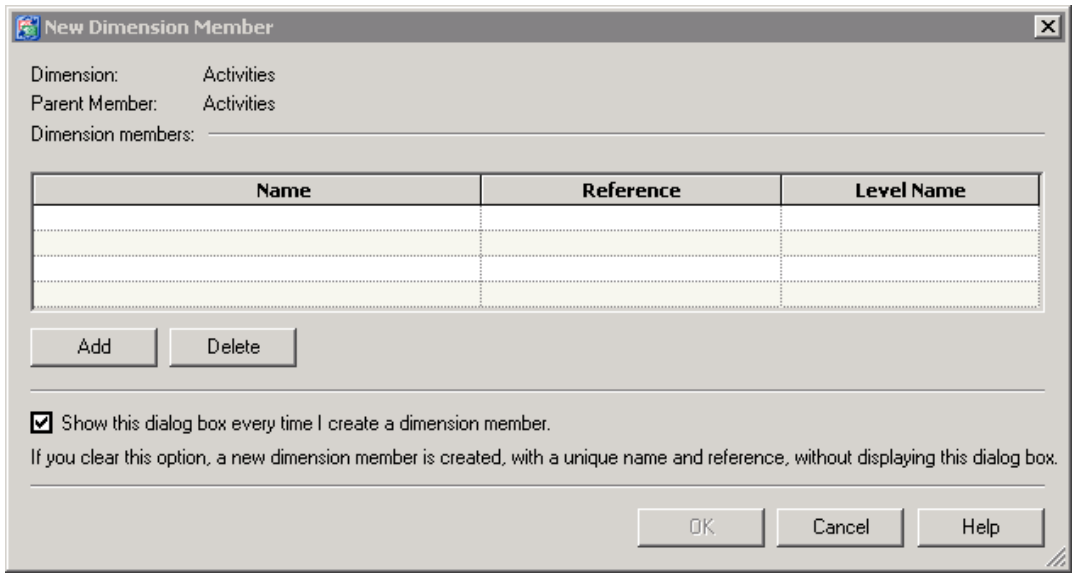
- "Edit Dimension Dialog Box" on page 207
- Chapter 24, "What Are Dimensions?," on page 195

New Dimension Member Dialog Box

About the New Dimension Member Dialog Box

In the New Dimension Member dialog box, you can name a new dimension member and you can specify other information about the dimension member.

Note: The availability of these features depends on your permissions.



How to Access the New Dimension Member Dialog Box

On the Dimensions view, select an item and select **Edit** ⇨ **New Dimension Member**.

See Also

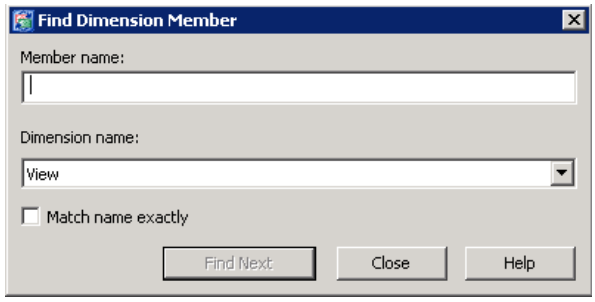
- "Create a Dimension Member" on page 212
- Chapter 24, "What Are Dimensions?," on page 195

Find Dimension Member Dialog Box

In the Find Dimension Member dialog box, you can quickly find dimension members when creating a new account.

How to Access the Find Dimension Member Dialog Box

Open a model and select **Edit** ⇨ **New Account**. When the New Account Wizard opens, click **Find**.



Find a Dimension Member

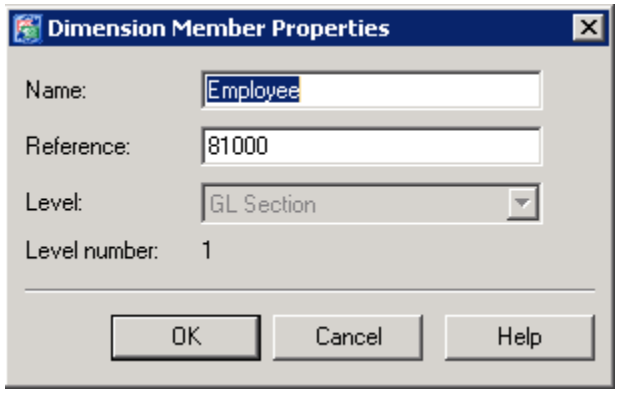
1. Type the Member name.
2. Select the dimension from the Dimension name menu.
3. (Optional) Select **Match name exactly** to find only exact matches with the member name that you type.

- If you don't select **Match name exactly**, then you can use an asterisk (*) as a wildcard to stand for 0 or more of any characters. And, the member name that you type is implicitly appended with an asterisk so that, for example, typing Equipment matches Equipment Depreciation and Equipment Expenses (it is the same as typing Equipment*).
- If you select **Match name exactly**, then only exact matches are made and there are no wildcards. If you type a name with an asterisk, then it will match only a member whose name itself contains an asterisk. And, the name that you type is not implicitly appended with an asterisk wildcard.

Dimension Member Properties Dialog Box

About the Dimension Member Properties Dialog Box

In the Dimension Member Properties dialog box, you can review or change information about a dimension member.



The availability of these features depends on your permissions.

How to Access the Dimension Member Properties Dialog Box

On the Dimensions view, select a dimension member and select **Edit** ⇨ **Item Properties**.

See Also

Chapter 24, “What Are Dimensions?,” on page 195

Chapter 26

How To: Dimensions

Create a Dimension

When you create a model, you specify the model's dimensions and you can not change them afterwards. When you specify the model's dimensions, you can create a new dimension.

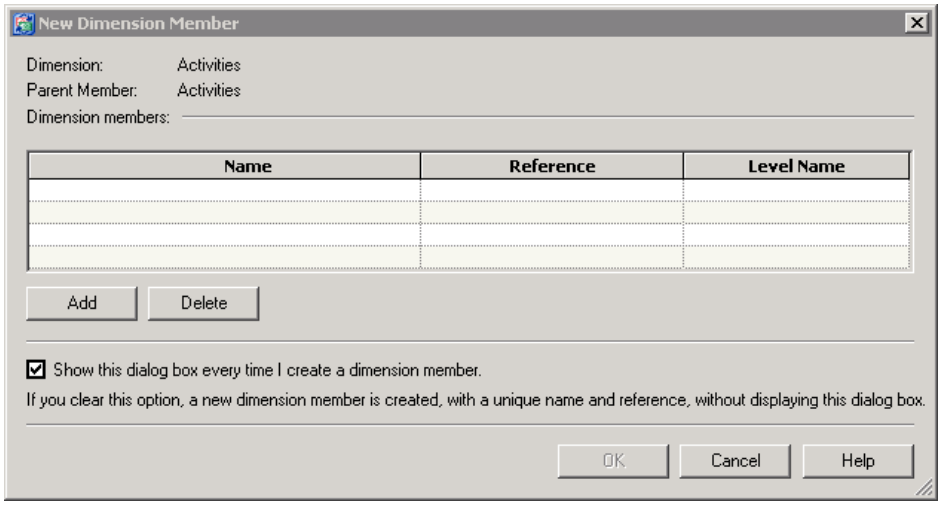
See Also

- "Create a Model" on page 104
- "New Dimension Dialog Box" on page 206

Create a Dimension Member

1. On the Dimensions page, select a dimension or a dimension member within which to create the new dimension member.
2. Select **Edit** ⇨ **New Dimension Member**.

   The New Dimension Member dialog box appears.



3. Click **Add**.

   A new row is added to the **Dimension members** list. The row contains default information.
4. Click in the **Name** column and type the name of the new dimension member.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.
5. Click in the **Reference** column and type the reference.

   A default reference is created from the dimension member's name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.
6. Click in the **Level Name**column and select a level name.
7. Select or clear the **Show this dialog box every time I create a dimension member** option.
8. Click **Add**.
9. On a module page, create the account that corresponds to the dimension member.

See Also

- "New Dimension Member Dialog Box" on page 208
- Chapter 24, "What Are Dimensions?," on page 195

Create a Dimension Attribute

See "Create a Dimension Attribute" on page 310.

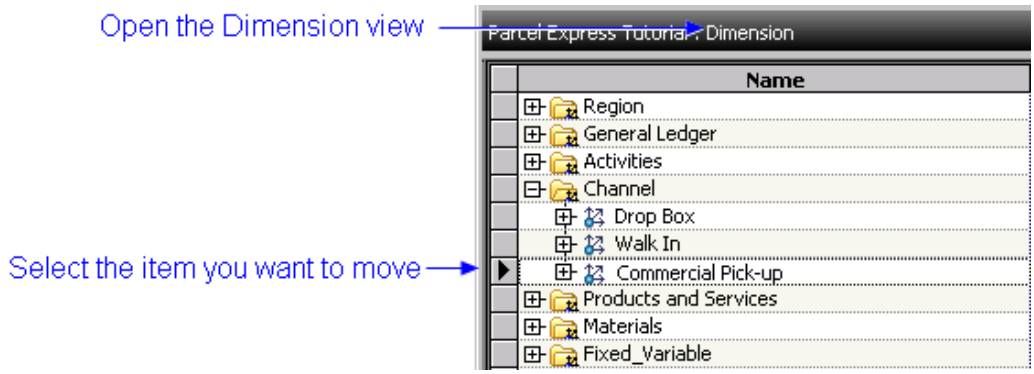
Reorder Dimension Members

Overview

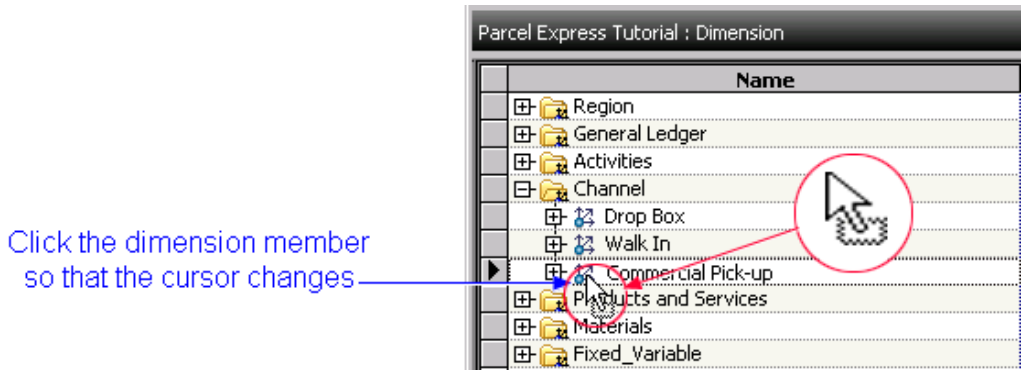
You can use the mouse, keyboard, or staging tables to change the order of dimension members.

Using the Mouse

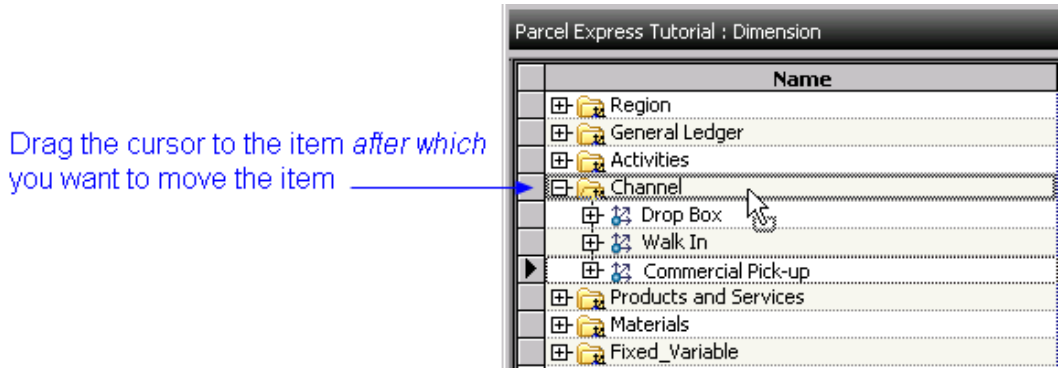
1. Go to the Dimensions view and select the dimension member that you want to reorder.



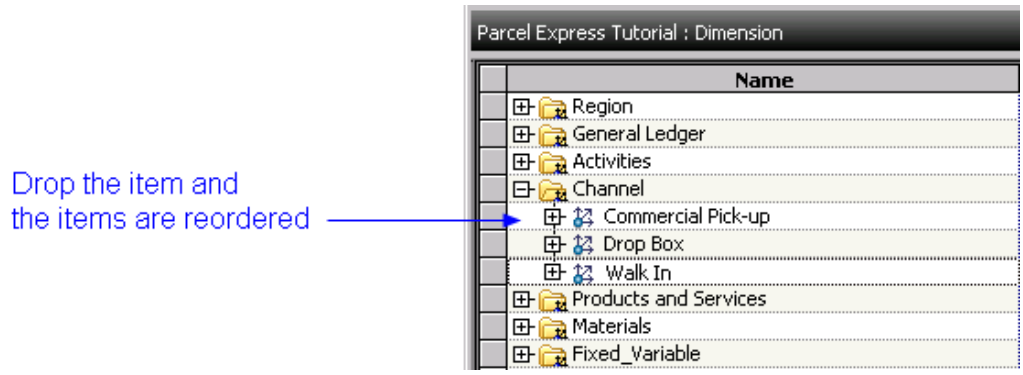
2. Click the dimension member so that the cursor changes to the following :



3. Drag the cursor to the dimension member after which you want to move the selected item.



4. Drop the item and the dimension members are reordered.



Note: If you drag a dimension member out of its parent, then the system assumes that you want to change the parent of the dimension member.

Using the Keyboard

To reorder dimension members using the keyboard, do the following:

1. Go to the Dimensions view.
2. Use the up or down arrow keys to navigate to the dimension member that you want to reorder.
3. Press **CTRL + D** to select the dimension member.
4. Use the up or down arrow keys to navigate to the dimension member after which you want to place the dimension member.
5. Press **CTRL + R** to drop the item.

   The item is moved to the new position.

   Note: If you drag a dimension member out of its parent, then the system assumes that you want to change the parent of the dimension member.

Using Staging Tables

To specify the display order of dimension members using staging tables, use the DisplayOrder field in the DimensionMember table.

You might consider numbering dimension members with increments of 10 so that you can easily interprose dimension member rows that are out of order in the staging table.

For example, in the following picture the dimension member, Land Distribution, is assigned DisplayOrder=15 so that it displays immediately after AirDistribution (DisplayOrder=10) and before Expedite Shipping (DisplayOrder=20).

DimRef	Reference	Name	ParentRefer	DisplayOrder
Act	Air Distribution	Air Distribution		10
Act	Expedite Shipping	Expedite Shipping		20
Act	Inspect Packaging	Inspect Packaging		30
Act	Land Distribution	Land Distribution		15
Chnl	Customer Pickup	Customer Pickup		10
Chnl	Drop Box	Drop Box		20
Chnl	Storefront	Storefront		30
GL	Equipment Expens	Equipment Expenses		10
GL	Operating Expense	Operating Expenses		20
GL	Wages	Wages		30

For information on staging tables, see Chapter 13, “About Staging Tables,” in *SAS Cost and Profitability Management: Data Administration Guide*..

See Also

“Change the Parent of a Dimension Member” on page 215

Change the Parent of a Dimension Member

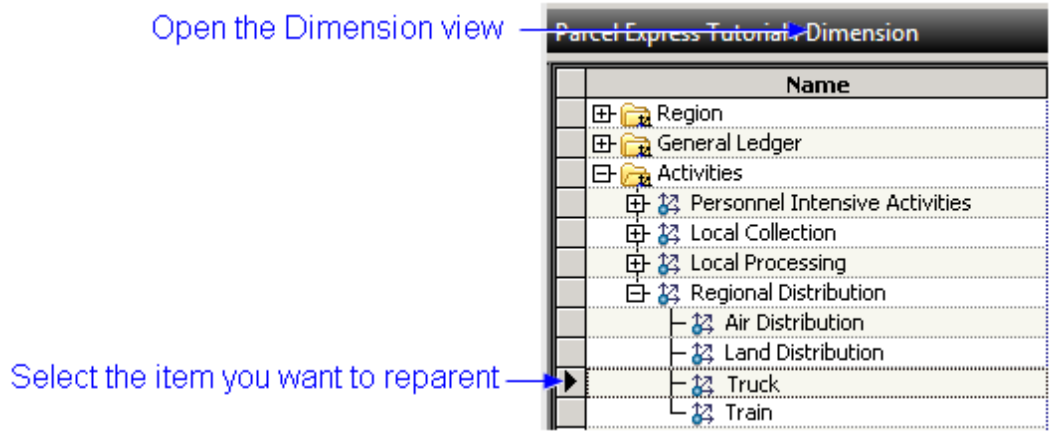
Overview

You can use the mouse or keyboard to change the parent of a dimension member by moving the dimension member to its new parent.

Note: You can move a dimension member to a new parent only in the same dimension.

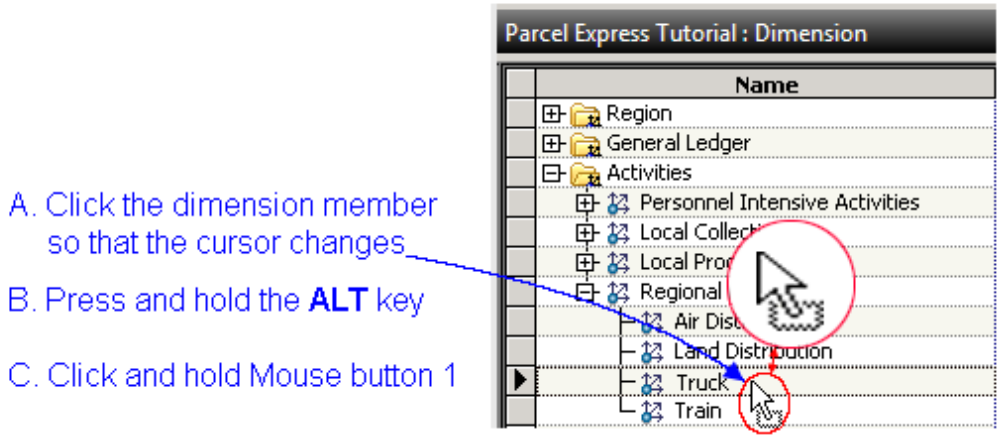
Using the Mouse

1. Go to the Dimensions view and select the dimension member that you want to reparent.



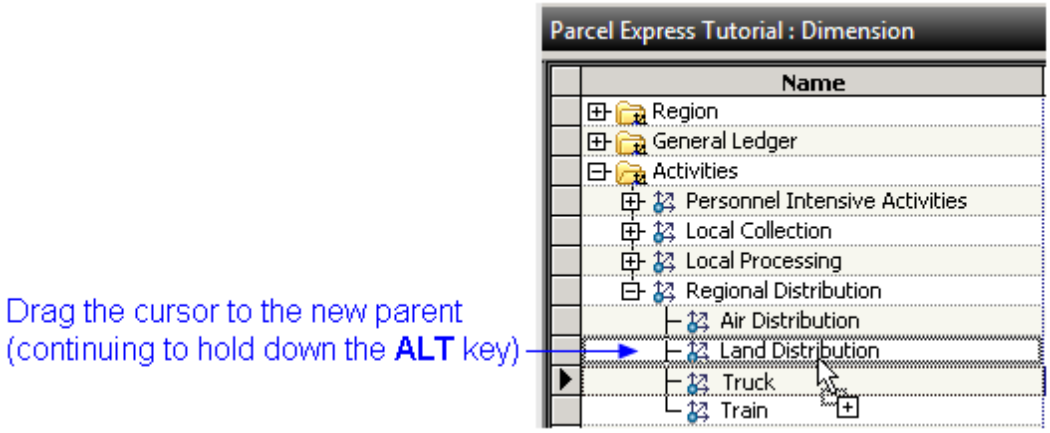
2. Click the dimension member so that the cursor changes to the following: 

3. Press and hold the **ALT** key, and then click and hold **Mouse button 1**.



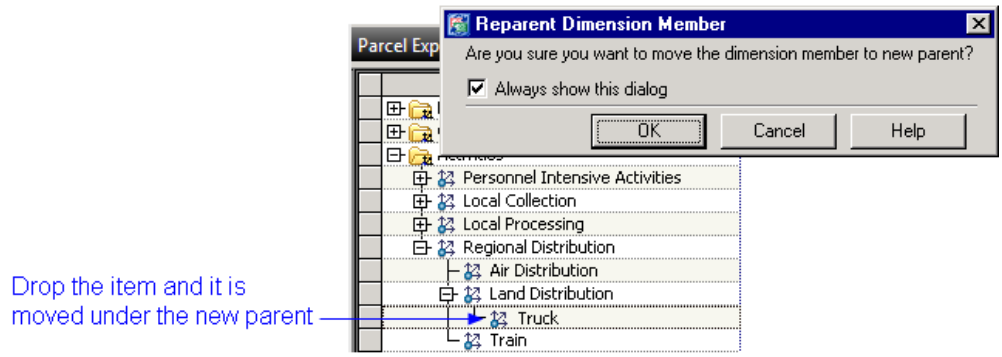
Note: It is important to press the ALT key before clicking the mouse button.

4. Drag the cursor to the new parent (continuing to hold down the ALT key).



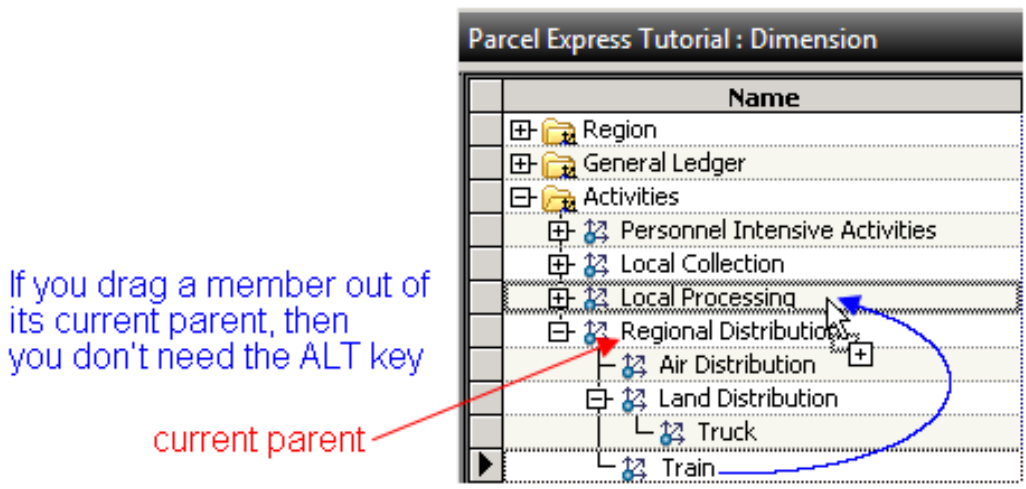
5. Drop the item.

   You are asked to confirm that you want to re-parent the item. If you confirm, then the item is moved under the new parent.



6. The dimension member is always added at the end as the last child of the new parent. If this is not where you want it, you can reorder the dimension member under its new parent.

   Note: If you drag a dimension member out of its current parent, then the system assumes that you want to change the parent of the dimension member and it is not necessary to hold the ALT key.



Using the Keyboard

To re-parent an account using the keyboard, do the following:

1. Go to the Dimensions view.
2. Use the up or down arrow keys to navigate to the dimension member that you want to re-parent.
3. Press **CTRL** + **D** to select the dimension member.
4. Use the up or down arrow keys to navigate to the dimension member that is to be the parent.
5. Press **CTRL** + **Shift** + **R** to drop the item.

   You are asked to confirm that you want to re-parent the item. If you confirm, then the item is moved under the new parent.

Using Staging Tables

You can specify the parent of a dimension member with the ParentReference field in the DimensionMember table.

Note: However, if a model already exists, you can not change the parent of a dimension member by re-importing a table with a different value for its ParentReference field—if the values are different, the imported value is ignored.

For information on staging tables, see Chapter 13, "About Staging Tables," in *SAS Cost and Profitability Management: Data Administration Guide*..

See Also

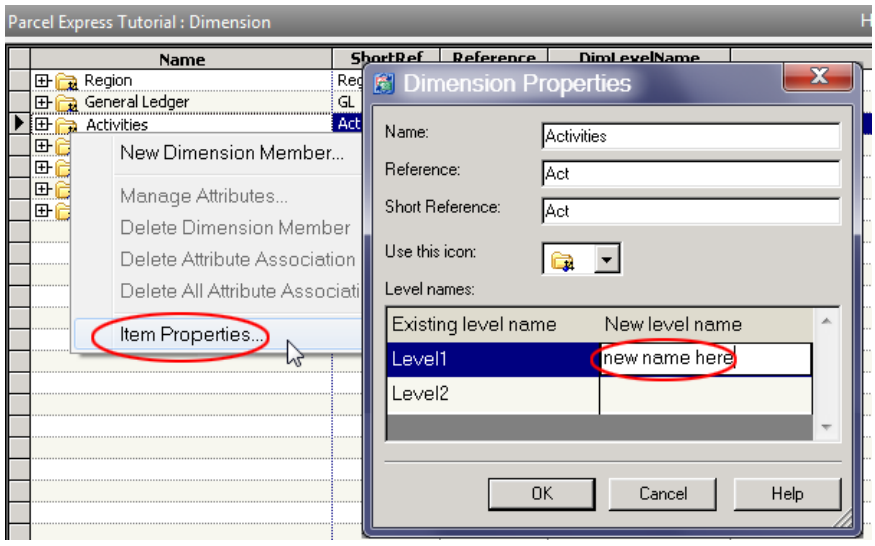
"Reorder Dimension Members" on page 213

Change Dimension Level Names

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**.



See Also

"Dimension level naming conventions" on page 70

Attach Attributes to Dimension Members

See Chapter 34, “Attributes on Dimension Members,” on page 277.

Part 9

Accounts

Chapter 27

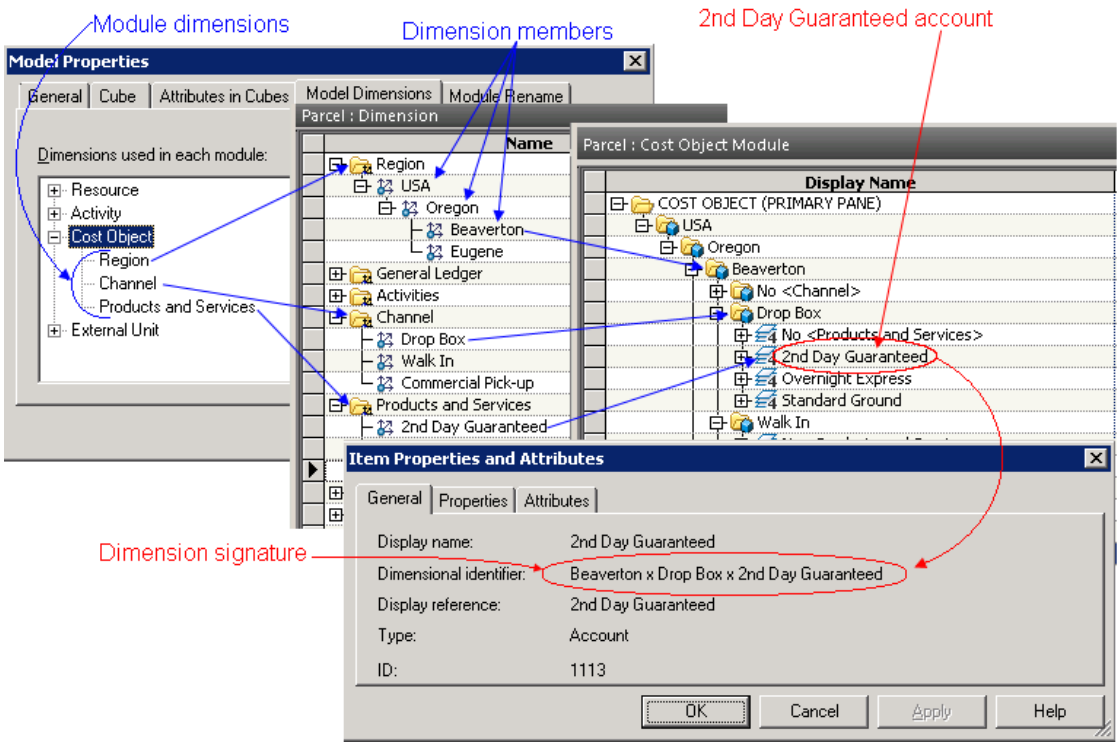
What are Accounts?

Introduction

An account is an intersection of dimension members—one dimension member from each dimension in a module—and is the bearer of costs in a model. In the following picture you can see that the account 2nd Day Guaranteed is the intersection of the dimension members Beaverton x Drop Box x 2nd Day Guaranteed. Those dimension members are, in turn, members of the dimensions Region, Channel, and Products and Services respectively.

You can notice that the display name of the account, 2nd Day Guaranteed, is the name of the last member in the intersection of the dimension members when the order of dimension members is the order of their containing dimensions—Region, Channel, Products. The order of dimensions is the order in which you define them when you create a model.

Don’t be confused by the fact that the display name of an account is the name of its last dimension member into thinking that the account just is that last dimension member. To repeat, an account is the intersection of dimension members. In addition to a display name, each account has an intersection name which is includes the name of each dimension member in the intersection. For example, the intersection name of 2nd Day Guaranteed is Beaverton x Drop Box x 2nd Day Guaranteed.



Note: An account must contain a dimension member from every dimension in a module. However, you can omit specifying a dimension member from a particular dimension by selecting instead the "None" dimension member that the system automatically generates for each dimension.

Accounts are periodic elements. The accounts in one period/scenario may exist in another period/scenario. And, when an account (same dimension signature) exists in multiple period/scenarios, its cost in one period/scenario is independent of its cost in another period/scenario.

Accounts as Dimension-Member Crossings

Introduction

This topic discusses accounts as dimension-member crossings with a simple example. The following picture shows a model whose Resource module has two dimensions: Dimension1 and Dimension2.

Dimension1 has the following dimension members:

AAA

- **111**
- **222**

BBB

- **333**

Dimension2 has the following dimension members:

CCC

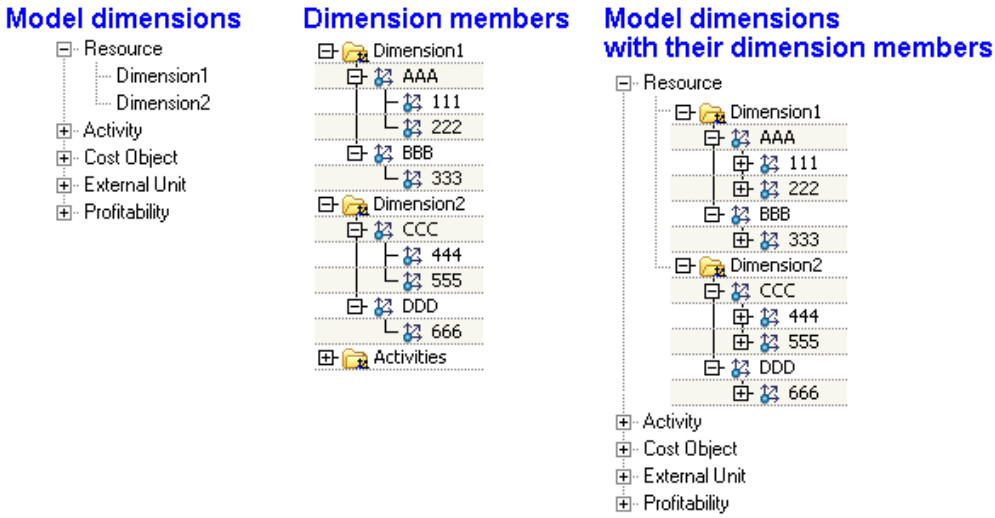
- **444**

- **555**

DDD

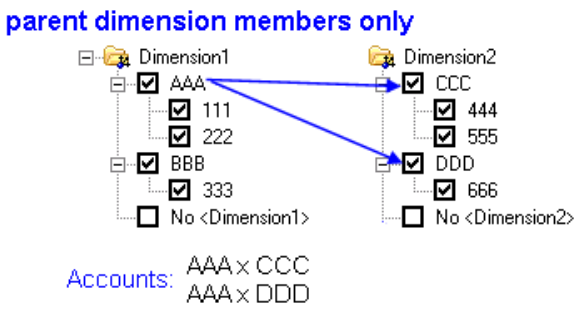
- **666**

Note: The picture also shows both dimensions along with their dimension members underneath them. Such a view is not available in the product UI but was created using an image editor.



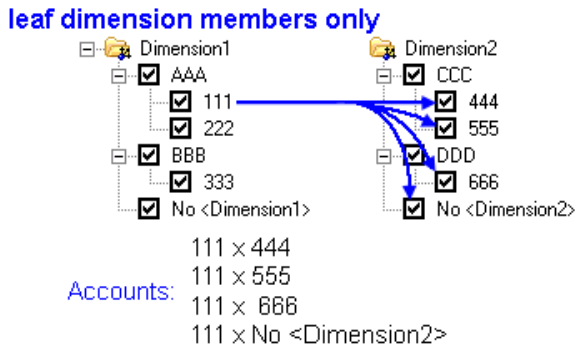
Parent Dimension Members Only

The following picture shows the accounts that are created as a result of selecting a number of dimension members and also selecting **parent dimension members only**. Notice that even though leaf members 111, 222, 333, 444, 555, and 666 are selected, they are not included in any intersections because of the **parent dimension members only** selection.



Leaf Dimension Members Only

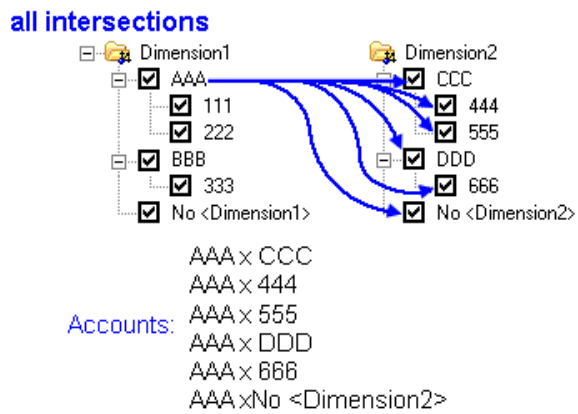
The following picture shows a sample of the accounts that are created as a result of selecting a number of dimension members and also selecting **leaf dimension members only**. Notice that even though parent members AAA, BBB, CCC, and DDD are selected, they are not included in any intersections because of the **leaf dimension members only** selection.



Note: The selections **leaf dimension members only** and **parent dimension members only** do not apply to the "No" dimension member. If you select the "No" dimension member (for example, No <Channel>) then it is included in dimension crossings regardless of whether you select **leaf dimension members only** or **parent dimension members only**. Another way to put this is that the "No" dimension member is treated as both a parent node and as a leaf node.

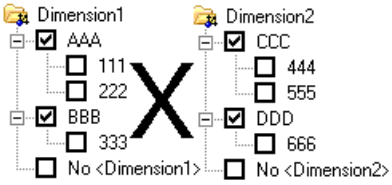
All Intersections

The following picture shows a sample of the accounts that are created as a result of selecting a number of dimension members and also selecting all intersections.

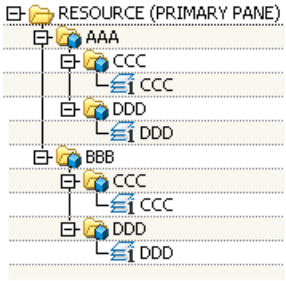


The following picture shows all the accounts that are created as a result of the same selections as above. Notice that **all intersections** is not just the union of **parent dimension members only** and **leaf dimension members only**. The accounts highlighted in yellow under **all intersections** are accounts that are not created when you select either **parent dimension members only** or **leaf dimension members only**. An example of such an account highlighted in yellow is AAA x 444. Notice that it includes both a parent dimension member, AAA, and a leaf dimension member, 444.

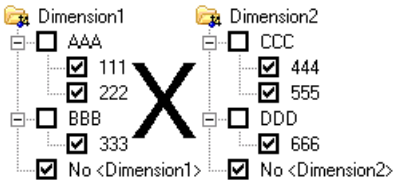
parent dimension members only



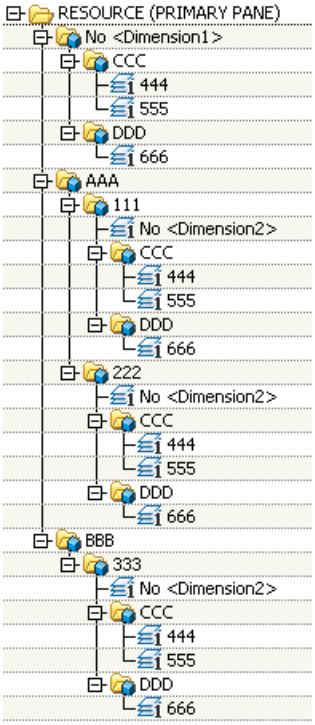
AAA x CCC
AAA x DDD
BBB x CCC
BBB x DDD



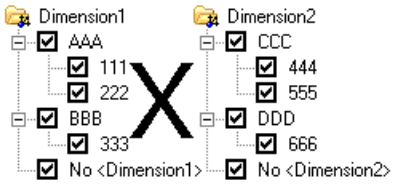
leaf dimension members only



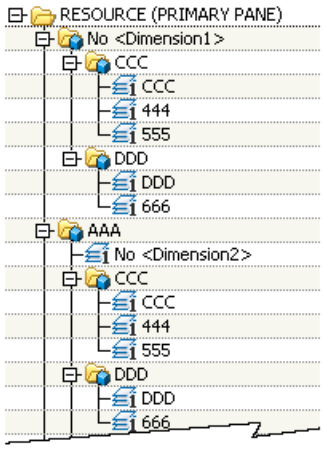
111 x 444
111 x 555
111 x 666
222 x 444
222 x 555
222 x 666
333 x 444
333 x 555
333 x 666
111 x No <Dimension2>
222 x No <Dimension2>
333 x No <Dimension2>
No <Dimension1> x 444
No <Dimension1> x 555
No <Dimension1> x 666



all intersections



111 x 444
111 x 555
111 x 666
222 x 444
222 x 555
222 x 666
333 x 444
333 x 555
333 x 666
111 x No <Dimension2>
222 x No <Dimension2>
333 x No <Dimension2>
No <Dimension1> x 444
No <Dimension1> x 555
No <Dimension1> x 666
AAA x CCC
AAA x 444
AAA x 555
AAA x DDD
AAA x 666
AAA x No <Dimension2>
111 x CCC
111 x DDD
222 x CCC
222 x DDD
BBB x CCC
BBB x 444
BBB x 555
BBB x DDD
BBB x 666
BBB x No <Dimension2>
333 x CCC
333 x DDD
No <Dimension1> x CCC
No <Dimension1> x DDD

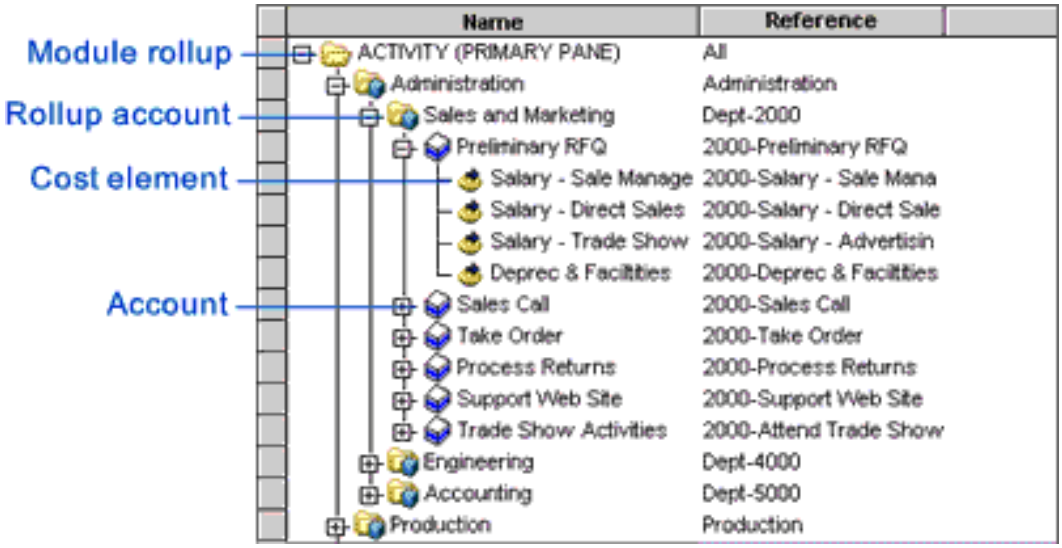


(too many accounts to show)

Rollup Accounts and Module Rollups

A rollup account is a cluster of accounts or a cluster of other rollup accounts that are related by function, department, location, or group. The cost of a rollup account is the sum of costs for all of the accounts and rollup accounts in the immediate subordinate level. Each module contains a module rollup. A module rollup is the highest level in the module. A module rollup represents all the accounts and rollup accounts in the module.

Each module contains a module rollup. A module rollup is the highest level in the module. A module rollup represents all the accounts and rollup accounts in the module.



Accounts contain cost elements. For information on cost elements, see Chapter 30, "What Are Cost Elements?," on page 259.

Using All Intersections

As discussed previously, the **all intersections** selection can create accounts that include both a parent node and a leaf node. (See "All Intersections" on page 226.)The question arises, then, as to why you might want to do this. To provide an answer, we first show some accounts created with the **leaf dimension members** option. The following picture shows the following four leaf-node-only accounts:

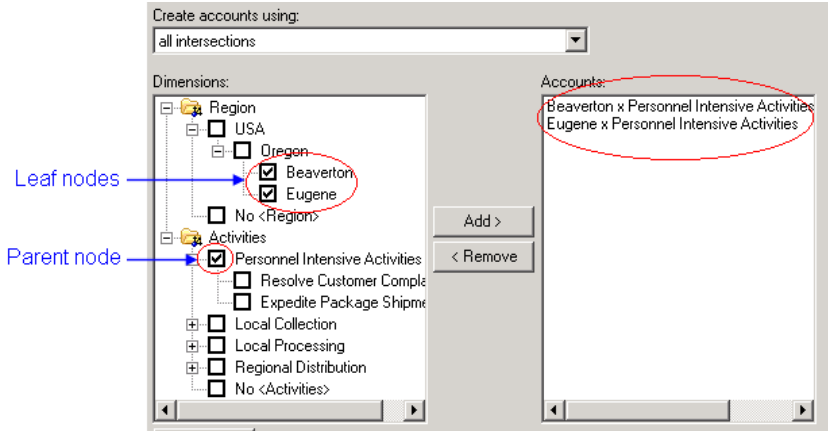
Beaverton x Resolve Customer Complaints	Eugene x Resolve Customer Complaints
Beaverton x Expedite Package Shipments	Eugene x Expedite Package Shipments

Display Name	IntsctnName	Cost
ACTIVITY (PRIMARY PANE)		$3,647,900.00
USA	USA x All	$3,647,900.00
Oregon	Oregon x All	$3,647,900.00
Beaverton	Beaverton x All	$1,930,900.00
Personnel Intensive Activities	Beaverton x Personnel Intensive Activities	$218,201.72
Resolve Customer Complaints	Beaverton x Resolve Customer Complaints	$128,901.18
Expedite Package Shipments	Beaverton x Expedite Package Shipments	$89,300.54
Local Collection	Beaverton x Local Collection	$338,040.02
Local Processing	Beaverton x Local Processing	$716,731.93
Regional Distribution	Beaverton x Regional Distribution	$1,682,395.54
Eugene	Eugene x All	$1,717,000.00
Personnel Intensive Activities	Eugene x Personnel Intensive Activities	$187,783.21
Resolve Customer Complaints	Eugene x Resolve Customer Complaints	$121,992.50
Expedite Package Shipments	Eugene x Expedite Package Shipments	$65,790.71
Local Collection	Eugene x Local Collection	$351,422.14
Local Processing	Eugene x Local Processing	$743,326.79
Regional Distribution	Eugene x Regional Distribution	$1,504,169.86

Next, the following picture shows how to add two mixed-node accounts with the **all intersections** option:

Beaverton x Personnel Intensive Activities (leaf node x parent node)

Eugene x Personnel Intensive Activities (leaf node x parent node)



And, the following picture shows those two mixed-node accounts added to the module

Display Name	IntsctnName	Cost
ACTIVITY (PRIMARY PANE)		$3,647,900.00
USA	USA x All	$3,647,900.00
Oregon	Oregon x All	$3,647,900.00
Beaverton	Beaverton x All	$1,930,900.00
Personnel Intensive Activities	Beaverton x Personnel Intensive Activities	$218,201.72
Personnel Intensive Activities	Beaverton x Personnel Intensive Activities	$0.00
Resolve Customer Complaints	Beaverton x Resolve Customer Complaints	$128,901.18
Expedite Package Shipments	Beaverton x Expedite Package Shipments	$89,300.54
Local Collection	Beaverton x Local Collection	$338,040.02
Local Processing	Beaverton x Local Processing	$716,731.93
Regional Distribution	Beaverton x Regional Distribution	$1,682,395.54
Eugene	Eugene x All	$1,717,000.00
Personnel Intensive Activities	Eugene x Personnel Intensive Activities	$187,783.21
Personnel Intensive Activities	Eugene x Personnel Intensive Activities	$0.00
Resolve Customer Complaints	Eugene x Resolve Customer Complaints	$121,992.50
Expedite Package Shipments	Eugene x Expedite Package Shipments	$65,790.71
Local Collection	Eugene x Local Collection	$351,422.14
Local Processing	Eugene x Local Processing	$743,326.79
Regional Distribution	Eugene x Regional Distribution	$1,504,169.86

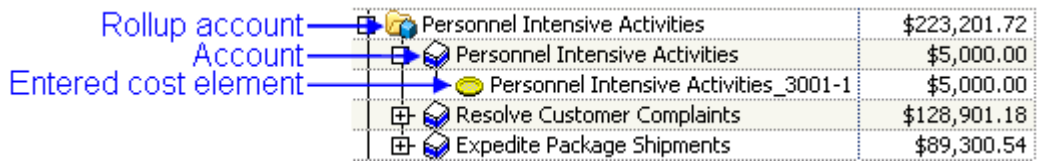
Because these accounts have just added to the module, they initially do not have any cost. So, after adding an entered cost element to each of those accounts and recalculating, the costs and rollup costs are as shown in the following picture.

For information on entered cost elements, see "Introduction" on page 259.

Display Name	IntsctnName	Cost
ACTIVITY (PRIMARY PANE)		$3,662,900.00
USA	USA x All	$3,662,900.00
Oregon	Oregon x All	$3,662,900.00
Beaverton	Beaverton x All	$1,935,900.00
Personnel Intensive Activities	Beaverton x Personnel Intensive Activities	$223,201.72
Personnel Intensive Activities	Beaverton x Personnel Intensive Activities	$5,000.00
Personnel Intensive Activities_3001-1		$5,000.00
Resolve Customer Complaints	Beaverton x Resolve Customer Complaints	$128,901.18
Expedite Package Shipments	Beaverton x Expedite Package Shipments	$89,300.54
Local Collection	Beaverton x Local Collection	$338,040.02
Local Processing	Beaverton x Local Processing	$716,731.93
Regional Distribution	Beaverton x Regional Distribution	$1,682,395.54
Eugene	Eugene x All	$1,727,000.00
Personnel Intensive Activities	Eugene x Personnel Intensive Activities	$197,783.21
Personnel Intensive Activities	Eugene x Personnel Intensive Activities	$10,000.00
Personnel Intensive Activities_3002-1		$10,000.00
Resolve Customer Complaints	Eugene x Resolve Customer Complaints	$121,992.50
Expedite Package Shipments	Eugene x Expedite Package Shipments	$65,790.71
Local Collection	Eugene x Local Collection	$351,422.14
Local Processing	Eugene x Local Processing	$743,326.79
Regional Distribution	Eugene x Regional Distribution	$1,504,169.86

In the preceding picture, you can notice two things:

- The cost for the rollup account, Personnel Intensive Activities, includes the cost of the account, Personnel Intensive Acitivities and two other accounts: Resolve Customer Complaints and Expedite Package Shipments. Do not confuse the rollup account, Personnel Intensive Activities with the account, Personnel Intensive Activities.



- The cost of the account, Personnel Intensive Activities comes (in this example) from an entered cost element. The cost of the account does not include the cost of the two other accounts: Resolve Customer Complaints and Expedite Package Shipments. The account, Personnel Intensive Activities, is not a rollup account.

Creating and Deleting Accounts

An account corresponds to dimension members. You must create the dimension member before you can create the account that corresponds to the dimension member.

When you delete an account, the corresponding dimension member still exists, and the dimension member will appear in any cube that you generate. To remove the dimension member from a cube, you must delete the dimension member.

Note: When you delete an account, the model to which it belongs can be viewed but cannot be edited while the delete is in progress.

Guidelines for Creating Accounts

Guidelines for Creating Accounts

- An account can consist of the intersection of at most 30 user-defined dimensions (including both structural dimensions and attribute dimensions).
- Combine similar accounts.

  If your organization's general ledger lists the details of travel expenses, such as airfare, hotel, or entertainment, you might want to combine these expenses into one travel account. Are the particular costs incurred together, or are multiple costs caused by the same factor? If so, you might want to combine such costs. Also, if some accounts have small costs, you might want to combine the small accounts into larger accounts that represent more general categories.

  Combining accounts into a single account can make creating and maintaining a model easier. However, you must ensure that combining accounts does not hinder business users from obtaining the information that they need.
- Group related accounts.

  Group accounts into roll-up accounts if the accounts have similar functionality or are linked to similar activities. For example, the following ungrouped accounts could be grouped as shown:

Ungrouped Accounts	Grouped Accounts
Magazine Advertising Inserts	Advertising
Direct Mail Advertising	Print
In-Store Advertising Circulars	Magazine Inserts
Radio Advertising	Direct Mail
TV Advertising	In-Store Circulars
Yellow Pages Advertising	Radio
	TV
	Yellow Pages

Grouping accounts is similar to the advice concerning the use of multiple dimensions in Chapter 6, Dimensions. A group, and the levels within a group, provide business users with more flexibility to analyze costs at different levels. In the previous example, business users can examine the individual costs of adverting through Magazine Inserts, Direct Mail, and In-Store Circulars. Business users can also examine the total cost of these methods at the Print level, and the total cost of advertising, regardless of the method.

- In OLAP analysis, the lowest level that can be examined in a cube is the accounts; cost element costs are rolled up into accounts and cannot be examined individually. Therefore, ensure that cost information that must be available to business users is in accounts, not in cost elements.

Guidelines for Creating Cost Object Accounts

Before you create cost object accounts, consider the following:

- Name and organize the cost objects so that they are familiar to the business users.

Guidelines for Creating Activity Accounts

Before you create activity accounts, consider the following:

- Name and organize the activity accounts so that they are familiar to the business users.
- Distinguish an activity from a task and a process from an activity. A task provides too much detail and a process provides too little detail.

  Follow the 5-50 rule:

  If a potential activity consumes less than 5% of anyone's time, then it is probably a task. Combine tasks to create an activity or several activities. For example, the tasks of opening correspondence, typing correspondence, and applying postage can be combined into the activity of Communicate with Customers.

  However, understanding the task level is important because it clarifies the scope of an activity, it clarifies the boundaries between adjacent activities, and it helps people in your organization relate to the activity.

  If a potential activity consumes more than 50% of anyone's time, then it is probably a business process (or business macro process). Divide the activity into finer detail to identify the activities that are involved in the process. For example, the process of Secure Facility can be divided into the activities of Patrol the Grounds, Monitor Automobile Traffic, and Issue Security Badges.

- An activity account must be active; it is not an arbitrary accounting bucket. The action is denoted by naming an activity account using a verb-noun phrase, such as Process Order or Enter Invoices.
- An activity account is a cost category that management is concerned about.
- Ensure that all required activity accounts are defined.

  To determine your organization's activities, conduct surveys with the people in your organization.
- Create an activity account only if management needs to know details about the activity to make decisions.
- An activity account has one driver that is used for assigning costs.

Guidelines for Creating Resource Accounts

Before you create resource accounts, consider the following:

- Name and organize the resource accounts so that they are familiar to the business users. Create the references for resource accounts and cost elements so that the references match the account numbers in your organization's chart of accounts.

See Also

"New Account Dialog Box" on page 235

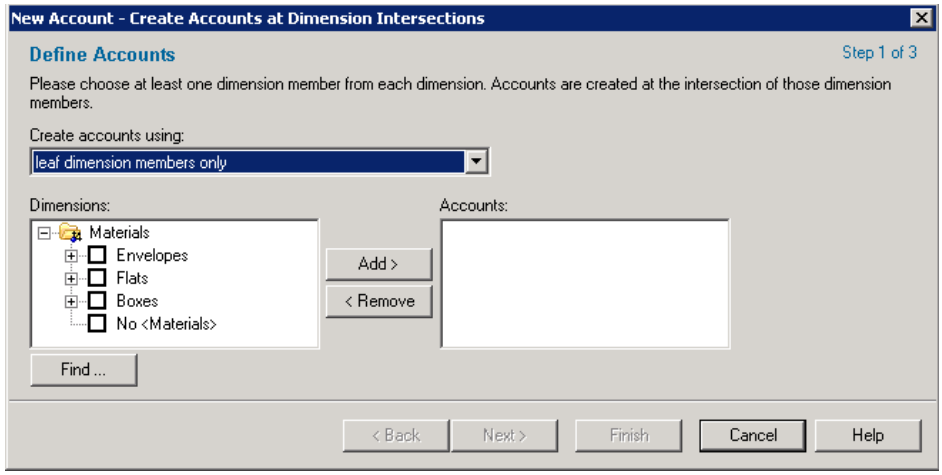
Chapter 28

Windows for Accounts

New Account Dialog Box

About the New Account Dialog Box

Use the New Account dialog box to create an account as an intersection of dimension members in a module.



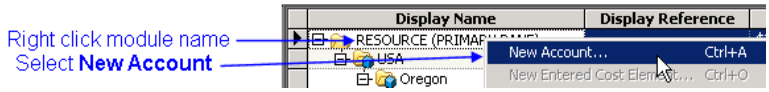
See Also

- Chapter 27, "What are Accounts?," on page 223
- "Create Accounts" on page 241

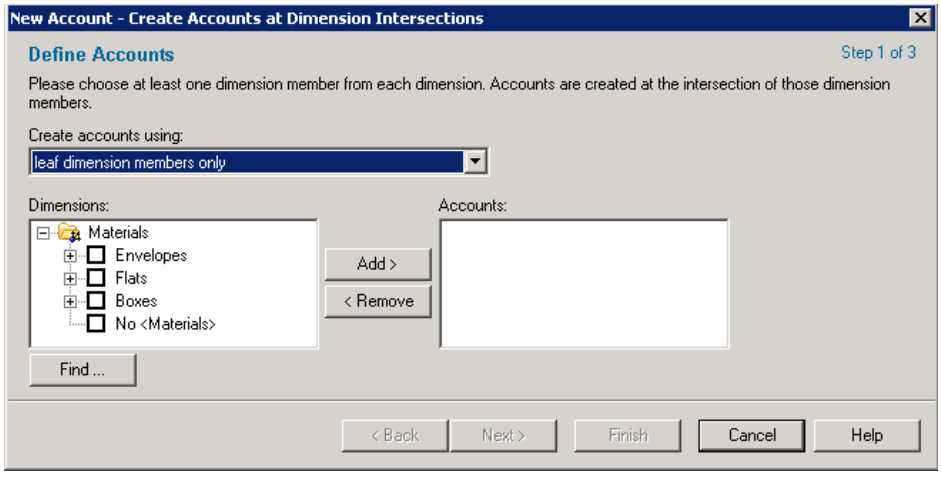
How to Access the New Account Dialog Box

To open the New Account dialog box:

1. Open a module view. See "Introduction to Modules" on page 179.
2. Do one of the following:
   - Select **Edit** ⇨ **New Account**
   - Right-click the module name and select **New Account**.



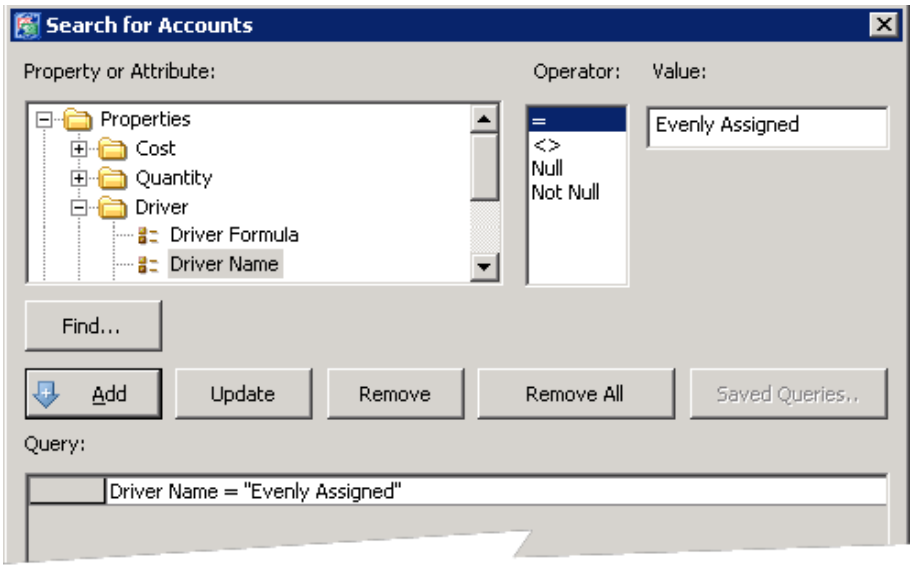
The New Account dialog opens.



Search for Accounts Dialog Box

About the Search for Accounts Dialog Box

In the Search for Accounts dialog box, you can specify criteria to build and update a query to search for accounts. This is useful when you build a model.



How to Access the Search for Accounts Dialog Box

Open a model and select **Edit** ⇨ **Search Accounts** or click the Search Accounts icon  .

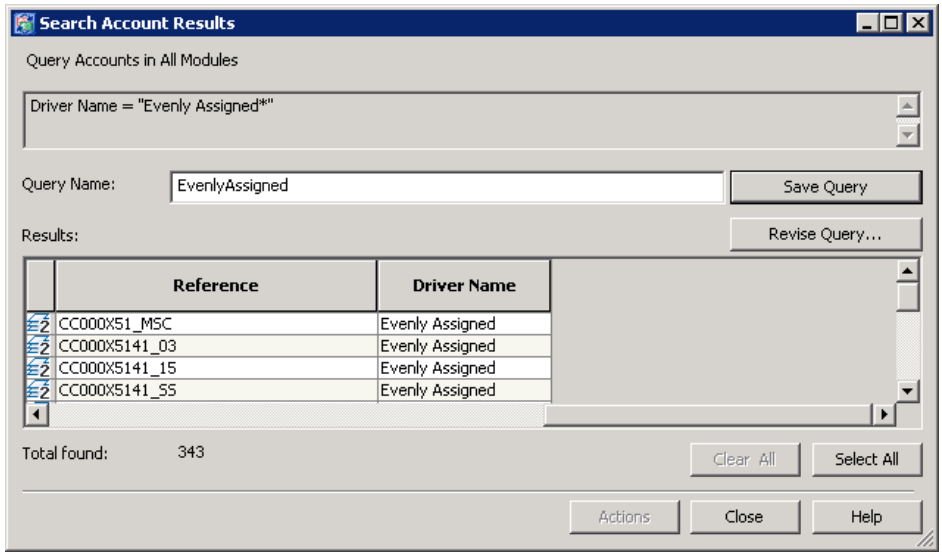
See Also

- "Search for Accounts" on page 248
- "Search Account Results Dialog Box" on page 238
- "Save Queries for Account Search" on page 252
- "Find a Property or Attribute Dialog Box" on page 303

Search Account Results Dialog Box

About the Search Account Results Dialog Box

In the Search Account Results dialog box, you can see the results of the query that you built in the Search for Accounts dialog box. The final search criteria are displayed in the non-editable Query box.



How to Access the Search Account Results Dialog Box

From the Search for Accounts dialog box, click **Search** to initiate a search. When the search completes, the results are displayed..

See Also

- "Search for Accounts" on page 248
- "Search for Accounts Dialog Box" on page 237
- "Save Queries for Account Search" on page 252
- "Find a Property or Attribute Dialog Box" on page 303

Item (Account) Properties and Attributes Dialog Box

About the Item Properties and Attributes Dialog Box

In the Item Properties and Attributes dialog box, you can see the properties and attributes of an account.

Note: The availability of these features depends on your permissions.

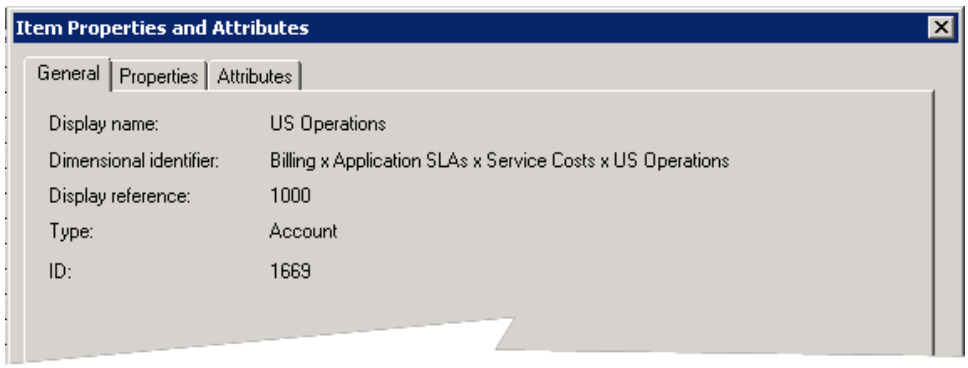
The Item Properties and Attributes dialog box contains the following tabs:

- "General Tab" on page 239
- "Properties Tab" on page 239
- "Attributes Tab" on page 240

How to Access the Item Properties and Attributes Dialog Box

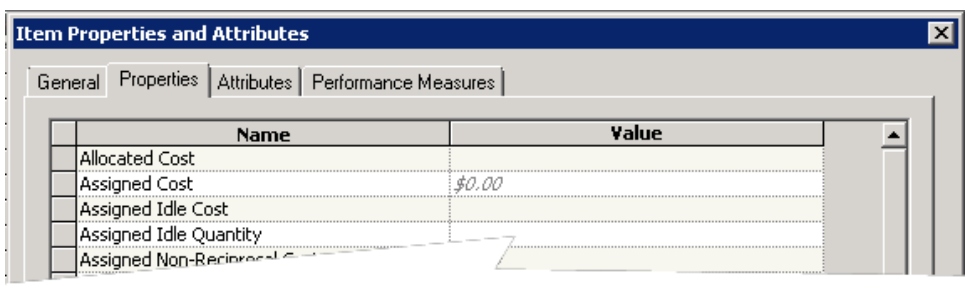
In a module view, select an item and then select **Edit** ⇨ **Item Properties**.

General Tab



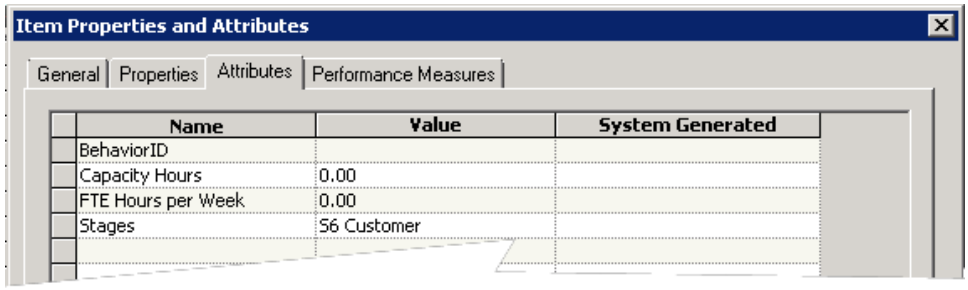
The General tab displays the account name, reference, display reference, ID, and identifies whether it is an account or rollup account.

Properties Tab



The Properties tab displays the values of properties for the account. If the value is modifiable by a user, then you can modify it here.

Attributes Tab



The Attributes tab displays the values of attributes that are attached to the account and allows you to modify the values.

An attribute on an account is **System Generated** when it is added by the system to the account as a result of your having added the attribute to a dimension member that is in the dimension signature of that account.

Note: If you change the value of a system-generated attribute from its default value, then it is no longer a system-generated attribute. See "System-generated versus user-entered attribute values" on page 285.

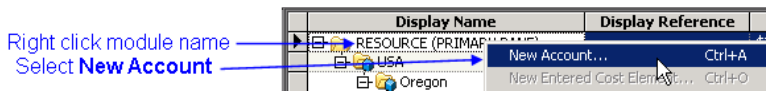
Chapter 29

How To: Accounts

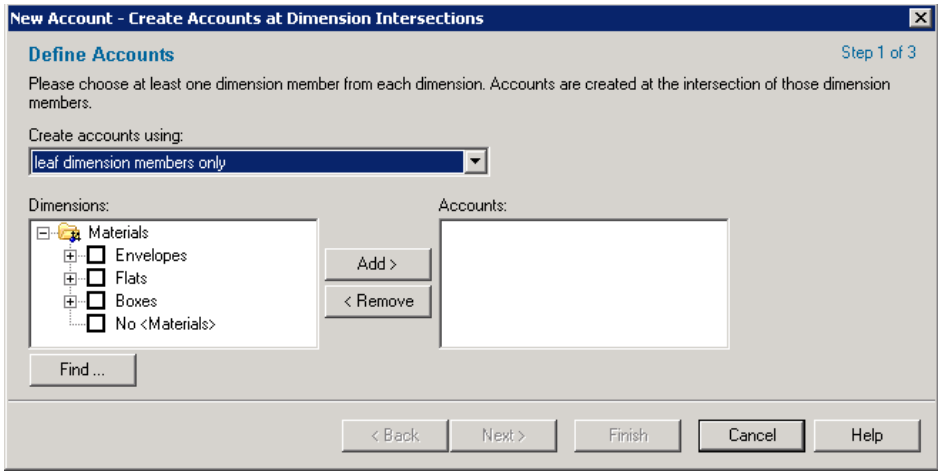
Create Accounts

To create an accounts:

1. Open a module view. See "Introduction to Modules" on page 179.
2. Do one of the following:
   - Select **Edit** ⇨ **New Account**
   - Right-click the module name and select **New Account**.



The New Account dialog opens.



3. From the list of **Dimensions**, select at least one dimension member from each dimension.

   TIP To quickly locate a dimension member, click **Find** in the New Account Wizard. The Find Dimension Member dialog box appears.

   Accounts are created at the intersection of the dimension members that you select.

   See "Introduction" on page 223.

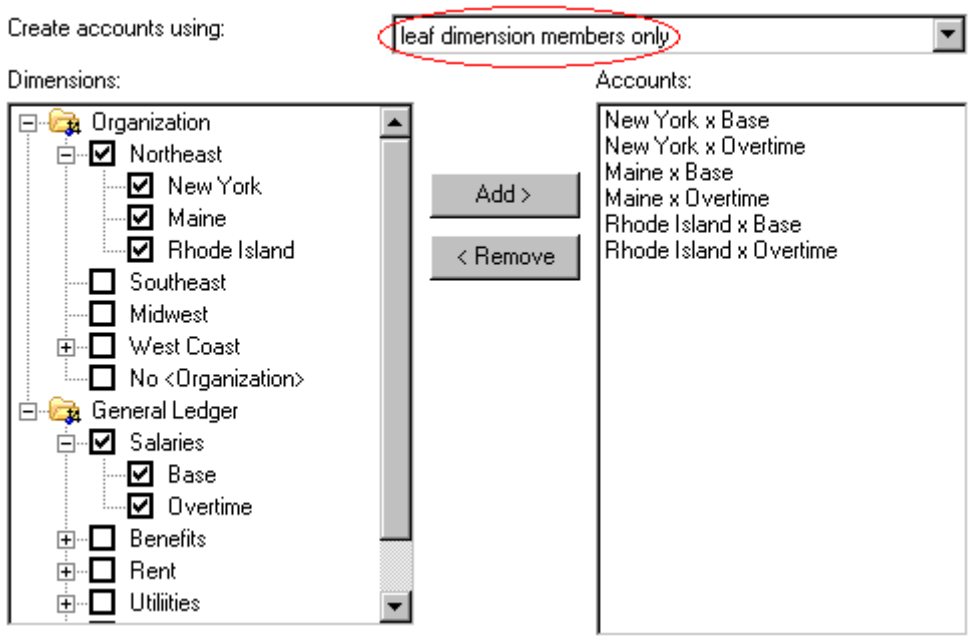
4. From the **Create accounts using** drop-down list, select one of the following:

   - leaf dimension members only
   - parent dimension members only
   - all intersections

   Note: These selections do not apply to the "No" dimension member. If you select the "No" dimension member (for example, No <Channel>) then it is included in dimension crossings regardless of whether you select **leaf dimension members only**, **parent dimension members only**or **all intersections**.

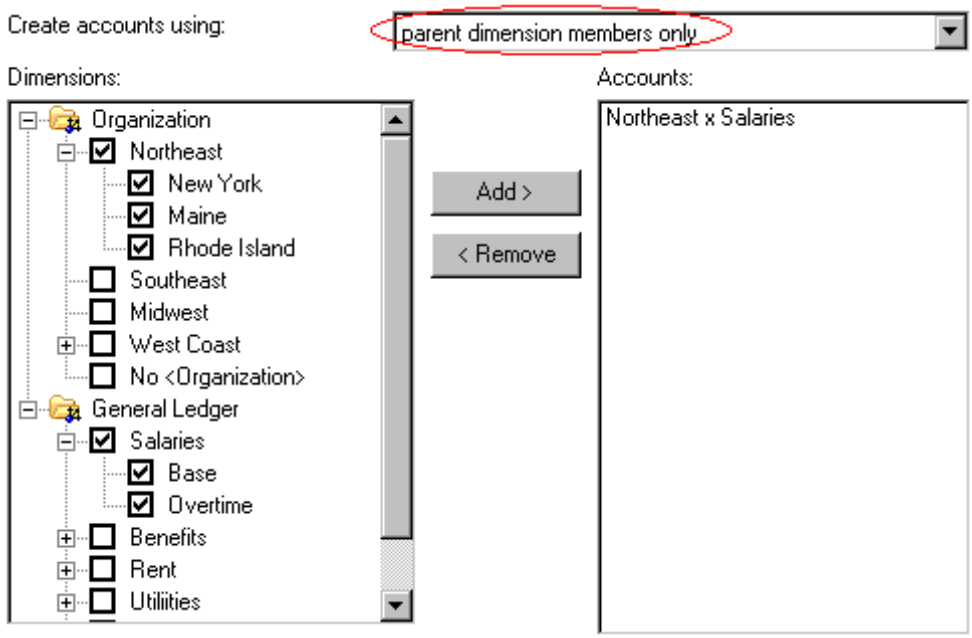
   See "Accounts as Dimension-Member Crossings" on page 224.

   leaf dimension members only
   creates accounts only for the intersections of the lowest dimension members in each dimension, as shown:



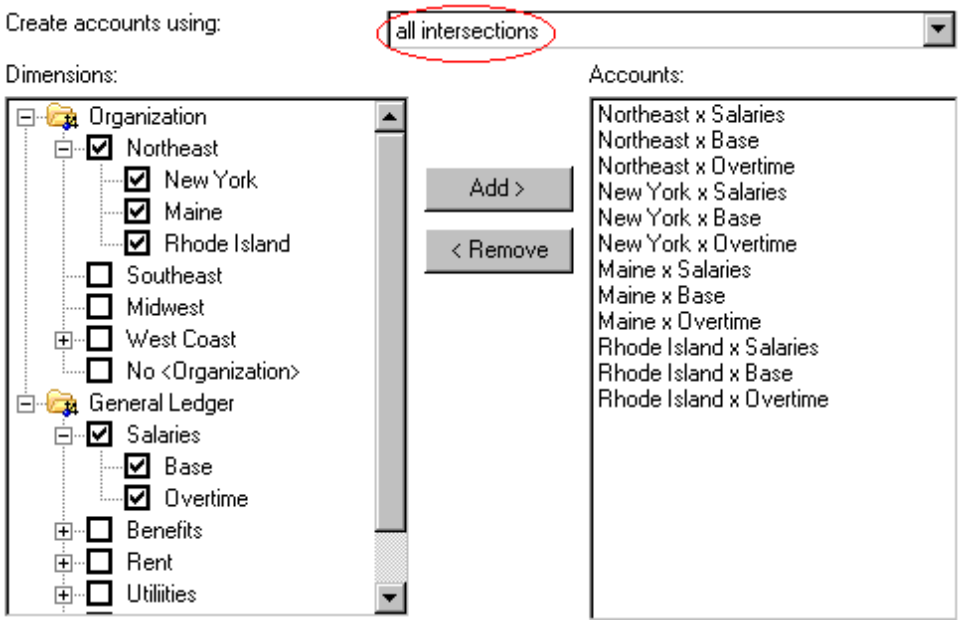
parent dimension members only

creates accounts only for the intersections of the highest dimension members



all intersections

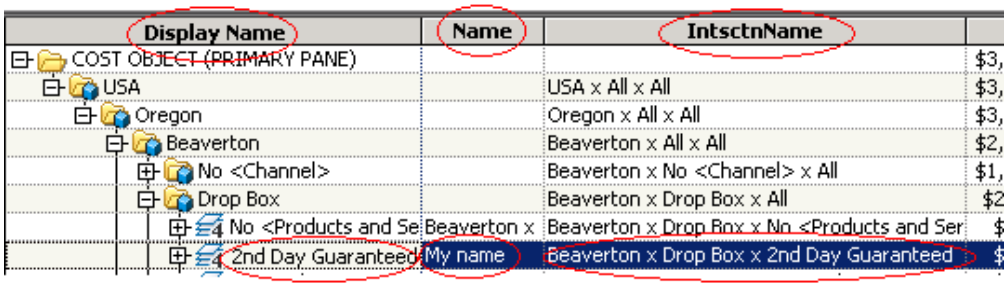
creates accounts for the intersections of all the dimension members in each dimension, as shown:



5. To change the name of an account, click in the **Name** column and type a new name. The name must follow the naming conventions. See "General naming conventions" on page 69.

   Note: You can change the name of an account but you can not change its display name—these are two different properties. The following picture makes the difference clear.

   - The system uses the display name in its graphical representation of the hierarchical structure of a module.
   - The name is a string of your choosing that you can add to a column layout. See "What is a Column Layout?" on page 319.
   - The intersection name shows the dimension members whose intersection constitutes an account.

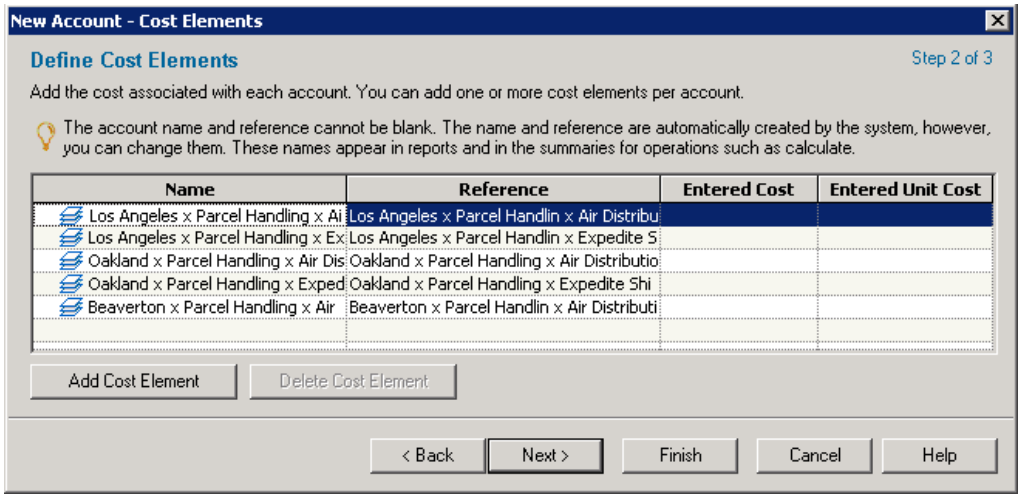


6. To change the reference of an account, click in the **Reference** column and type a new reference.

   The reference must follow the reference conventions. See "Account reference conventions" on page 74,

7. Click **Finish** or click **Next**.

   If you click **Next**, then the New Account — Cost Elements dialog appears. This dialog allows you to create cost elements for accounts.



8. To create a cost element for an account, select the account and click **Add Cost Element**.

   See also:

   - "New Entered Cost Element Dialog Box" on page 263
   - Chapter 30, "What Are Cost Elements?," on page 259
   - "Create an Entered Cost Element" on page 265

   To change the name of a cost element, click in the **Name** column and type a new name. The name must follow the naming conventions. See "Entered cost element naming conventions" on page 72.

   To change the reference of a cost element account, click in the **Reference** column and type a new reference. The reference must follow the reference conventions. See "Entered cost element reference conventions" on page 75,

9. Click **Finish**.

   The accounts are created as intersections of the selected dimension members.

Reorder Accounts

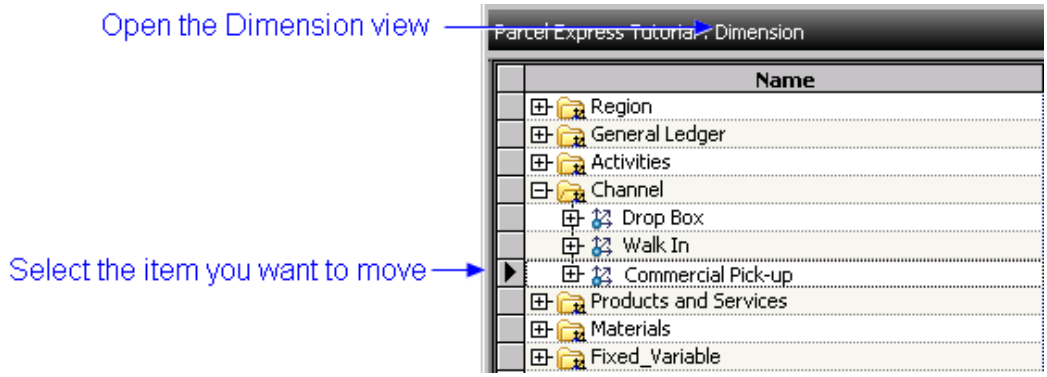
Overview

You can control the display order of accounts by reordering dimension members on the Dimensions view. The order that you establish is preserved when you export and import a model.

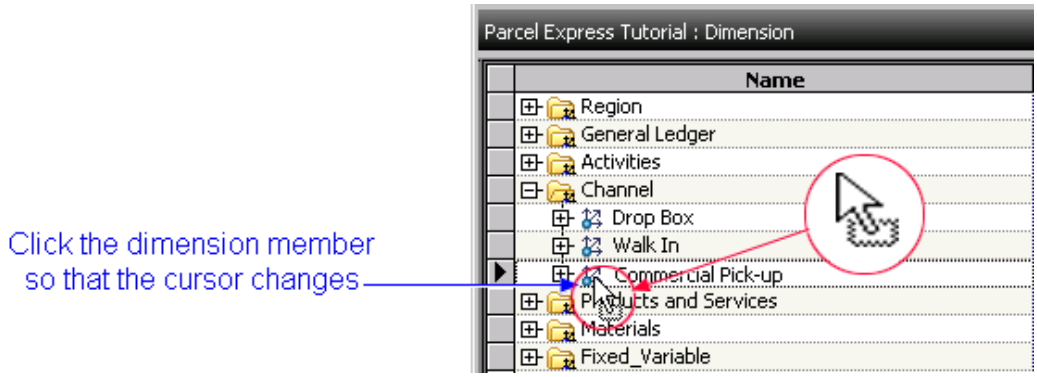
You can reorder accounts using the mouse, keyboard, or staging tables.

Using the Mouse

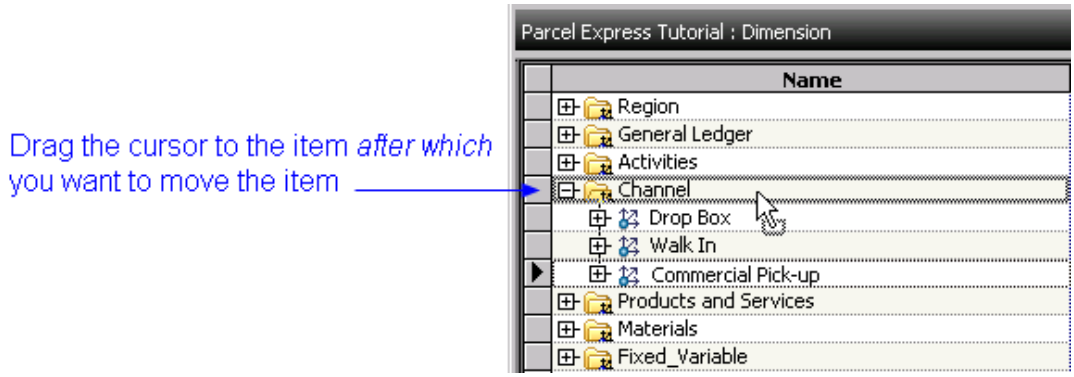
1. Go to the Dimensions view and select the dimension member that you want to reorder.



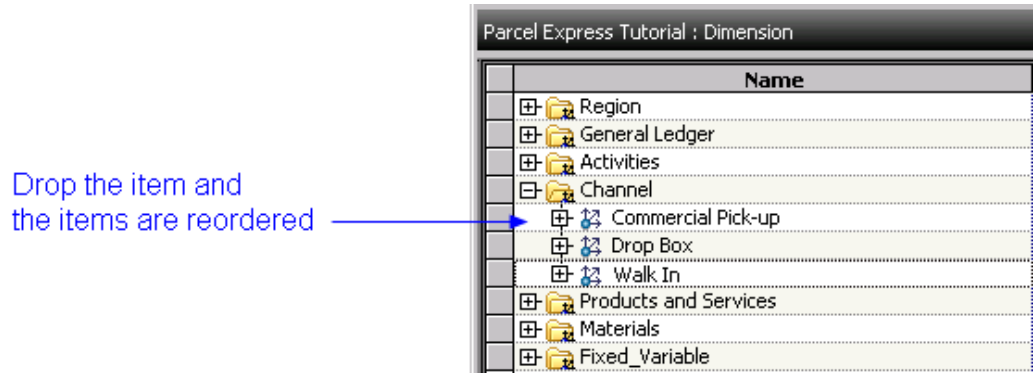
2. Hover over the dimension member until the cursor changes to the following:



3. Drag the cursor to the dimension member after which you want to move the selected item.

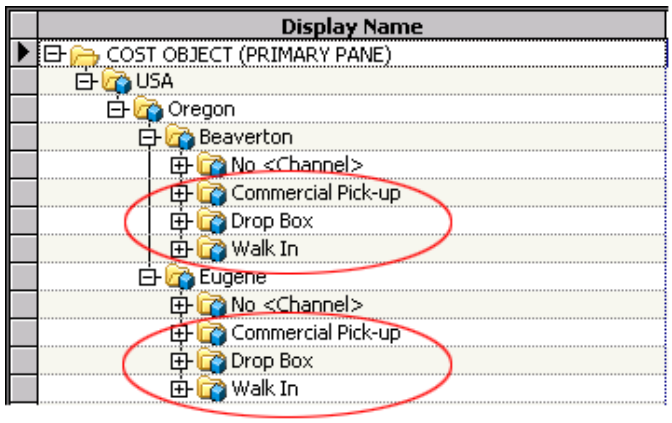


4. Drop the item and the dimension members are reordered.

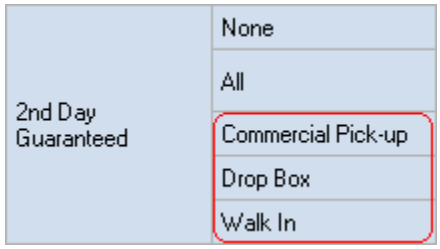


Note: If you drag a dimension member out of its parent, then the system assumes that you want to change the parent of the dimension member. For more information on re-parenting, see Changing the Parent of a Dimension Member.

When you open the appropriate assignments pane, you can notice that the accounts are reordered.



And the new order is reflected in cubes:



Using the Keyboard

To reorder an account using the keyboard, do the following:

1. Go to the Dimensions view.
2. Use the up or down arrow keys to navigate to the dimension member that you want to reorder.
3. Press **CTRL + D** to select the dimension member.
4. Use the up or down arrow keys to navigate to the dimension member after which you want to place the dimension member.

5. Press **CTRL + R** to drop the item.

   The item is moved to the new position.

See Also

"Change the Parent of a Dimension Member" on page 215

Using Staging Tables

To specify the display order of dimension members using staging tables, use the DisplayOrder field in the DimensionMember table.

For information on staging tables, see Chapter 13, "About Staging Tables," in *SAS Cost and Profitability Management: Data Administration Guide*..

Search for Accounts

Invoke the Search Accounts Dialog Box

To search for accounts, open a model and do either of the following:

- Select **Edit** ⇨ **Search Accounts**
- Click the Search Accounts icon 

Note: If a network user from a different domain (than that of the SAS Cost and Profitability Management client machine) logs in to the client machine to search for accounts, then that user must be granted WRITE access to the [ClientInstallpath]\bin folder for the operation to succeed.

Build a Query to Search for Accounts

1. Select a Property or Attribute.
2. Select an Operator and specify a Value (numeric, Boolean, text, enumerated values, sets) to add criteria to your query.
3. Click **Add** to add the line to your query. The combined search criteria display in the Query box.
4. Repeat Steps 1 through 3 as necessary to build your query.
5. Filter the search results by selecting options in the Search for and Search in fields. You can select the **Use * as wildcard** check box and limit the result count in multiples of 50.
6. Click **Search**. The Search Account Results dialog box appears.

   You can update an existing query to search for accounts.

Update Your Query to Search for Accounts

1. Select the line in the Query box that you want to change and modify the criteria as necessary.

2. Click **Update** to display the updated search criteria in the Query box.
3. To remove a single line from the query, select the line in the Query box that you want to delete and click **Remove**. To remove all the lines of the query, click **Remove All**.

   TIP To quickly recall the last query that you used, click **Get Last**.

Sort the Search Results

To sort the results by column, click the column heading.

Perform an Action on the Accounts That Are Found

Overview

Select one or more accounts from the list of search results, and click **Actions** to perform any of the following actions:

- "Go to an Account" on page 249
- "Add Accounts" on page 249
- "Delete Accounts" on page 249
- "Add Attributes and Values" on page 250
- "Remove Attributes" on page 250
- "Mark Accounts as Profitability Management Behaviors" on page 251
- "Unmark Accounts as Profitability Management Behaviors" on page 251
- "Delete Assignments" on page 251

Go to an Account

1. Select an account from the list of results.
2. Click **Actions** ⇨ **Go to Account** to go to the selected account.

Add Accounts

1. Select one or more accounts from the list of results.
2. Click one of the following:

   Actions ⇨ **Add Accounts in Left Pane**

   Actions ⇨ **Add Accounts in Right Pane**

   Note: These actions are disabled on the **Actions** menu if the appropriate pane is not already open.

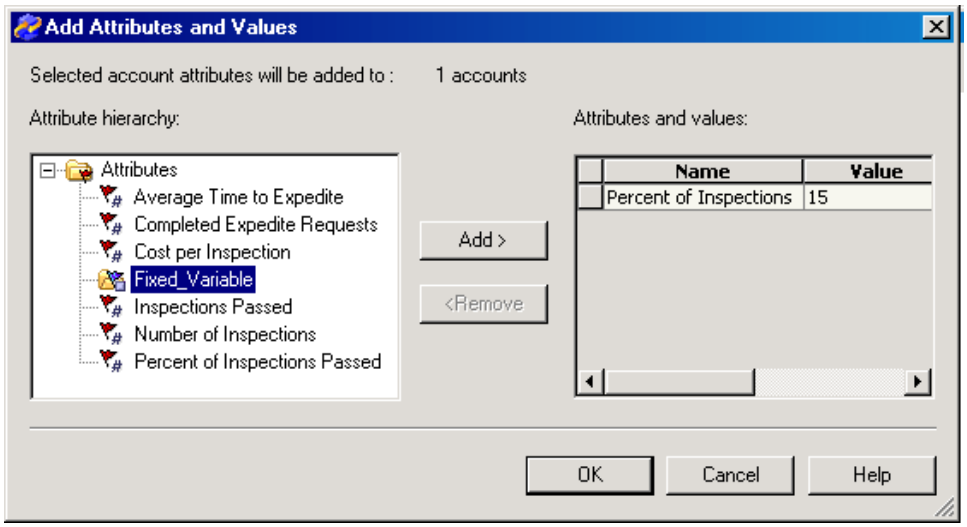
Delete Accounts

1. Select one or more accounts from the list of results.
2. Click **Actions** ⇨ **Delete Accounts** to delete the selected accounts. You are asked to confirm the deletion before the actual deletion takes place.

Note: When you delete an account, the model to which it belongs can be viewed but cannot be edited while the delete is in progress.

Add Attributes and Values

1. Select one or more accounts from the list of results.
2. Click **Actions** ⇨ **Add Attributes and Values**. The Add Attributes and Values dialog box opens.



3. Select each attribute that you want to add to the selected accounts, and then click Add.
4. Enter a new value in the **Value** column if you want to change the value of the attribute.

   Note: The value that is displayed in the **Value** column is the default value for an attribute -not its current value even if the account already has that attribute. An existing attribute is reset to its default value unless you specify a new value.

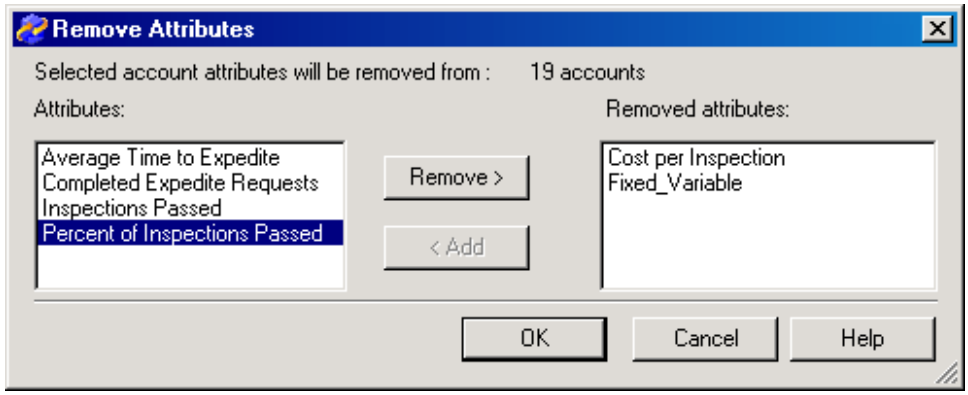
   Note: You cannot change Boolean attributes or calculated attributes. A calculated attribute retains its formula. If you want to change the formula of a calculated attribute, you must go to the Attributes view and change the formula from there.
5. Click **OK**. The attributes (along with their values) are added to the selected accounts.

   Note: This method of adding attributes to accounts is an efficient method for adding stage attributes to multiple accounts to create multi-stage contribution cubes.

Remove Attributes

1. Select one or more accounts from the list of results.
2. Click **Actions** ⇨ **Remove Attributes**.

   The Remove Attributes dialog box opens, showing all the attributes from all selected accounts.



3. Select each attribute that you want to remove from the selected accounts, and click **Remove**.

   If you have selected multiple accounts, and an attribute that you select for removal belongs to more than one of the selected accounts, then the attribute is removed from all of those accounts.

4. Click **OK**. The attributes are removed from the selected accounts.

Mark Accounts as Profitability Management Behaviors

1. Select one or more accounts from the list of results.
2. Click **Actions** ⇨ **Mark Accounts as Behaviors**. A message is displayed if the action is successful.

   Note: Rollup accounts cannot be marked as behaviors.

Once you have marked accounts as behaviors, you can publish them to SAS Profitability Management.

See also:

- "Mark Accounts as Behaviors" on page 729
- Chapter 25, "Working with SAS Profitability Management," in *SAS Cost and Profitability Management: Data Administration Guide*

Unmark Accounts as Profitability Management Behaviors

1. Select one or more accounts from the list of results.
2. Click **Actions** ⇨ **Unmark Accounts as Behaviors**. A message is displayed if the action is successful.

See also:

- "Mark Accounts as Behaviors" on page 729
- Chapter 25, "Working with SAS Profitability Management," in *SAS Cost and Profitability Management: Data Administration Guide*

Delete Assignments

1. Select one or more accounts from the list of results.
2. Click one of the following:

   Actions ⇨ **Delete Incoming Assignments** to delete assignments coming into the selected accounts.

Actions ⇨ Delete Outgoing Assignments to delete assignments going out of the selected accounts.

Actions ⇨ Delete Incoming and Outgoing Assignments to delete both sorts of assignments.

Note: These options are not available for rollup accounts. You are asked to confirm the deletion before the actual deletion takes place.

See Also

- "Search for Accounts Dialog Box" on page 237
- "Search Account Results Dialog Box" on page 238
- "Save Queries for Account Search" on page 252

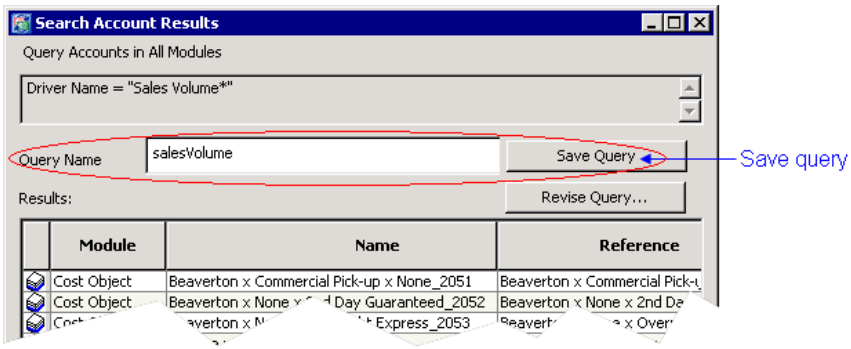
Save Queries for Account Search

Queries for searching for accounts can be long, complicated, and tedious to construct. Now you can save your queries, retrieve them for subsequent use, export them, and import them.

Save a Query

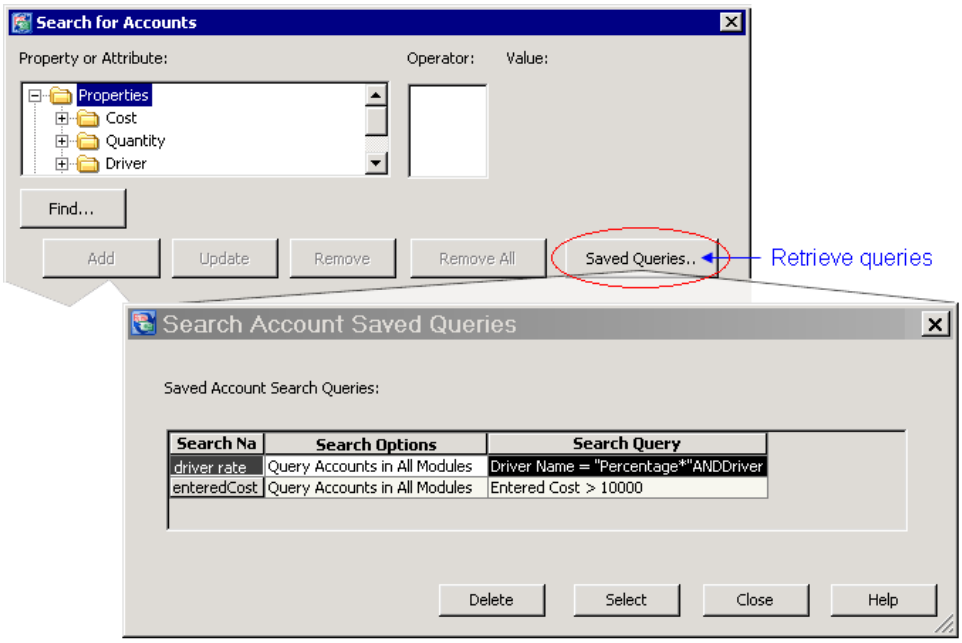
Once a query has returned the results that you want, click **Save Query** on the Search Account Results dialog box to save the query.

A saved query is not associated with a particular model or a particular user but rather with the machine on which it is saved. Any SAS Cost and Profitability Management user who uses that machine can see all the queries saved on that machine. In order to use the queries on another machine, you can export the queries and import them on the other machine.



Retrieve Queries

To retrieve a query for subsequent use, click **Saved Queries** on the Search for Accounts dialog box. All those queries that have been saved on that machine are displayed for selection and reuse.

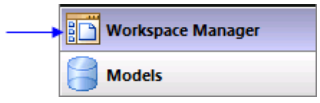


Export Account Search Queries

In order to use the queries on another machine, you can export the queries and import them on the other machine. When you export, all the queries saved on the machine are exported regardless of which users saved them. The exported queries may be imported by any user.

To export your queries:

1. Go to the Workspace Manager.



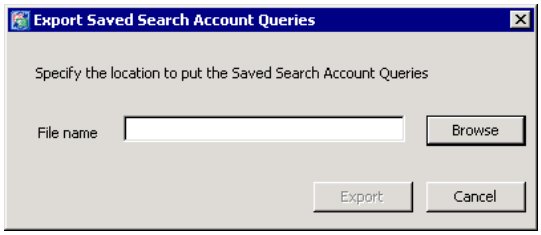
2. Select **Files** ⇨ **Export** ⇨ **Account Queries**.

   The Export Saved Search Account Queries dialog box opens.

3. Specify the name of the Registry file to which the queries are to be saved and click **Export**.

   Note:

   - It is not necessary to specify the file extension because it is added automatically. The default extension is .reg.
   - If you do not type a path before the file name, then the file is saved in <*installation directory*>\SASCostandProfitabilityManagementClient\8.3\bin.
   - You can click the Browse button to browse to the directory in which you want to save the export file. Again, it is not necessary to specify the file extension because it is added automatically. The default extension is .reg.

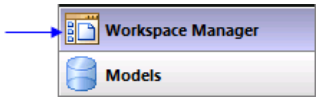


Import Account Search Queries

When you import queries on a machine, any SAS Cost and Profitability Management user who uses that machine can see all the imported queries. When you import saved queries, the imported queries replace any queries that were previously saved on that machine.

To import queries:

1. Go to the Workspace Manager.

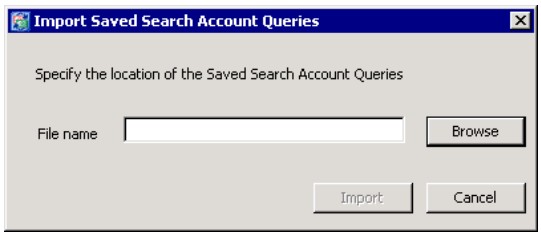


2. Select **Files** ⇨ **Import** ⇨ **Account Queries**.

   The Import Saved Search Account Queries dialog box opens.

3. Specify the name of, or browse to the file to be imported and click **Import**.

   Note: When you import saved queries, the imported queries replace any queries that were previously saved on that machine.



See Also

- "Search for Accounts Dialog Box" on page 237
- "Search Account Results Dialog Box" on page 238

Go to an Account

You can quickly go to an account that is displayed in the left assignments pane or the right assignments pane.

1. Select an account in the left assignments pane or the right assignments pane.
2. Select **Edit** ⇨ **Go To Account**.

The selected account is displayed in the primary pane. You can display the source accounts and the destination accounts for the selected account.

See Also

"Assignments Panes" on page 433

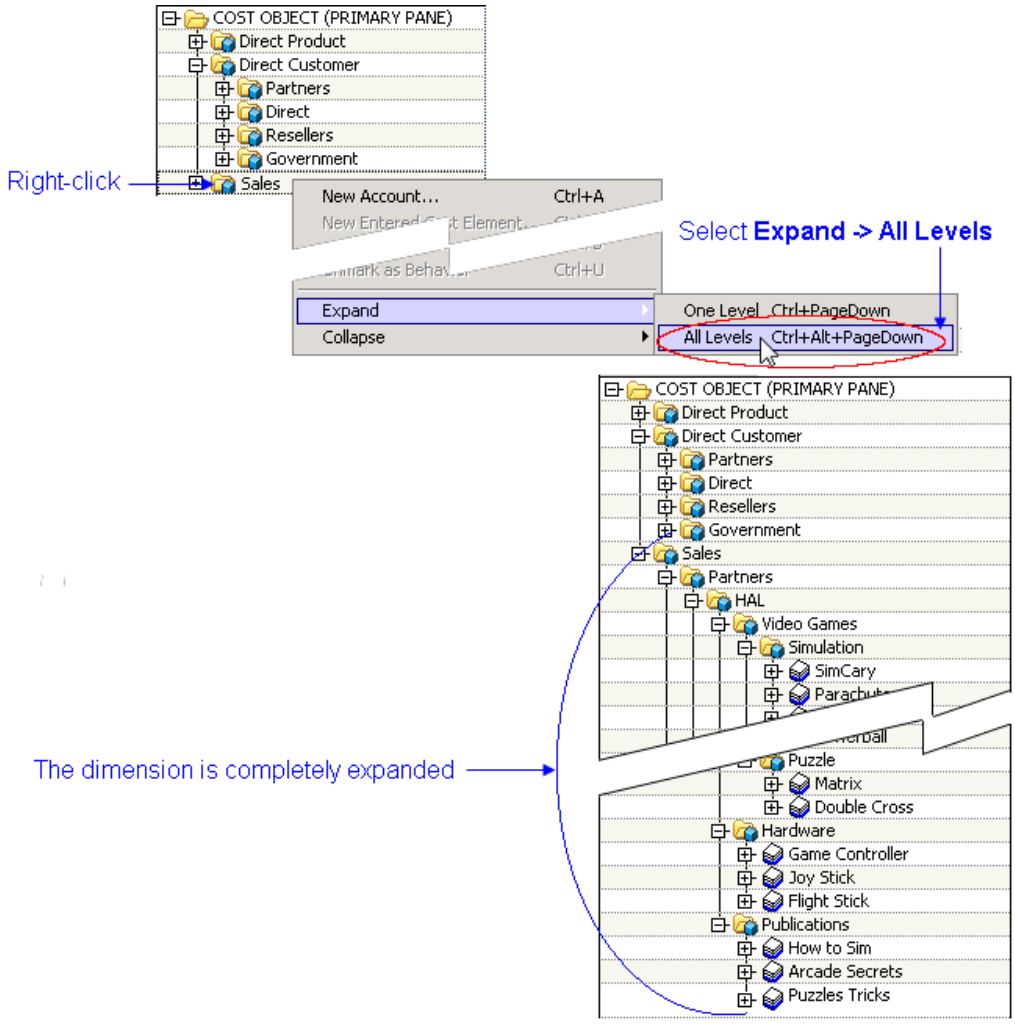
Mark an Account for What-If Analysis

See "Mark Accounts for the X-Axis" on page 606.

Expand All Levels

To expand all dimension levels to see the accounts at every level, do either of the following:

- Select a module name (Resource, Activity, Cost Object, External Object), a dimension, or a dimension member. Then, select **View** ⇨ **Expand** ⇨ **All Levels** from the menu bar.
- Right-click a module, dimension, or dimension member, and select **Expand** ⇨ **All Levels** from the pop-up menu.



Collapse All

To collapse all dimension levels so that no accounts are visible, do either of the following:

- Select a module name, dimension, or dimension member that is fully or partially expanded, and then select **View** ⇨ **Collapse** ⇨ **All Levels** from the menu bar.
- Right-click a module, dimension, or dimension member that is fully or partially expanded, and then select **Collapse** ⇨ **All Levels** from the pop-up menu.

Note:

- You can only Expand All one at a time on a view. If an Expand All in in process, you can not do another Expand All until the first one has finished.
- If you change models during an Expand All function, the expansion is canceled.

Part 10

Cost Elements

Chapter 30

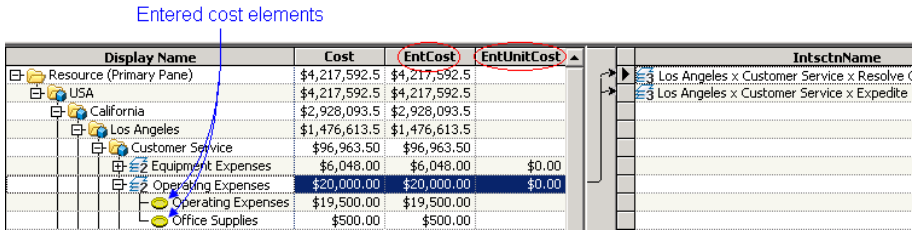
What Are Cost Elements?

Introduction

Cost elements are the bearers of cost in a model. The cost in an account is the sum of its entered cost elements plus its received cost elements:

Entered cost elements

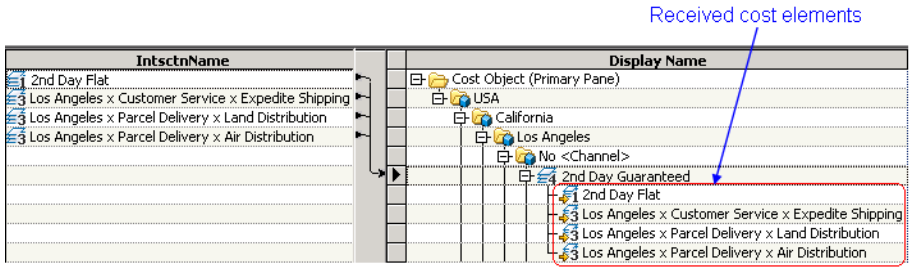
An entered cost element is one that you create and to which you assign costs directly. An entered cost element is represented by the following icon: . All of the cost in a model originates with entered cost elements.



See "Entered Cost Elements" on page 260.

Received cost elements

A received cost element is one that you create indirectly by creating an assignment from one account to another. The cost that the destination account receives from that source account is a received cost element



See "Received Cost Elements" on page 262.

Cost elements are periodic elements. The cost elements in one period/scenario may exist in another period/scenario. And, when a cost element (with the same name and reference in the same module) exists in multiple periods/scenarios, its cost in one period/scenario is independent of its cost in another period/scenario.

See Also

"Create an Entered Cost Element" on page 265

Entered Cost Elements

Introduction

An entered cost element is one that you create and to which you assign costs directly by giving it one or both of the following kinds of cost:

Entered Cost
A lump sum that you give to a cost element.

EnteredUnitCost
The unit cost that you assign to a cost element.

If you specify EnteredUnitCost, then the cost of the cost element is determined by the following formula:

```
Cost = EnteredCost + EnteredUnitCost x Parent.TDQ
```

All of the cost in a model originates with entered cost elements. That cost is distributed to accounts via drivers that work by either push or pull:

push
A driver that works by push either:

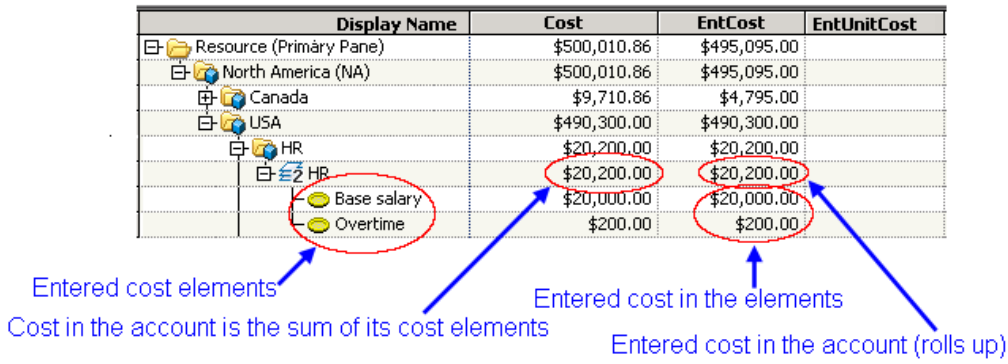
- distributes a fixed amount of cost (either Entered Cost in the account or cost received from another account) to other accounts. Drivers that work by push are: Evenly assigned, Percentage, and Standard. For an example of how the Percentage driver pushes cost to other accounts, see "Percentage Driver" on page 363.
- or, it distributes cost to other accounts by multiplying the Entered Unit Cost in the account by the number of units outputted by that account. The Standard driver can distribute costs by this method. For an example of a Standard driver using this method, see "Example 1: Entered Unit Cost with Fixed Driver Quantities" on page 371.

pull

In the case of pull, cost is determined by demand (hence, it is often referred to as "demand flow"). Cost is "pulled" by the quantity required at the final destination. The source account's Entered Unit Cost is multiplied by the number of units demanded. In more detail, the source account's Entered Unit Cost is multiplied by the driver's variable driver quantity (DQV) and by the Destination.TDQ. For an example, see "Example 2: Entered Unit Cost with Variable Driver Quantities and SoldQty" on page 374. For more on pull, see "Variable Driver Quantities" on page 401.

Entered Cost

Entered Cost is a lump sum that you give to a cost element. The following picture shows an account with two cost elements both of which have entered cost: one of $20,000 and the other of $200. Notice that the Cost in the account is the sum of its cost elements. Also notice that the Entered Cost of the account is a roll up (sum) of the Entered Cost of its cost elements.

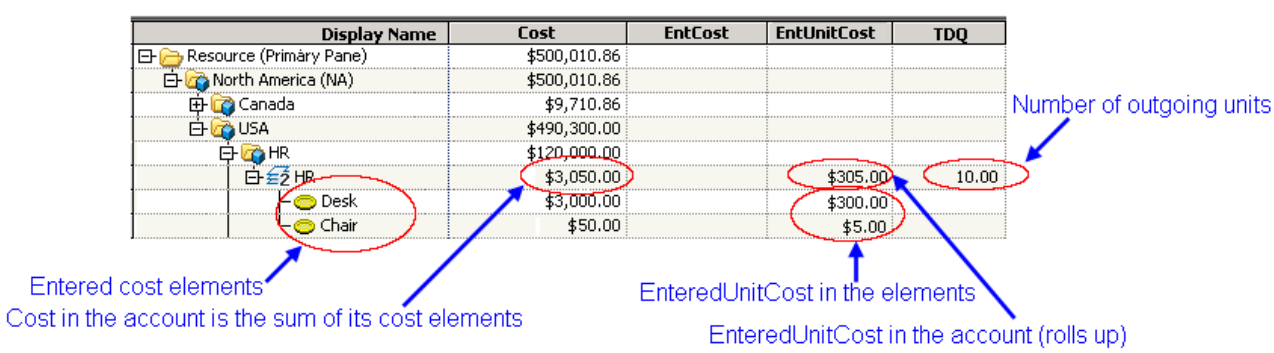


Display Name	Cost	EntCost	EntUnitCost
Resource (Primary Pane)	$500,010.86	$495,095.00	
North America (NA)	$500,010.86	$495,095.00	
Canada	$9,710.86	$4,795.00	
USA	$490,300.00	$490,300.00	
HR	$20,200.00	$20,200.00	
HR	$20,200.00	$20,200.00	
Base salary	$20,000.00	$20,000.00	
Overtime	$200.00	$200.00	

Entered Unit Cost

EnteredUnitCost is a user-assigned cost per unit. The total cost depends on the number of units. The total cost of an entered cost element is calculated by the following formula:

```
Cost = EnteredCost + EnteredUnitCost x Parent.TDQ
```



Display Name	Cost	EntCost	EntUnitCost	TDQ
Resource (Primary Pane)	$500,010.86			
North America (NA)	$500,010.86			
Canada	$9,710.86			
USA	$490,300.00			
HR	$120,000.00			
HR	$3,050.00		$305.00	10.00
Desk	$3,000.00		$300.00	
Chair	$50.00		$5.00	

For two examples of using Entered Unit Cost, see:

- "Example 1: Entered Unit Cost with Fixed Driver Quantities" on page 371

- "Example 2: Entered Unit Cost with Variable Driver Quantities and SoldQty" on page 374

Received Cost Elements

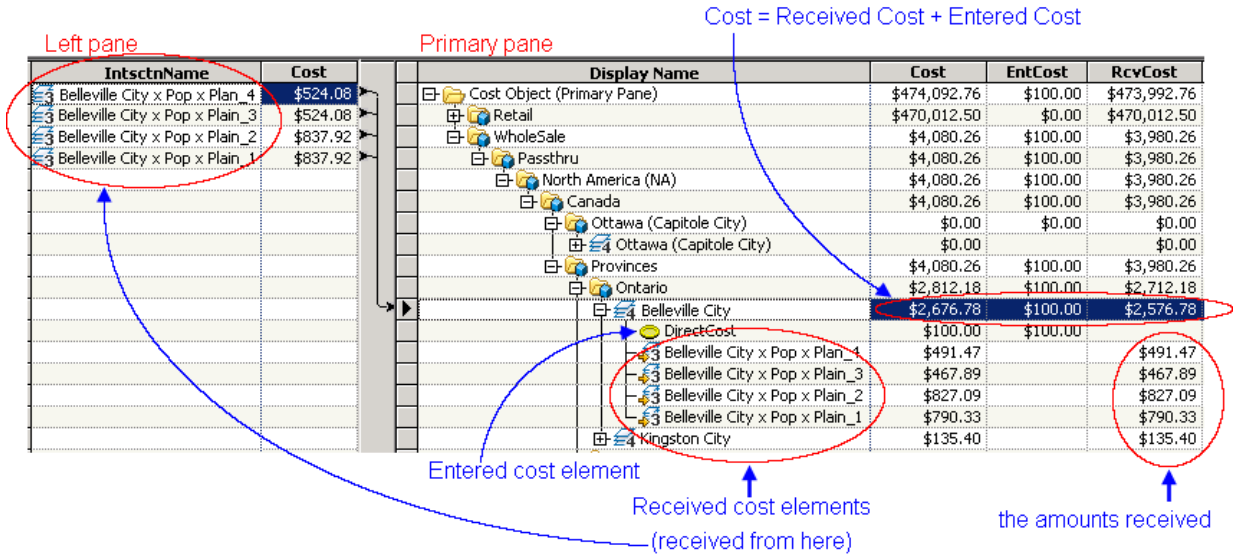
A received cost element is one that you create indirectly by creating an assignment from one account to another. The cost that the destination account receives from that source account is a received cost element.

A received cost element is represented by the following kinds of icon: 1 2 3. The number in the icon indicates the number of the module from which the cost is received.

The following picture shows four received cost elements. You can see that:

- the received cost elements result from assignments from accounts that are shown in the left assignments pane.
- the cost received by each assignment is shown in the RcvCost column of the primary pane.
- the account that receives the received cost elements has an entered cost element named DirectCost.
- the Cost in the account that receives the received costs is calculated by the formula:

```
Cost = Entered Cost + Received Cost
```



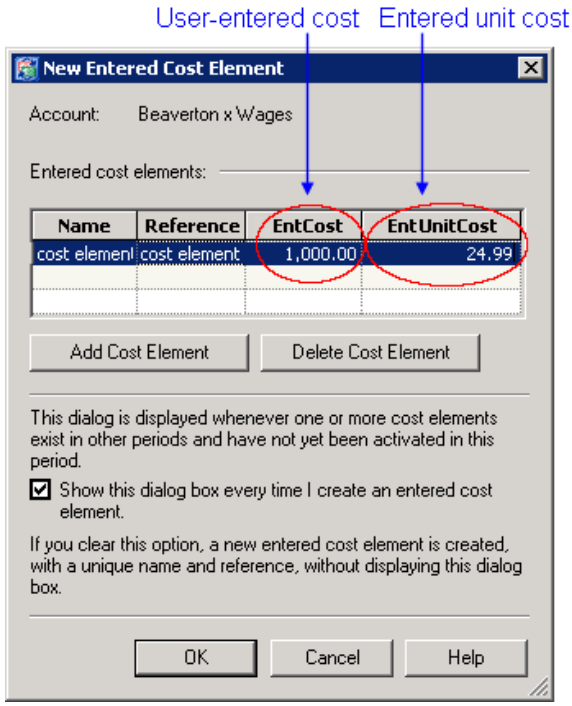
Chapter 31

Windows for Cost Elements

New Entered Cost Element Dialog Box

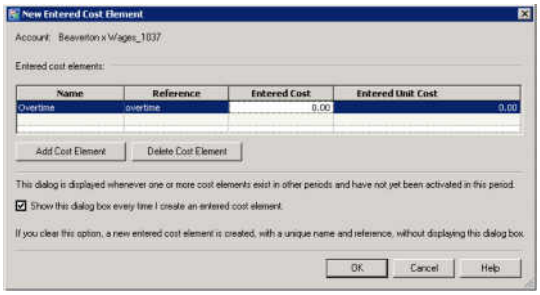
About the New Entered Cost Element Dialog Box

In the New Entered Cost Element dialog box, you can create one or more entered cost elements.



How to Access the New Entered Cost Element Dialog Box

In a module view, select an account, right-click, and select **Edit ⇨ New Entered Cost Element**.



See Also

- Chapter 32, “How To: Cost Elements,” on page 265
- Chapter 30, “What Are Cost Elements?,” on page 259
- “Create an Entered Cost Element” on page 265

Chapter 32

How To: Cost Elements

Create an Entered Cost Element

Overview

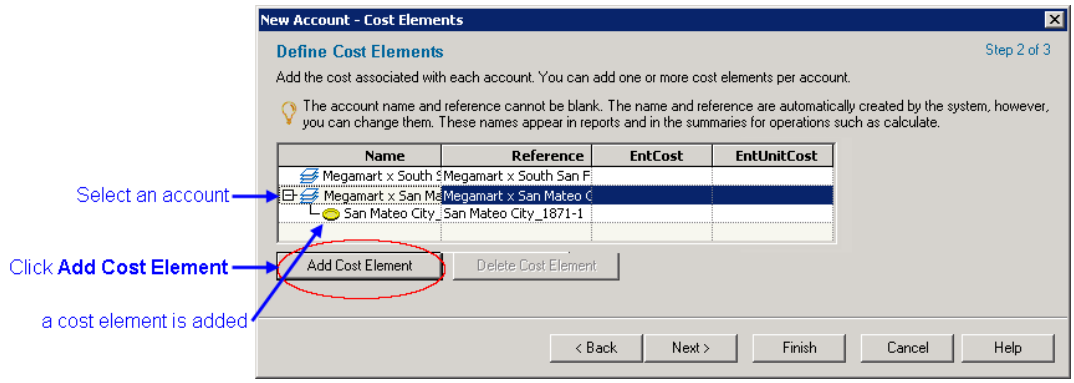
You can create an entered cost element in two ways:

- "When Creating an Account" on page 265
- "On an Existing Account in a Module Page" on page 266

When Creating an Account

1. Open the New Account dialog to create an account. See "Create Accounts" on page 241.
2. Select an account from the list of accounts on the second page of the New Account dialog. See "New Account Dialog Box" on page 235.
3. Click **Add Cost Element**.

   A new entered cost element appears.



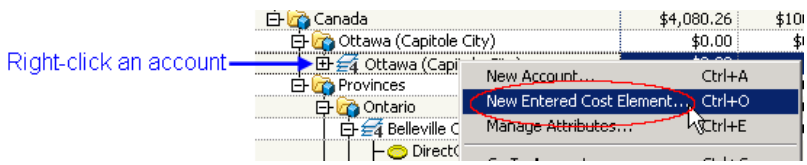
4. You can change the Name, Reference, Entered Cost, and Entered Unit Cost of the new entered cost element.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

   The reference must follow the reference conventions. See "Reference Conventions" on page 74.

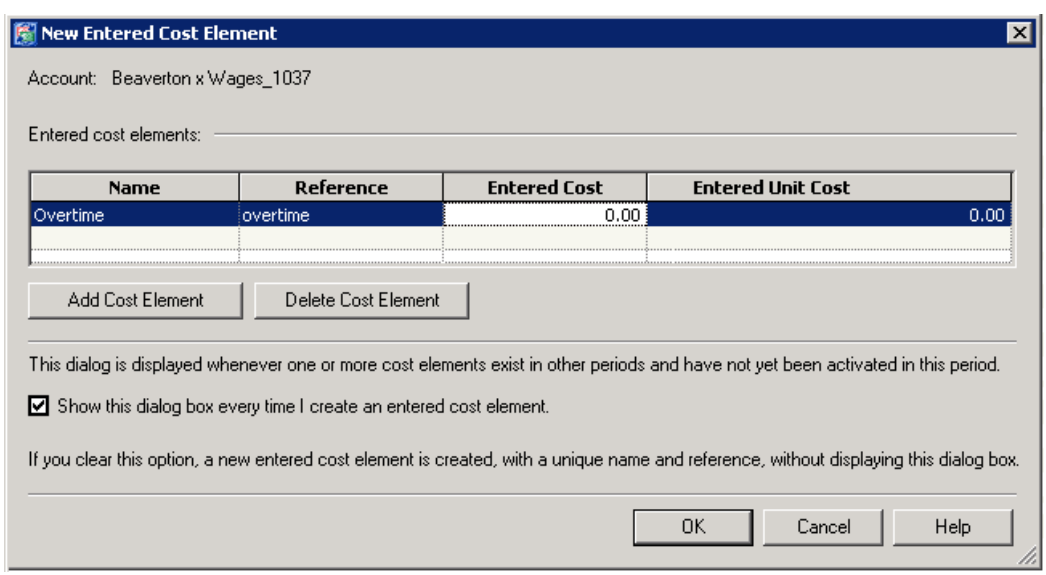
On an Existing Account in a Module Page

1. In a module, right-click an account (not a roll-up account).



2. Select **Edit** ⇨ **New Entered Cost Element**.

   The New Entered Cost Element dialog box appears.



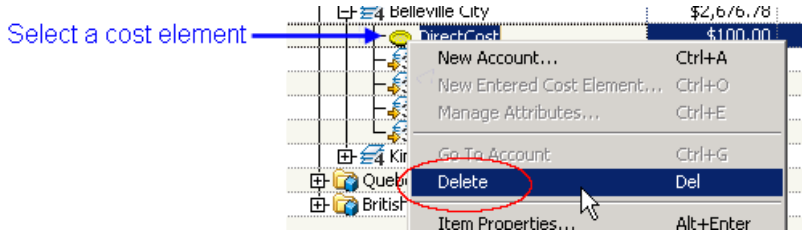
3. You can change the Name, Reference, Entered Cost, and Entered Unit Cost of the new entered cost element.

The name must follow the naming conventions. See "Naming Conventions" on page 69.

The reference must follow the reference conventions. See "Reference Conventions" on page 74.

Delete an Entered Cost Element

1. In a module, right-click an entered cost element and select **Delete**..



2. Click **Delete Cost Element**.

Mark an Entered Cost Element for What-If Analysis

See "Mark Entered Cost Elements as Independent Variables" on page 607.

Part 11

Attributes

Chapter 33

Types of Attributes

Introduction

About Attributes

An attribute is a label that is added to an account. Each attribute is a characteristic that is used for analysis or for a calculated value, such as in a formula for a calculated attribute or calculated driver. An attribute conveys information about the item to which the attribute is added.

You create an attribute with the Attributes view. You add and delete (manage) the attributes that have been added to an account with a module view.

Attributes are periodic elements. An attribute that is attached to an account in one period/scenario may or may not be attached to the same account in another period/scenario. And, when a value attribute is attached to the same account in multiple periods/scenarios, its value in one period/scenario is independent of its value in another period/scenario.

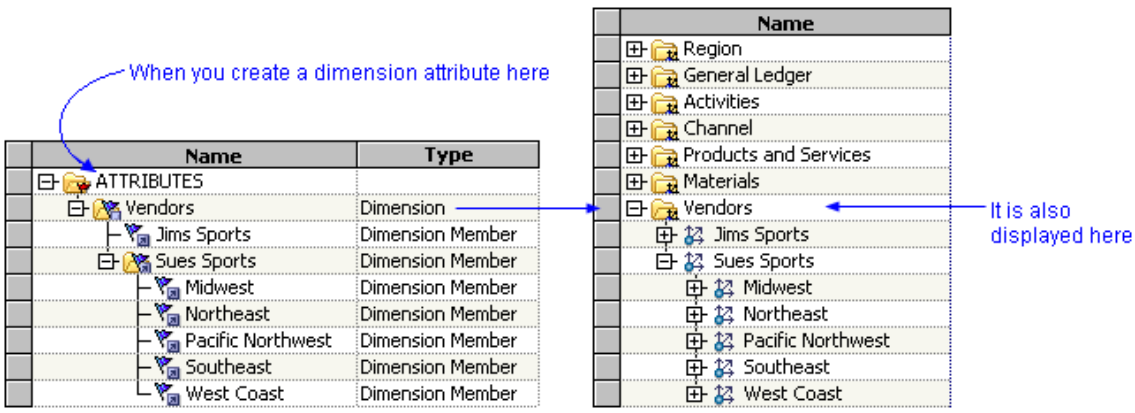
Attribute Folders

An attribute folder organizes attributes. Generally, an attribute folder can contain any type of attribute, and it can contain multiple types of attributes, because the folder itself has no type. However, a dimension attribute folder can contain only dimension member attributes or other dimension value attributes. In addition, dimension member attributes must be contained in a dimension attribute folder.

You cannot explicitly create a dimension attribute folder. When you explicitly create a new dimension attribute, SAS Cost and Profitability Management automatically creates a dimension attribute folder in which dimension member attributes are contained. If you create a new dimension attribute when a dimension attribute folder is selected, the new dimension attribute is automatically created as a dimension member attribute. You cannot change this attribute type.

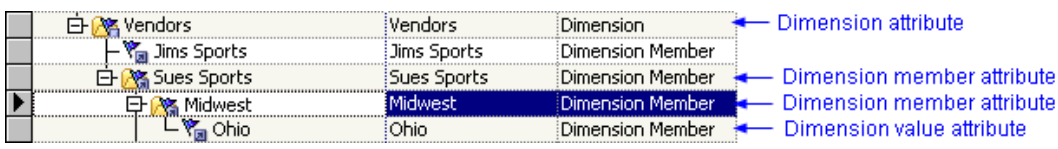
Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes

A dimension attribute is a type of dimension and is included in and displayed in cubes. When you create a dimension attribute on the Attributes view, it is automatically displayed on the Dimensions view, as shown:

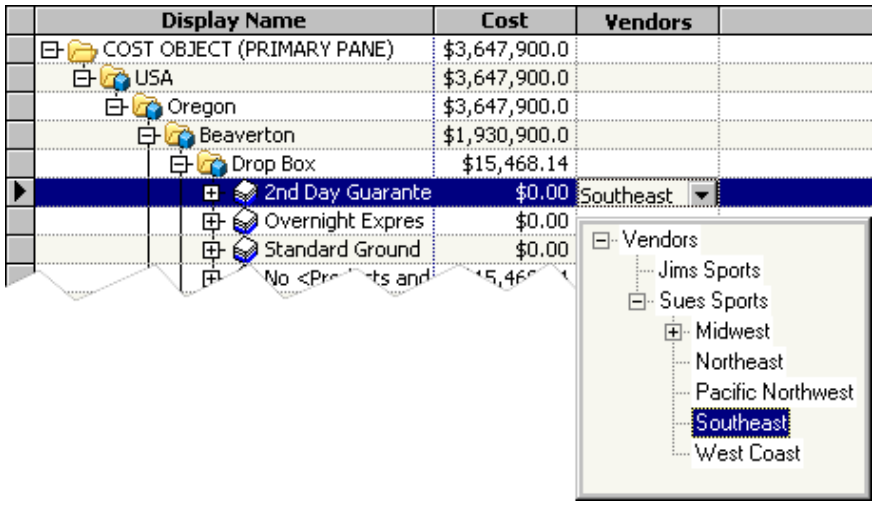


The leaf node of a dimension attribute is called a dimension value attribute. It is the dimension value attribute that applies to an account. Whereas a dimension member attribute contains a dimension value attribute, a dimension value attribute cannot contain other attributes.

If you attempt to create another attribute within a dimension value attribute, the dimension value attribute automatically becomes a dimension member attribute. For example, you can see in the following that if you create an **Ohio** attribute inside the **Midwest** attribute, then the **Midwest** attribute automatically becomes a dimension member attribute:



Dimension value attributes can facilitate data entry because a drop-down list of possible attribute values is displayed in the interface. For example, from the previous picture, the following drop-down list is displayed when a user sets the value of the **Vendors** attribute:



Dimension attributes, dimension member attributes, and dimension value attributes provide business users with roll-up values for OLAP analysis that differ from the values available from the structural dimensions. Using dimension attributes, you enhance a model by classifying or organizing information in ways that will help business users analyze model results.

For example, dimension attributes are commonly used to indicate which resources are fixed or variable and which activities are value-added or non-value-added. Other typical dimension attributes include core, sustaining, and discretionary; strategic and non-strategic; and primary and secondary.

Numeric Attributes

A numeric attribute stores a number. What the number represents is determined by the unit of measure. The unit of measure is text that provides meaning to the unit. For example, the unit of measure could be cases, pounds, kilograms, and so on. Without a unit of measure, a numeric attribute has little meaning. Alternatively, the name of the numeric attribute can indicate the unit of measure, as in Miles Shipped or Number of Pallets.

A numeric attribute can serve purposes such as:

- A numeric attribute can track a models financial measures or non-financial measures.
- A numeric attribute can track activity performance over time.
- A numeric attribute can be used for What-If analysis.

The data that a numeric attribute contains can be used in calculated drivers and in calculated attributes.

See Also

- "Using Numeric Attributes in a Formula" on page 467

- "Properties That Can Be in Formulas" on page 495

Calculated Attributes

Overview

A calculated attribute stores a number that is calculated from numeric properties of a model and/or from other calculated attributes, based on an formula that you define. (Examples of calculated attributes)

Property values and attribute values are derived from the item to which the attribute is attached. For example, the value of a calculated numeric attribute that is based on the property Cost depends on the item to which the attribute is attached, because each item has a different cost.

Unlike the formula for a calculated driver, which remains the same for all periods, the formula for a calculated attribute can change with each period.

Example: Average Cost of Processing a Customer Order

The following example illustrates how calculated attributes can be used in a model. Equations often include numeric properties and numeric attributes.

Suppose that a company provides order-processing services to other companies. To determine the average cost of processing an order, the cost (the propertyCost) is divided by the number of orders processed (the user-defined numeric attribute Orders Processed), as follows:

Cost/Orders Processed

Example: Average Number of Cases Loaded per Employee

Suppose that a company loads goods onto trucks for shipping. Management wants to determine how many cases are loaded per full-time employee. The number of cases loaded (the user-defined numeric attribute Cases Loaded) is divided by the number of full-time employees who are loading cases (the user-defined numeric attribute FTE), as follows:

```
Cases Loaded/FTE
```

See Also

- Chapter 51, "Formulas," on page 457
- "Using Numeric Attributes in a Formula" on page 467
- "Properties That Can Be in Formulas" on page 495

Tag Attributes

A tag attribute is either added to an account or it is not. A tag attribute does not store a value.

You can use the HasAttribute function in the formula for a calculated driver, rule-based driver or calculated attribute to test whether an account has a particular tag attribute or not.

Note: In previous releases of SAS Cost and Profitability Management, tag attributes were called Boolean attributes.

The maximum length of a text attribute is 2048 characters.

Text Attributes

A text attribute stores alphabetic and numeric characters. A text attribute provides information about the item to which the attribute is added. For example, a text attribute named Cost Analysis can indicate whether a resource account is a fixed cost or a variable cost. Or, a text attribute named Life Cycle can indicate where a cost object account belongs in a life cycle: analysis, start-up, entry, build, mature, decline, or withdrawal.

The maximum length of a text attribute is 2048 characters.

Stage Attributes

Overview

Stages are used for creating the Multi-stage Contributions cube. To define stages, you create a dimension value attribute for each stage, and then you assign these attributes to accounts. You must create these dimension value attributes within a dimension attribute named Stages. You can give a dimension value attribute any name, but the name must conform to the attribute naming conventions (see "Attribute naming conventions" on page 70.).

For more information, see "Add Stage Attributes to Accounts" on page 313.

Note:

- All the accounts for a given stage must belong to the same module.
- Stage names and references must begin with an alphabetic character (letter).

Note: Previous releases of SAS Cost and Profitability Management allowed stage names and references to begin with a numeric character. In SAS Cost and Profitability Management 7.2, this is not allowed. If you have a stage name or reference that begins with a numeric character, then you must change it to begin with an alphabetic character. For models using renamed stages, update cube configurations to re-select the dimensions that are included in the cube and

regenerate any multi-stage contributions cubes. (Regenerating cubes is not necessary if you are using Microsoft Analysis Services for cubes.)

- The stage name is used to define a dimension name in cubes. The stage name is concatenated with a user-defined dimension name to form the actual dimension in the cube.
- A Stages attribute example is Stage2 Activity, where Stage2 is the stage name and Activity is the user-defined dimension. If 2 is used in place of Stage2, then the generated cube dimension name becomes 2 Activity.

To see the stages in the Multi-stage Contributions cube, you must specify the properties for the cube. This is true regardless of which method you use for creating the Stages attributes.

Guidelines for Adding the Stages Attributes to Accounts

Follow these guidelines when you add Stages attributes to accounts:

- Add only one Stages attribute (or one member of a Stages attribute) to an account.
- Do not skip an account that you need for analysis.

  For example, suppose that an IT Salaries account is assigned to an IT Help Desk account, and that the IT Help Desk account is subsequently assigned to a Design Products account. Ensure that a different Stages attribute is added to each account. Do not skip the IT Help Desk account.
- Do not add Stages attributes to accounts that you do not need for analysis.

  Every account that has a Stages attribute is included in the Multi-stage Contributions cube. The cube can be generated more quickly if unnecessary accounts are omitted.
- To avoid performance problems, do not exceed six to eight stages.

See Also

"Add Stage Attributes to Accounts" on page 313

Chapter 34

Attributes on Dimension Members

General Description

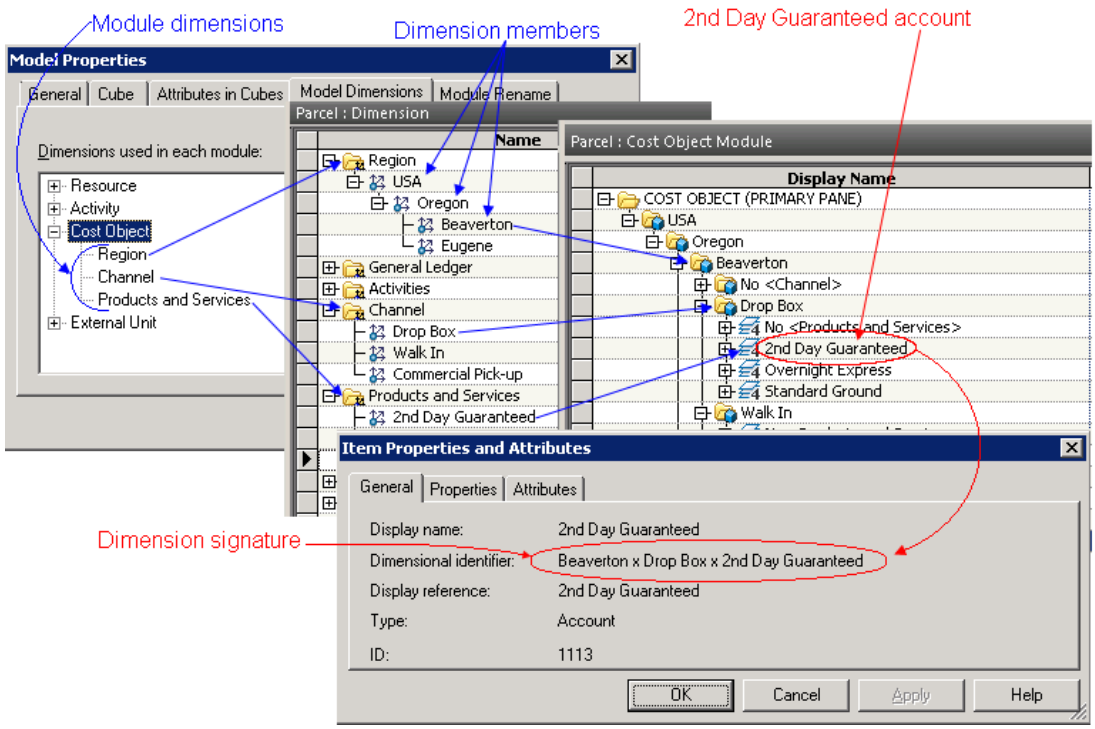
Introduction

Setting an attribute on a dimension member sets the attribute on multiple accounts automatically.

Setting an attribute on a dimension member is an indirect method of setting the attribute on all the accounts whose dimension signature includes that dimension member. One reason that it is important to be able easily to set attributes on accounts is because having attributes on accounts is particularly important for generating rule-based drivers. Setting attributes on dimension members provides a method of quickly setting those attributes on many accounts.

To understand how setting an attribute on a dimension member works, first remember that an account is an intersection of dimension members. An account is defined by that intersection, which is called its dimension signature (or alternatively, its dimension identifier). For example, in the following picture you can see that the account 2nd Day Guaranteed is the intersection of the dimension members Beaverton x Drop Box x 2nd Day Guaranteed. Those dimension members are, in turn, members of the dimensions Region x Channel x Products and Services, respectively.

Note: The display name of the account, 2nd Day Guaranteed, is the name of the last member in the intersection of the dimension members when the order of dimension members is the order of their containing dimensions—Region, Channel, Products. The order of dimensions is the order in which you define them when you create a model.



Now, if you set an attribute on the dimension member Beaverton, you would automatically set the attribute on the 2nd Day Guaranteed account because its dimension signature is Beaverton x Drop Box x 2nd Day Guaranteed. And, in fact, you would automatically set the attribute on all the accounts whose dimension signature includes Beaverton.

An Example

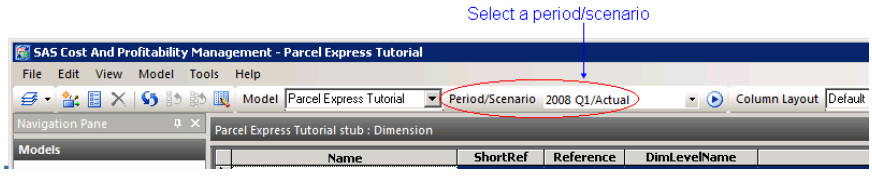
Let's look at an example.

1. Create an attribute:

   You can use any existing attribute, but for this example suppose that you create a numeric attribute named AttributeOnADimensionMember with a default value of 100, as shown in the following picture.

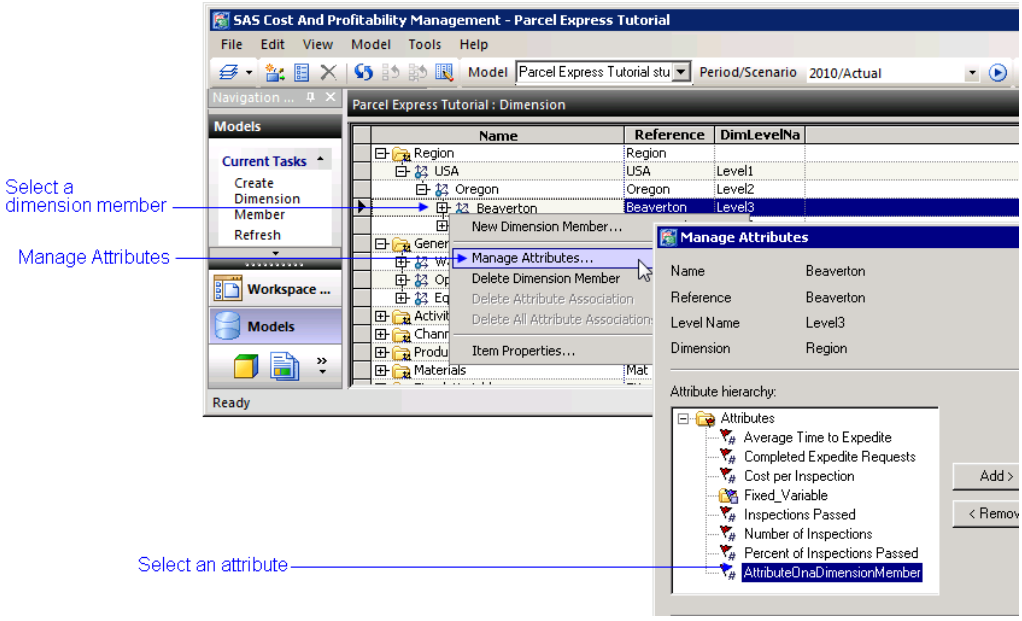
Name	Reference	Type	UoM	DefaultValue
ATTRIBUTES	ATTRIBUTES			
Average Time to Expedite	Average Time to Expedite	Numeric	Hours	
Completed Expedite Requests	Completed Expedite Requests	Numeric	Completed Request	
Cost per Inspection	Cost per Inspection	Numeric	Dollars	
Fixed_Variable	FV	Dimension		
Inspections Passed	I_P	Numeric	Units Passed	
Number of Inspections	Num of Ins	Numeric	Inspections	
Percent of Inspections Passed	Percent of Inspections Passed	Numeric	Percentage	
AttributeOnADimensionMember	AttributeOnADimensionMember	Numeric	Credits	100.00

2. Go to the model's Dimension view, and select the period in which the attribute is to be applied.



3. Select a dimension member, and select **Manage Attributes** to attach the attribute to the dimension member.

For this example, attach the attribute, AttributeOnADimensionMember, to the Beaverton dimension member in the Region dimension.



In fact, there are two ways in which to attach an attribute to a dimensioin member. And, you can modify the columns in the Dimension view to display attributes. For more information, see "Attach an Attribute to a Dimension Member" on page 287.

4. The result is that the attribute, AttributeOnADimensionMember, is applied to every account whose dimension signature includes Beaverton, as shown in the following picture.

Parcel : Cost Object Module

Display Name	AttributeOnADimensionMember
COST OBJECT (PRIMARY PANE)	
USA	
Oregon	
Beaverton	
No <Channel>	
2nd Day Guaranteed	100.00
Overnight Express	100.00
Standard Ground	100.00
Drop Box	
No <Products and Services>	100.00
2nd Day Guaranteed	100.00
Overnight Express	100.00
Standard Ground	100.00
Walk In	
No <Products and Services>	100.00
2nd Day Guaranteed	100.00
Overnight Express	100.00
Standard Ground	100.00
Commercial Pick-up	
No <Products and Services>	100.00
2nd Day Guaranteed	100.00
Overnight Express	100.00
Standard Ground	100.00
Eugene	
No <Channel>	
Drop Box	
No <Products and Services>	
2nd Day Guaranteed	
Overnight Express	
Standard Ground	
Walk In	
Commercial Pick-up	

You can see that the attribute with its default value is now on every account with the dimension member of Beaverton, but it is not on accounts with the dimension member Eugene. And, note, moreover, that the preceding picture shows only the Cost Object module. As a result of attaching dimension attributes to accounts, other modules that have accounts whose dimension signature includes Beaverton would also have attributes attached to those accounts. In short, an attribute that you initially assigned to a dimension member is now on multiple accounts.

Frequently Asked Questions

FAQ

- "Importing and exporting attributes on dimensions" on page 286

What type of attributes can you attach to dimension members?

You can attach any type of attribute to a dimension member, including dimension attributes and calculated attributes.

Name	Reference	Type
ATTRIBUTES	ATTRIBUTES	
Text attribute	Text attribute	Text
Tag attribute	Tag attribute	Tag
Numeric attribute	Numeric attribute	Numeric
Calculated numeric attribute	Calculated numeric attribute	Numeric
Dimension attribute	Dimension attribute	Dimension

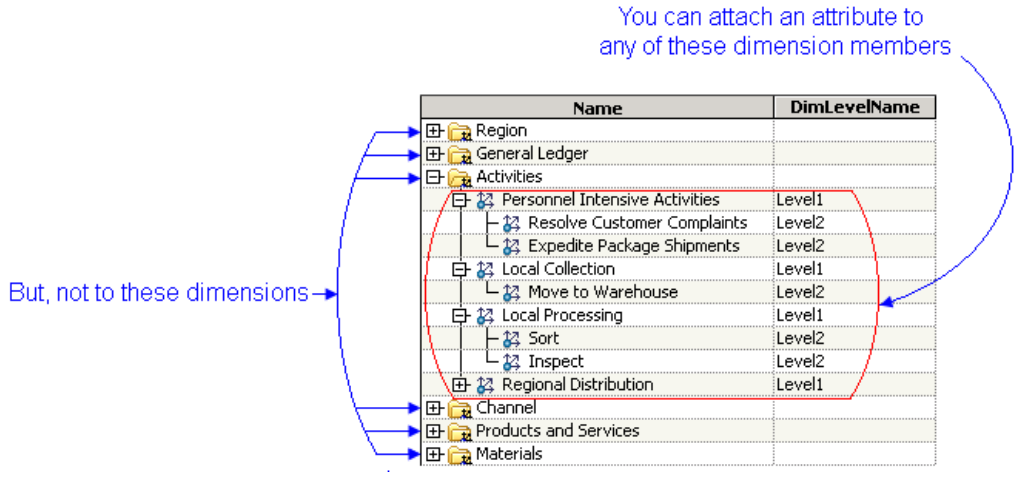
So, you can attach any of the following types of attributes to a dimension member:

- Text
- Tag (Boolean)
- Numeric (including calculated numeric)
- Dimension

Where precisely can you attach the attributes?

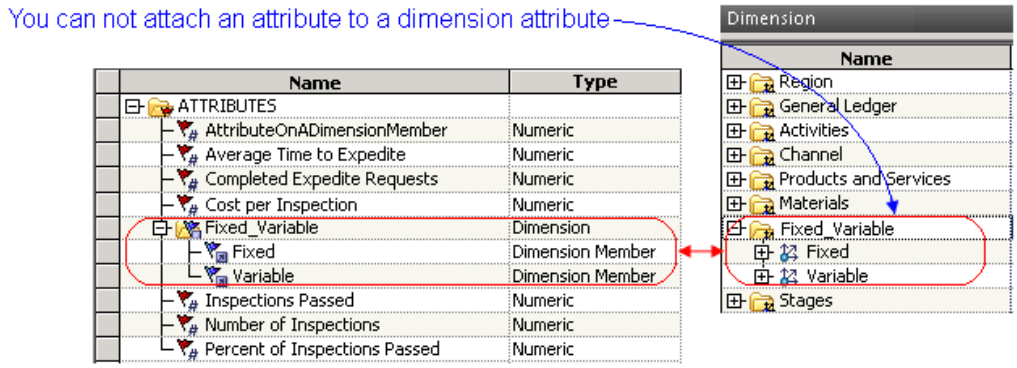
- **You can attach an attribute to any dimension member.**
- **You can not attach an attribute to a dimension.**

  For example, in the following picture, you can attach an attribute to (among others) the dimension members Personnel Intensive Activities (Level1) and Resolve Customer Complaints (Level2), but you can not attach an attribute to the Activities dimension.



- **You can not attach an attribute to a dimension attribute.**

  You can attach an attribute only to a structural dimension. Thus, for example, in the Parcel Express Tutorial model, you can not attach an attribute to the FixedVariable dimension attribute.

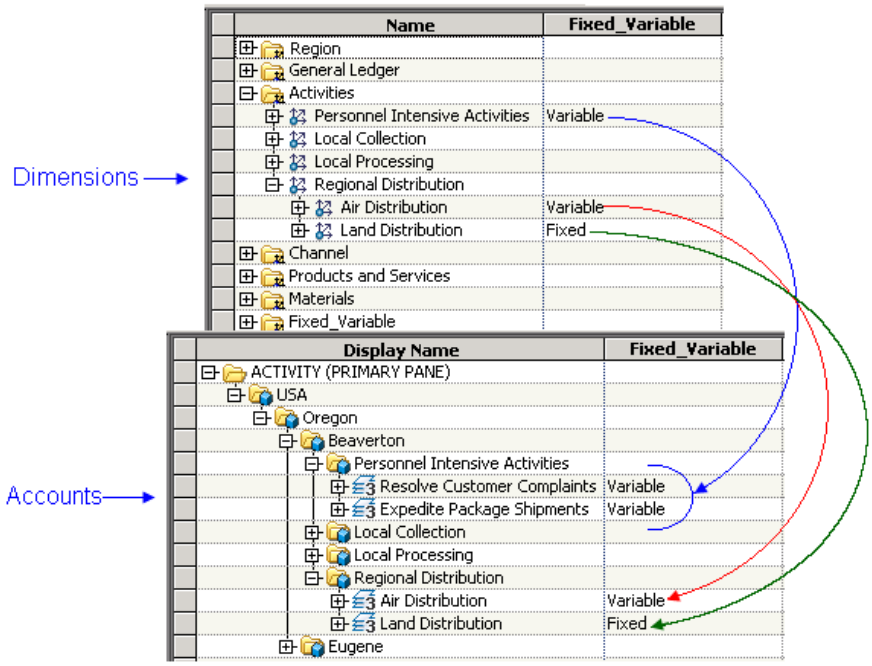


Do not be confused by terminology. A dimension attributute is called "dimension attribute" because it is like an ordinary (structural) dimension in that it is an element of an OLAP cube. Inside an OLAP cube there is no distinction between dimension attributes and structural dimensions. They are both dimensions of a measure.

However, whereas you can not attach an attribute to a dimension attribute, you can attach a dimension attribute to a dimension member. For example, in the following picture, the dimension attribute Fixed_Variable is attached to the following dimension members:

- Personnel Intensive Activities (with a value of Variable)
- Air Distribution (with a value of Variable)
- Land Distribution (with a value of Fixed)

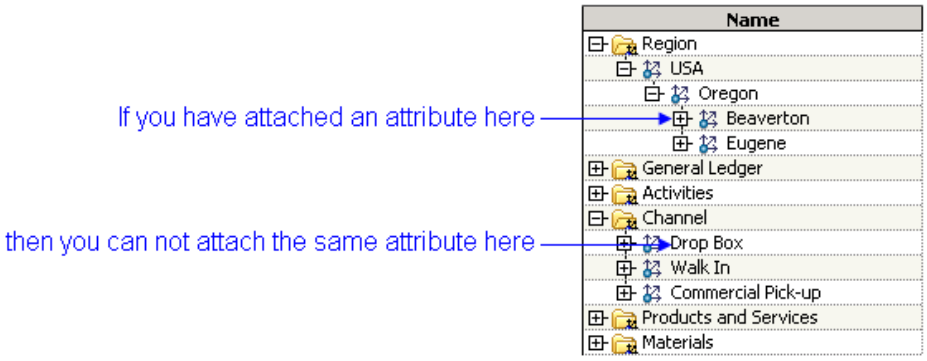
The picture also shows the result of attaching the attribute Fixed_Variable to accounts. You can see that the attribute is attached to four different accounts in the Beaverton region. (It would also be attached to the appropriate accounts in the Eugene region, but those accounts are not shown.)



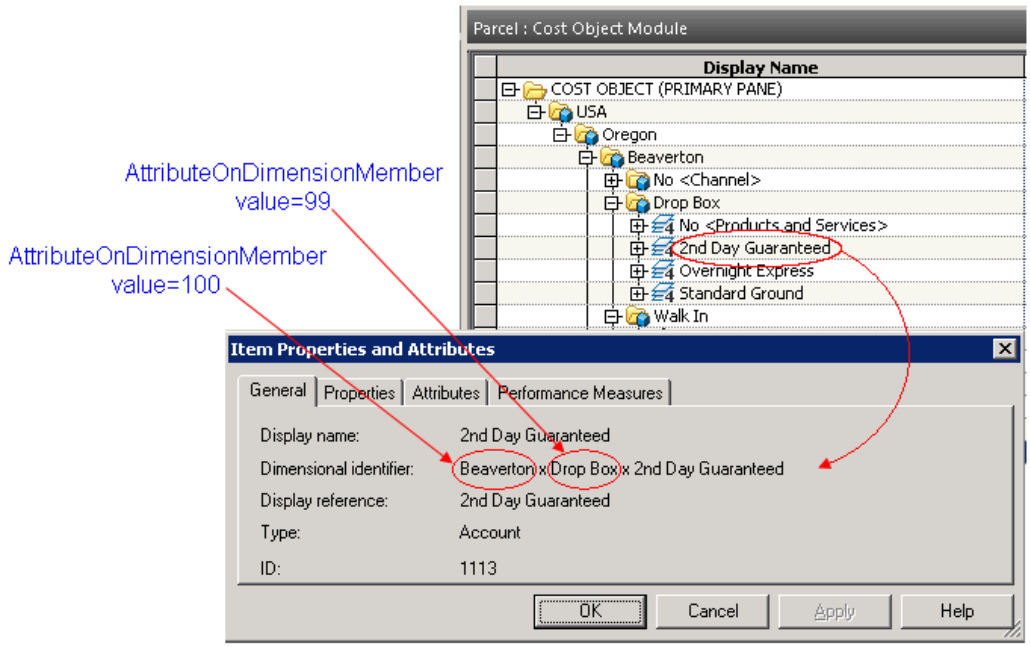
- **You can attach a particular attribute to the dimension members of only one dimension in a model.**

  For example, in the following picture, if you have attached a particular attribute to the Beaverton dimension member (which is in the Region dimension), then you can

not attach the same attribute to the Drop Box dimension because it is in the Channel dimension.



The reason for this restriction is to avoid conflicting attribute assignments. Suppose that you could attach the same attribute to dimension members in different dimensions. Take, for example, our previous example of the 2nd Day Guaranteed account whose dimension signature is Beaverton x Drop Box x 2nd Day Guaranteed. Suppose that the attribute AttributeOnDimensionMember has the value 100 when it is attached to the Beaverton dimension member. And, suppose that the same attribute has the value 99 when it is attached to the Drop Box dimension member. Then, the 2nd Day Guaranteed account will get the AttributeOnDimensionMember attribute from both the Beaverton dimension member and the Drop Box dimension member. But, in that case, what is its value? Is it 100 (from Beaverton) or 99 (from Drop Box)?



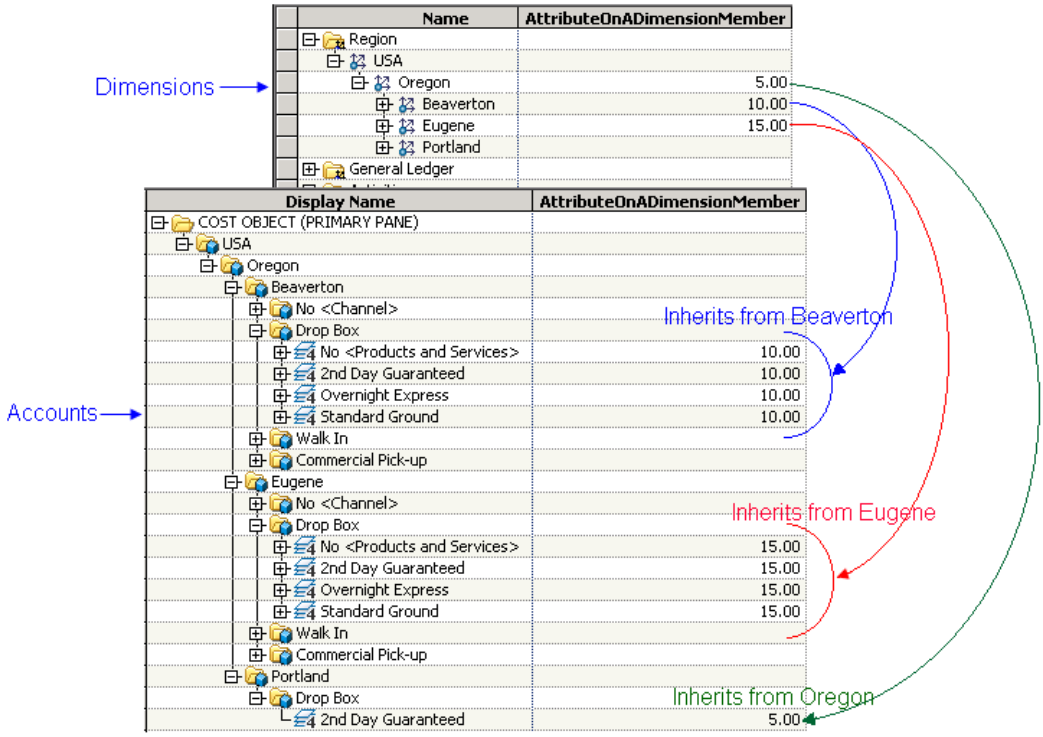
Attribute associations are periodic

When you attach an attribute to a dimension member, you attach it in a particular period. Consequently, a dimension member can have a particular attribute in one period but not in another. And, it can have one value in a particular period and a different value in another period.

How are attribute values inherited from dimension members?

When an attribute on a dimension member is applied to an account, and when there are additional attributes on dimension members in the same dimension hierarchy, then the first dimension member found up the hierarchy from the account is used.

For example, in the following picture the value of the attribute AttributeOnDimensionMember is 5 on the account, Portland x Drop Box x 2nd Day Guaranteed. It is 5 because there is no attribute on the Portland dimension member, so the account inherits the value 5 from the attribute AttributeOnDimensionMember on the parent dimension member, Oregon. The value of the attribute, AttributeOnDimensionMember, is 10 on the accounts that inherit from the Beaverton dimension member. The value is 15 on the accounts that inherit from the Eugene dimension member.



Name	AttributeOnADimensionMember
Region	
USA	
Oregon	5.00
Beaverton	10.00
Eugene	15.00
Portland	
General Ledger	

Display Name	AttributeOnADimensionMember
COST OBJECT (PRIMARY PANE)	
USA	
Oregon	
Beaverton	
No <Channel>	
Drop Box	
No <Products and Services>	10.00
2nd Day Guaranteed	10.00
Overnight Express	10.00
Standard Ground	10.00
Walk In	
Commercial Pick-up	
Eugene	
No <Channel>	
Drop Box	
No <Products and Services>	15.00
2nd Day Guaranteed	15.00
Overnight Express	15.00
Standard Ground	15.00
Walk In	
Commercial Pick-up	
Portland	
Drop Box	
2nd Day Guaranteed	5.00

What are the default values for such attributes?

Because attributes on dimension members are ordinary attributes, once they have been applied to accounts, they have the same default values as ordinary attributes:

- For attributes with a user-defined default value for the active period/scenario, the attribute is initialized with the user-defined default value.
- Numeric attributes that do not have a user-defined default value are initialized with zero. Calculated numeric attributes are initialized with the result of their formula.
- Text attributes are initialized with an empty string.
- Tag attributes and dimension attributes do not have a value, so they are not initialized with a default value.

Despite the existence of default values, an error occurs during calculation if an attribute is referenced in a formula and both of the following conditions are true:

- the attribute has no user-defined default value, and

- the attribute is not attached to the account whose formula is being evaluated (for the period being calculated).

If either of the preceding conditions is false, then no error occurs during calculation. That is, no error occurs if either of the following conditions is true (the negations, respectively, of the preceding conditions):

- the attribute has a user-defined default value (even if it is not attached to the account whose formula is being evaluated), or
- the attribute is attached to the account whose formula is being evaluated (even if it has no user-defined default value).

System-generated versus user-entered attribute values

When an attribute on a dimension member is applied to an account, the account receives the attribute with its default value, if one is defined for that period/scenario. The following question then arises:

- What happens if you edit the attribute on that account to modify the value? Does the attribute on the account once again receive its default value, or does it retain its modified value?
- The answer is that it retains its modified value.

SAS Cost and Profitability Management distinguishes between system-generated attribute values and user-entered attribute values:

- A system-generated attribute value is the value that an attribute on an account receives when the attribute is applied to the account in virtue of having been attached to a dimension member.
- A user-entered attribute value is the value that an attribute on an account receives when you, the user, modifies the attribute value.

  Note:

  - An attribute value is considered to be user-entered even if you overwrite the attribute value with exactly the same value. It is sufficient to type into the attribute value field, no matter what you type.
  - You can not change the attribute value of a system-generated tag attribute. That is, if you attach a tag attribute to a dimension member, then it is automatically applied to every account whose dimension signature includes that dimension member and you can not remove it from any such account except by removing it from the dimension member (which automatically removes it from every account whose dimension signature includes that dimension member). If you want to apply a tag attribute to some accounts and not others, then you can do so on individual accounts.

Once an attribute value is marked as user-entered, the only way in which it can be returned to system-generated (if you want to return to using its default value) is by doing the following:

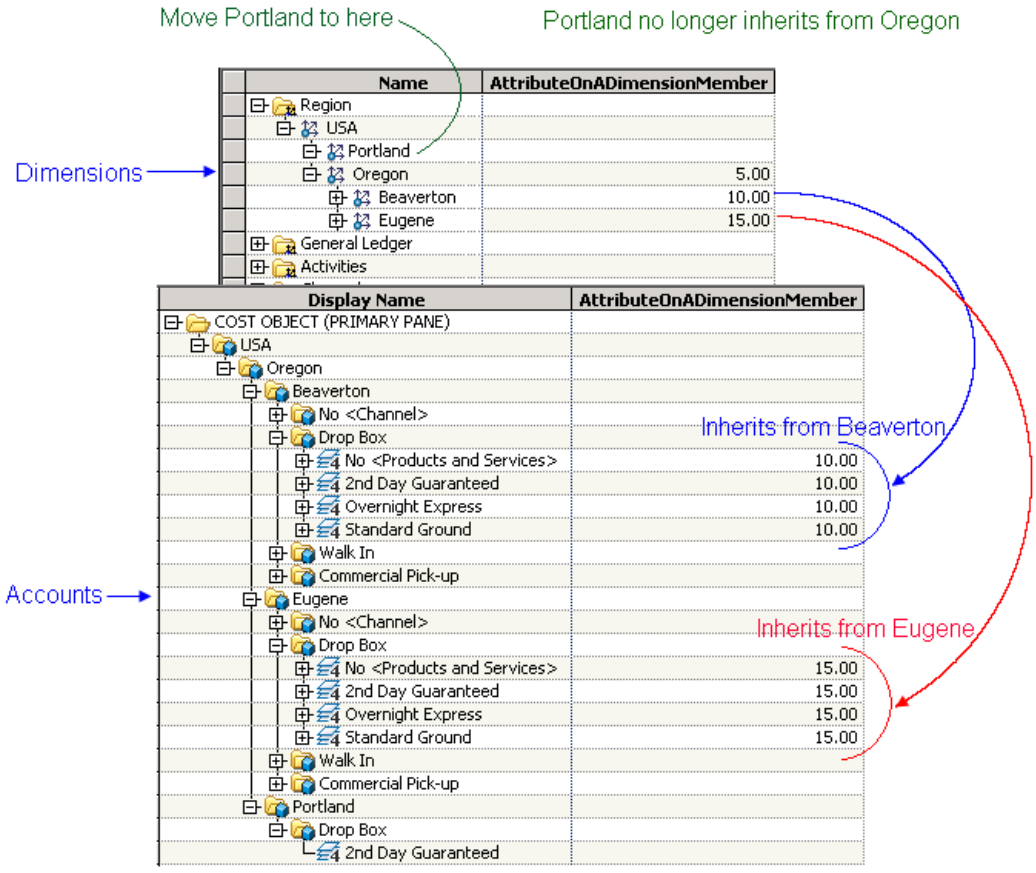
1. Remove the attribute from the dimension member. (That is, remove the association between the attribute and the dimension member—this does not mean deleting the attribute itself.)
2. Re-attach the attribute to the dimension member with the default value desired.

Note: Tag attributes and dimension attributes do not have a value, so the question of their default value does not arise.

What happens when a dimension member is re-parented?

When you re-parent a dimension member, accounts that have that dimension member in their signature can no longer inherit attributes from the old parent of that dimension member (assuming that the old parent is another dimension member). They now inherit attributes from the new parent (assuming, again, that the new parent is another dimension member).

For example, in the following picture, Portland is moved out from under Oregon to be directly under USA. This means that accounts whose dimension signature includes Portland no longer can inherit from any attributes on the Oregon dimension member.



Importing and exporting attributes on dimensions

When you export a model, the associations of attributes with dimension members are exported. If you subsequently import the same model, then dimension members have the same attributes that they had when the model was exported, as do the accounts to which those attributes apply.

See Also

- "Attach an Attribute to a Dimension Member" on page 287
- "Remove Attributes from Dimension Members" on page 291

Attach an Attribute to a Dimension Member

Introduction

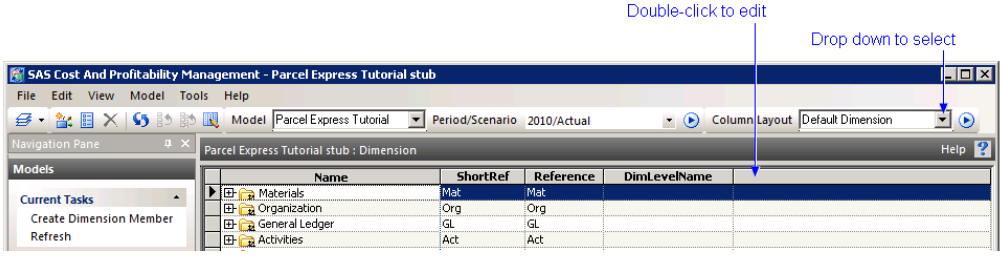
Attaching attributes to dimension members is done in two stages:

- "Define the Attribute Columns to be Displayed in the Dimensions View" on page 287
- "Attach Attributes to Dimension Members" on page 289

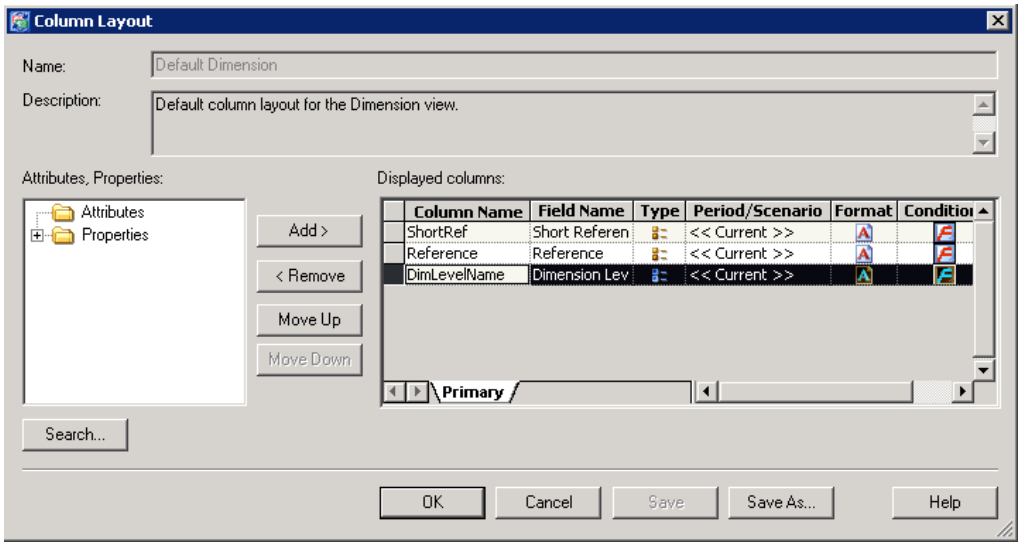
Define the Attribute Columns to be Displayed in the Dimensions View

1. Select **Model** ⇨ **Dimensions**.
2. Select **Model** ⇨ **Column Layout** ⇨ **Edit Columns** to open the Column Layout dialog box.

   Note: You can also double-click the column header.



The Column Layout dialog opens, showing the layout for the Dimensions view.



3. Select and add the attributes that you want to display in the column layout so that you can attach them to dimension members.

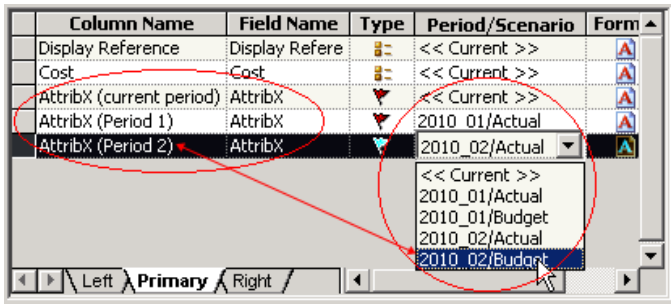
   For each attribute that you select, specify the following:

Column Name

The column name is the name that is displayed in the Dimensions view to identify an attribute on a dimension member. It is an arbitrary name that you can assign to enable you to recognize an attribute. It might identify the period and scenario in which the attribute is assigned so that you can distinguish different instances of the same attribute applied to different periods and scenarios.

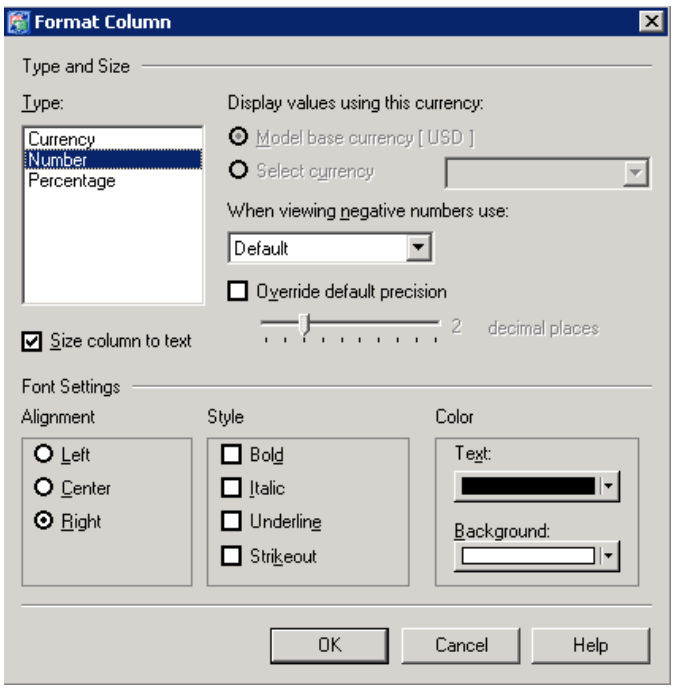
Period/Scenario

The period and scenario in which this attribute is applied when it is applied to an account. Attributes on dimension members are periodic. They can have different values in different periods and scenarios.



Format

Click the format icon to open the Format Column dialog to format the column display of the attribute.



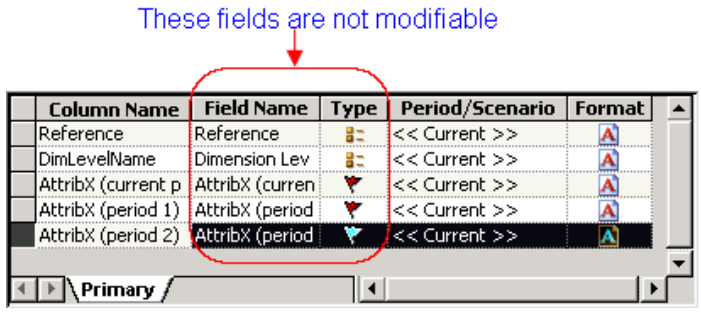
Note: The following fields of the Column Layout dialog are not modifiable. They are for information only:

Field Name

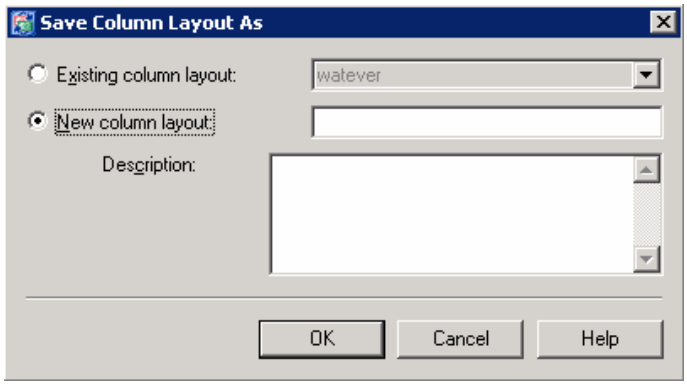
The Field Name is the name of the attribute that you assigned to it when you created it.

Type

	Tag attribute (Boolean)
	Dimension attribute
	Numeric attribute
	Text attribute

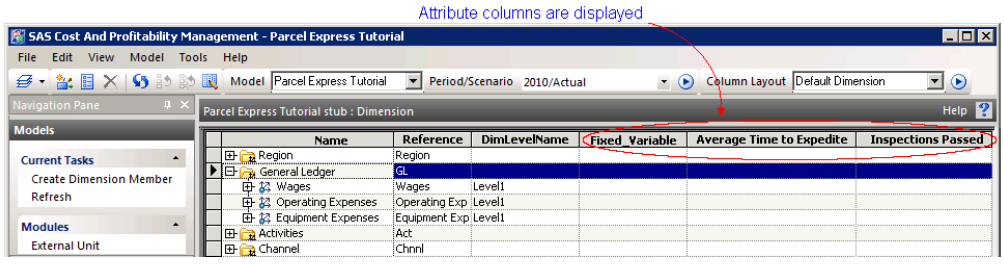


4. Click **Save As** if you want to save the column layout.



5. Click **OK**.

   The Attribute Layout dialog closes and returns to the Dimensions view, with the attribute columns displayed that you just defined.



Note: The attributes are not yet attached to any dimension members. You have only determined what attribute columns to display.

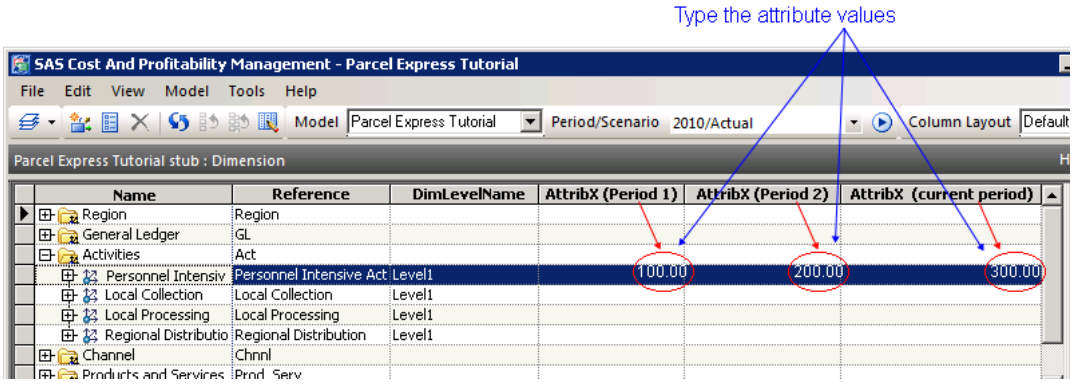
Attach Attributes to Dimension Members

There are two ways to attach an attribute to a dimension member:

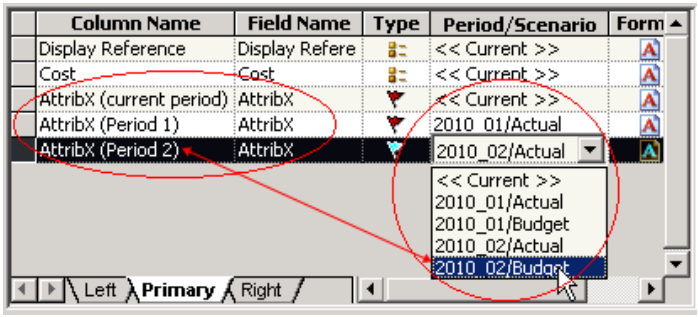
- Attach attributes using the Dimensions view
- Attach attributes using the Manage Attributes dialog

Attach attributes using the Dimensions view

After defining the attribute columns to be displayed in the Dimensions view, type the attribute value (or select the value for a dimension attribute) in the column cell for a dimension member.



Note: The period to which the value that you type applies is the period specified for this column in the Column Layout dialog, as shown for example in the following picture:

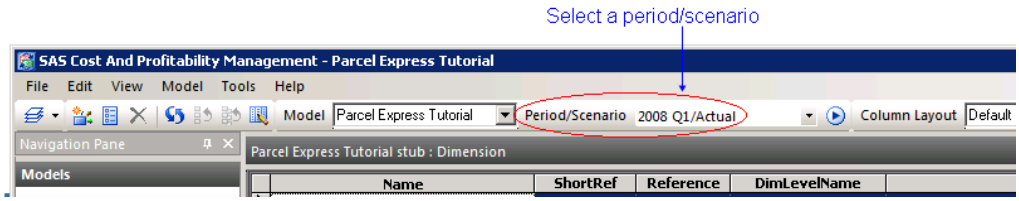


Note: You can not use this method for attaching a calculated numeric attribute (because you don't enter its value—it is calculated)

Attach attributes using the Manage Attributes dialog

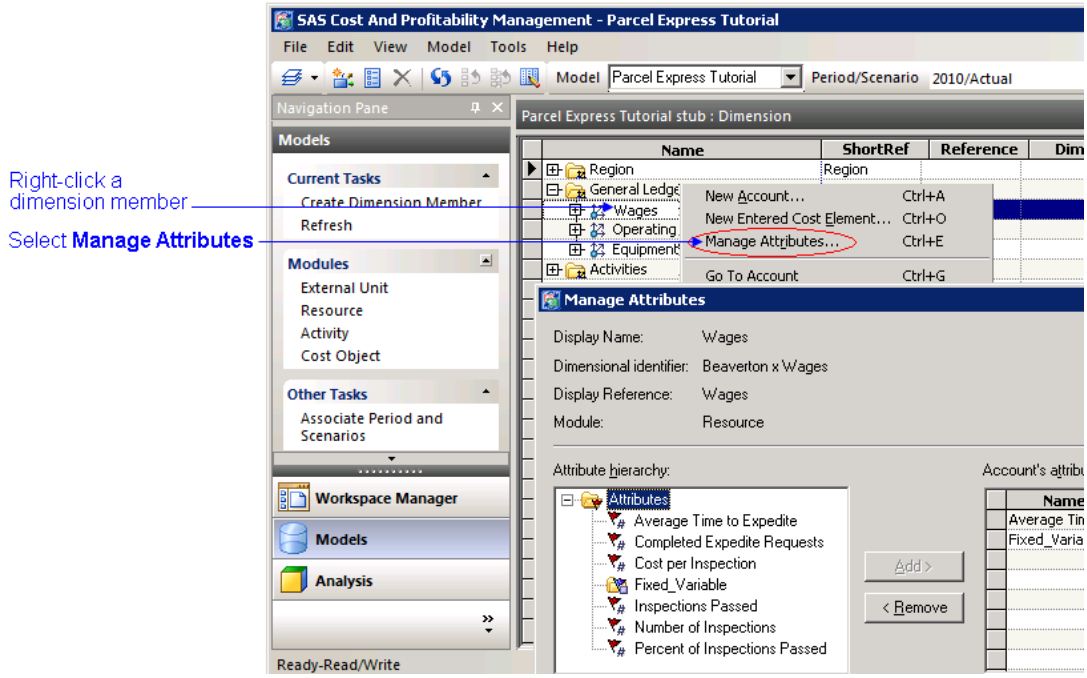
To use the Manage Attributes dialog:

1. Select **Model** ⇨ **Dimensions**.
2. Select a period/scenario association. The period/scenario that you select is the period/scenario in which any attribute that you add will apply.



3. Select a dimension member, right-click, and select **Manage Attributes**.

   The Manage Attributes dialog opens.



4. Select the attributes that you want to attach to the dimension member.
5. Click **OK**.

See Also

"Remove Attributes from Dimension Members" on page 291

Remove Attributes from Dimension Members

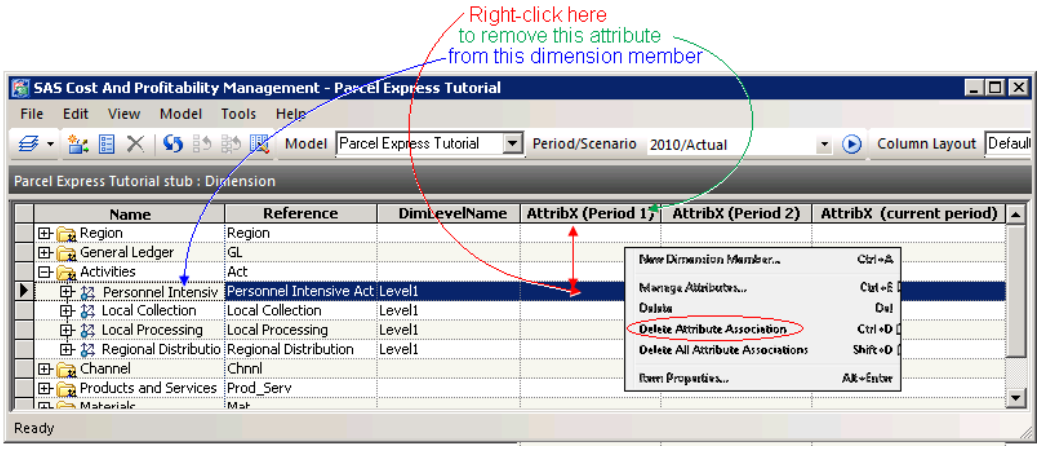
There are three ways in which you can remove the association of attributes with dimension members:

- Remove a particular attribute from a dimension member
- Remove all attributes from a dimension member
- Remove attributes using the Manage Attributes dialog

Remove a particular attribute from a dimension member

To remove a particular attribute from a dimension member:

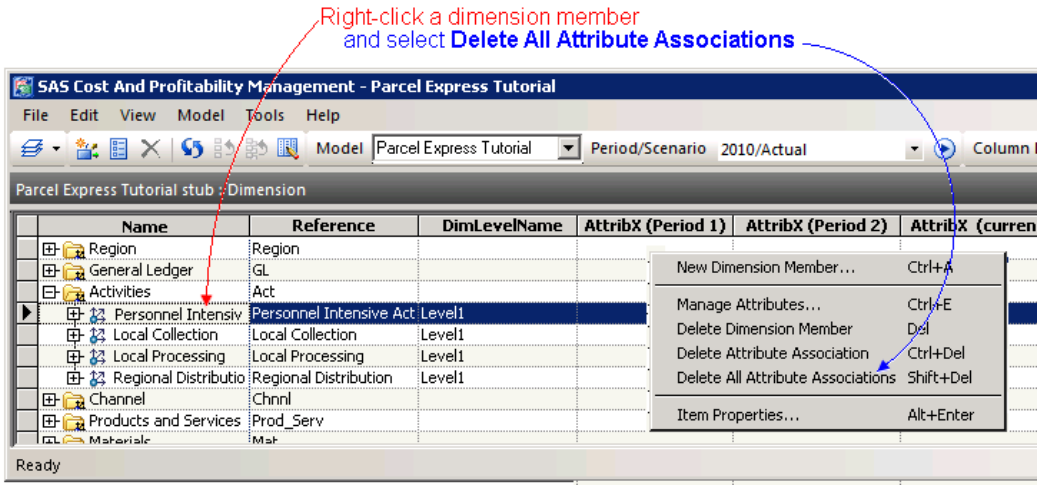
1. Select **Model** ⇨ **Dimensions** to open the Dimensions view.
2. Right-click on the cell where the column for the attribute to be deleted intersects the row of the dimension member from which the attribute is to be removed (see the following picture), and select **Delete Attribute Association**.



Remove all attributes from a dimension member

To remove all attributes from a dimension member:

1. Select **Model** ⇨ **Dimensions** to open the Dimensions view.
2. Right-click on the dimension member from which you want to remove all attributes, and select **Delete All Attribute Associations**.

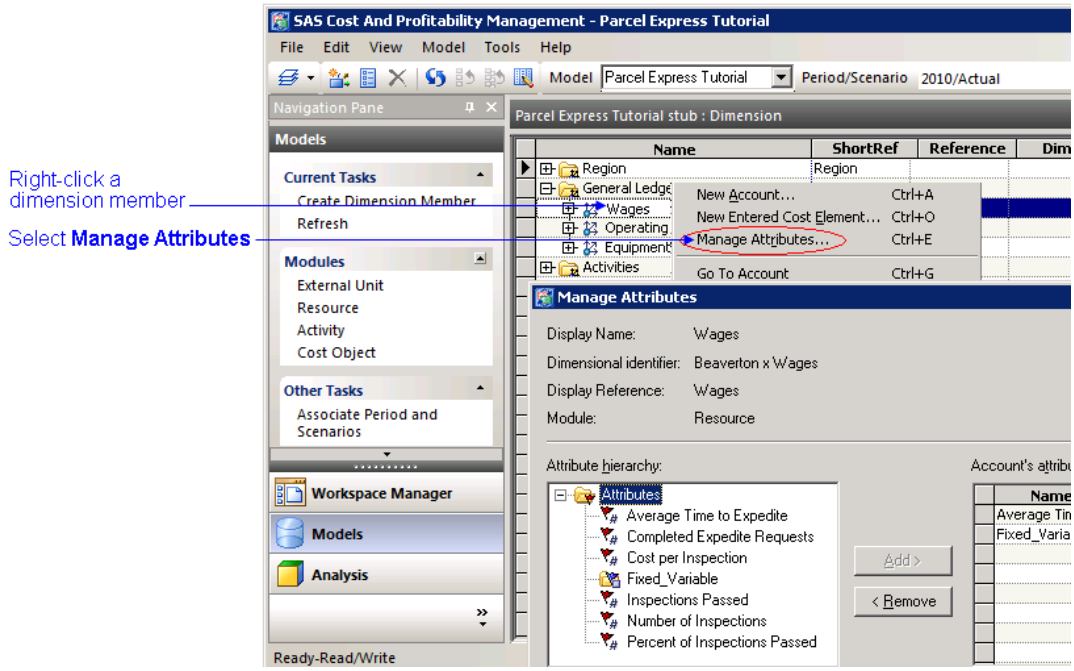


Remove attributes using the Manage Attributes dialog

To use the Manage Attributes dialog:

1. Select **Model** ⇨ **Dimensions** to open the Dimensions view.
2. Select a dimension member, right-click, and select **Manage Attributes**.

   The Manage Attributes dialog opens.



3. Select the attribute that you want to remove from the dimension member, and then click **Remove**.

4. Click **OK** to close the Manage Attributes dialog.

Chapter 35

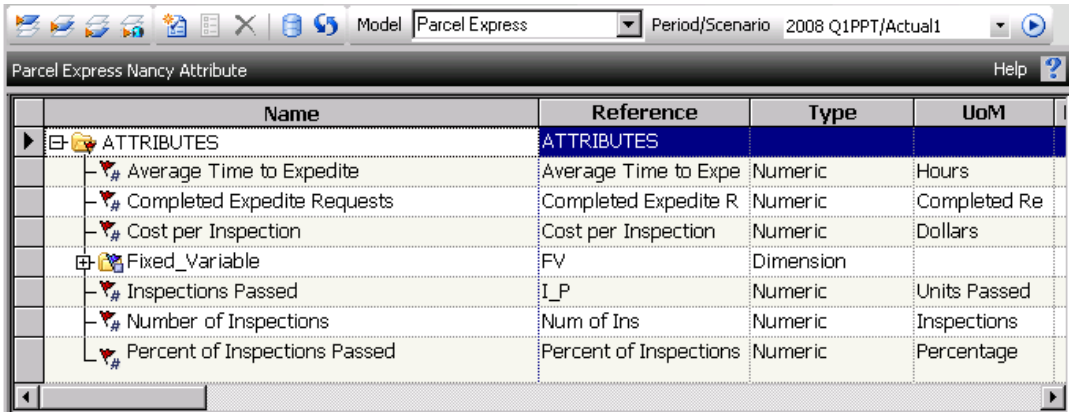
Windows for Attributes

Attributes View

About the Attributes View

The availability of this feature depends on your permissions.

On the Attributes view, you can manage a model's attributes.



Name	Reference	Type	UoM
ATTRIBUTES	ATTRIBUTES		
Average Time to Expedite	Average Time to Expe	Numeric	Hours
Completed Expedite Requests	Completed Expedite R	Numeric	Completed Re
Cost per Inspection	Cost per Inspection	Numeric	Dollars
Fixed_Variable	FV	Dimension	
Inspections Passed	I_P	Numeric	Units Passed
Number of Inspections	Num of Ins	Numeric	Inspections
Percent of Inspections Passed	Percent of Inspections	Numeric	Percentage

Note: You cannot directly edit the information on the Attributes view.

See Also

- Chapter 33, “Types of Attributes,” on page 271
- Chapter 36, “How To: Attributes,” on page 307
- Chapter 34, “Attributes on Dimension Members,” on page 277

How to Access the Attributes View

Open a modeland select **Model** ⇨ **Attributes View**.

See Also

- Chapter 33, “Types of Attributes,” on page 271
- “Create an Attribute” on page 307
- Chapter 36, “How To: Attributes,” on page 307

New Attribute Dialog Box

About the New Attribute Dialog Box

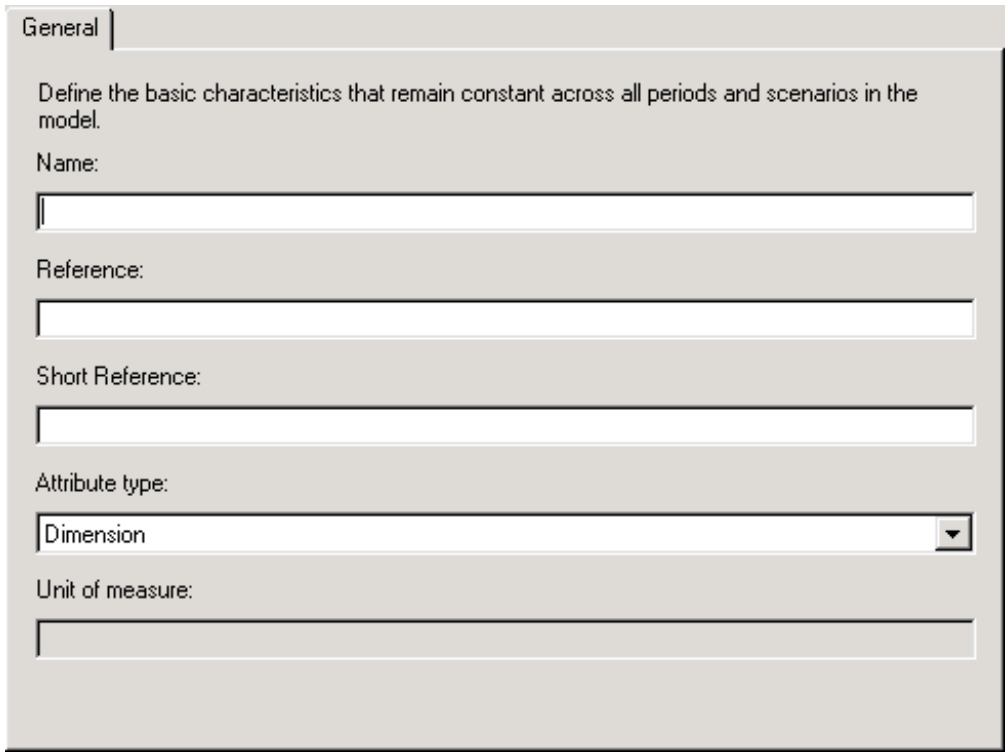
In the New Attribute dialog box, you can name a new attribute and you can specify other information about the attribute. You can create a dimension attribute folder.

Note: The availability of these features depends on your permissions.

How to Access the New Attribute Dialog Box

On the Attributes view, select either an attribute folder or **ATTRIBUTES** and select **Edit ⇨ New Attribute**.

General Tab



Specify the following general information for an attribute:

Name
See "Attribute naming conventions" on page 70.

Reference
See "Attribute reference conventions" on page 75.

Short Reference
A short reference is required only when the type of attribute is Dimension.

Attribute type

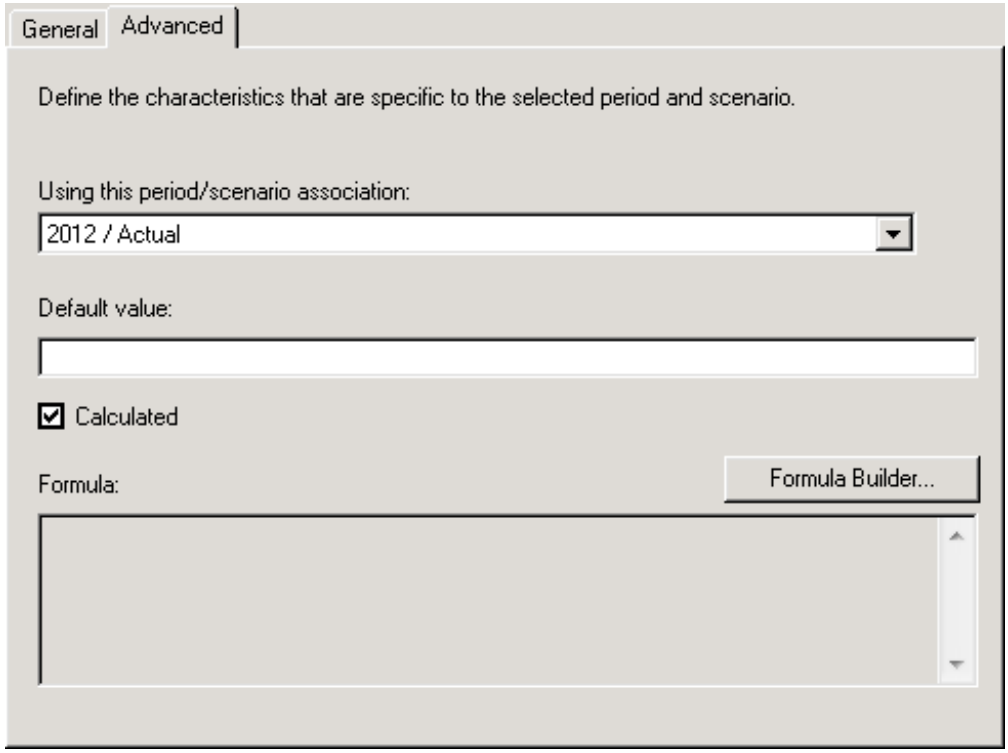
See Chapter 33, "Types of Attributes," on page 271.

Note: If you select **Dimension** when either **ATTRIBUTES** or an existing dimension attribute is selected, you will create a dimension attribute folder, not a dimension attribute.

Unit of measure

A unit of measure is required only for a numeric attribute.

Advanced Tab (for a Numeric Attribute)



The Advanced tab is available only for numeric attributes.

Using this period/scenario association

The value of a numeric attribute is periodic—it can vary from one period to another.

Default value

(Optional) Specify the value that this attribute is to have unless it is overridden when the attribute is attached to a particular account.

Note: A numeric attribute can have a default value even if the attribute is not attached to any account. The default value can be used, for example, in a driver formula. See "Using Numeric Attributes in a Formula" on page 467.

Calculated

Select this option for a calculated driver. See Chapter 44, "Calculated Driver," on page 393.

Click **Formula Builder** to define a formula. See "Formula Builder Dialog Box" on page 471.

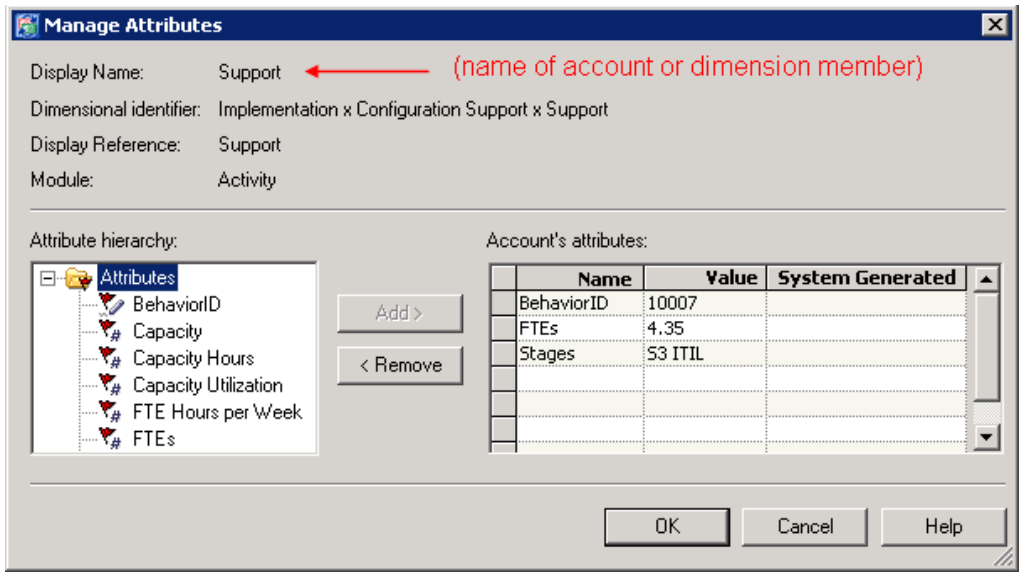
The Formula Builder dialog box appears.

TIP You can also modify the formula in the Formula box.

Manage Attributes Dialog Box

About the Manage Attributes Dialog Box

In the Manage Attributes dialog box, you can add or remove attributes and change their value.



The availability of these features depends on your permissions.

How to Access the Manage Attributes Dialog Box

- In a module view, select an account and select **Edit** ⇨ **Manage Attributes**.
- In the Dimensions view, select a dimension member and select **Edit** ⇨ **Manage Attributes**.

  Adding an attribute to a dimension member is an indirect way of adding the attribute to all the accounts that include that dimension member in its dimension signature. See Chapter 34, “Attributes on Dimension Members,” on page 277.

See Also

- “Attributes View” on page 296
- Chapter 33, “Types of Attributes,” on page 271
- Chapter 34, “Attributes on Dimension Members,” on page 277

System Generated Attributes

An attribute on an account is **System Generated** when it is added by the system to the account as a result of your having added the attribute to a dimension member that is in the dimension signature of that account.

Note: If you change the value of a system-generated attribute from its default value, then it is no longer a system-generated attribute. See "System-generated versus user-entered attribute values" on page 285.

See Also

Chapter 34, "Attributes on Dimension Members," on page 277

Attribute Properties Dialog Box

About the Attribute Properties Dialog Box

In the Attribute Properties dialog box, you can review or change information about an attribute.

The availability of these features depends on your permissions.

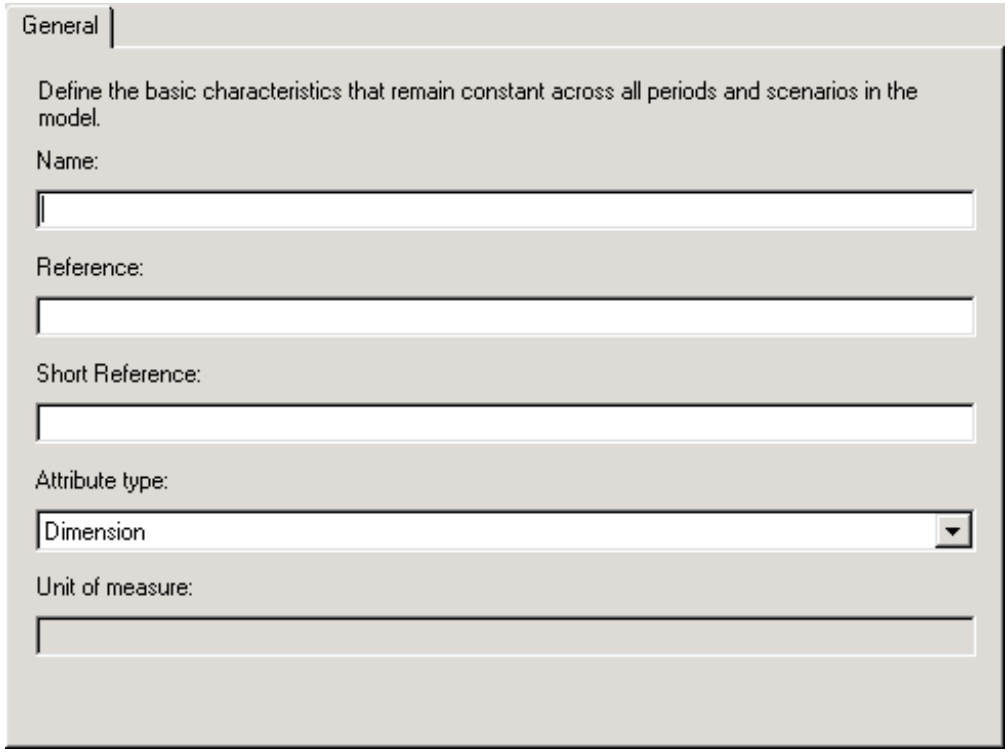
How to Access the Attribute Properties Dialog Box

On the Attributes view, select an attribute, and select **Edit** ⇨ **Item Properties**.

See Also

"Attributes View" on page 296

General Tab



Specify the following general information for an attribute:

Name
See "Attribute naming conventions" on page 70.

Reference
See "Attribute reference conventions" on page 75.

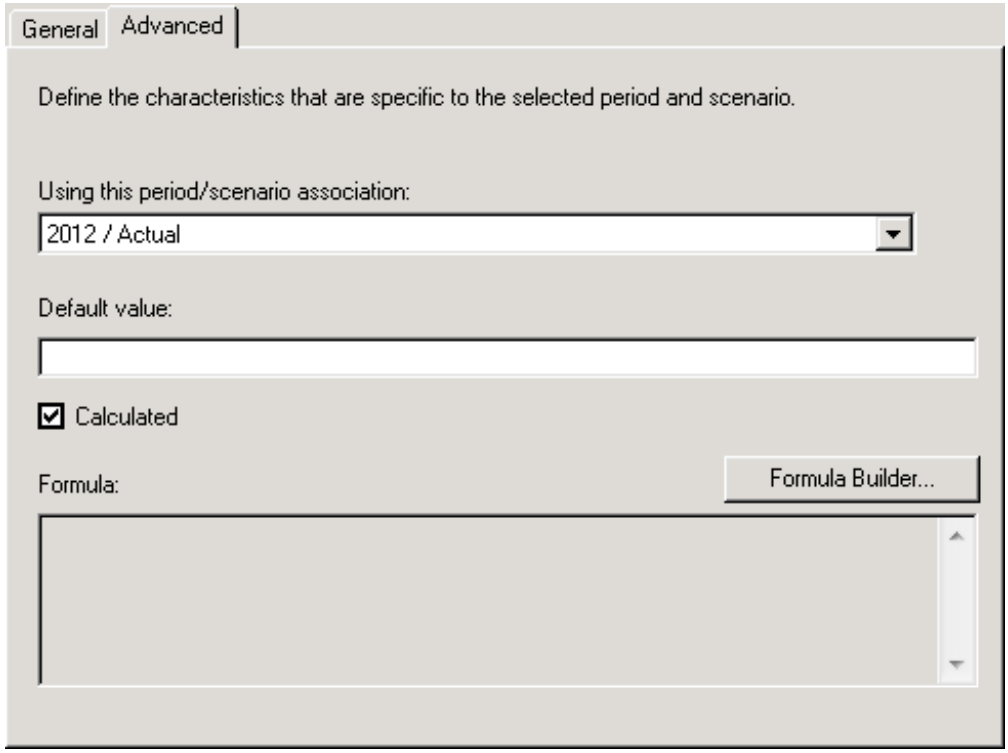
Short Reference
A short reference is required only when the type of attribute is Dimension.

Attribute type
See Chapter 33, "Types of Attributes," on page 271.

Note: If you select **Dimension** when either **ATTRIBUTES** or an existing dimension attribute is selected, you will create a dimension attribute folder, not a dimension attribute.

Unit of measure
A unit of measure is required only for a numeric attribute.

Advanced Tab (for a Numeric Attribute)



The Advanced tab is available only for numeric attributes.

Using this period/scenario association
The value of a numeric attribute is periodic—it can vary from one period to another.

Default value
(Optional) Specify the value that this attribute is to have unless it is overridden when the attribute is attached to a particular account.

Note: A numeric attribute can have a default value even if the attribute is not attached to any account. The default value can be used, for example, in a driver formula. See "Using Numeric Attributes in a Formula" on page 467.

Calculated
Select this option for a calculated driver. See Chapter 44, "Calculated Driver," on page 393.

Click **Formula Builder** to define a formula. See "Formula Builder Dialog Box" on page 471.

The Formula Builder dialog box appears.

TIP You can also modify the formula in the Formula box.

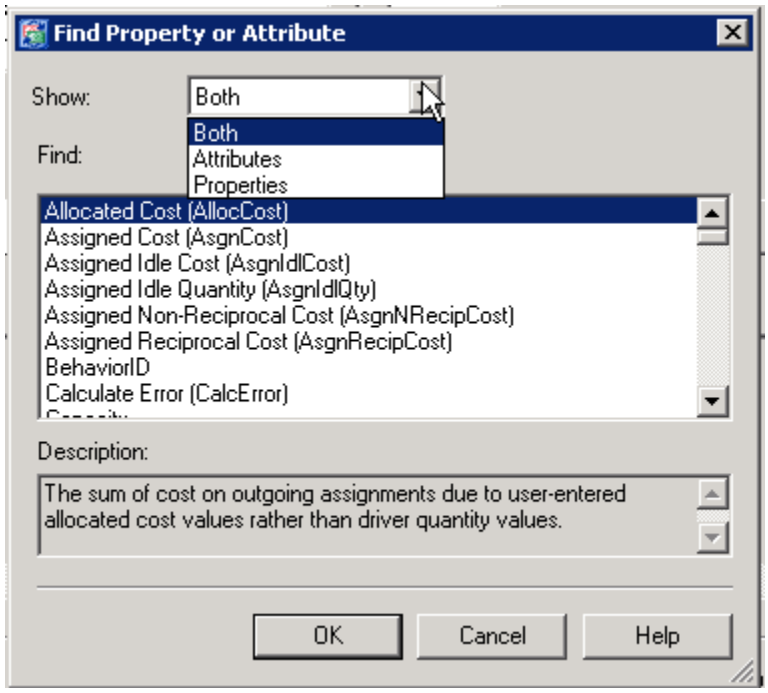
See Also

"Calculated Attributes" on page 274

Find a Property or Attribute Dialog Box

About the Find a Property or Attribute Dialog Box

Use this dialog to find a property or attribute that you want to select in order to build a search query in the Search for Accounts dialog box.



How to Access the Find a Property or Attribute Dialog Box

In the Search for Accounts dialog box, click **Find**.

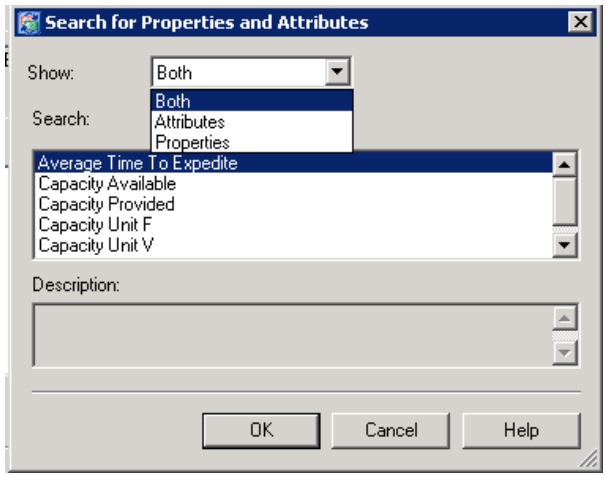
See Also

- "Search for Accounts Dialog Box" on page 237
- "Search Account Results Dialog Box" on page 238

Search for Properties and Attributes Dialog Box

About the Search for Properties and Attributes Dialog Box

In the Search for Properties and Attributes dialog box, you can find data to display when you add a column, or you can find a property when you review the properties of an item or when you review the attributes of an item.



How to Access the Search for Properties and Attributes Dialog Box

Do one of the following:

- In the Column Layout dialog box, click **Search**.

  See "Column Layout Dialog Box" on page 335.

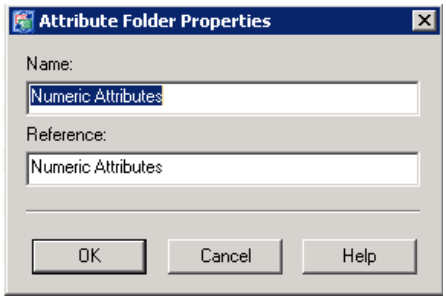
- In the Item Properties dialog box, click **Search**.

  See " Item (Account) Properties and Attributes Dialog Box" on page 239.

Attribute Folder Properties Dialog Box

About the Attribute Folder Properties Dialog Box

In the Attribute Folder Properties dialog box, you can review or change information about an attribute folder.



The availability of this feature depends on your permissions.

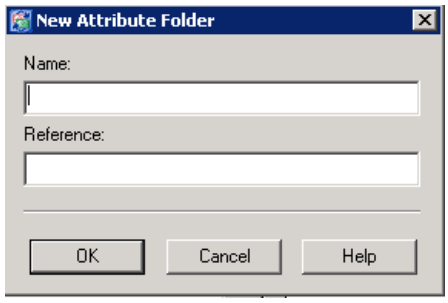
How to Access the Attribute Folder Properties Dialog Box

On the Attributes view, select a folder, and select **Edit** ⇨ **Item Properties**.

New Attribute Folder Dialog Box

About the New Attribute Folder Dialog Box

In the New Attribute Folder dialog box, you can name a new folder and you can specify the folder's reference.



Note: The availability of these features depends on your permissions.

How to Access the New Attribute Folder Dialog Box

On the Attributes view, select an item and select **Edit** ⇨ **New Folder**.

See Also

- "Attributes View" on page 296
- "Naming Conventions" on page 69
- "Reference Conventions" on page 74

Chapter 36

How To: Attributes

Create an Attribute

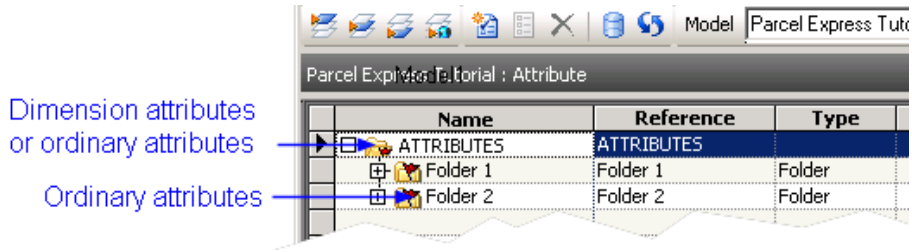
1. Open a model and select **Model** ⇨ **Attributes Page**.

   The Attributes page appears.

2. Select the folder in which the attribute is to go.

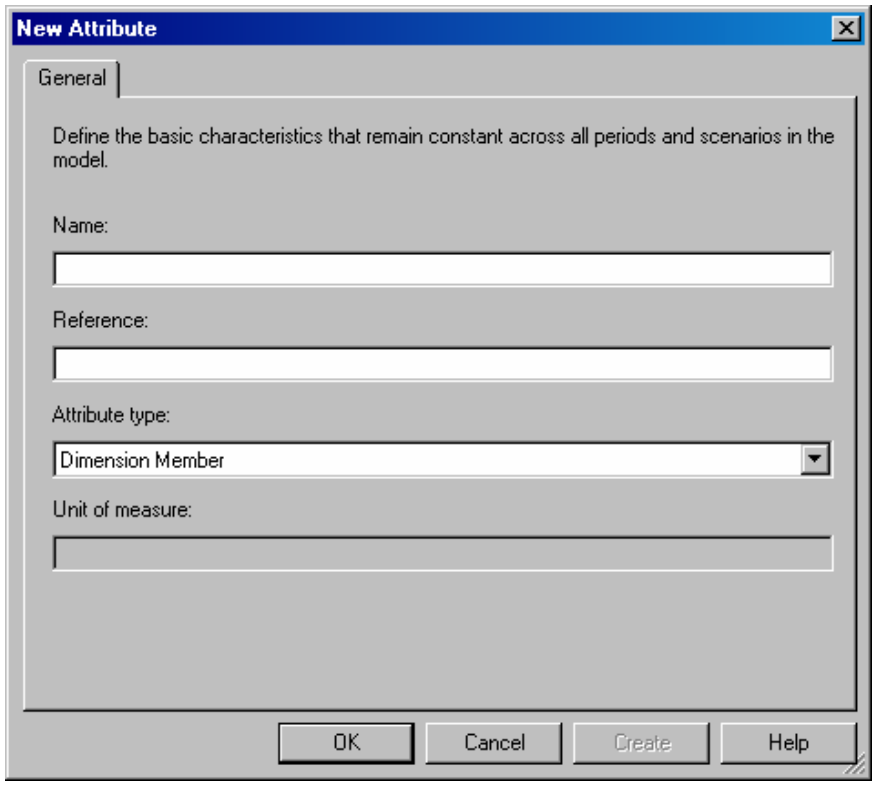
   A dimension attribute can go in the **ATTRIBUTES** folder or in an existing dimension attribute.

   Other attributes can go in the **ATTRIBUTES** folder or in another attribute folder.



3. Select **Edit** ⇨ **New Attribute**.

   The New Attribute dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

5. Type the **Reference**.

   An attribute reference is required only for dimension attributes and is used in public views. See the chapter on "Public Views" in the *SAS Activity-B ased Management Data Administration Guide*. A reference must follow the reference conventions. See "Reference Conventions" on page 74.

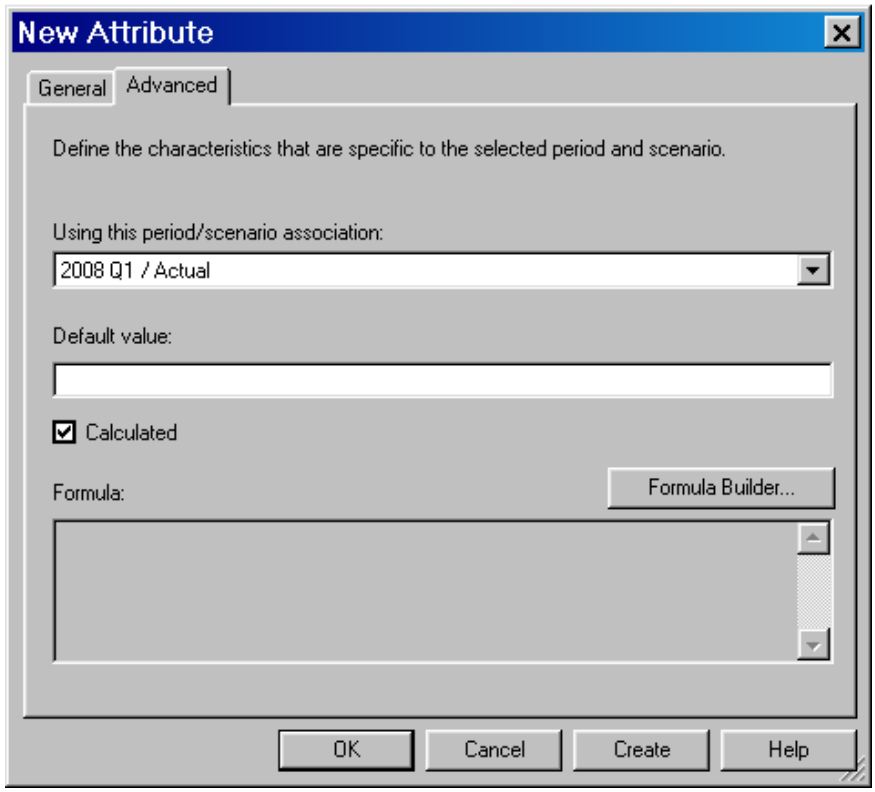
6. Select the **Attribute type**.

   Note: If you select **Dimension** when either **ATTRIBUTES** or an existing dimension attribute is selected, you will create a dimension attribute folder, not a dimension attribute.

   For a numeric attribute only:

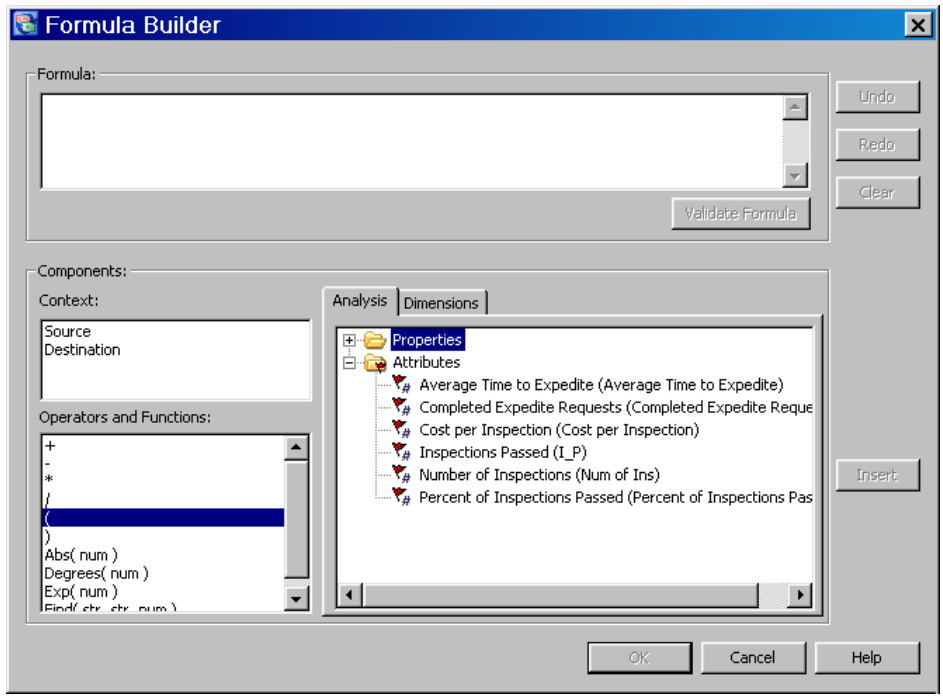
7. Type the **Unit of measure**.

8. Click the **Advanced** tab.



9. From the **Using this period/scenario association** drop-down list, select a period/scenario association.
10. Type the **Default value**.
11. If this is a calculated attribute, perform the following steps:
    a. Select the **Calculated** option.
    b. Click **Formula Builder**.

    The Formula Builder dialog box appears.



TIP You can also modify the formula in the **Formula** box.

c. Select an item from the list of **Numeric properties**, from the list of **Operators**, or from the list of **Attributes**.

d. Click **Insert**.

The item is added to the formula.

TIP You can also type the formula.

e. Continue to insert items until you have built the entire formula.

f. To remove the last item that you inserted, click **Undo**.

g. To delete the entire formula, click **Clear**.

h. Click **Test**.

If no errors are displayed, the syntax of the formula is valid.

See Also

- Chapter 33, “Types of Attributes,” on page 271
- “Add an Attribute to an Account” on page 311
- “Specify an Attribute's Value” on page 312
- “Show the Accounts To Which an Attribute Has Been Added” on page 314
- “Remove an Attribute from an Account” on page 315

Create a Dimension Attribute

1. Select **ATTRIBUTES** or an existing dimension attribute.

2. Select **Edit** ⇨ **New Attribute**.

   The New Attribute dialog box appears. See "New Attribute Dialog Box" on page 297.

See Also

"Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes" on page 272

Create an Attribute Folder

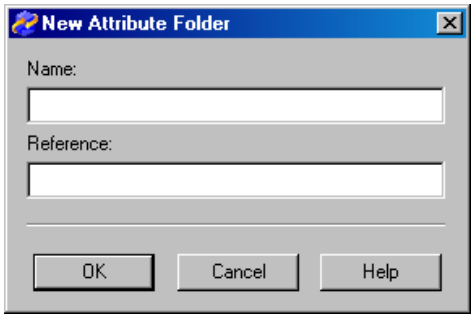
1. Open a model and select **Model** ⇨ **Attributes Page**.

   The Attributes page appears.

2. Select an attribute folder (other than a dimension attribute folder) within which to create the new folder.

3. Select **Edit** ⇨ **New Folder**.

   The New Attribute Folder dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. See "Naming Conventions" on page 69.

5. Type the **Reference**.

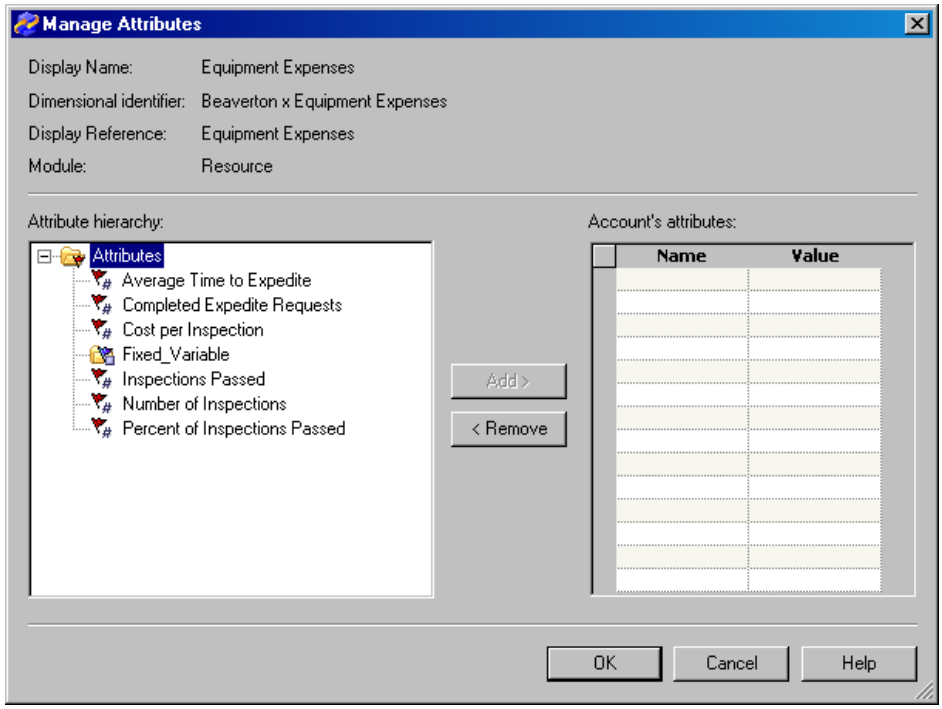
   A default reference is created from the name. If you change the reference, the new reference must follow the reference conventions. See "Reference Conventions" on page 74.

Add an Attribute to an Account

Note: This topic describes how to add an attribute directly to an account. You can also add an attribute to accounts indirectly by adding the attribute to a dimension member that is included in an account's dimension signature. For information, see Chapter 34, "Attributes on Dimension Members," on page 277.

1. On a module page, select a account, and select **Edit** ⇨ **Manage Attributes**.

   The Manage Attributes dialog box appears.



2. From the **Attribute hierarchy** list, select an attribute.
3. Click **Add >**.

   The attribute is added to the **Account's attributes** list.

See Also

"Specify an Attribute's Value" on page 312

Add an Attribute to a Dimension Member

You can add an attribute to a dimension member as an indirect way of adding the attribute to account. If you add an attribute to a dimension member, then the system automatically adds the attribute to all the accounts whose dimension signature includes that dimension member.

See Chapter 34, "Attributes on Dimension Members," on page 277.

Specify an Attribute's Value

1. On a module page, select an account, and select **Edit** ⇨ **Manage Attributes**.

   The Manage Attributes dialog box appears.
2. From the **Account's attributes** list, select an attribute.
3. Set the **Value**.

   The type of value that you can specify depends on the type of attribute.

See Also

- "Show the Accounts To Which an Attribute Has Been Added" on page 314
- "Manage Attributes Dialog Box" on page 299

Add Stage Attributes to Accounts

Overview

Adding stage attributes to accounts requires two steps:

1. Create stage attributes
2. Add stage attributes to accounts

To Create Stage Attributes:

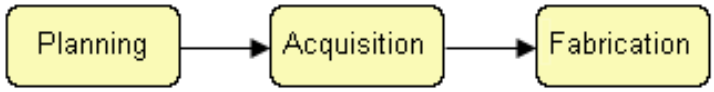
1. Open a model and select **Model** ⇨ **Attributes view**. The Attributes view opens.
2. On the Attributes view, select **Edit** ⇨ **New Attribute** (or click the **New Attribute** button). The New Attribute dialog box opens.
3. In the New Attribute dialog box, specify **Stages** as the name of the new attribute and select **Dimension** as its Attribute Type, then click **OK**.
4. On the Attributes view, select the Stages dimension (that you created in Step 3) and select **Edit** ⇨ **New Attribute** (or click the **New Attribute** button). The New Attribute dialog opens again.
5. From the New Attribute dialog box, name the new attribute (which represents a stage) and select **Dimension Member** as the Attribute Type.
6. Repeat steps 4 and 5 for as many times as you want stages - one dimension member attribute per stage.

   The direction of cost flow between stages is determined by the display order of stage attributes, so you can assign any name and reference to stages that you want. For example, in the following picture you can see that the display order of three stage attributes is:

   1. Planning
   2. Acquisition
   3. Fabrication

Name	Reference	Type
ATTRIBUTES (PRIMARY PANE)		
Stages	Stages	Dimension
Planning	Planning	Dimension Member
Acquisition	Acquisition	Dimension Member
Fabrication	Fabrication	Dimension Member

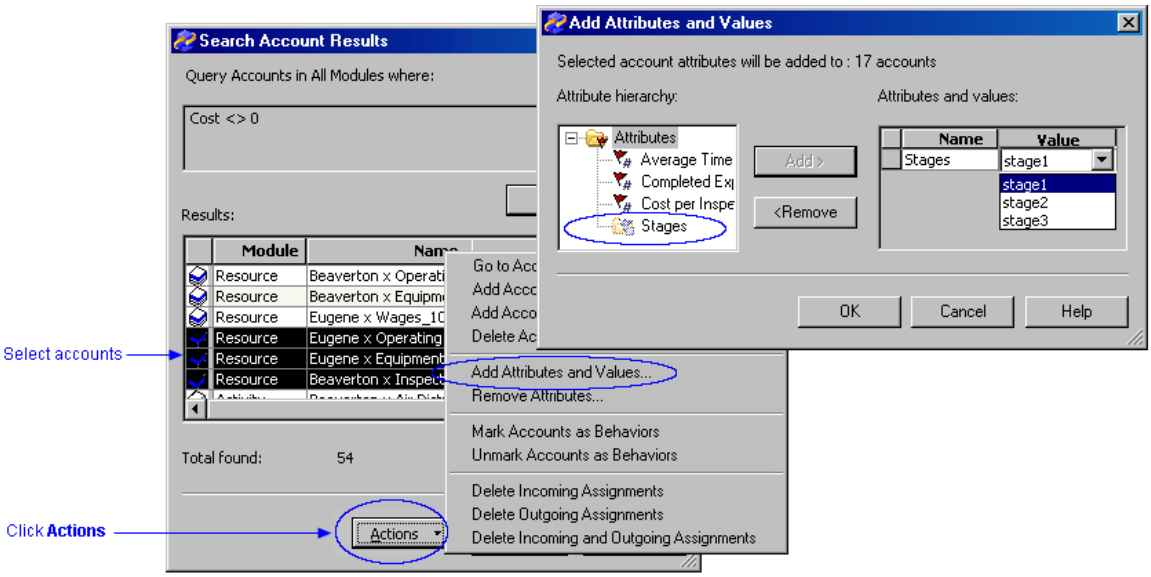
   Consequently, *Planning* flows to *Acquisition* which flows to *Fabrication* because that is their display order.



To Add Stage Attributes to Accounts, Do One of the Following:

- Add attributes from the Manage Attributes dialog box.
- Search for accounts and add attributes to accounts that are found.

Note: All of the accounts for a stage must belong to the same module (Resource, Activity, Cost Object, or xternal Unit). The following picture shows adding the stage1 attribute to accounts that are found as a result of searching for accounts.



See Also

"Stage Attributes" on page 275

Show the Accounts To Which an Attribute Has Been Added

You can divide the Attributes page to see the accounts to which an attribute has been added.

1. Select **Model** ⇨ **Attributes**.

   The Attributes page appears.

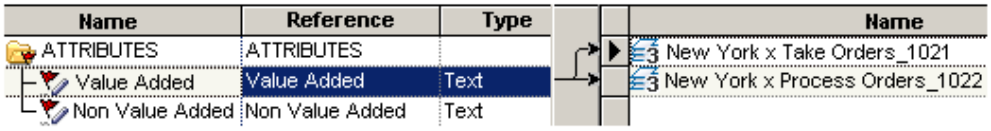
2. Select **Model** ⇨ **Assignments** ⇨ **Show Right Assignments Pane**.

   The right assignments pane is displayed.

3. Select an attribute.

4. Select **Model** ⇨ **Assignments** ⇨ **Show Right**.

   If the attribute has been added to any accounts, then those accounts are displayed.



Remove an Attribute from an Account

1. On a module page, select a account, and select **Edit** ⇨ **Manage Attributes**.

   The Manage Attributes dialog box appears.
2. From the **Account's attributes** list, select an attribute.
3. Click **Remove**.

   The attribute is removed from the **Attribute hierarchy** list.

Delete an Attribute

1. Open a model and select **Model** ⇨ **Attributes Page**.
2. Select an attribute.
3. Select **Edit** ⇨ **Delete**.

Change the Properties of an Attribute

1. Open a model and select **Model** ⇨ **Attributes Page**.
2. Select an attribute.
3. Select **Edit** ⇨ **Item Properties**.

   The Attribute Properties dialog box appears. See "Attribute Properties Dialog Box" on page 300.

Mark a Numeric Attribute for What-If Analysis

See "Mark Numeric Attributes as Independent Variables" on page 608.

Part 12

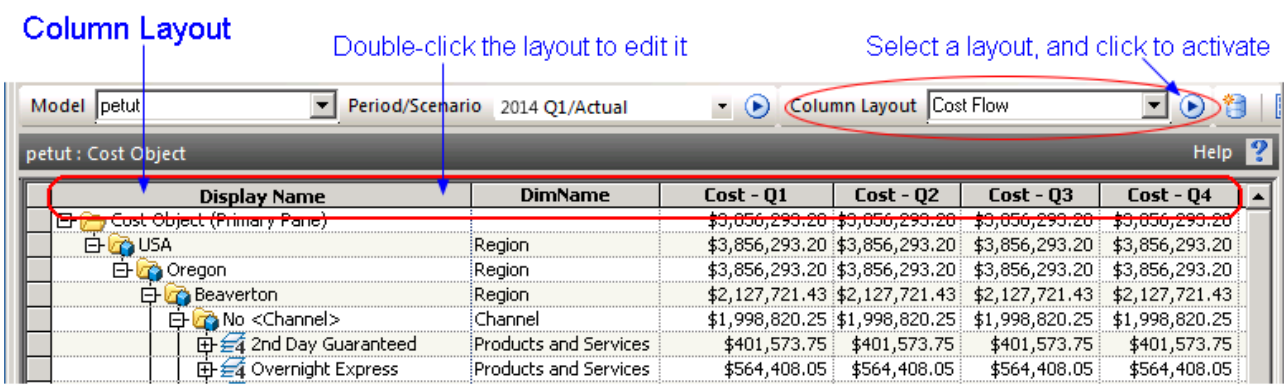
Column Layouts

Chapter 37

Working with Column Layouts

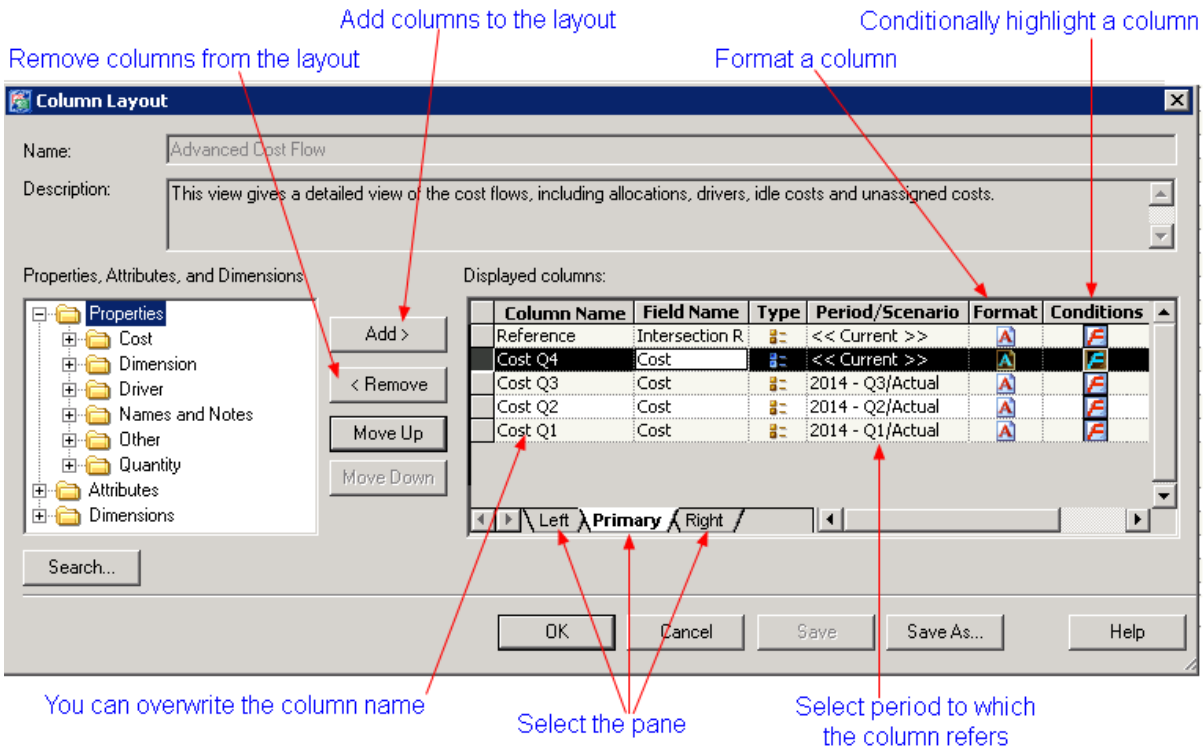
What is a Column Layout?

SAS Cost and Profitability Management displays information in a grid. A column layout is a set of column headers for a grid along with the specification for how information in those columns should appear.



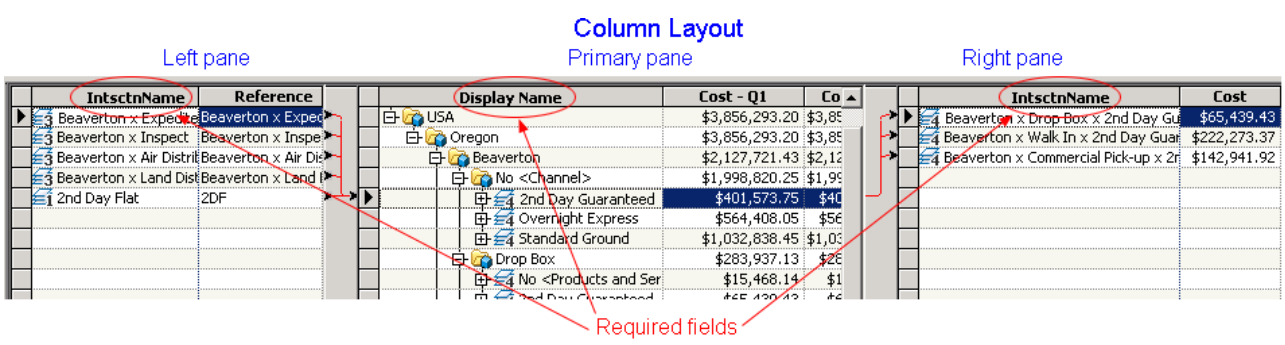
Using the Column Layout dialog ("Column Layout Dialog Box" on page 335) you can customize and save your own column layouts for the following:

- Module views
- Dimensions view

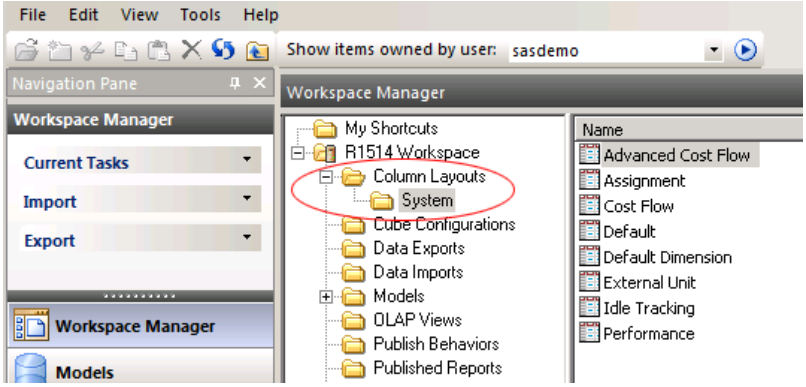


It is important to understand that a column layout for a module view specifies the columns for all the panes of the module view:

- Left pane
- Primary pane
- Right pane



In addition to custom column layouts that you define, the system provides a number of layouts for your use. You can find all the column layouts in the Column Layouts folder of the Workspace manager. See "Workspace Manager" on page 53.



You can import and export your saved custom column layouts:

- "Import a Column Layout" on page 679
- "Export a Column Layout" on page 679

You can also set the default column layout for each module in a model and for the Dimensions view of a model:

- See "Specify a Default Column Layout" on page 351.

System Column Layouts

SAS Cost and Profitability Management provides the following predefined column layouts. You can not modify these layouts, but you can add to or remove columns from them and then save the layout by another name for future use as a custom column layout.

Note: **Intersection Name** is a required column which cannot be removed in the Left and Right panes. **Display name** is a required column in the Primary pane and also cannot be removed.

Default Dimension

This view is presented by default when you open the Dimensions view.

Primary pane
Name
ShortRef
Reference
DimLevelName

Default (Module View)

This view is presented by default when you view a module. It contains minimal column information.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
Reference	Display Reference	Reference
Cost	Cost	Cost

Cost Flow

This view is designed to display the properties needed to explain the flow of cost through the model. It is optimized for the three-pane view.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
DrvName	DrvName	DrvCost
Cost	Cost	DrvQtyCalc

Advanced Cost Flow

This view gives a detailed view of the cost flows, including allocations, drivers, idle costs and unassigned costs. It is optimized for the primary and right panes.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
	DrvName	DrvCost
	Cost	DrvAllocCost
	AllocCost	DrvIdlCost
	DrvnCost	
	IdlCost	
	AsgnIdlCost	
	UnAsgnCost	

Assignment

This view is used when making assignments and entering or verifying properties of the assignment relationship. It is optimized for the primary and left panes.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
Reference	Reference	Reference
DrvName	DrvName	DQF
Cost	TDQCalc	DQV
		DWF
		DWV
		DrvQtyCalc
		DrvAllocCost

IdleTracking

This view is used for managing idle cost flows. It is optimized for the primary and right panes only.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
	DrvName	IdlQtyUE
	TDQCalc	DrvQtyCalc
	TDQUE	DQF
	IdlQty	DQV
	UnAsgnQty	DWF
		DWV

Performance

This view is used when investigating performance of accounts such as profitability and account costs and unit costs. It is intended to be a primary pane view only.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
Cost	Revenue	Cost
UnitCost	Cost	UnitCost
	Profit	
	SoldQty	
	SoldCost	
	UnitRevenue	
	UnitCost	
	UnitProfit	

External Unit

This view contains columns for EntCost and EntUnitCost. All cost in a model originates with these two properties.

Left pane	Primary pane	Right pane
(Intersection Name)	(Display Name)	(Intersection Name)
Reference	UsedQty	Cost
Cost	EntCost	UnitCost
	EntUnitCost	DrvCost
	Cost	

Understanding Properties in a Column Layout

Overview

This chapter discusses how to understand the significance of properties when they are added to a column layout.

Note: With the ability to add attributes to dimension members, you can also add properties to column layouts for the Dimensions view (see Chapter 34, "Attributes on Dimension Members," on page 277). But, in this chapter, we are concerned only with adding properties to column layouts for assignments panes.

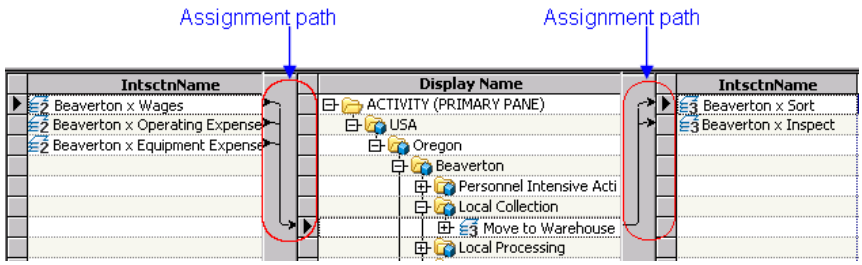
Assignments panes provide a window into accounts. The primary assignments pane shows all the accounts in the module to which the pane is attached. The left and right assignments panes provide a restricted (filtered) view of accounts:

- The left assignments pane shows accounts making assignments to the primary pane.
- The right assignments pane shows accounts receiving assignments from the primary pane.

The primary assignments pane shows accounts in one module only. Given the existence of intra-modular assignments (assignments within the same module), the right and left assignments panes can show accounts in more than one module.

Properties fall into the following categories:

assignment properties
refer to a particular assignment between accounts. Assignments are displayed in the narrow area between assignments panes, as shown in the following picture:



In a column layout, given that you cannot add properties in between assignments panes, the following question arises: if you attach an assignment property to one assignments pane, does it refer to an assignment to the left of that pane or to the right of it? The answer, as we shall see in more detail shortly, is to the left.

Outgoing properties
refer to costs flowing out of an account, across assignment paths, into one or more destination accounts. Outgoing properties are relational. They attach to an account, but they refer to cost flows to other accounts.

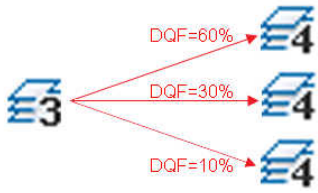
Incoming properties
refer to costs flowing into an account, across an assignment path, from one or more source accounts. Incoming properties are also relational. They refer to cost flows from other accounts.

Account properties
refer to accounts apart from any flow into or out of the account.

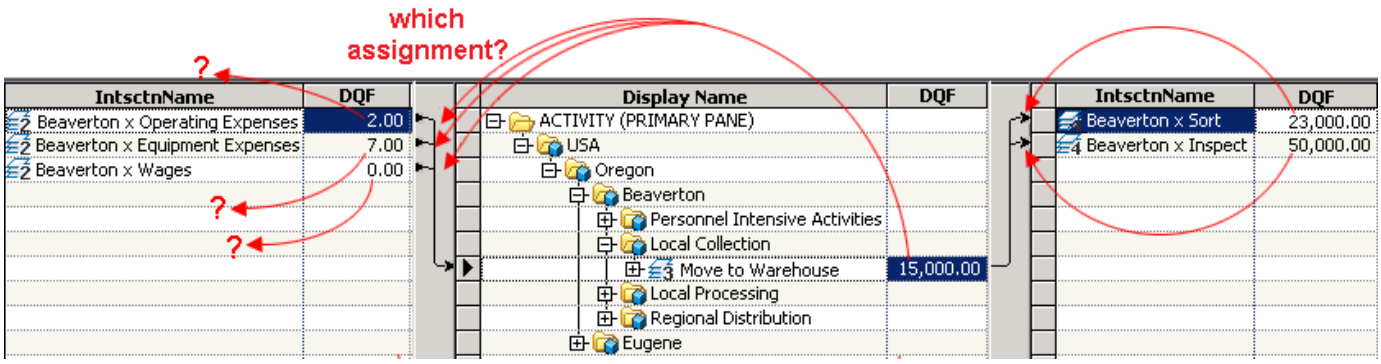
Driver properties
refer to drivers apart from any particular assignment between accounts.

Assignment Properties

An *assignment property* describes a flow between accounts. Driver Quantity Fixed (DQF), for example for a percentage driver, specifies the percentage of an account's cost that flows to each of its destination accounts. The DQF for one destination account can be, say, 60%, and 30% for a different destination account, and 10% for another.



Given that you attach properties only to assignments panes and not to the flow between panes, it makes most sense to attach assignment properties to the right assignments pane. The following picture shows why. Notice that in the right assignments pane, you can clearly see to which assignment path each DQF of 23,000 and 50,000, respectively, applies. By contrast, in the primary pane, you can see the DQF of 15,000, but to which incoming assignment does it apply? When there is more than one incoming assignment (in this case there are three), you cannot tell from the primary assignments panes to which assignment it applies. And the situation is even worse in the left assignments pane because you can't see any of the incoming assignments. So, in short, attach assignment properties to the right assignments pane.



The following table lists assignment properties. The second column shows the properties whose value you can set in an assignments pane.

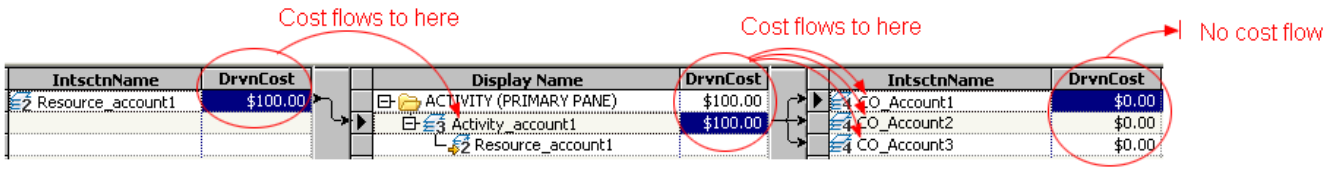
Table 37.1 *Assignment Properties*

Property	Set By User?
Driver Allocated Cost	Yes
Driver Cost	
Driver Driven Cost	

Property	Set By User?
Driver Driven Quantity	
Driver Idle Cost	
Driver Percentage	
Driver Quantity Basic	
Driver Quantity Calculated	
Driver Quantity Fixed (DQF)	Yes
Driver Quantity Variable (DQV)	Yes
Driver Used Cost	
Driver Weight Fixed (DWF)	Yes
Driver Weight Variable (DWV)	Yes
Idle Driver Quantity	
Idle Driver Quantity UE	Yes
Idle Percentage	
Is Reciprocal	
Reciprocal Id	

Outgoing Properties

An *outgoing property* refers to a cost flow going out of accounts across assignment paths to destination accounts. The following picture shows the DrvnCost property attached to the left, primary, and right assignments panes. In every case, regardless of which pane the property is attached to, it shows something going out of an account to destination accounts.



The following table lists outgoing properties. The second column lists the properties whose value you can set in an assignments pane. With two exceptions, you can not set outgoing properties.

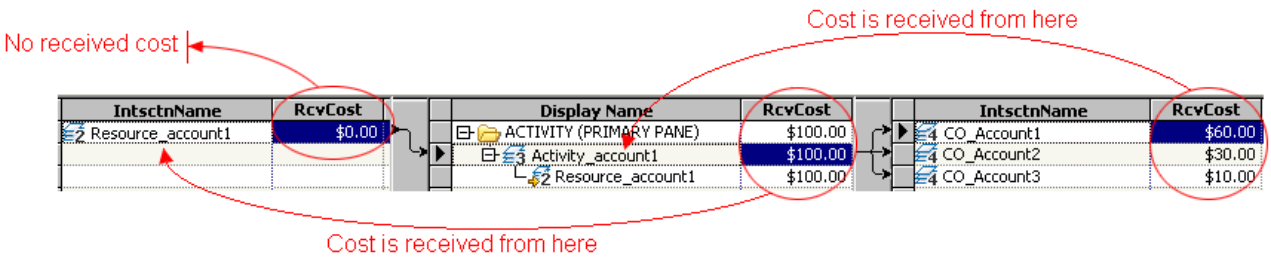
Table 37.2 *Outgoing Properties*

Property	Set By User?
Allocated Cost	
Assigned Cost	
Assigned Idle Cost	
Assigned Idle Quantity	
Assigned Non-Reciprocal Cost	
Assigned Reciprocal Cost	
Calculate Error	
Drivable Cost	
Driven Cost	
Driven Quantity	
Driver Rate	
EnteredUnitCost	Yes
Has Idle Cost	
Has Used Cost	
Idle Cost	
Idle Quantity	
Total Driver Quantity (TDQ)	
Total Driver Quantity Basic (TDQBasic)	
Total Driver Quantity Calculated (TDQCalc)	
Total Driver Quantity UE (TDQUE)	Yes
Unassigned Cost	
Unassigned Quantity	
Unit Cost	

Property	Set By User?
Used Cost	
Used Quantity	

Incoming Properties

An *incoming property* refers to cost flows coming into accounts, across assignment paths, from source accounts. The following picture shows Received Cost attached to the left, primary, and right assignments panes. In every case, regardless of which pane the property is attached to, the property shows costs flowing into an account from source accounts.



The following table lists incoming properties.

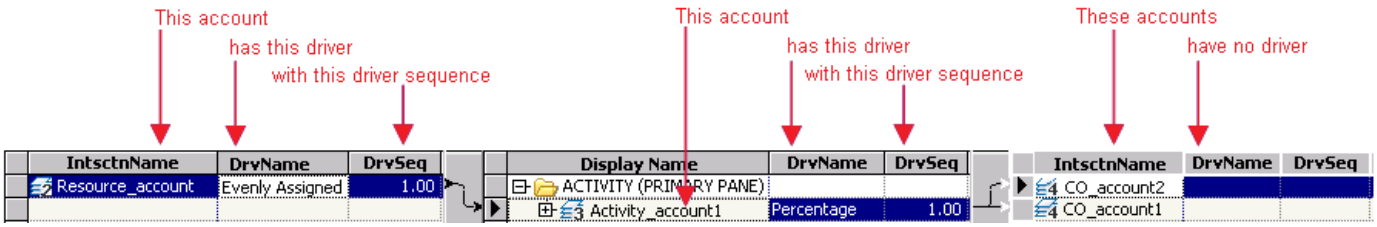
Table 37.3 *Incoming Properties*

Property	Set By User?
Has Assignments	
Received Allocated Cost	
Received Cost	
Received Driven Cost	
Received Idle Cost	
Received Non-Reciprocal Cost	
Received Reciprocal Cost	
Received Used Cost	

Driver Properties

Driver properties are properties of a driver that are independent of any particular assignment. For example, Driver Name is the same for a given driver in all its assignment paths.

The following picture shows two driver properties (DrvName and DrvSeq) attached to all three assignments panes.



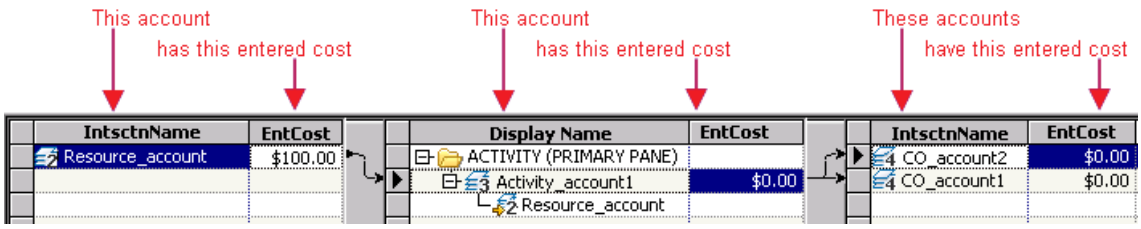
The following table lists driver properties. You can set driver properties only in driver dialogs, not in assignments panes.

Table 37.4 *Driver Properties*

Property
Driver Formula
Driver Name
Driver Sequence Number
Driver Type
Idle Flow Method
Unique Driver Quantities
Use Fixed Quantities
Use Variable Quantities
Use Weighted Quantities

Account Properties

Accounts properties are properties of an account considered by itself, apart from any flow into or out of the account. For example, Entered Cost is an account property and not an outgoing property because it may or may not flow out of the account to which it is attached. The following picture shows the account property (EntCost) attached to all three assignments panes.



The following table lists account properties.

Table 37.5 *Account Properties*

Property	Set By User?
Cost	
Entered Cost	Yes
Entered Unit Cost	Yes
Has Attributes	
Has Entered Cost	
Has Notes	
Is Behavior	Yes
Is Reciprocal	
Model Note	Yes
Periodic Note	Yes
Profit	
Output Quantity	
Output Quantity UE	Yes
Reciprocal Id	
Revenue	Yes
Sold Cost	
Sold Quantity	Yes
Unit Profit	
Unit Revenue	

Summary

In thinking about properties, you might find useful the following rule of thumb: if a property is prefixed with “Driver”, then it is either an assignment property or a driver property. If it is prefixed with “Received”, then it is an incoming property. And, if it is prefixed with neither “Driver” nor “Received”, then it is either an outgoing property or an account property.

The following table lists the properties in alphabetical order and specifies the type of property for each one.

Table 37.6 *Property Type*

Property	Outgoing	Incoming	Assignment	Account	Driver
Allocated Cost	✓				
Assigned Cost	✓				
Assigned Idle Cost	✓				
Assigned Idle Quantity	✓				
Assigned Non-Reciprocal Cost	✓				
Assigned Reciprocal Cost	✓				
Calculate Error	✓				
Cost				✓	
Drivable Cost	✓				
Driven Cost	✓				
Driven Quantity	✓				
Driver Allocated Cost			✓		
Driver Cost			✓		
Driver Driven Cost			✓		
Driver Driven Quantity			✓		
Driver Formula					✓
Driver Idle Cost			✓		
Driver Name					✓
Driver Percentage			✓		
Driver Quantity Basic			✓		
Driver Quantity Calculated			✓		

Property	Outgoing	Incoming	Assignment	Account	Driver
Driver Quantity Fixed (DQF)			✓		
Driver Quantity Variable (DQV)			✓		
Driver Rate	✓				
Driver Sequence Number					✓
Driver Type					✓
Driver Used Cost			✓		
Driver Weight Fixed (DWF)			✓		
Driver Weight Variable (DWV)			✓		
Entered Cost				✓	
Entered Unit Cost				✓	
Has Assignments		✓			
Has Attributes				✓	
Has Entered Cost				✓	
Has Idle Cost	✓				
Has Notes				✓	
Has Used Cost	✓				
Idle Cost	✓				
Idle Driver Quantity			✓		
Idle Driver Quantity UE			✓		
Idle Flow Method					✓
Idle Percentage			✓		
Idle Quantity	✓				
Is Behavior				✓	
Is Reciprocal			✓	✓	
Model Note				✓	
Model Note				✓	
Output Quantity				✓	
Output Quantity UE				✓	
Periodic Note				✓	

Property	Outgoing	Incoming	Assignment	Account	Driver
Profit				✓	
Received Allocated Cost		✓			
Received Cost		✓			
Received Driven Cost		✓			
Received Idle Cost		✓			
Received Non-Reciprocal Cost		✓			
Received Reciprocal Cost		✓			
Received Used Cost		✓			
Reciprocal Id			✓	✓	
Revenue				✓	
Sold Cost				✓	
Sold Quantity				✓	
Total Driver Quantity (TDQ)	✓				
Total Driver Quantity Basic (TDQBasic)	✓				
Total Driver Quantity Calculated (TDQCalc)	✓				
Total Driver Quantity UE (TDQUE)	✓				
Unassigned Cost	✓				
Unassigned Quantity	✓				
Unique Driver Quantities					✓
Unit Cost	✓				
Unit Profit				✓	
Unit Revenue				✓	
Use Fixed Quantities					✓
Use Variable Quantities					✓
Use Weighted Quantities					✓
Used Cost	✓				
Used Quantity	✓				

Chapter 38

Windows for Column Layouts

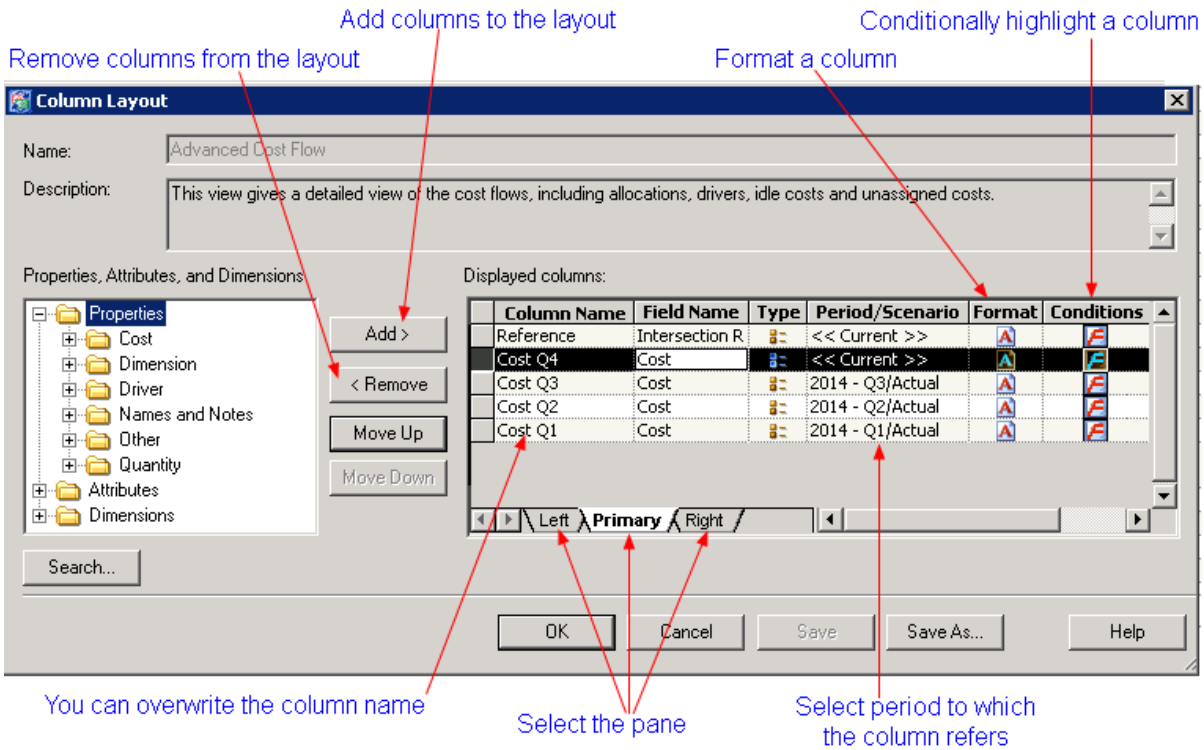
Column Layout Dialog Box

About the Column Layout Dialog Box

Use the Column Layout dialog box to manage the columns in the:

- Module view (see "What Is a Module?" on page 179)
- Dimensions view (see "Dimensions View" on page 205)

Rows in the Column Layout dialog box (from top to bottom) represent columns (from left to right) in a module view or dimensions view.



You can use the Column Layout dialog box to:

- "Create a Column Layout" on page 339
- "Remove a Column" on page 341
- "Order Columns" on page 342
- "Format a Column" on page 342
- "Conditionally Highlight Cells in a Column" on page 346
- "Select the Period to which the Column Refers" on page 350
- "Save a Column Layout" on page 350
- "Apply a Saved Column Layout" on page 350
- "Specify a Default Column Layout" on page 351

Note: The **Display Name** column, which is the left-most column in the column layout, is not displayed in this dialog box. The Display Name column is required, so you cannot remove it, change it, or reorder it.

Icons

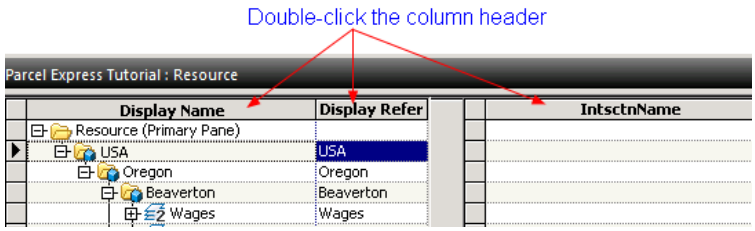
Icons indicate whether each item in the column is a property, an attribute, or a dimension:

This icon	Represents this
	Property

This icon	Represents this
	Attribute
	Dimension

How to Access the Column Layout Dialog Box

- In a Module view or the Dimensions view, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**, or double-click the column header.



See Also

- Chapter 37, “Working with Column Layouts,” on page 319
- Chapter 38, “Windows for Column Layouts,” on page 335
- “Understanding Properties in a Column Layout” on page 324

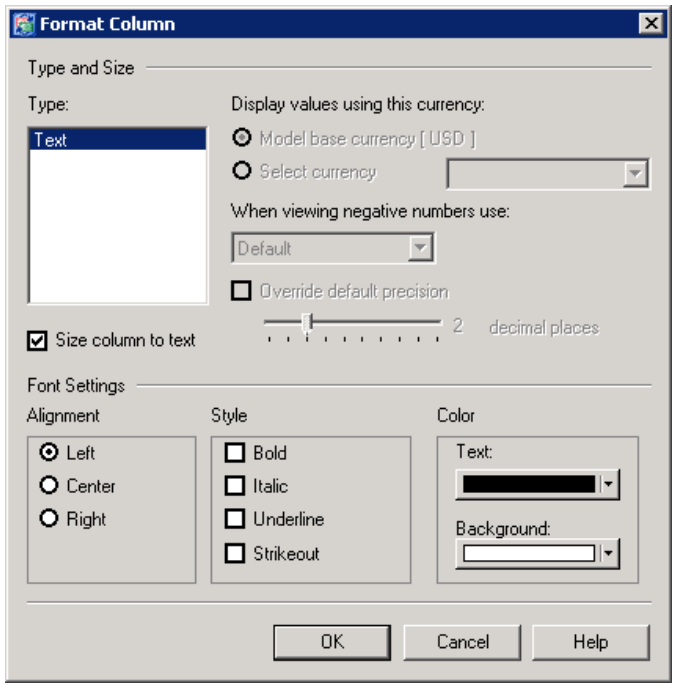
Format Column Dialog Box

In the Format Column dialog box, you can format the information in the columns in a module view, and in the Dimensions view.

How to Access the Format Column Dialog Box

In the Column Layout dialog box, select a row and click the .

The Format Column dialog box appears:



See Also

"Column Layout Dialog Box" on page 335

See Also

- "Format a Column" on page 342
- "Format a Column" on page 342

Save Column Layout As Dialog Box

About the Save Column Layout As Dialog Box

In the Save Column Layout As dialog box, you can name a column layout.

See Also

"Column Layout Dialog Box" on page 335

How to Access the Save Column Layout As Dialog Box

Do one of the following:

- In a module view, select **Model** ⇨ **Column Layout** ⇨ **Save As**.
- In the Column Layout dialog box, click **Save As**.

Chapter 39

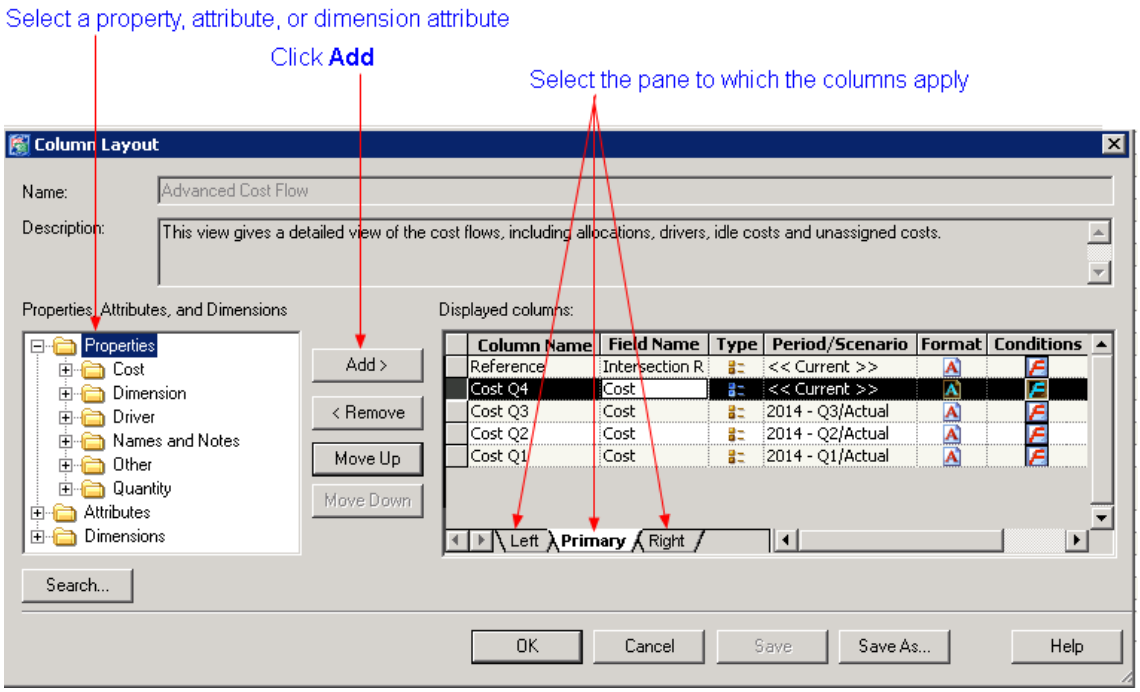
How To: Column Layouts

Create a Column Layout

1. On a module view or Dimensions view, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

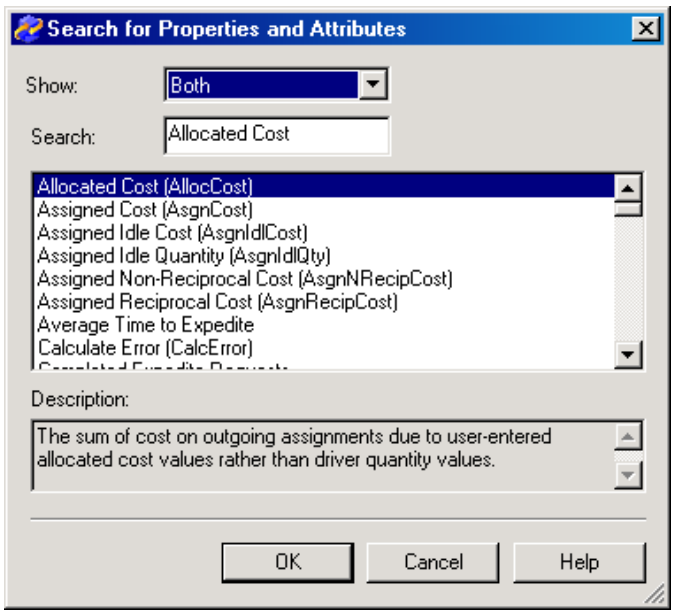
TIP Alternatively, you can double-click a column heading.

The Column Layout dialog box appears.



2. From the list of **Displayed columns**, select the pane from the appropriate tab (**Left**, **Primary**, or **Right**) to which you want to add the column.
3. From the list of **Properties, Attributes, and Dimensions**, select an item.
4. To search for an item, do the following:
   a. Click **Search**.

      The Search for Properties and Attributes dialog box appears.



   b. From the **Show** drop-down list, select the items to display.
   c. To rapidly move through the list of items, type a phrase in the **Search** box.

As you type text, the item that most closely matches the text is selected.

d. Select an item from the list.

The **Description** of the item is displayed.

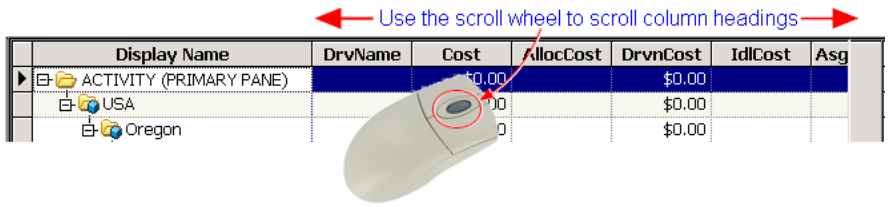
Note: Some items, such as attributes, might not have a description.

e. Click **OK**.

5. Click **Add >**.
6. To rename the column, click the **Column Name** of the new row, and type a new name.

   This text is displayed in the column heading of the column layout.
7. To select a different period/scenario association, click in the **Period/Scenario** column, and select a period/scenario association.
8. Select **Save As** to create the layout with a new name.

Note: If your column layout has too many column headings to be able to view in a window all at once, you can use the scroll wheel of the mouse to scroll the column headings.



See Also

"Understanding Properties in a Column Layout" on page 324

Remove a Column

1. On a module page, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

   The Column Layout dialog box appears.
2. Select the **Module** to which this column layout applies.
3. From the list of **Displayed columns**, select the pane from the appropriate tab (**Left**, **Primary**, or **Right**) from which you want to remove the column.
4. Select a row (which represents a column in the view).
5. Click **< Remove**.

   The row (column) is removed from the list of **Displayed columns**.

Order Columns

1. On a module page, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns** .

   The Column Layout dialog box appears.
2. Select the **Module** to which this column layout applies.
3. From the list of **Displayed columns**, select the pane from the appropriate tab (**Left**, **Primary**, or **Right**) that you want to order.
4. Select a row (which represents a column in the view).
5. To move the row up, click **Move Up**.

   The column will be moved to the left.
6. To move the row down, click **Move Down**.

   The column will be moved to the right.

   TIP You can drag columns in a module view to change the order of columns.

Format a Column

Overview

When you add a column to a module view or to the dimensions view, SAS Cost and Profitability Management assigns default formatting to the columns.

The default formatting is based on the following factors:

- the type of underlying data
- your Windows Regional Options
- the default currency
- the display precision that you specified in your options

You can change some of the formatting for individual columns. You can specify a columns general appearance, such as text color, background color, cell alignment, and font style. Additionally, you can specify the currency and the format type. Changing the currency or the format type affects how the underlying data is displayed on the screen; it does not change the currency or the format type of the underlying data itself.

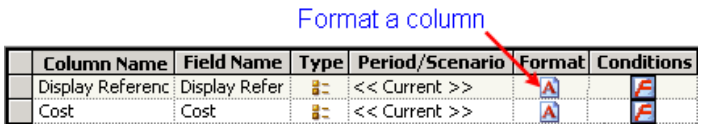
Formatting affects the display of information in the grid; cubes on the OLAP view display the model's base currency in the view title bars. Measures are formatted as numbers.

How to Format

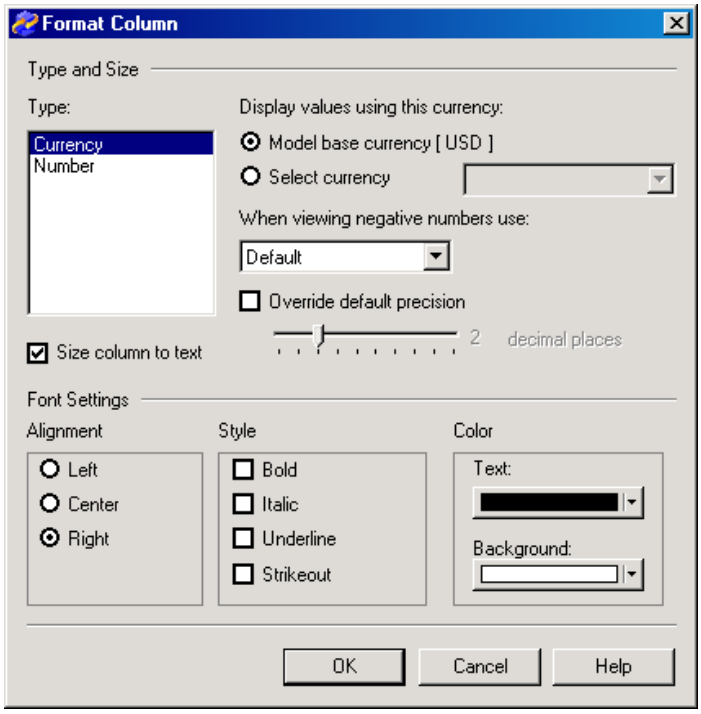
1. In a module view, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

   The Column Layout dialog box appears.

2. In a row that represents the column that you want to format, click .



The Format Column dialog box appears.



3. In the **Type and Size** section, select the **Type** of data.
4. For **Display values using this currency**, select or clear an option.
5. From the **When viewing negative numbers use** drop-down list, select or clear an option.
6. Specify the precision:

   Select or clear the **Override default precision** option.

   Set a value for the number of **decimal places**.
7. To quickly set the width of a column so that the column is wide enough to display all data, select the **Size column to text** option.
8. In the **Font settings** section, select or clear options for **Alignment**, **Style**, and **Color**.

Color, Style, and Alignment

To highlight important information, you can specify the foreground color and the background color for a column. Also, you can specify the font style, such as bold or italic, and the alignment of data within a column, such as right-aligned or left-aligned.

To ensure that columns are not too wide or too narrow for their contents, you can specify that column widths should change according to the width of each column's contents.

Currency Type

The Currency type displays column values as currency. A column can display either the model's base currency or a currency that was specified in an exchange rates table. You can select any available currency, including the currencies that are not specified in an exchange rates table. If you select a currency that does not have an exchange rate, all of the column values are displayed as zeros.

Number Type

The Number type displays column values as numbers.

Percentage Type

The Percentage type displays column values by multiplying each value by 100 and then appending the percentage symbol (%).

Note: The Percentage type pertains only to individual column values. The percentage that each column value contributes to the total of all the column values is not calculated.

Text Type

The Text type displays column values as text.

Checkbox Type

The Checkbox type displays column values as check boxes. Each cell is either checked or not checked.

Available Format Types

The format types that are available for a specific column depend on the data in the column. For example, a column that contains text cannot be formatted to display as percentages or as numbers because the underlying data is not numeric. The following table lists the format types for underlying data:

Underlying data	Available format types
Text	Text
Cost or rate	Currency (default) Number
Number or quantity	Number (default) Currency Percentage

Underlying data	Available format types
Percentage	Percentage (default) Number
Boolean	Checkbox

Negative Numbers

You can specify how negative numbers are displayed. The default appearance for negative numbers is based on your Windows regional settings.

Precision

Precision is the number of decimal places that are displayed for numbers in a column. Your user options affect precision.

TIP Before you add new columns, set the default precision in your user options. Then, set the precision for any column in which you want the precision to be different from the default precision.

Note: For the following properties, if you select the Percentage type, each column value can have only two decimal places:

- Driver Percentage
- Idle Percentage

Currency Formatting

The number of decimal places that you specify in your options overrides the No. of digits after decimal setting in Windows Regional Options.

Change this	in this dialog box.	Example
Positive currency format	Customize Regional Options	1,1 F
Negative currency format	Customize Regional Options	-1,1 F
Decimal symbol	Customize Regional Options	,
No. of digits after decimal	Customize Regional Options	2
Digit grouping symbol	Customize Regional Options	a space

Change this	in this dialog box.	Example
Digit grouping	Customize Regional Options	123 456 789

Displaying the Currency Symbol

You can add columns on the Resource module view, the Activity module view, and the Cost Object module view to display different currencies in the same display.

You can choose to display the currency symbol as part of the number formatting, or in column headings. If you choose to display the currency symbol in column headings, then you can choose to display the currency symbol, the currency code, both, or neither. You might want to display only the currency code if your computer's fonts do not support currency symbols.

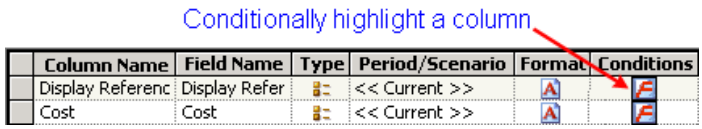
Conditionally Highlight Cells in a Column

Overview

With conditional highlighting of cells, you can change the text and background color of the cells in a column that satisfy conditions that you specify. For example, you can highlight in red all the Cost cells that contain a negative value. Or, you can highlight text cells with a particular content.

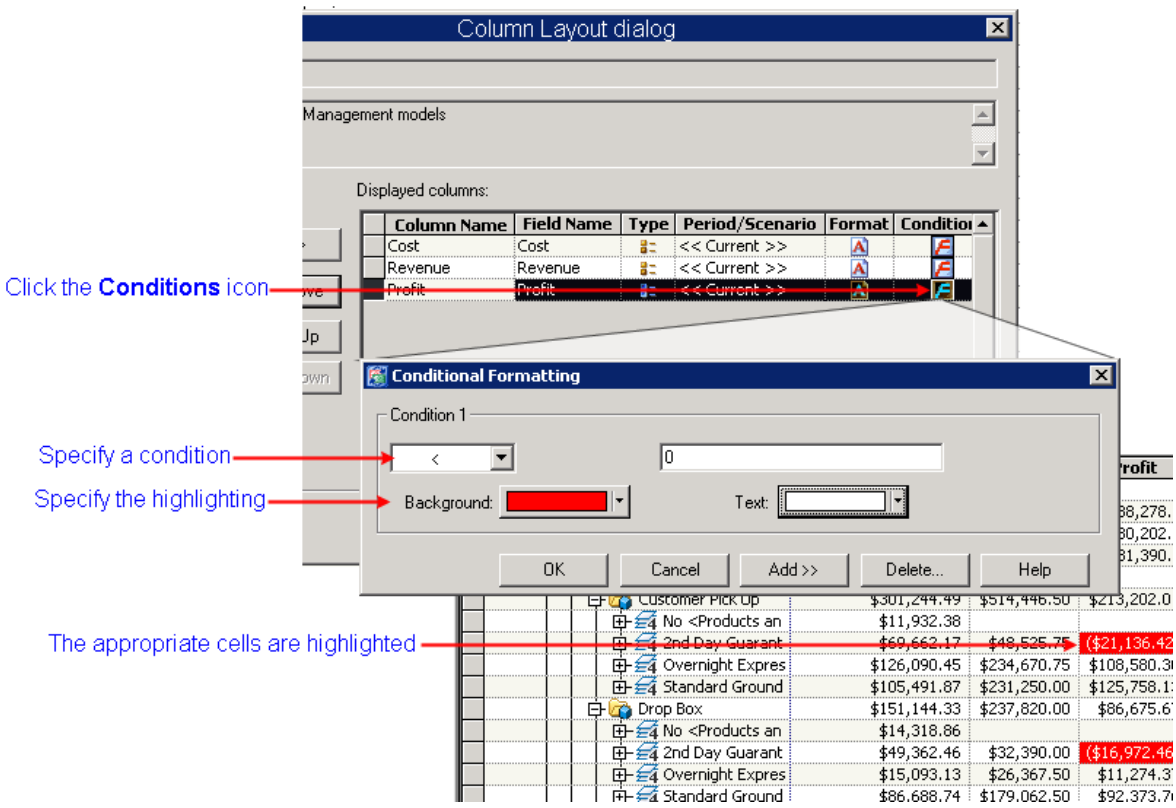
To conditionally highlight cells in a column:

1. Open the Column Layout dialog. See "Column Layout Dialog Box" on page 335.
2. Select a column whose cells you want to highlight.
3. Click the Conditions icon.



The Conditional Formatting dialog opens.

4. Specify a condition that determines which cells are to be highlighted. For allowable conditions, see "Conditions" on page 349.
5. Specify the highlighting that is to be applied to each cell whose data satisfies the condition specified. You can specify:
   - Background color
   - Text color
6. Click **Add** to select additional cells for highlighting. For each set of additional cells selected, you can specify different highlighting. See "Highlight Additional Cells" on page 347.



Highlight Additional Cells

For each column, you can select more than one set of cells for highlighting. For each set of cells selected, you can specify different highlighting.

To select more than one set of cells in a column for highlighting, do the following:

1. From the Column Layout dialog box, click the Conditions icon

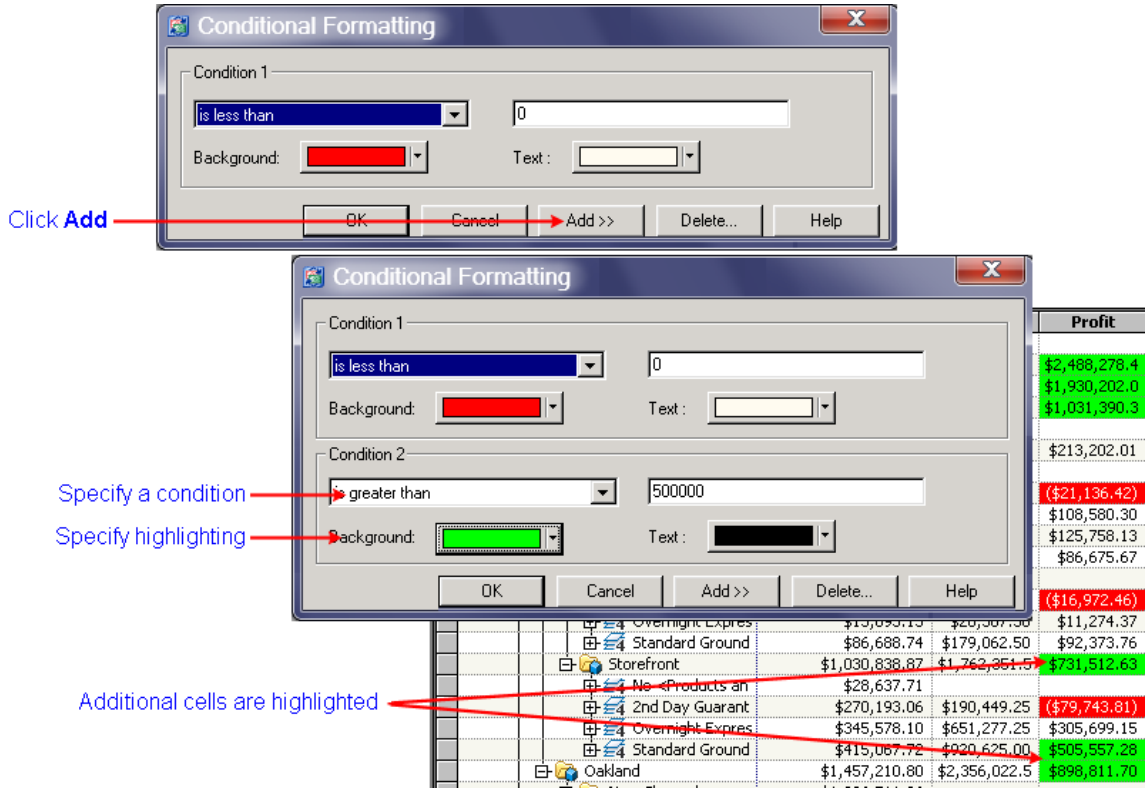
   The Conditional Formatting dialog opens.

2. Specify a condition that determines which cells are to be highlighted.
3. Specify the highlighting that is to be applied to each cell whose data satisfies the condition specified.
4. Click **Add** to select additional cells for highlighting. For each set of additional cells selected, you can specify different highlighting.

   The following picture shows highlighting cells in the Profit column as follows:

Cells	Condition	Highlighting
Set 1	is less than 0	Background: red Text: white

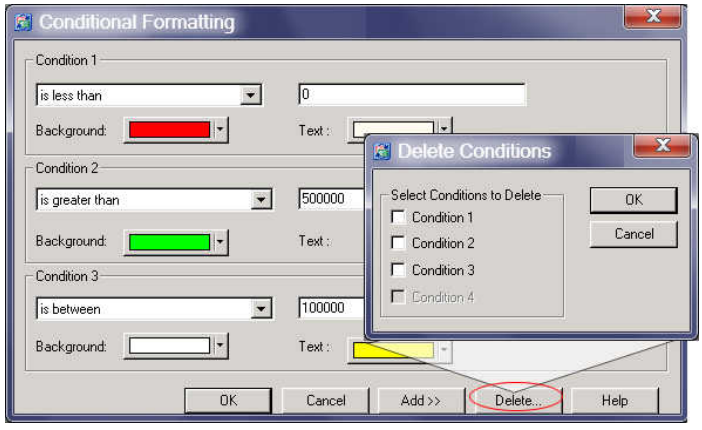
Cells	Condition	Highlighting
Set 2	is greater than 500,000	Background: green Text: black



Remove Highlighting

To remove the highlighting from cells:

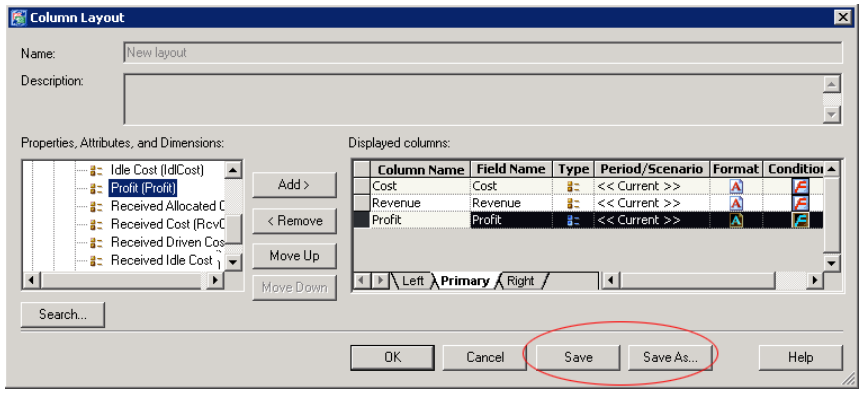
1. On the Conditional Formatting window, click **Delete**.
2. Select the condition to be deleted.
3. Click **OK**.



Save the Highlighting

To save the conditional highlighting specifications, select **Save** or **Save As** from the Column Layout dialog.

The conditional highlighting specifications are saved along with the column layout. They persist with the column layout if the column layout is exported and subsequently imported.



Conditions

The conditions that you can specify to select cells for highlighting depend on the type of column selected. The conditions are as follow:

Numeric Columns

You can select cells in a numeric data column by using the following conditions:

- is equal to
- is not equal to
- is between
- is greater than
- is greater than or equal to
- is less than
- is less than or equal to

Text Columns

You can select cells in a text data column by using the following conditions:

- is equal to
- is not equal to

Boolean Property Columns

You can select cells in a Boolean property column (such as HasAttribute) by using the following conditions:

- is checked
- is unchecked

Tag Attribute Columns

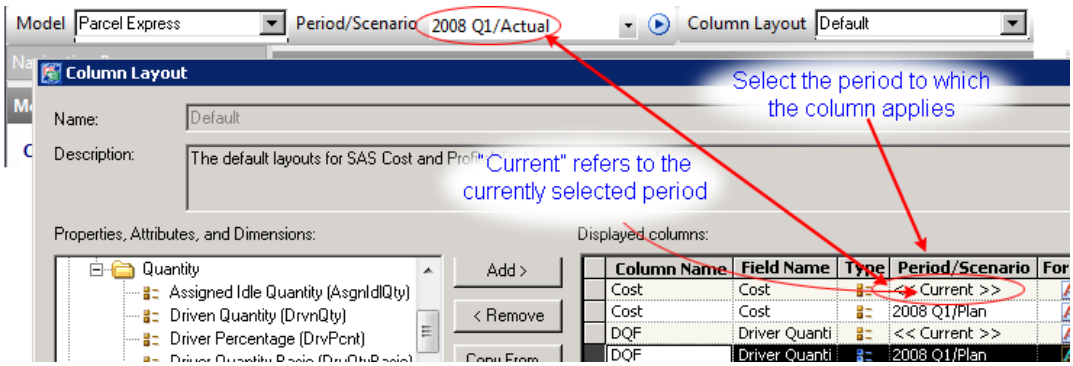
You can select cells in a tag attribute column by using the following condition:

- is checked

Select the Period to which the Column Refers

From the **Period/Scenario** drop-down menu, select the period/scenario association to which the data in the column refers.

Note: **<<Current>>** refers to the period/scenario association currently displayed in the module view or dimensions view to which the column layout is being applied.



Save a Column Layout

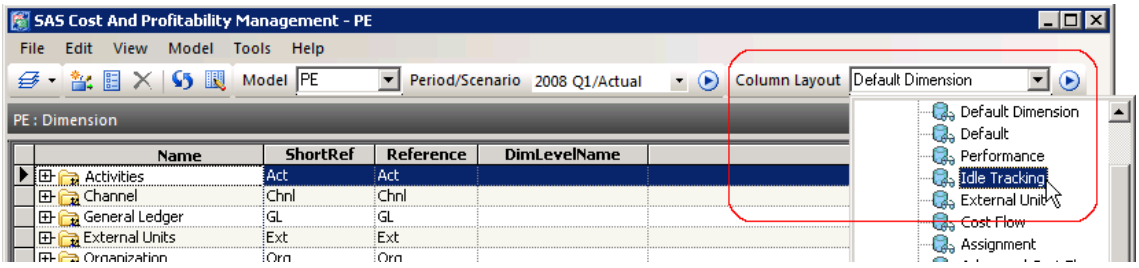
Select **Model** ⇨ **Column Layout** ⇨ **Save**.

The Save Column Layout As dialog box appears.

Apply a Saved Column Layout

Note: You can apply a saved column layout to a module view or a dimensions view.

1. From the **Column Layout** drop-down list, select a column layout.



2. Click  .

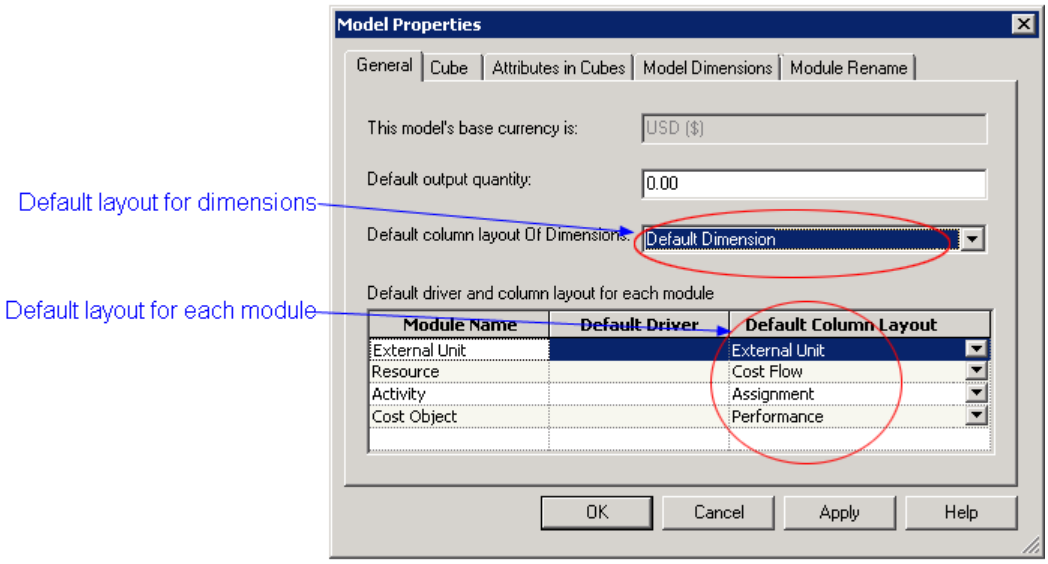
Specify a Default Column Layout

To specify the column layout that is displayed by default when you:

- view a module
- are in the Dimensions view

1. Select **Model** ⇨ **Properties**
2. Click the **General** tab.
3. **for modules**
   Select a column layout for each module in the **Default driver a column layout for each module** field.

   for dimensions
   Specify a column layout in the **Default column layout for dimensions** field.



Once you are in the Dimensions view or are viewing a module, you can activate a different column layout. However, when you open the model again, the default column layout is displayed when you go to the Dimensions view or view a module.

Import or Export a Column Layout

- "Import a Column Layout" on page 679
- "Export a Column Layout" on page 679

Part 13

Drivers

Chapter 40

Introducing Drivers

What Is a Driver?

Overview

A driver distributes cost in an account to one or more destination accounts.

There are four types of drivers, each of which is described in detail in the following chapters.:

- Chapter 41, “Evenly Assigned Driver,” on page 359
- Chapter 42, “Percentage Driver,” on page 363
- Chapter 43, “Standard Driver,” on page 367
- Chapter 44, “Calculated Driver,” on page 393

Drivers are structural elements. They exist in every period/scenario in a model. And, if a driver has a formula or a rule formula, then those formulas are the same in every period/scenario. However, the evaluation of a formula can result in different values in different period/scenarios. And, a particular driver that is attached to an account in one period/scenario may not be attached to the same account in a different period/scenario.

For all drivers, the basic mechanism for distributing costs is by calculating the following three properties:

Driver Quantity Calculated (DrvQtyCalc)
the number of units going to each destination account

Driver Rate (DrvRate)
the cost per unit for every destination account

Driver Driven Cost (DrvDrvnCost)
the cost going to each destination account. It is calculated by multiplying DrvQtyCalc (the number of units) times DrvRate (the cost per unit)

DrvQtyCalc

The following table shows for each type of driver what properties it uses for calculating DrvQtyCalc—the number of units going to each destination account—and what formula it uses for the calculation:

Type of Driver	Properties Used	Formula
Evenly assigned	DQF	DrvQtyCalc = DQF = 1 DQF is automatically assigned a value of 1 for every assignment path of the driver. DQF is not changeable by a user for this driver.
Percentage	DQF	DrvQtyCalc = DQF
Standard	DQF, DWF DQV, DWV TDQ	DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
Calculated	See Chapter 55, "Properties That Can Be In Formulas," on page 495.	DrvQtyCalc = the value of the driver formula

DrvRate

For every driver type:

```
DrvRate = DrvblCost / TDQ
```

where:

```
DrvblCost = Cost - AllocatedCost
```

- DrvblCost (Drivable Cost) is the cost that is distributed by the driver.
- Cost is the cost in the account.
- AllocatedCost is cost that is distributed to destination accounts in lump sums rather than by the driver mechanism. (See "User-Entered Cost Allocation" on page 400.)

and where:

```
TDQ = TDQCalc + SoldQuantity
```

- TDQ (Total Driver Quantity) is the total quantity (of whatever measure) that flows out of an account via its driver to any number of destination accounts..
- TDQCalc is the sum of DrvQtyCalc for all the destinations of a given account. DrvQtyCalc is the quantity on an assignment to an individual destination..
- SoldQuantity is a user-entered value for an account..

DrvDrvnCost

For every driver type:

```
DrvDrvnCost = DrvQtyCalc x DrvRate
```

DrvDrvnCost (the cost driven by an assignment) is calculated by multiplying DrvQtyCalc (the number of units on the assignment) times DrvRate (the cost per unit).

Guidelines for Creating Drivers

Before you create drivers, consider the following points:

- A formula for a driver can contain at most 32K characters.
- Minimize the use of unique driver quantities. Unique driver quantities can use a lot of memory. They increase processing time, and they do not provide a calculation advantage. See "Unique and Non-Unique (Shared) Driver Quantities" on page 405.
- Create drivers whose quantities or volumes are already being collected or cost little to measure.
- When you collect driver data, ensure that the data is current, available, and accurate. Ask individuals in your organization to verify the data's accuracy.

Chapter 41

Evenly Assigned Driver

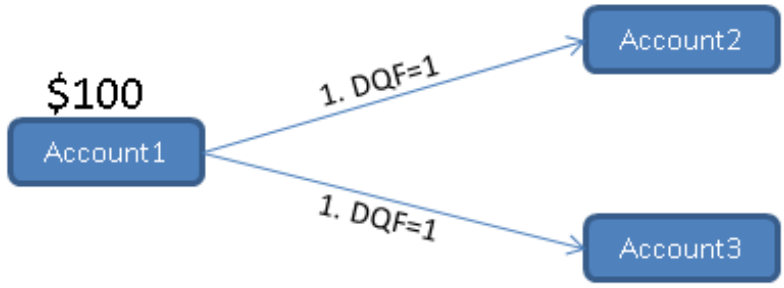
Overview

An evenly assigned driver distributes equal portions of cost to every destination account.

The evenly assigned driver is often used as a first approximation of the costs that flow between accounts when actual driver quantities are not known. When actual driver quantities are known, you can replace this driver with a driver that is more accurate.

System Assigns DQF

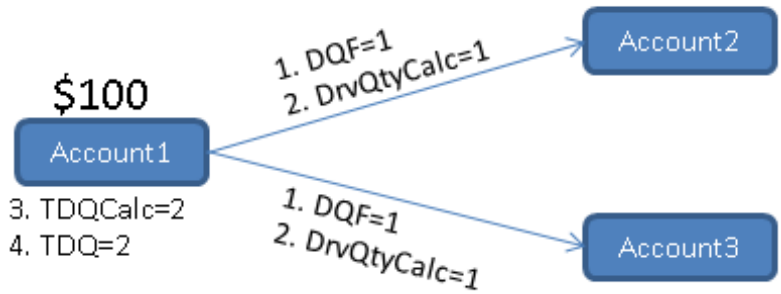
The following picture shows Account1 distributing $100 in equal portions to Account2 and Account3.



Step 1: The system automatically sets DQF to 1 for each assignment to a destination account. DQF represents the number of units going to a destination account.

Note: The driver quantities DQF, DQV, DWF, and DWV are preset for the evenly assigned driver. You cannot change them.

Calculate DrvQtyCalc, TDQCalc, and TDQ



Step 2: DrvQtyCalc represents the total units going to a destination account. In the case of the evenly assigned driver, DrvQtyCalc=DQF. So, in this case DrvQtyCalc=1.

Note: Actually, the evenly assigned driver uses the same formula for calculating DrvQtyCalc as the standard driver, which formula is the following (see "What Is a Standard Driver?" on page 367):

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

In the case of the evenly assigned driver, DQF=1, DWF=1, DQV=0, and DWV=1. So, by substitution:

```
DrvQtyCalc = (1 x 1) + (0 x 1 x Dest.TDQ)
```

Then:

```
DrvQtyCalc = 1
```

Step 3: TDQCalc is the total units going out to all destination accounts. In other words, it is the sum of DrvQtyCalc for all destination accounts. In this example, DrvQtyCalc is 1 for each of two destination accounts, so TDQCalc= 2.

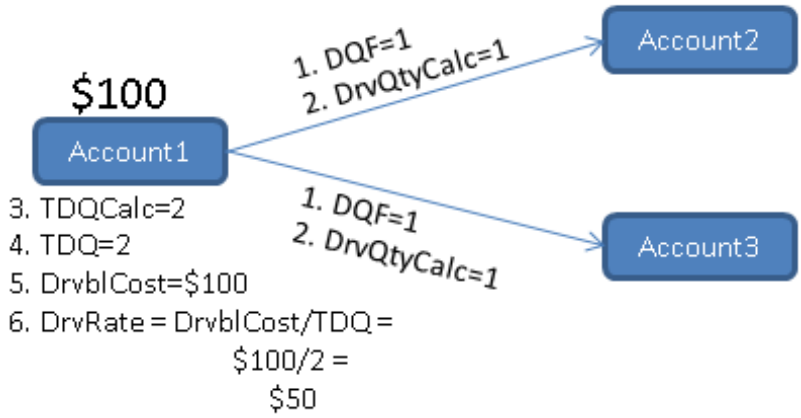
Step 4: If we ignore complications which are discussed in other chapters, Total Driver Quantity (TDQ) equals TDQCalc. So, TDQ=2.

Note: TDQ can exceed TDQCalc if either is the case:

- SoldQty is specified for the source account, in which case **`TDQ=SoldQty +TDQCalc`**.
- TDQUE (Total Driver Quantity User-Entered) is specified as greater than TDQ, in which case **`TDQ=TDQUE`** (TDQUE overrides TDQ).

  Specifying TDQUE as greater than TDQ indicates that there is idle capacity—more units on the source account than are actually distributed to target accounts (the total units distributed is represented by TDQCalc).

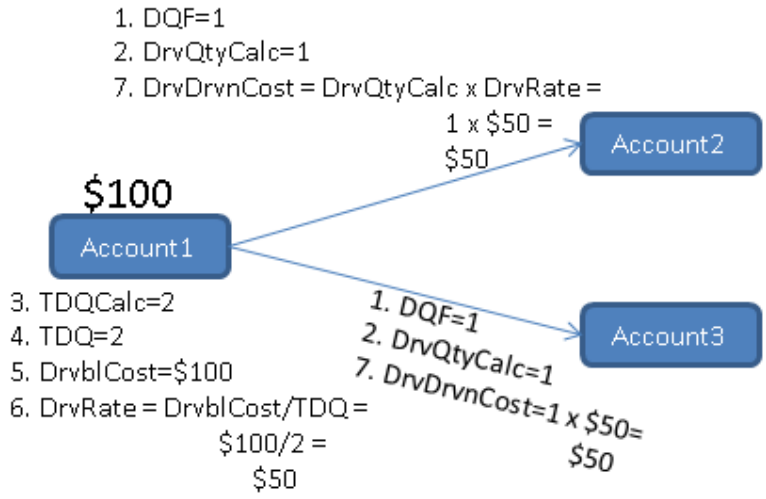
Calculate DrvblCost and DrvRate



Step 5. Drivable Cost (DrvblCost) equals Cost minus Allocated Cost. (See "User-Entered Cost Allocation" on page 400.) In the case of this simple example, there is no allocated cost, so Drivable Cost equals the cost in the account. In this example, DrvblCost=$100.

Step 6. Driver Rate is the cost per unit for any destination account. It is the Drivable Cost divided by the total units going to all destination accounts (DrvblCost/TDQ). In this example, DrvRate = $100/2 = $50.

Calculate DrvDrvnCost



Step 7. Driver Driven Cost is the cost that is distributed on an assignment to a particular destination account. It is calculated by multiplying DrvQtyCalc (the number of units going to a destination account) times DrvRate (the cost per unit for any destination account). In this example, DrvDrvnCost = 1 x $50 = $50 for each of the assignments.

Chapter 42

Percentage Driver

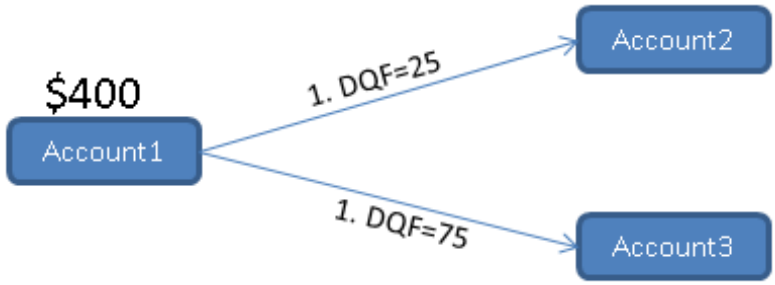
Percentage Driver

Overview

A percentage driver distributes cost to destination accounts in fixed percentages. The percentage driver is often used as a first approximation of the costs that flow between accounts when actual driver quantities are not known.

User Specifies DQF

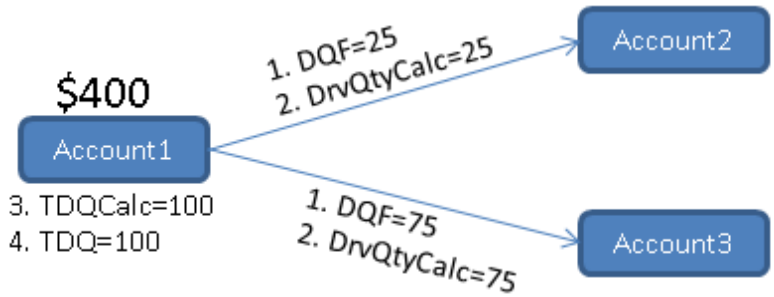
The following picture shows Account1 distributing 25% of $400 to Account2 and 75% of the same $400 to Account3.



Step 1: You set DQF to the percentage of cost that is to be distributed to each destination account. The system issues an error if the total percentage exceeds 100 percent and a warning if it is less than 100 percent.

Note: The driver quantities DWF, DQV, and DWV are set by the system for the percentage driver. You cannot change them.

Calculate DrvQtyCalc, TDQCalc, and TDQ



Step 2: DrvQtyCalc represents the percentage of cost going to a destination account. In the case of the percentage driver, DrvQtyCalc=DQF. So, for one assignment DQF=25, and for the other assignment DQF=75.

Note: Actually, the percentage driver uses the same formula for calculating DrvQtyCalc as the standard driver, which formula is the following (see "What Is a Standard Driver?" on page 367):

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

In the case of the percentage driver, DWF=1, DQV=0, and DWV=1. So, by substitution:

```
DrvQtyCalc = (DQF x 1) + (0 x 1 x Dest.TDQ)
```

Then:

```
DrvQtyCalc = DQF
```

Step 3: TDQCalc is the total percentage of cost going to all destination accounts. In other words, it is the sum of DrvQtyCalc for all destination accounts. In this simple example, TDQCalc is 100. An error is issued if TDQCalc>100, and a warning is issued if TDQCalc<100.

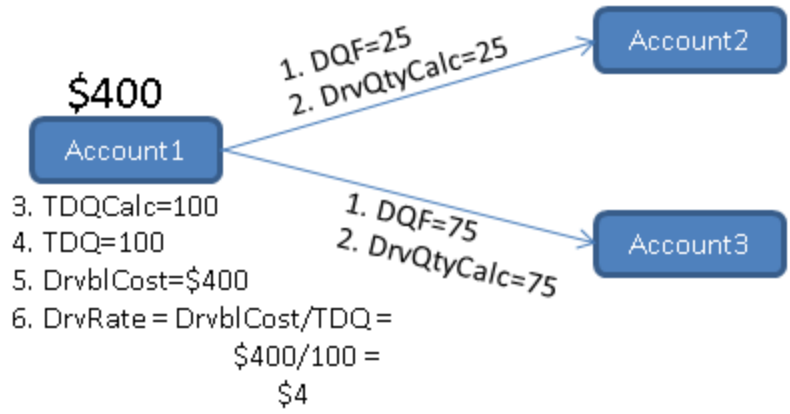
Step 4: If we ignore complications which are discussed in other chapters, Total Driver Quantity (TDQ) equals TDQCalc.

Note: TDQ can exceed TDQCalc if either is the case:

- SoldQty is specified for the source account, in which case **`TDQ=SoldQty +TDQCalc`**.
- TDQUE (Total Driver Quantity User-Entered) is specified as greater than TDQ, in which case **`TDQ=TDQUE`** (TDQUE overrides TDQ).

  Specifying TDQUE as greater than TDQ indicates that there is idle capacity—more units on the source account than are actually distributed to target accounts (the total units distributed is represented by TDQCalc).

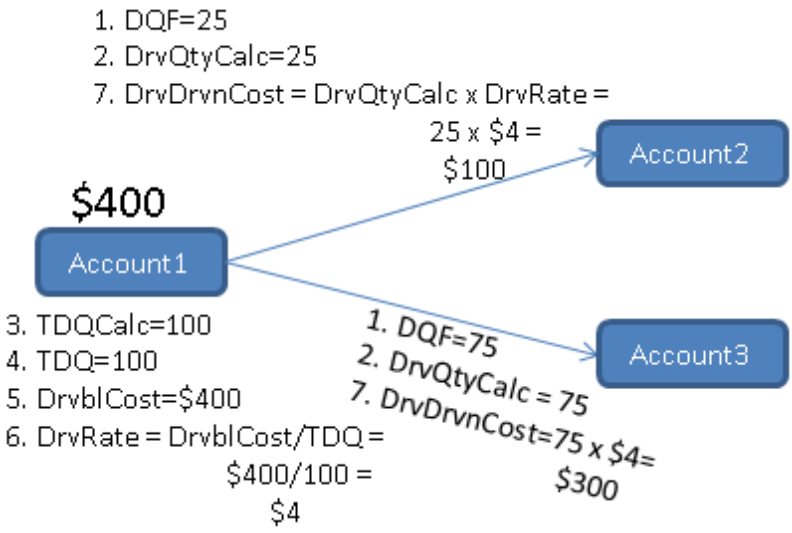
Calculate DrvblCost and DrvRate



Step 5. Drivable Cost (DrvblCost) equals Cost minus Allocated Cost. (See "User-Entered Cost Allocation" on page 400) In the case of this simple example, there is no allocated cost, so Drivable Cost equals the cost in the account: DrvblCost=$400.

Step 6. Driver Rate is the cost per unit for any destination account. In the case of the percentage driver, Driver Rate is the cost per percentage point. It is Drivable Cost divided by the total percentage going to all destination accounts (DrvblCost/TDQ). In this example, DrvRate = $400/100 = $4. That is, $4 is distributed for each percentage point that is to be distributed to a destination account.

Calculate DrvDrvnCost



Step 7. Driver Driven Cost is the cost that is distributed on an assignment to a particular destination account. It is calculated by multiplying DrvQtyCalc (the total percentage going to a particular destination account) times DrvRate (the cost for each percentage point). For one assignment, DrvDrvnCost = 25 x $4 = $100. For the other assignment, DrvDrvnCost = 75 x $4 = $300.

You can see that the math involved for a percentage driver is essentially the same for the evenly assigned driver.

Example of a Percentage Driver

The following example illustrates how a percentage driver can be used in a model.

Suppose that an organization decides that it's not practical to determine the actual number of hours that clerical personnel spend on various tasks per year. The cost to determine the actual number of hours outweighs the benefit. To approximate the annual number of hours, management asks the clerical personnel to record their actual hours for one week. From these recordings, a percentage is assigned to each task. For this organization, a percentage is reasonably accurate and acceptable.

Chapter 43

Standard Driver

What Is a Standard Driver?

Introduction

The standard driver is the most flexible type of driver. Unlike the evenly assigned and percentage drivers, which use only DQF to distribute cost, the standard driver can use any or all of the following properties:

Driver Quantity Fixed (DQF)
See "Fixed Driver Quantities" on page 400. Also, see"Variable Driver Quantities and Fixed Driver Quantities" on page 403.

Driver Weight Fixed (DWF)
See "Driver Weights" on page 404.

Driver Quantity Variable (DQV)
Demand flow is the calculation of the quantities involved in assignments. Demand flow is always calculated before cost flow. That is, before costs can be calculated, it is necessary to calculate the quantities of things involved. Whereas cost flows from left to right (for example, from the Resource module to the Activities module to the Cost Object module) demand flows from right to left (that is, before calculating how much cost flows to a destination account, it is necessary to know how much output is required at the destination). This should become clear in the course of the examples.

See "Variable Driver Quantities" on page 401. Also, see"Variable Driver Quantities and Fixed Driver Quantities" on page 403.

Driver Weight Variable (DWV)
See "Driver Weights" on page 404.

Driver Formula (Formula)
See "Calculated Driver" on page 393.

Like the evenly assigned and percentage drivers, the standard driver calculates DrvDrvnCost (the cost going to each destination account) by determining the value of two other properties:

Driver Quantity Calculated (DrvQtyCalc)
the number of units going to each destination account

A standard driver uses the following fornula to calculate DrvQtyCalc:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

where:

DQF	Driver Quantity Fixed
DWF	Driver Weight Fixed
DQV	Driver Quantity Variable
DWV	Driver Weight Variable
Dest.TDQ	Total Driver Quantity on the Destination account

Driver Rate (DrvRate)
the cost per unit for any destination account

A standard driver uses the same formula to calculate Driver Rate as the other two drivers:

```
DrvRate = DrvblCost/TDQ
```

For the sake of simplicity, the following example of a standard driver uses only DQF to determine the distribution of costs.

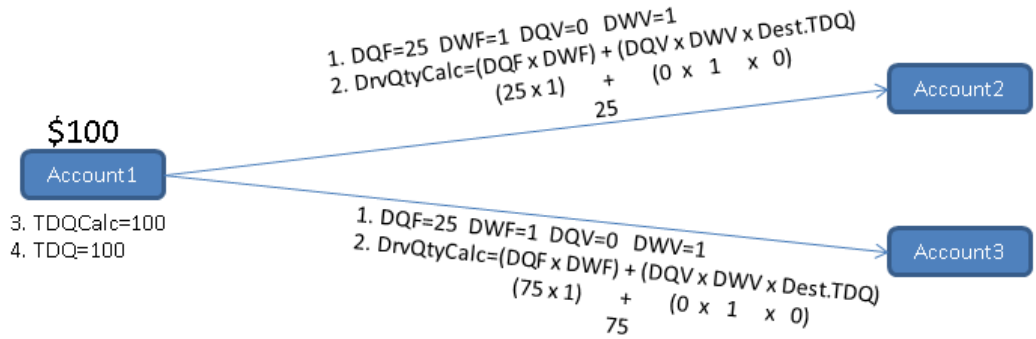
User Specifies Quantities

Step 1: You set DQF to determine how much cost is to be distributed for each assignment to a destination account. DQF can represent any unit of measurement—number of hours, amount of electricity, volume of weight. By adding the total of whatever unit is distributed to all destination accounts, the system determines what portion is distributed to each individual account and distributes cost accordingly.

For this simple example, we assume that you as a user do not set DQV, so the system assigns DQV a value of zero.

And, for this example we assume that you do not attach any weights to the particular assigenments. So, the system assigns 1 to each of DWF and DWV so that those properties have no effect in the formula for the distribution of cost.

Calculate DrvQtyCalc, TDQCalc, and TDQ



Step 2: DrvQtyCalc represents the total units going to a destination account. In the case of a standard driver, the formula for calculating DrvQtyCalc is

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

Because this is a simple example, Dest.TDQ=0. Remember that TDQ is total units going to destination accounts. And, in this example, neither Account2 nor Account3 has any outgoing assignments. So, TDQ is zero for both Account2 and Account3. And, since Account2 and Account 3 are the destination accounts of Account1, Dest.TDQ for Account1 is zero.

By substitution for the one assignment we have:

```
DrvQtyCalc = (25 x 1) + (0 x 1 x 0)= 25
```

and for the other assignment:

```
DrvQtyCalc = (75 x 1) + (0 x 1 x 0)= 75
```

Step 3: TDQCalc is the total units going to all destination accounts. In other words, it is the sum of DrvQtyCalc for all destination accounts. In this example, TDQCalc = 25 + 75 = 100.

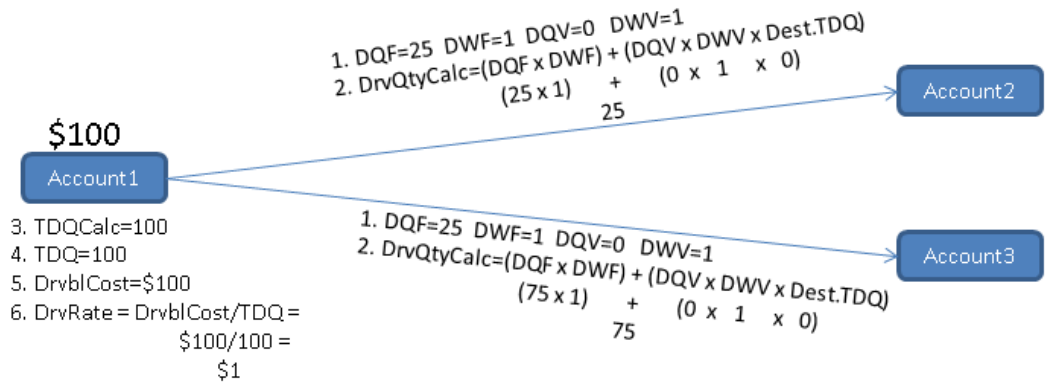
Step 4: If we ignore complications which are discussed in other chapters, Total Driver Quantity (TDQ) equals TDQCalc.

Note: TDQ can exceed TDQCalc if either is the case:

- SoldQty is specified for the source account, in which case **`TDQ=SoldQty +TDQCalc`**.
- TDQUE (Total Driver Quantity User-Entered) is specified as greater than TDQ, in which case **`TDQ=TDQUE`** (TDQUE overrides TDQ).

  Specifying TDQUE as greater than TDQ indicates that there is idle capacity—more units on the source account than are actually distributed to target accounts (the total units distributed is represented by TDQCalc).

Calculate DrvblCost and DrvRate

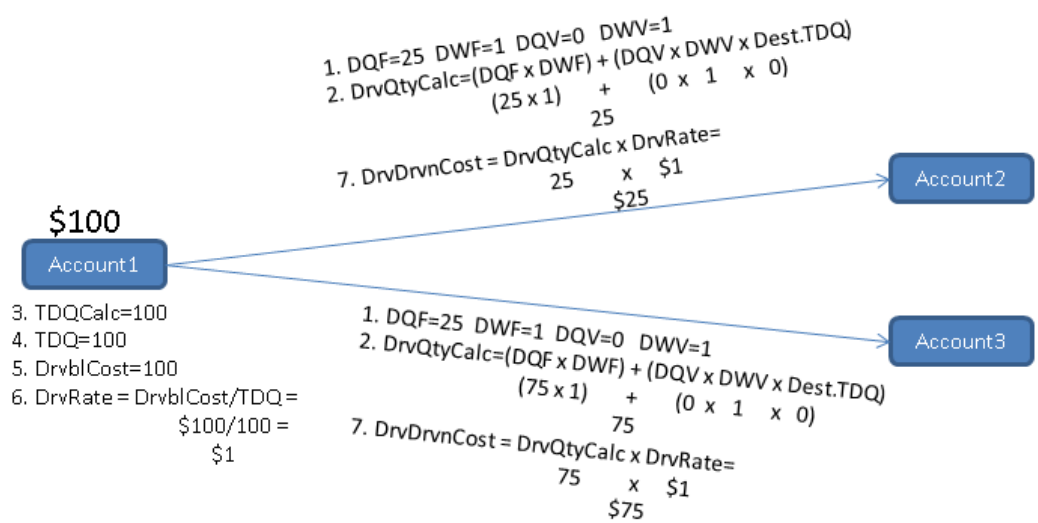


Step 5. Drivable Cost (DrvblCost) equals Cost minus Allocated Cost. (See "User-Entered Cost Allocation" on page 400) In this example, there is no allocated cost, so Drivable Cost equals the cost in the account. DrvblCost=$100.

Step 6. Driver Rate is the cost per unit for any destination account. It is Drivable Cost divided by the total number of units going to all destination accounts (DrvblCost/TDQ). In this example, $1 of cost is distributed for each unit that is distributed to a destination account.

```
DrvRate = $100/100 = $1
```

Calculate DrvDrvnCost



Step 7. Driver Driven Cost is the cost that is distributed on an assignment path to a particular destination account. It is calculated by multiplying DrvQtyCalc (the number of units going to a particular destination account) times DrvRate (the cost per unit for any destination account).

So, for one assignment:

```
DrvDrvnCost = 25 x $1 = $25
```

and for the other assignment:

```
DrvDrvnCost = 75 x $1 = $75
```

Examples of a Standard Driver

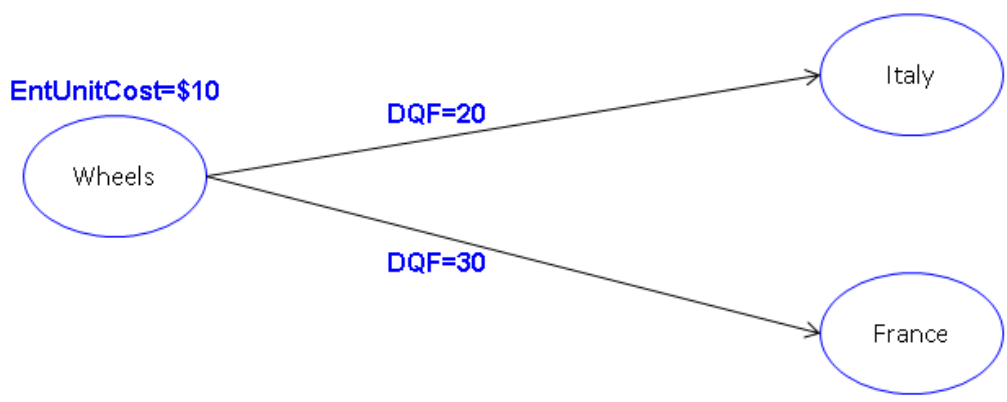
Example 1: Entered Unit Cost with Fixed Driver Quantities

Overview

For fixed quantities, the cost that flows out of an account does not depend on downstream demand—how much output is demanded from the account.. It depends only on how much is specified to flow out of the account, which is specified as a fixed quantity.

A. User-Entered Values

The following picture shows a very simple example of a standard driver distributing wheels to two destinations. One assignment distributes 20 wheels (DQF=20) to Italy. A second assignment distributes 30 wheels (DQF=30) to France. The Entered Unit Cost of each wheel is $10.



B. Calculate DrvQtyCalc, TDQCalc, and TDQ

The next diagram shows the following calculations:

1. DrvQtyCalc represents how much is driven by an individual assignment. The general formula for calculating DrvQtyCalc is the following:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

In the case of this simple example, DQV=0 so the second part of the formula drops out. And, for this example, no weight is specified so DWF=1. So, in the case of this simple example:

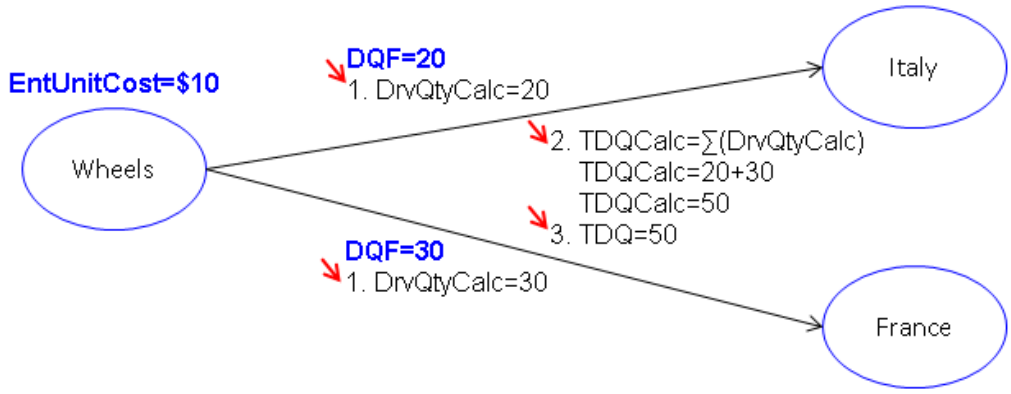
```
DrvQtyCalc = DQF
```

2. TDQCalc is the sum of DrvQtyCalc for all the assignments of a particular driver. So, if one assignment distributes 20 wheels to Italy and a second assignment distributes 30 wheels to France, then TDQCalc=50.

3. In general, TDQ is calculated according to the following formula:

```
TDQ = TDQCalc + SoldQty
```

In the case of this simple example there is no SoldQty, so TDQ=TDQCalc=50

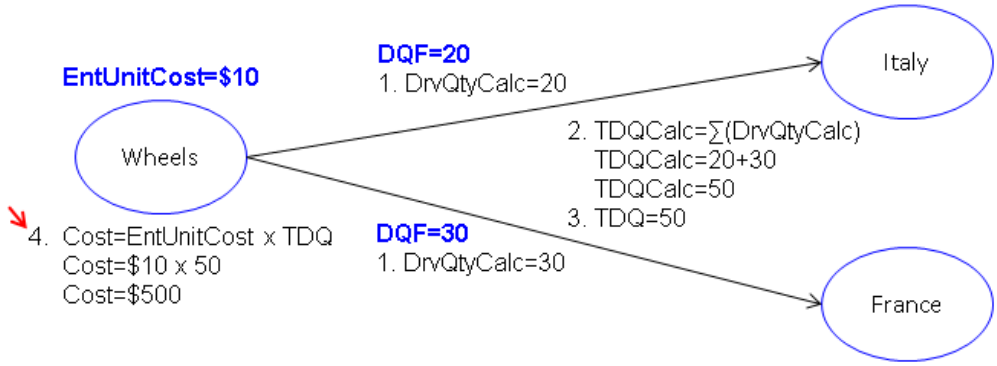


C. Caculate Cost

In general, the cost in an account is calculated according to the following formula:

```
Cost = EnteredCost + EnteredUnitCost x TDQ
```

In the case of this example the Wheels account has no EnteredCost, so its cost is EnteredUnitCost times TDQ (or $10 times 50). As you would expect, the total cost of wheels is determined by multiplying the cost of each wheel times the number of wheels.



D. Calculate DrvblCost and DrvRate

Now let's see how the cost is distributed to each of the destination accounts.

5. In general, DrvblCost is calculated according to the following formula:

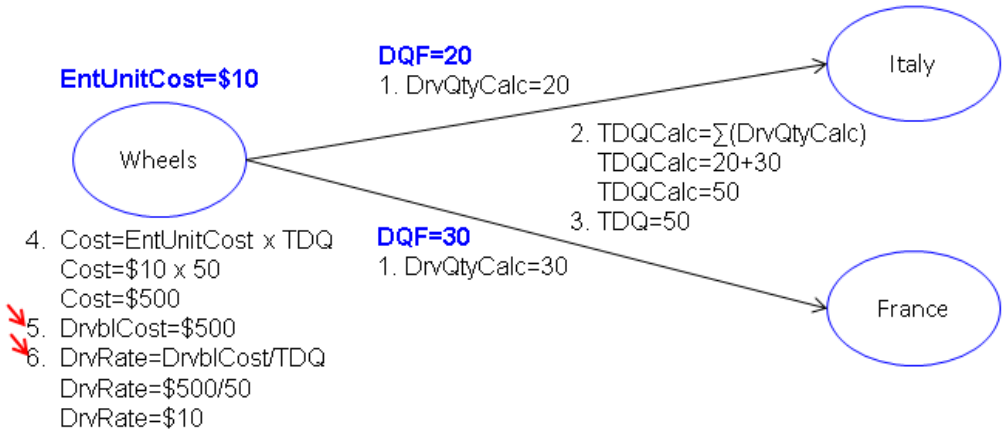
```
DrvblCost = Cost - AllocatedCost
```

AllocatedCost is a cost that is distributed to destination accounts in lump sums rather than by the mechanism of a driver. In the case of this example there is no AllocatedCost, so DrvblCost=Cost=$500.

6. Driver Rate is the dollar amount going to each destination account per unit. Driver rate is calculated according to the following formula:

```
DrvRate = DrvblCost/TDQ
```

Notice that in this case DrvRate=EnteredUnitCost=$10. In general, DrvRate=EnteredUnitCost unless the Cost to be distributed (DrvblCost) is reduced by sending some of the cost to destination accounts in the form of AllocatedCost.



E. Calculate DrvDrvnCost and Cost

7. DrvDrvnCost represents the cost driven on a particular assignment path. It is calculated according to the following formula:

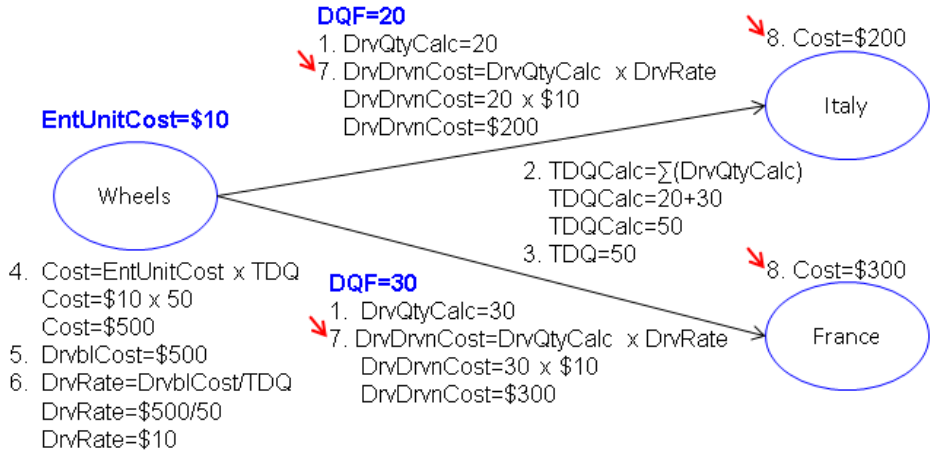
```
DrvDrvnCost = DrvQtyCalc x DrvRate
```

In other words, the cost driven on a particular assignment is the number of units on the assignment multiplied by the cost per unit.

8. The Cost on a destination account is calculated according to the following formula:

```
Cost = ReceivedCost + EnteredCost
```

For this example, the destination accounts have no EnteredCost in the form of Entered Cost Elements belonging to the destination accounts. So, the Cost on the destination accounts is the Cost that they receive from the source accounts.



Example 2: Entered Unit Cost with Variable Driver Quantities and SoldQty

Demand Flow

For variable quantities, the cost that flows out of an account depends on the downstream demand—how much output is demanded from the account to flow downstream. In concrete terms, how much is demanded from an account to flow downstream is represented by the destination account's TDQ (total driver quantity). You can see this from the formula for the account's DrvQtyCalc (the number of units flowing from the account to a destination account):

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

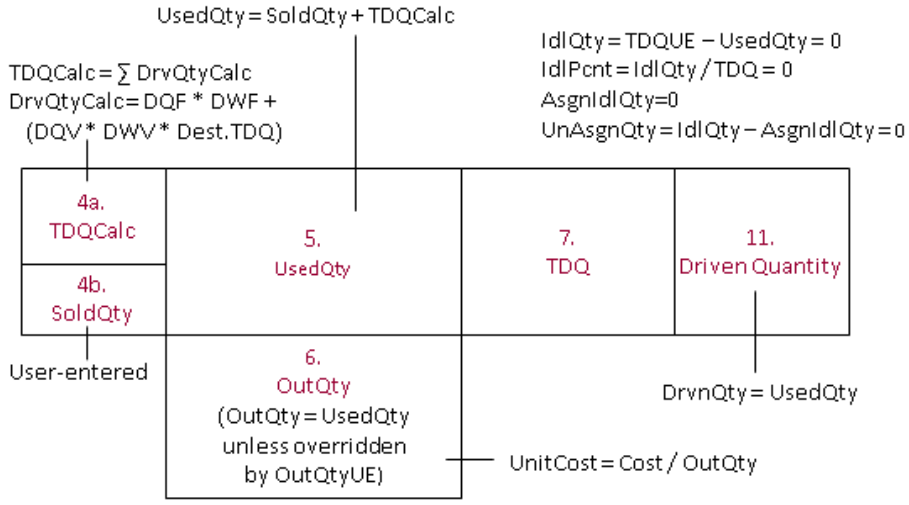
In this formula, you see that the variable quantity (DQV) is multiplied by Dest.TDQ (and by a driver weight if it exists). If you look at the diagram for account driver properties, you can see that (assuming TDQ is not overridden by TDQUE) the following formula holds:

```
TDQ=UsedQty=TDQCalc + SoldQty
```

That is, TDQ depends on two quantities:

- TDQCalc
- SoldQty

Figure 43.1 *Account Driver Properties (assuming that TDQUE=0)*



For this first example, the user enters the value of EntUnitCost, DQV, and SoldQty.

The problem: calculate how much is spent for cars and trucks respectively, given the following information:

50 cars were sold, and 5 trucks were sold	SoldQty(cars)=50 SoldQty(trucks)=5

Each car has 4 wheels, and each truck has 8 wheels	DQV(cars)=4 DQV(trucks)=8
Each wheel (regardless of car or truck) has 5 lug nuts	DQV=5
Each lug nut costs $0.05 per nut	EnteredUnitCost=$0.05

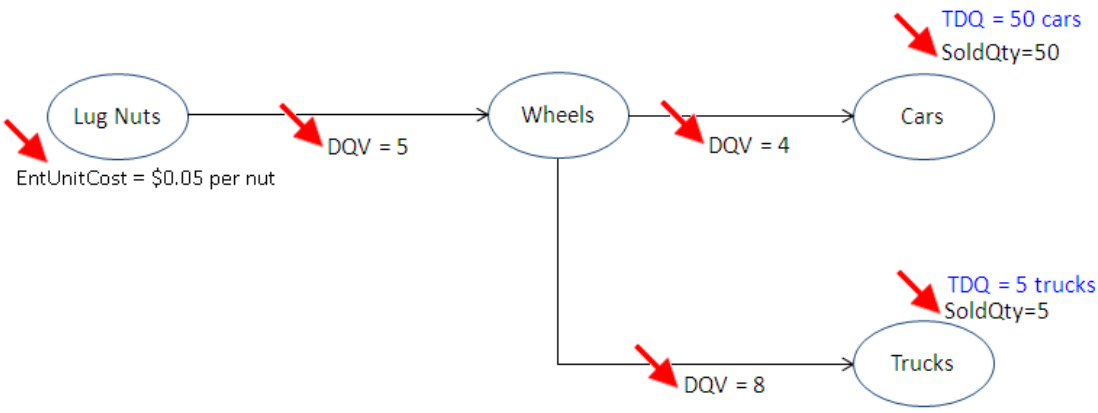
For an example with a variable quantity and Dest.TDQ instead of SoldQty, see "Examples 3: Entered Unit Cost with Variable Driver Quantity and Dest.TDQ" on page 382.

See also:

- "Diagram 3: Account Driver Properties " on page 529
- "Variable Driver Quantities" on page 401

A. Specify the User-Entered Values

The following picture shows the information that is entered by a user.



You can also see that **`TDQ(cars)=50`** and TDQ(trucks)=5. In this example, **`TDQ=SoldQty`** for cars and trucks respectively. This is explained as follows. (Refer to Figure 43.1 on page 374) Because there is no further outgoing flow of costs from either cars or trucks, it follows that their **`TDQCalc=0`**. So, given that **`UsedQty=TDQCalc +SoldQty`**, it follows that **`UsedQty=SoldQty`**. Furthermore, because **`TDQ=IdlQty +UsedQty`**, and because there is no Idle Quantity (for this example we assume that **`TDQUE=0`**), it follows that **`TDQ=SoldQty`**.

B. DrvQtyCalc for Each Assignment to Cars and Trucks

In general, DrvQtyCalc is calculated according to the formula:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

For this example we are assuming that DQF=0, so:

```
DrvQtyCalc = DQV x DWV x Dest.TDQ
```

And, because the driver from Wheels to Cars and Trucks is not weighted, **`DWV`**=1.

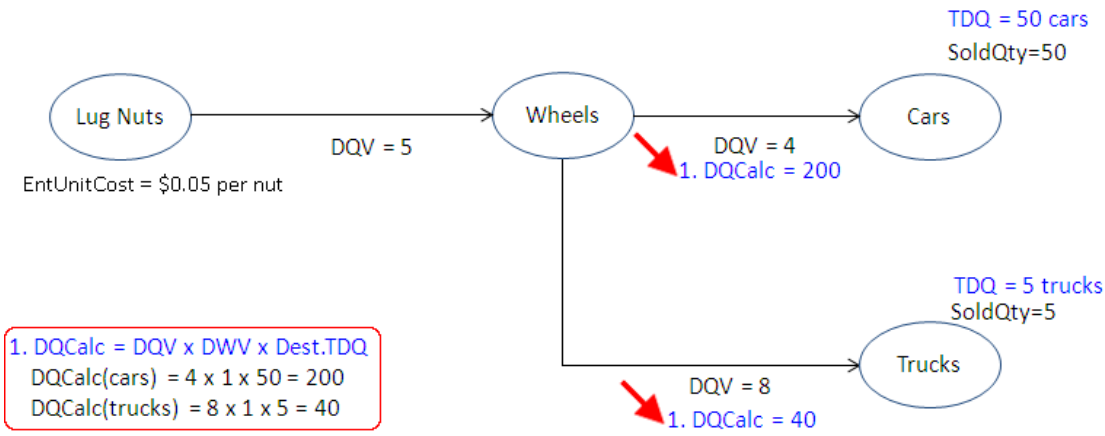
So, by substitution, for one assignment we have:

```
1. DrvQtyCalc = 4 x 1 x 50 = 200
```

and for the other assignment:

```
1. DrvQtyCalc = 8 x 1 x 5 = 40
```

Note: For the pictures that follow for this example, DrvQtyCalc is abbreviated as DQCalc in order to save space.



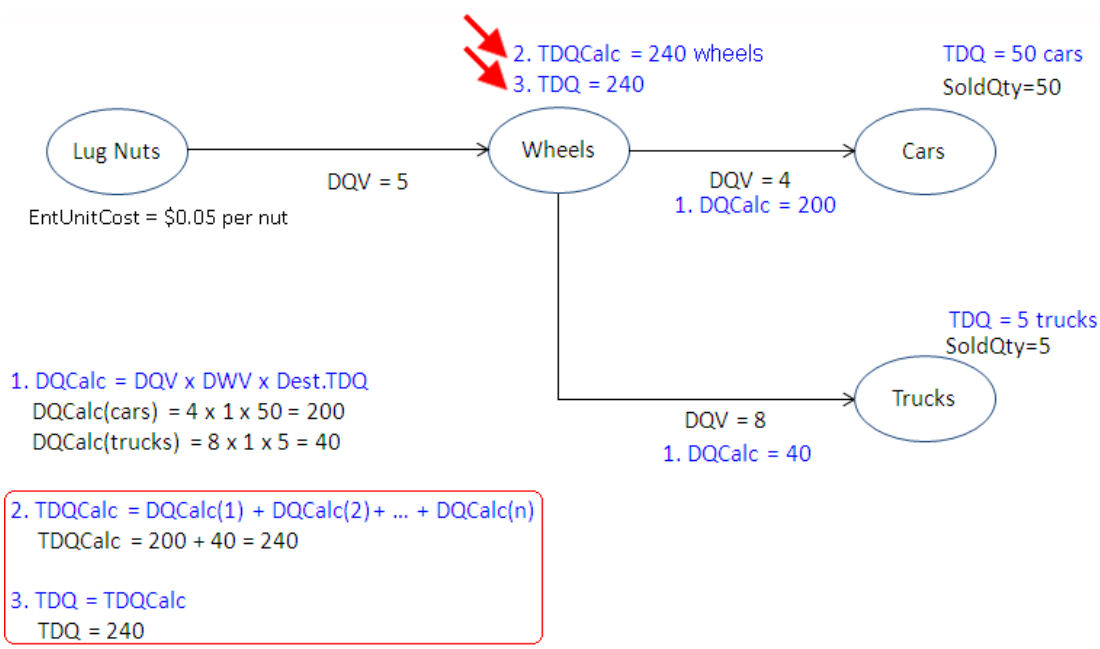
C. TDQCalc and TDQ for Wheels

TDQCalc for the Wheels account is the number of wheels for cars plus the number of wheels for trucks:

```
2. TDQCalc = DrvQtyCalc(cars)+DrvQtyCalc(trucks) = 200 + 40 = 240
```

And, given that **`TDQ=TDQCalc+SoldQty`**, and because Wheels itself has no SoldQty:

```
3. TDQ = TDQCal = 240
```



D. DrvQtyCalc for the Assignment to Wheels

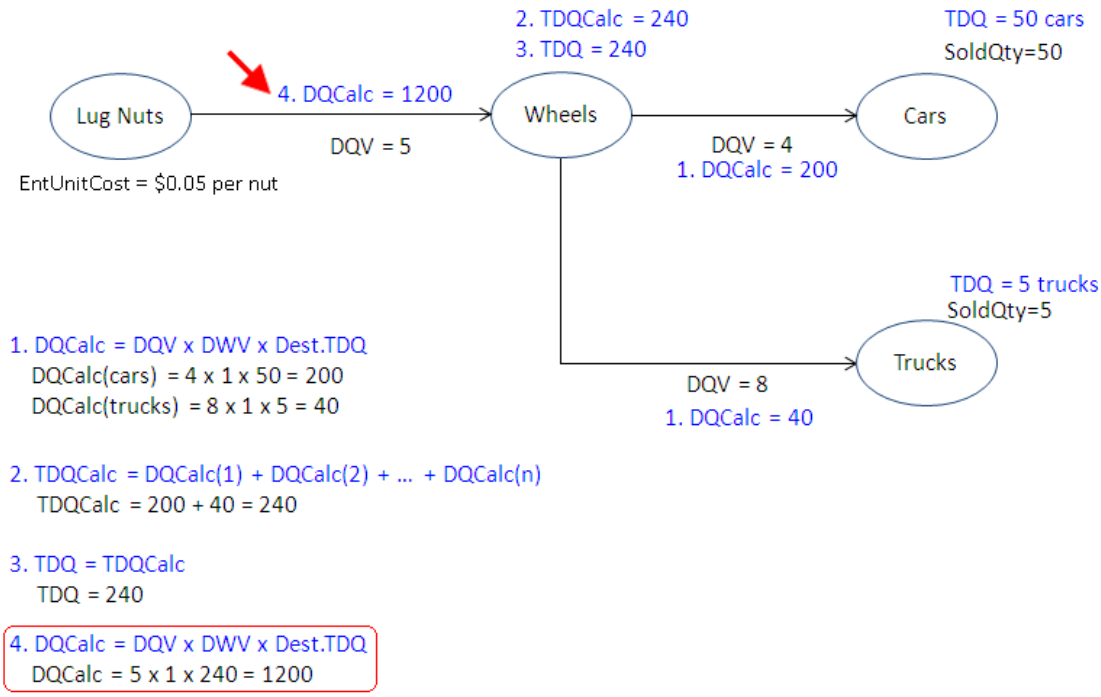
Remember that the general formula for calculating DrvQtyCalc is the following:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

In the case of the assignment from Lug Nuts to Wheels DQF=0, DWF=1 (the driver is not weighted), DQV=5 (there are 5 lug nuts per vehicle),and Dest.TDQ=240 (see above).

So, by substitution we have:

```
4. DrvQtyCalc = (0 x 1) + (5 x 1 x 240) = 1200
```



E. TDQCalc and TDQ for Lug Nuts

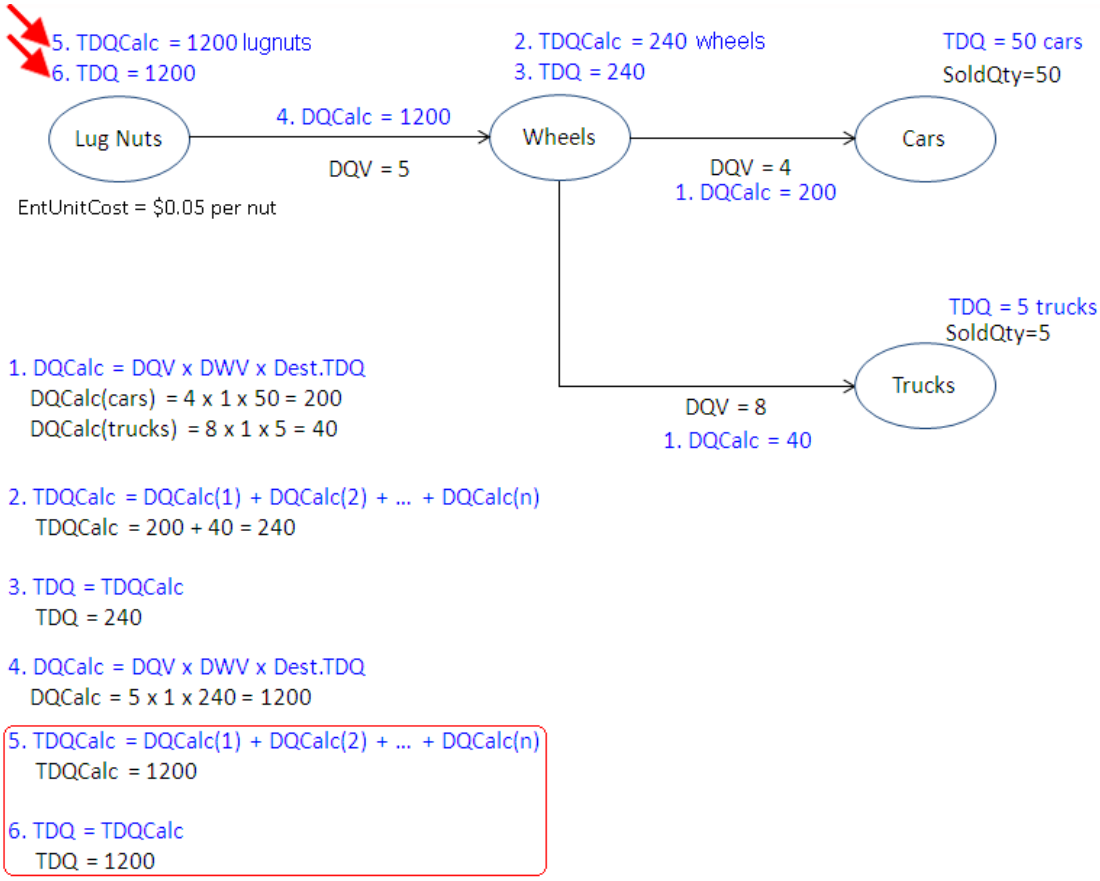
TDQCalc is the sum of DrvQtyCalc for the individual assignments, and since there is only one assignment from Lug Nuts to Wheels:

```
5. TDQCalc = 1200
```

TDQ = TDQCalc + SoldQty, and since SoldQty for LugNuts=0, it follows that:

```
6. TDQCalc = TDQ = 1200
```

Given the calculation of TDQCalc, and therefore also TDQ, we now know how many lug nuts are needed for both cars and trucks.



F. Cost and DrvblCost for Lug Nuts

Knowing how many lug nuts are needed in total, we can calculate the cost according to the formula

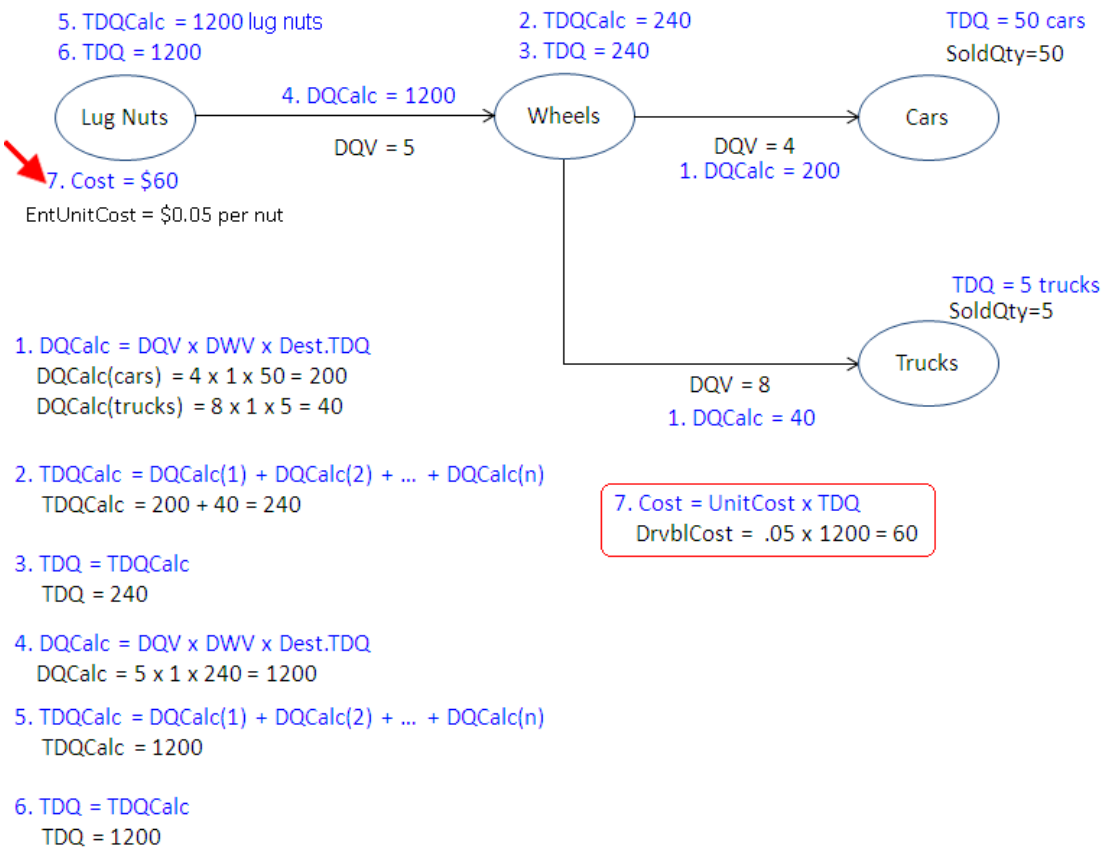
```
7.Cost = UnitCost x TDQ = $0.05 x 1200 = $60
```

Furthermore, because

```
DrvblCost = Cost - Allocated Cost
```

and, because there is no Allocated Cost in this example, it follows that:

```
DrvblCost = Cost = $60
```



G. DrvDrvnCost for Each Assignment to Cars and Trucks

Knowing the total cost, $60, we distribute the cost for cars and trucks respectively according to the formula

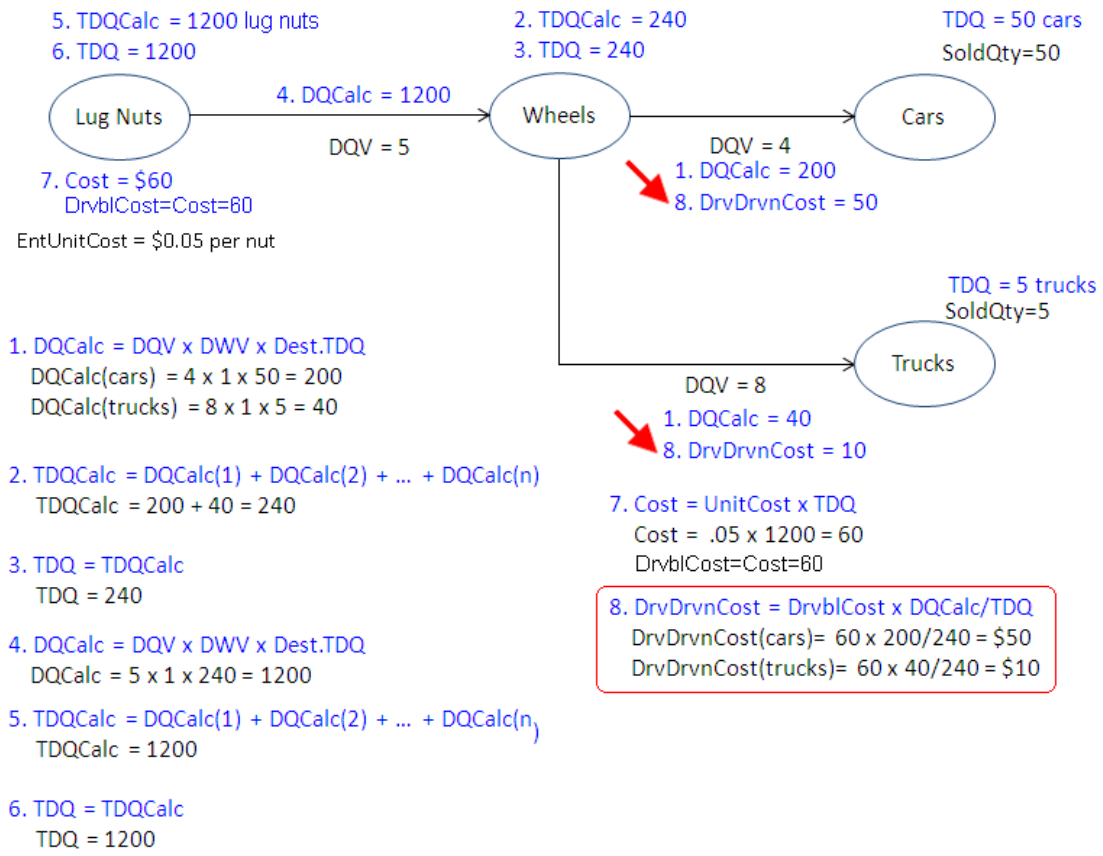
```
DrvDrvnCost = DrvblCost x DrvQtyCalc/TDQ
```

By subsitution for the first assignment:

```
8. DrvDrvnCost = $60 x 200/240 = $50
```

and for the second assignment:

```
8. DrvDrvnCost = $60 x 40/240 = $10
```



Note:

You might have noticed that the formula shown in this example for calculating DrvDrvnCost:

```
DrvDrvnCost = DrvblCost x (DrvQtyCalc/TDQ)
```

is different from the one shown earlier ("DrvDrvnCost" on page 356):

```
DrvDrvnCost = DrvQtyCalc x DrvRate
```

However, we can show that the two formulas are equivalent. If you remember that

```
DrvRate=DrvblCost/TDQ
```

then by substitution in the second formula we have:

```
DrvDrvnCost = DrvQtyCalc x (DrvblCost/TDQ)
```

and that is mathematically equivalent to:

```
DrvDrvnCost = DrvblCost x (DrvQtyCalc/TDQ)
```

More intuitively, whereas the standard formula calculates DrvDrvnCost as:

```
DrvDrvnCost = DrvQtyCalc (the number of units) times
DrvRate (the cost per unit)
```

the formula shown in this example calculates DrvDrvnCost as:

```
DrvDrvnCost = DvrblCost (cost distributed to all destinations)
times DrvQtyCalc / TDQ
(the ratio of units-to-a-particular-destination
to the total-number-of-units-to-all-destinations)
```

Or, more simply:

```
DrvDrvnCost = DvrblCost (cost distributed to all destinations)
times (DrvQtyCalc / TDQ)
(the proportion of costs to a particular destination)
```

H. Cost for Cars and Trucks

The cost for a given account is the sum of costs driven to that account (its ReceivedCost) plus any cost entered directly into the account:

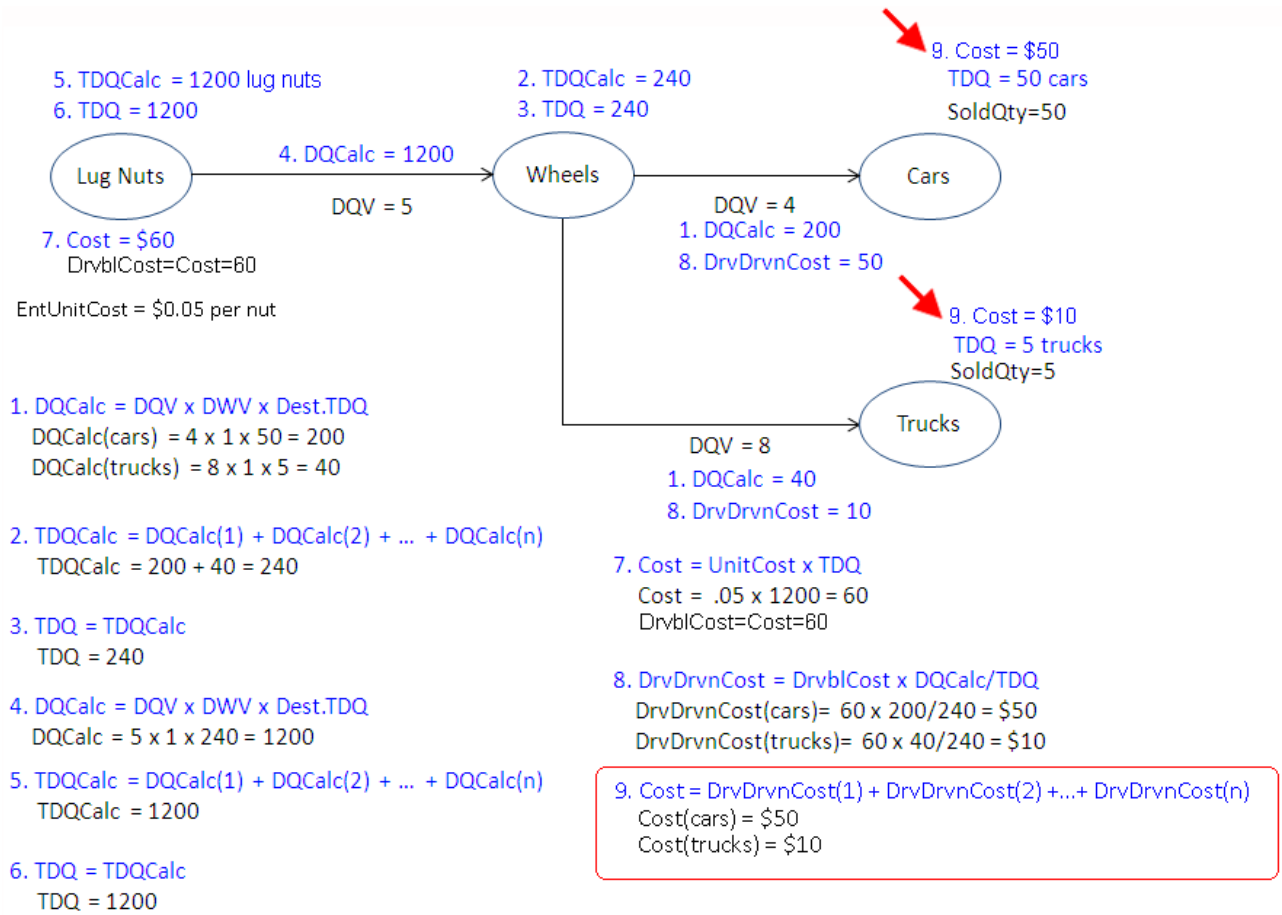
```
Cost = RcvCost + EnteredCost
```

Because, for this example, there is only one driver into each of the Cars and Trucks accounts, the cost is the DrvDrvnCost for that driver. So, for the first assignment:

```
9. Cost(Cars) = DrvDrvCost(Wheels) = $50
```

and for the other assignment:

```
9. Cost(Trucks) = DrvDrvCost(Wheels) = $10
```



Examples 3: Entered Unit Cost with Variable Driver Quantity and Dest.TDQ

Overview

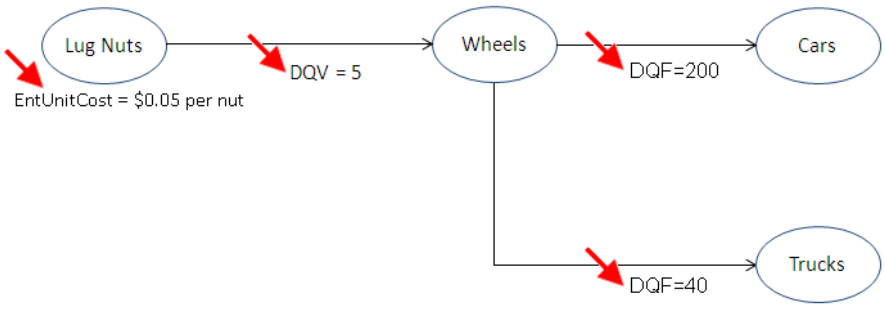
The previous example used DQV along with SoldQty to determine how much output was demanded from a source account. (See "Example 2: Entered Unit Cost with Variable Driver Quantities and SoldQty" on page 374) The example in this section uses DQV along with the destination account's TDQ to determine how much output is demanded from the source account. The problem: calculate how much is spent for cars and trucks respectively, given the following information:

Cars require 200 wheels in all	DQF(cars)=200
Trucks require 40 wheels in all	DQF(trucks)=40
Each wheel (regardless of car or truck) has 5 lug nuts	DQV=5
Each lug nut costs $0.05 per nut	EnteredUnitCost=$0.05

A. User-Entered Values

A user begins by specifying the following user-entered values:

EntUnitCost=$0.05	Each lug not costs $0.05
DQV=5	Five lug nuts are required for each wheel
DQF=200 and DQF=40	200 wheels are required in all for cars; and 40 wheels are required in all for trucks



B. How Many Wheels (DrvQtyCalc) for Cars and Trucks?

The general forumula for calculating the number of units going from a source account to a destination account is the following:

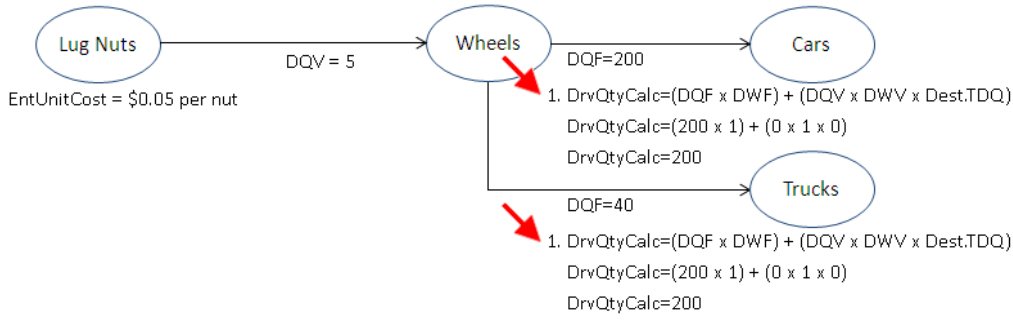
```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

Because there is no weighting factor for drivers in this example, DWF and DWV both equal 1. And, because cars and trucks are at the end of the line of assignments, they have no destination accounts and, so, their Dest.TDQ=0. Consequently, DrvQtyCalc=DQF as can be seen from the following:

```
DrvQtyCalc = (DQF x 1) + (0 x 1 x 0)

DrvQtyCalc = DQF
```

Therefore:

```
DrvQtyCalc(cars)=DQF=200

DrvQtyCalc(trucks)=DQF=40
```



C. How Many Wheels Are Demanded Altogether (TDQ)?

To calculate how many wheels are demanded altogether one must add the wheels for cars plus the wheels for trucks. In general:

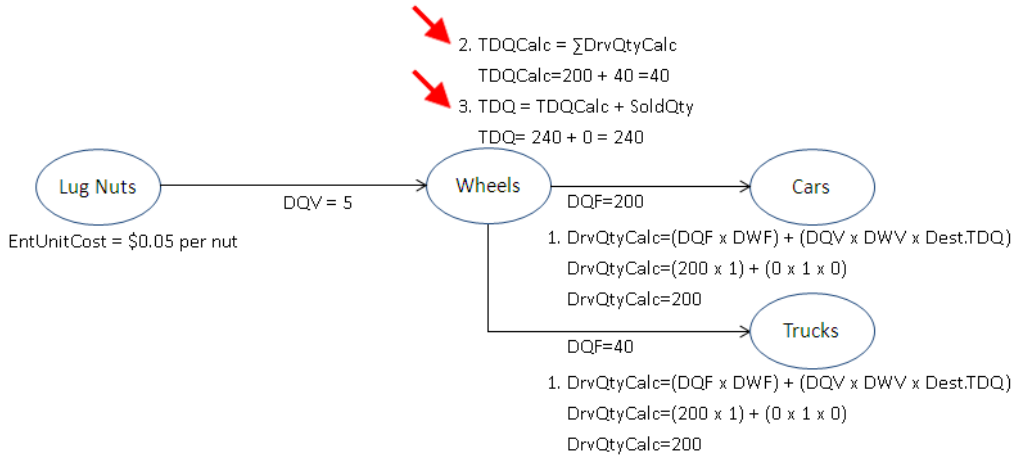
```
TDQCalc=∑DrvQtyCalc
TDQCalc=200+40
TDQCalc=200
```

In general, TDQ (the total units flowing from a source account) is calculated according to the following formula:

```
TDQ=TDQCalc+SoldQty
```

In the current example, there is no SoldQty, so:

```
TDQ=TDQCalc
TDQ=240
```



D. How Many Lug Nuts (DrvQtyCalc) Are Needed for the Wheels?

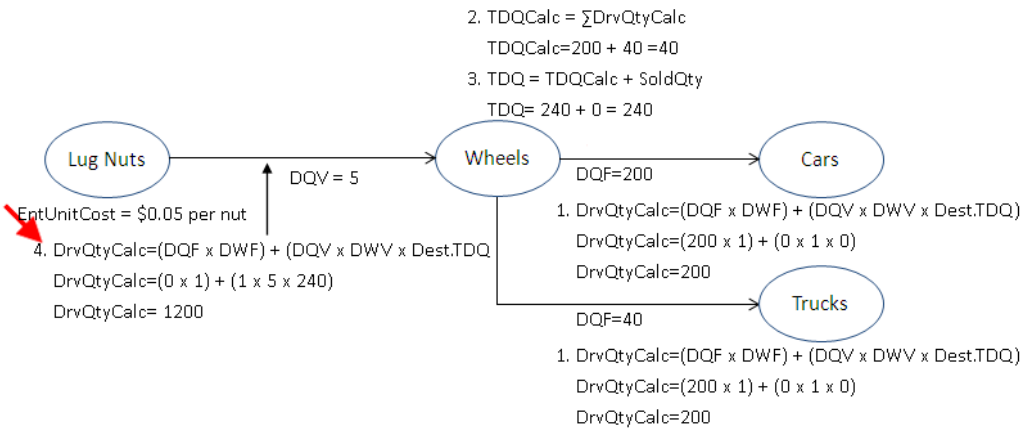
To calculate how many lug nuts are needed for the 240 wheels, one again uses the general formula:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

By substitution, where Dest.TDQ=240 is the number of wheels, and DQV=5 is the number of lug nuts required per wheel:

```
DrvQtyCalc=(0 x 1) + (5 x 1 x 240)
DrvQtyCalc=1200
```

So, 1200 lug nuts are required for 240 wheels.



E. What Does It Cost to Provide the Lug Nuts Demanded?

In general:

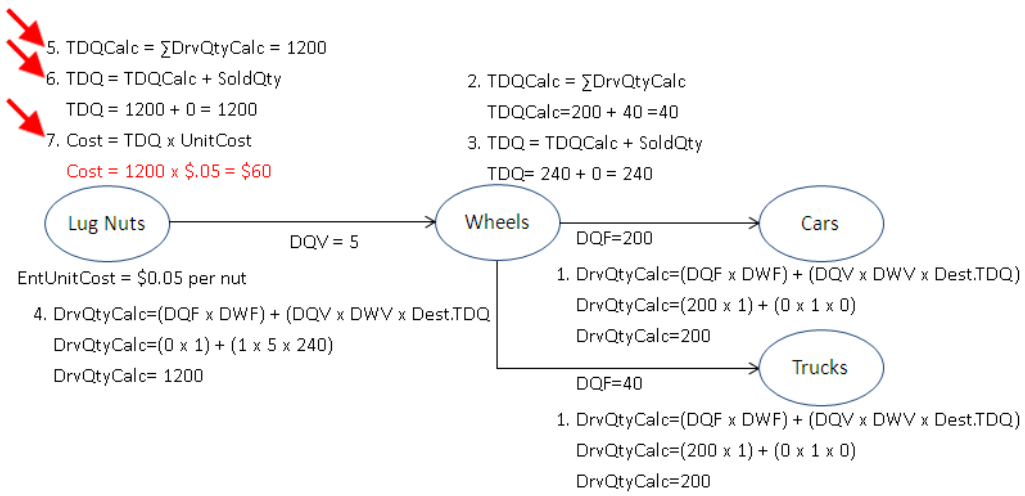
```
TDQCalc=∑DrvQtyCalc
```

And, because, in this example, lug nuts are used only for wheels, there is only one DrvQtyCalc to consider, so:

```
TDQCalc=DrvQtyCalc=1200
```

The cost of lug nuts is calculated by multiplying the number of lug nuts required by the unit cost of each lug nut:

```
Cost=1200 x $0.05 = $60
```



F. What Does It Cost to Provide the Wheels Demanded?

To recapitulate: first the cost of lug nuts was calculated by determining how many lug nuts are required downstream. This meant adding the number of wheels demanded for cars to the number of wheels demanded for trucks, and multiplying that sum by the

number of lug nuts required for each wheel. Now that the cost of lug nuts has been determined, we can calculate how that cost is distributed.

In general, the cost driven (distributed) from an account is calculated according to the following formula:

```
DrvDrvnCost=DrvQtyCalc x DrvRate
```

Because **`DrvRate=DrvblCost/TDQ`**, the following formula obtains:

```
DrvDrvnCost=DrvQtyCalcx(DrvblCost/TDQ)
```

By substitution, to determine DrvDrvnCost for lug nuts we have:

```
DrvDrvnCost=1200 x ($60/1200)
DrvDrvnCost=$60.
```

That is, all the cost in the Lug Nuts account is distributed to its one destination account, Wheels.

Next, to determine the cost of wheels for cars and trucks, the same general formula for DrvDrvnCost is applied to each one.

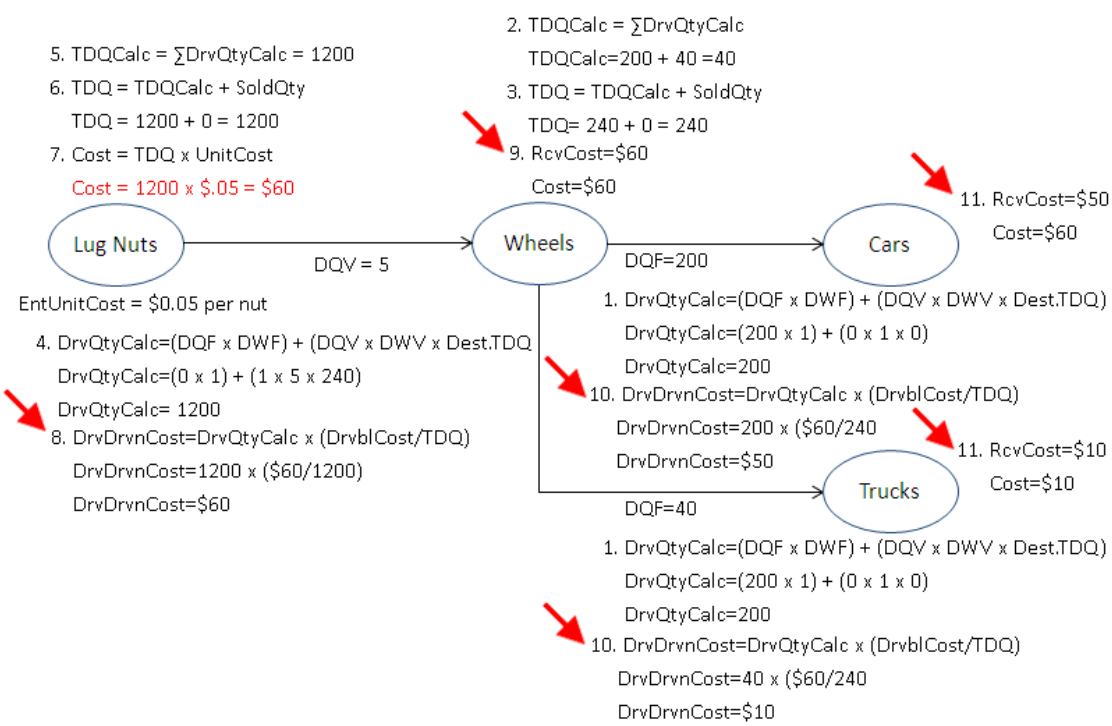
For wheels, we have:

```
DrvDrvnCost=DrvQtyCalc x (DrvblCost/TDQ)
DrvDrvnCost=200 x ($60/240)
DrvDrvnCost=$50
```

For trucks, we have:

```
DrvDrvnCost=DrvQtyCalc x (DrvblCost/TDQ)
DrvDrvnCost=40 x ($60/240)
DrvDrvnCost=$10
```

So, of the $50 spent on lug nuts, $40 went to car wheels and $10 went to truck wheels.

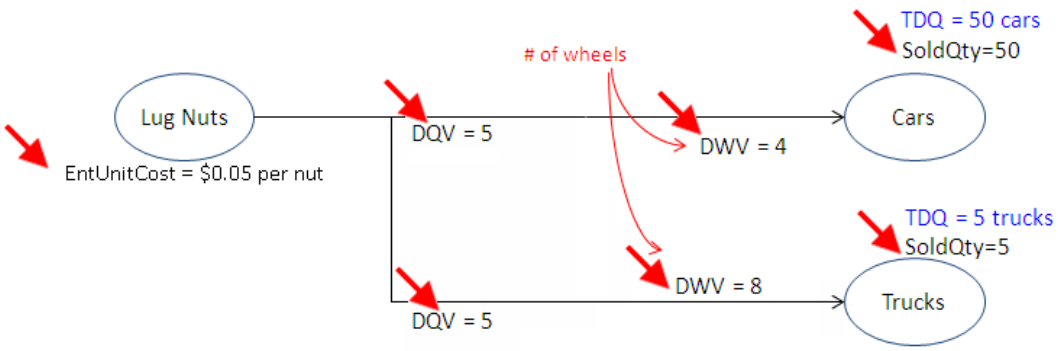


Example 4: Weighted Driver with Variable Driver Quantities

A. Specify User-Entered Values

Now we show an alternative method for distributing the cost of lug nuts to cars and trucks. Instead of creating an account for Wheels and having a separate driver from the Wheels account to each of the Cars and Trucks accounts, we now have a single driver from Lug Nuts with different weights for Cars and Trucks.

Driver weights are used when, in addition to driver quantities, some special characteristics need to be considered to distribute costs. For example, complexity of the product, length of the order (and so forth). In this example, the weight captures how many wheels are required for each of Cars and Trucks respectively, as shown in the following picture:

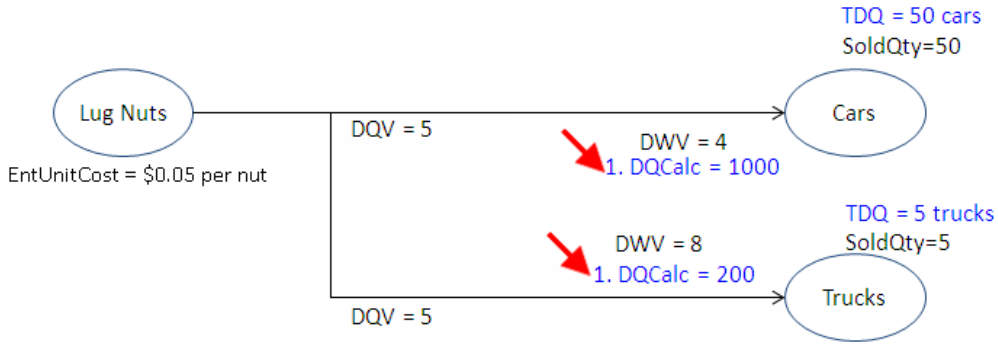


B. DrvQtyCalc for Each Assignment from Lug Nuts to Cars and Trucks

As before, we calculate DrvQtyCalc according to the formula

```
DrvQtyCalc = DQV x DWV x Dest.TDQ
```

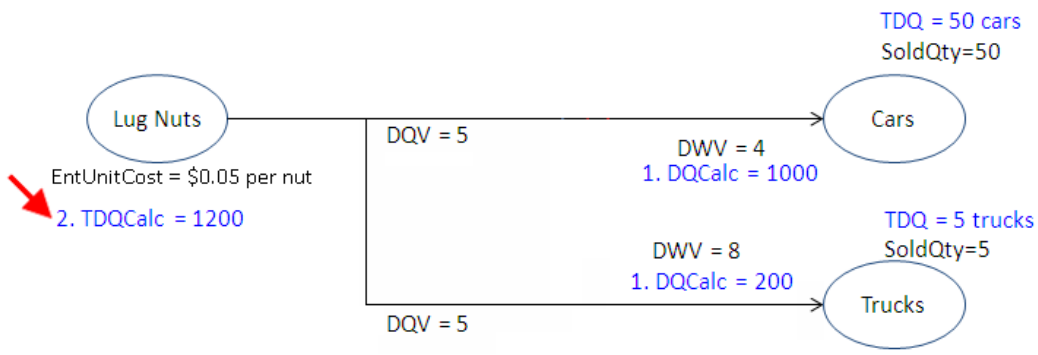
This time, however, there is a weight to be included in the calculation, as shown in the following picture:



1. DQCalc = DQV x DWV x Dest.TDQ
   DQCalc(cars) = 5 x 4 x 50 = 1000
   DQCalc(trucks) = 5 x 8 x 5 = 200

C. TDQCalc for Lug Nuts

As shown in the following picture, TDQCalc for Lug Nuts is the sum of DrvQtyCalc for cars plus DrvQtyCalc for trucks.

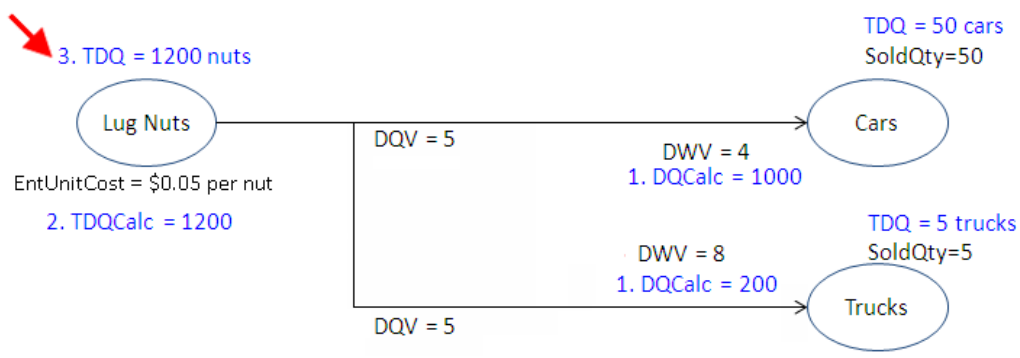


1. DQCalc = DQV x DWV x Dest.TDQ
   DQCalc(cars) = 5 x 4 x 50 = 1000
   DQCalc(trucks) = 5 x 8 x 5 = 200

2. TDQCalc = DQCalc(1) + DQCalc(2) + ... + DQCalc(n)
   TDQCalc = 1000 + 200 = 1200

D. TDQ for Lug Nuts

TDQ=TDQCalc+SoldQty, and because SoldQty=0, TDQ=TDQCalc. So, now we know how many lug nuts are needed in total for both cars and trucks.



1. DQCalc = DQV x DWV x Dest.TDQ
   DQCalc(cars) = 5 x 4 x 50 = 1000
   DQCalc(trucks) = 5 x 8 x 5 = 200

2. TDQCalc = DQCalc(1) + DQCalc(2) + ... + DQCalc(n)
   TDQCalc = 1000 + 200 = 1200

3. TDQ = TDQCalc
   TDQ = 1200

E. Cost and DrvblCost for Lug Nuts

Cost equals the total number of lug nuts times the unit cost for each lug nut:

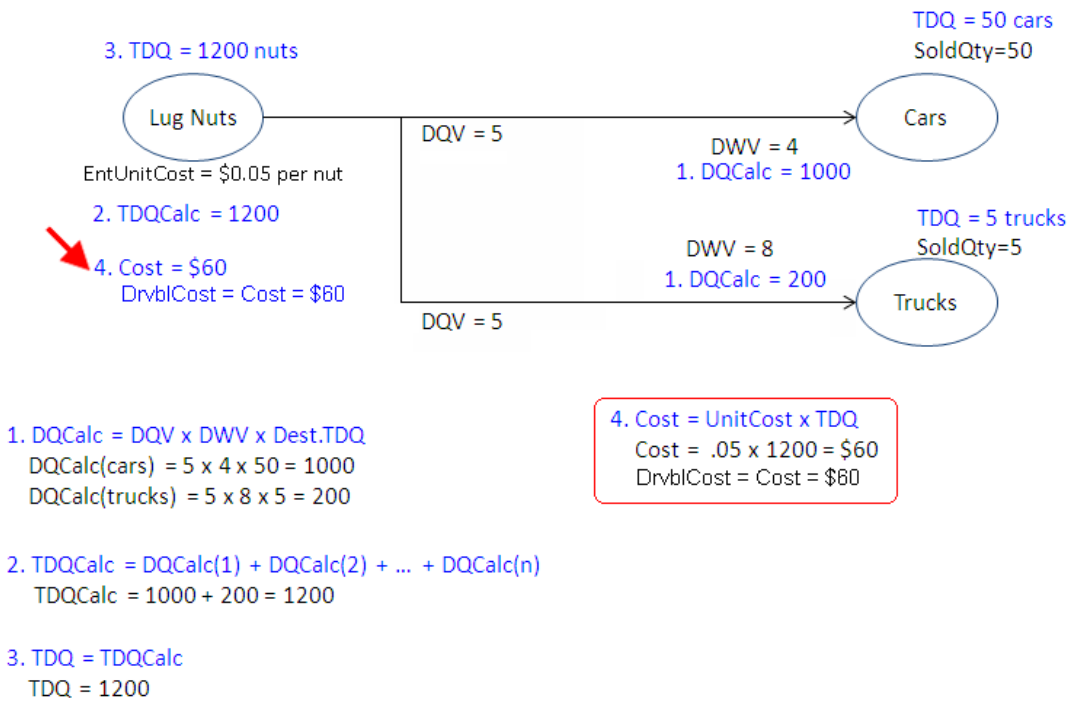
```
Cost = UnitCost x TDQ = $0.05 x 1200 = $60
```

because:

```
DrvblCost = Cost - Allocated Cost
```

and, because there is no Allocated Cost in this example, it follows that

```
DrvblCost=Cost
```



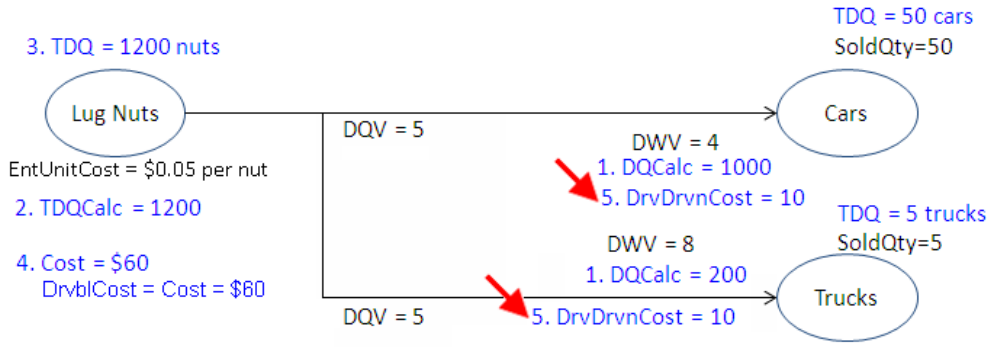
F. DrvDrvnCost for Each Assignment from Lug Nuts to Cars and Trucks

Knowing that drivable cost is $60, we can distribute the cost to cars and trucks respectively according to the formula:

```
DrvDrvnCost = DrvblCost x DrvQtyCalc/TDQCalc
```

as shown in the following picture:

Step F



1. DQCalc = DQV x DWV x Dest.TDQ
   DQCalc(cars) = 5 x 4 x 50 = 1000
   DQCalc(trucks) = 5 x 8 x 5 = 200

2. TDQCalc = DQCalc(1) + DQCalc(2) + ... + DQCalc(n)
   TDQCalc = 1000 + 200 = 1200

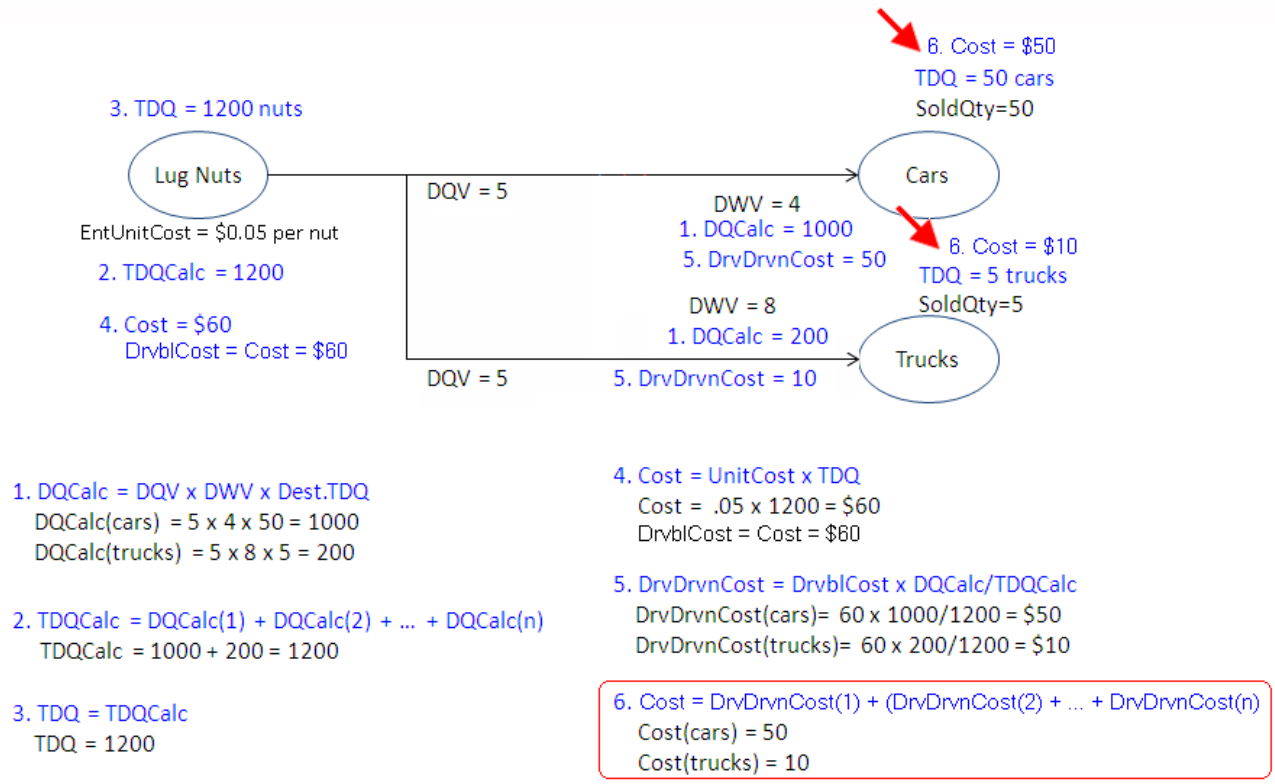
3. TDQ = TDQCalc
   TDQ = 1200

4. Cost = UnitCost x TDQ
   Cost = .05 x 1200 = $60
   DrvblCost = Cost = $60

5. DrvDrvnCost = DrvblCost x DQCalc/TDQCalc
   DrvDrvnCost(cars)= 60 x 1000/1200 = $50
   DrvDrvnCost(trucks)= 60 x 200/1200 = $10

G. Cost for Cars and Trucks

The cost for a given account is the sum of costs driven to that account by all the drivers into it, plus whatever entered cost is entered directly into the account. Because, for this example, there is no entered cost and there is only one driver going into each of the Cars and Trucks accounts, the cost is the DrvDrvnCost for that driver, as shown in the following picture:



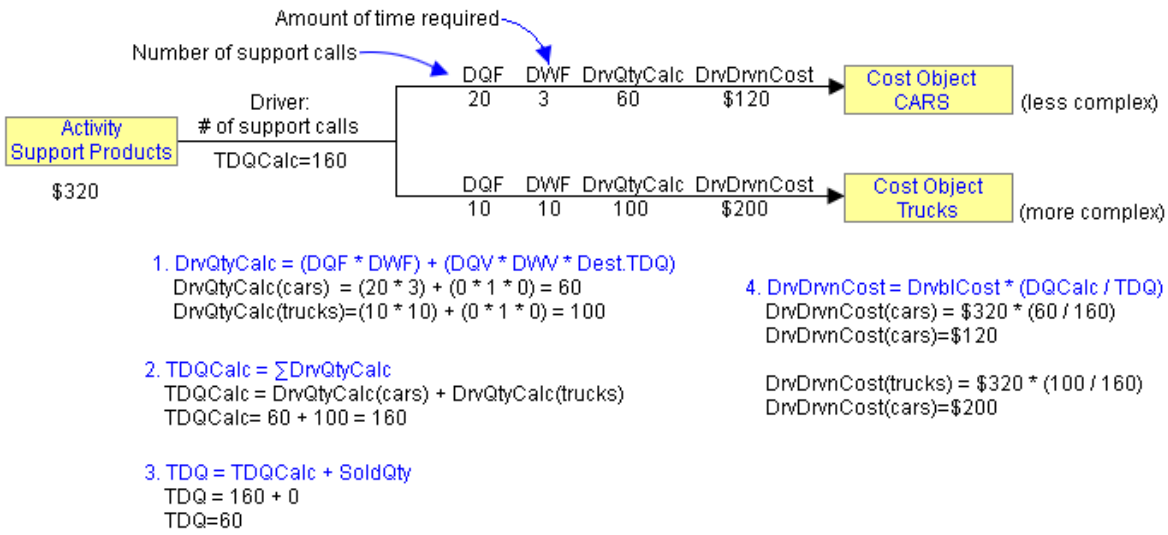
Example 5: Weighted Driver with Fixed Driver Quantities

For this example, we show the properties in a single picture. The problem for this example is to distribute the cost of support calls for two products: cars (a less complex product requiring less time to support) and trucks (a more complex product requiring more time to support). The data given for the problem is the following:

- $320 was spent on support calls for both cars and trucks combined.
- Whereas each support call for a car takes 3 minutes (DWF=3), each support call for a truck requires 10 minutes (DWF=10).

The following picture shows the calculation of the cost driven to each product (DrvDrvnCost) using the following formulas:

- **`DrvQtyCalc = (DQF x DWF)+(DQV x DWV x Dest.TDQ)`**
- **`TDQCalc = DrvQtyCalc(1) + DrvQtyCalc(2) +...+ DrvQtyCalc(n)`**
- **`TDQ = TDQCalc + SoldQty`**
- **`DrvDrvnCost = DrvblCost x (DrvQtyCalc/TDQ)`**

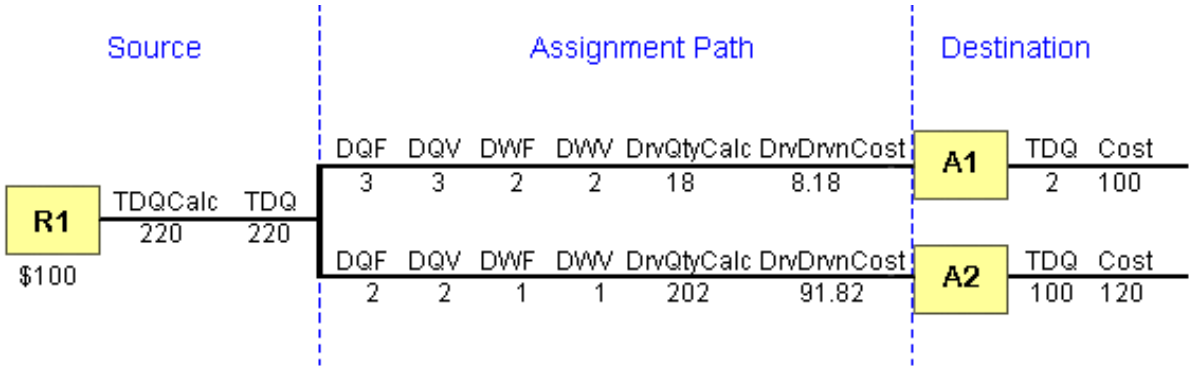


Example 6: Weighted Driver with Fixed and Variable Driver Quantities

For this example, we again show the properties in a single picture for the cost flow of a driver with two assignment paths, each using a different weight.

Once again, the picture shows the calculation of the cost driven to each product (DriverCost) using the following formulas:

- **`DrvQtyCalc = (DQF x DWF)+(DQV x DWV x Dest.TDQ)`**
- **`TDQCalc = DrvQtyCalc(1) + DrvQtyCalc(2) + ... + DrvQtyCalc(n)`**
- **`TDQ = TDQCalc + SoldQty`**
- **`DrvDrvnCost = DrvblCost x (DrvQtyCalc/TDQ)`**



1. DrvQtyCalc = (DQF * DWF) + (DQV * DWV * Dest TDQ)
   DrvQtyCalc(A1) = (3*2) + (3*2*2) = 18
   DrvQtyCalc(A2) = (2*1) + (2*1*100) = 202

2. TDQCalc=∑DrvQtyCal
   TDQCalc=18+202=220

3. TDQ=TDQCalc+SoldQty
   TDQ=220+0=220

4. DrvDrvnCost=DrvblCost x (DrvQtyCalc/TDQ)
   DrvDrvnCost(A1) = 100 x (18/220) = 8.18
   DrvDrvnCost(A2) = 100 x (202/220) = 91.82

Chapter 44

Calculated Driver

Calculated Driver

Introduction

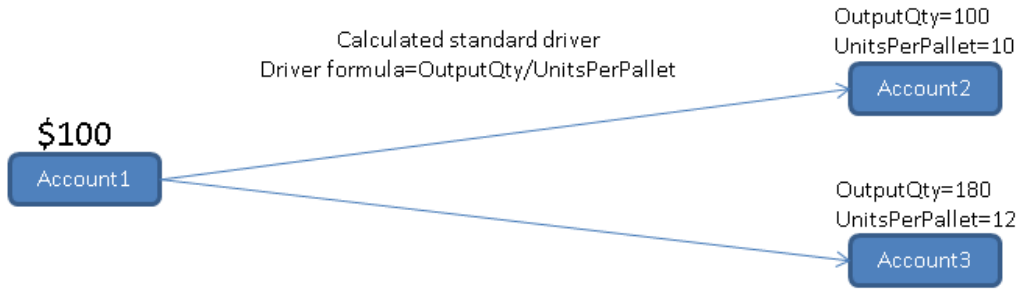
A calculated driver is a variety of standard driver. It is one that, instead of using the standard formula for calculating DrvQtyCalc, uses a formula—referred to as a *Driver Formula*— that you specify as a user. The formula can use any number of properties and attributes and can be as simple or as complicated as you want. See Chapter 55, “Properties That Can Be In Formulas,” on page 495.

For example, suppose that a company wants to calculate the shipping cost for products that are shipped on pallets. The shipping vendor's cost is based on how many pallets are shipped, not on how many units are produced. To determine the shipping cost, the output quantity (the property OutputQty) is divided by the number of products that a pallet can hold (the user-defined numeric attribute UnitsPerPallet), as follows:

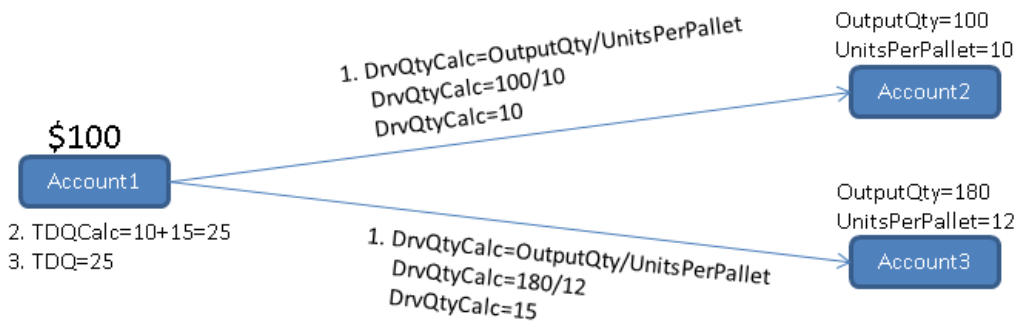
```
OutputQty/UnitsPerPallet
```

If the physical dimensions of the product change, then the number of units that a pallet can hold will change. The value of UnitsPerPallet can be changed accordingly. Also, the value of UnitsPerPallet can be different for each product, but the same equation can be used for multiple products.

The following picture shows a very simple example using a calculated driver, whose driver formula is **`OutputQty/UnitsPerPallet`**, to distribute part of $100 from Account1 to Account2 and the remaining part to Account3.



Calculate DrvQtyCalc, TDQCalc, and TDQ



Step 1: DrvQtyCalc represents the number of units going on an assignment to a destination account. In the case of a calculated driver, DrvQtyCalc is assigned the value of the driver's formula, **`OutputQty/UnitsPerPallet`**. In this case, OutputQty is a system property of each destination account, and UnitsPerPallet is a user-defined attribute assigned to each destination account.

- For the first assignment, DrvQty=100/10=10
- For the second assignment, DrvQty=180/12=15

Step 2: TDQCalc is the total number of units going to all destination accounts. In other words, it is the sum of DrvQtyCalc for all of the destination accounts.. In this example, TDQCalc is 25.

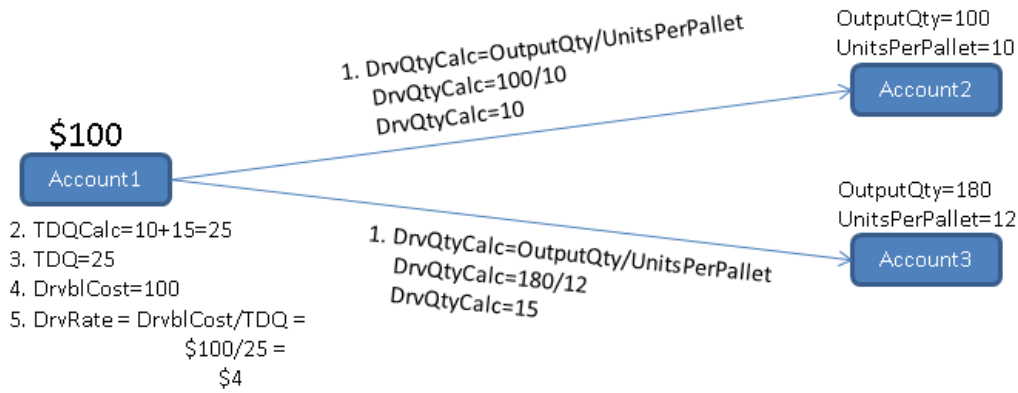
Step 3: If we ignore complications which are discussed in other chapters, Total Driver Quantity (TDQ) equals TDQCalc. So, for this example, TDQ=TDQCalc=25.

Note: TDQ can exceed TDQCalc if either is the case:

- SoldQty is specified for the source account, in which case **`TDQ=SoldQty +TDQCalc`**.
- TDQUE (Total Driver Quantity User-Entered) is specified as greater than TDQ, in which case **`TDQ=TDQUE`** (TDQUE overrides TDQ).

  Specifying TDQUE as greater than TDQ indicates that there is idle capacity—more units on the source account than are actually distributed to target accounts (the total units distributed is represented by TDQCalc).

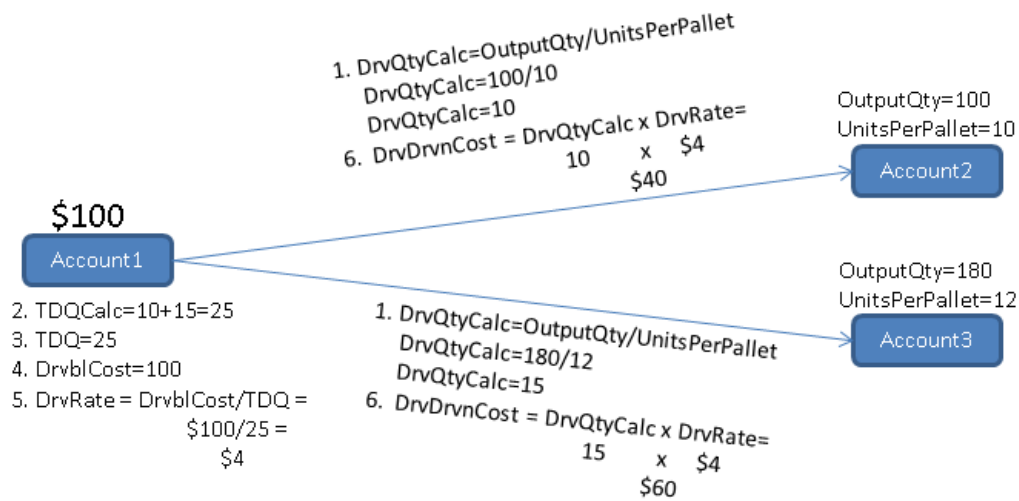
Calculate DrvblCost and DrvRate



Step 4. Drivable Cost (DrvblCost) equals Cost minus Allocated Cost. (See "User-Entered Cost Allocation" on page 400.) For this example, there is no allocated cost, so Drivable Cost equals the cost in the account=100.

Step 5. Driver Rate is the cost per unit for any destination account. It is the Drivable Cost divided by the total number of units going to all destination accounts (DrvblCost/TDQ). In this example, $4 is distributed for every unit that is distributed to a destination account.

Calculate DrvDrvnCost



Step 6. Driver Driven Cost is the cost that is distributed on an assignment to a particular destination account. It is calculated by multiplying DrvQtyCalc (the units going to a particular destination account) times DrvRate (the cost per unit for any destination account).

Further Information on Calculated Drivers

The formula for a calculated driver is not periodic; the formula remains the same for all periods.

By default, if a name in the formula of a calculated driver is not found belonging to the source account, then it is assumed to belong to the destination account. In more detail, the order of precedence for resolving references (for example "Foo") in the formula of a calculated driver is the following:

1. Assignment path
2. Destination account
3. Source account
4. Dimension reference
5. Attribute reference

The following properties are not valid for use in the formula of a calculated driver and, therefore, do not appear in the Formula Builder drop-down list:

- DriverQuantityBasic
- IdleDriverQuantity

The reason is that in an assignment these quantities depend on DrvQtyCalc. Consequently, a vicious circle would be created if the formula were to make DrvQtyCalc depend on these quantities. That is, DrvQtyCalc could not be determined without first determining these quantities, and these quantities could not be determined without first determining DrvQtyCalc.

The following examples illustrate how calculated drivers can be used in a model. These equations often include numeric properties and numeric attributes.

Note: You can use cost properties in the driver formula for a calculated driver, but you will need to use driver sequencing to ensure that the properties are non-zero. If you use the default sequence of 1, then all cost values will be zero except for Entered Cost.

See Also

- Chapter 51, "Formulas," on page 457
- "Using Numeric Attributes in a Formula" on page 467
- "Properties That Can Be in Formulas" on page 495

Examples of a Calculated Driver

Delivering a Product to Customers

Suppose that a company wants to calculate the cost of product delivery. To determine the delivery cost, the number of deliveries to each customer per month is multiplied by the average delivery time. Both of these attributes are user-defined numeric attributes.

```
DeliveriesPerMonth*AvgTimePerDelivery
```

A customer who prefers more deliveries per month might pay a different delivery cost than a customer who prefers fewer deliveries. Likewise, a customer who is farther away from the company might pay a different delivery charge than a customer who is closer.

Storage Costs for an Inventory of Finished Goods

Suppose that a company wants to calculate the cost of storing slow-moving products. To determine the storage cost, the output quantity (the property OutputQuantity) is divided by the speed at which a product sells (the user-defined numeric attribute InventoryTurns). The result of that calculation is multiplied by the volume of storage space that a product requires (the numeric attribute CuFtPerUnit), as follows:

```
(OutputQuantity/InventoryTurns)*CuFtPerUnit
```

Dividing OutputQuantity by InventoryTurns yields the average inventory level in units. In order to use CuFtPerUnit, the Resource module in the model must include costs, such as utilities or rent, that are based on per-cubic-foot values.

Carrying Costs of an Inventory of Finished Goods

Suppose that a company wants to add costs to a model to calculate the financial carrying cost of inventory. The previous equation could be altered to calculate the financial carrying cost of the inventory, not the physical storage cost:

```
(OutputQuantity/InventoryTurns)*Cost
```

Processing Customer Orders

Suppose that a company processes orders for other companies. The number of orders that are processed is only part of the important data. The company might need to consider the complexity of each customer's order.

To determine the cost of processing an order, the number of orders that are processed (the user-defined numeric attribute NumberOfOrdersProcessed) is multiplied by the complexity of each order (the user-defined numeric attribute OrderComplexityByCustomer), as follows:

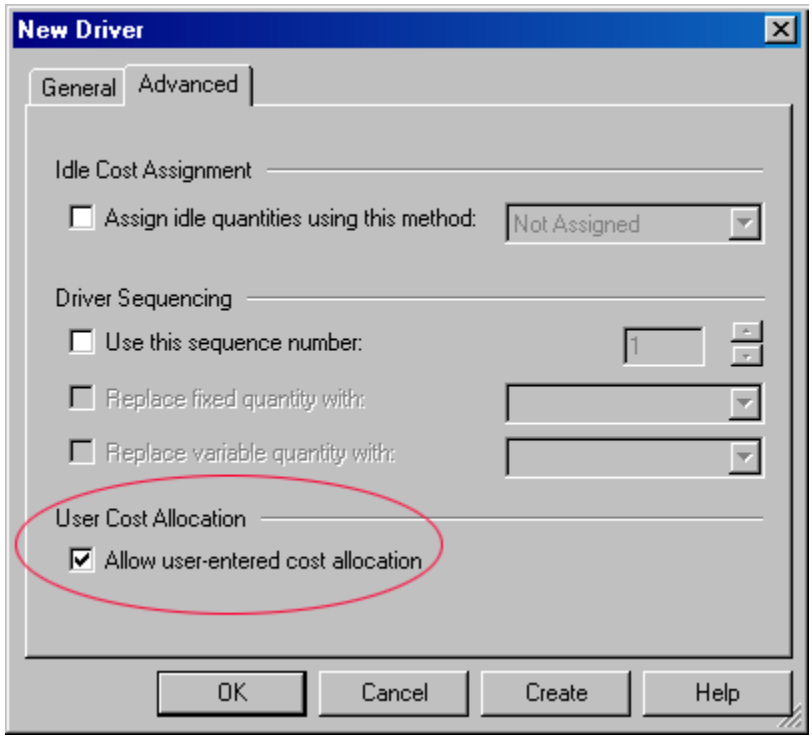
```
NumberOfOrdersProcessed*OrderComplexityByCustomer
```

For OrderComplexityByCustomer, a different value could be assigned to each customer, and the same equation could be used for all customers.

Chapter 45

Driver Topics

User-Entered Cost Allocation



User-entered cost allocation enables you to assign a specific cost to an assignment path by setting the Driver Allocated Cost (DrvAllocCost) property.

Some organizations allocate a specific cost to an account or department at the start of a budgetary period. This cost is then depleted during the budgetary period. This type of allocation is used in more traditional costing methodologies, but it is generally not part of the activity-based costing methodology. However, if the people within your organization are more familiar with this method of allocating costs, you can employ user-entered cost allocation to simulate this use.

See Also

- "Allocated Cost (AllocCost)" on page 747
- "Driver Allocated Cost (DrvAllocCost)" on page 758
- "Received Allocated Cost (RcvAllocCost)" on page 784

Fixed Driver Quantities

For fixed driver quantities, the cost that flows to a destination account from a source account does not depend on any property of the destination account. In other words, it does not depend on demand. The cost is "pushed" by the quantity supplied. The source

account's driver rate is multiplied by the fixed driver quantity (represented by the property DQF).

For example, suppose that $100.00 in the resource account Salary flows into two activity accounts: Take Orders and Process Orders. The costs that flow through a Standard, fixed-quantity driver are shown in the following figure:

The total driver quantity (TDQ) for Salary is 3.00 (2.00 for Take Orders plus 1.00 for Process Orders).

The driver rate that flows from Salary is calculated by dividing the total drivable cost of Salary by the total driver quantity (TDQ) of Salary: $100.00/3.00 = $33.33.

So, the cost for each account is calculated as follows:

Account	DrvRate x DQF	Cost
Take Orders	$33.33 x 2.00	$66.67
Process Orders	$33.33 x 1.00	$33.33

Note: All values are rounded.

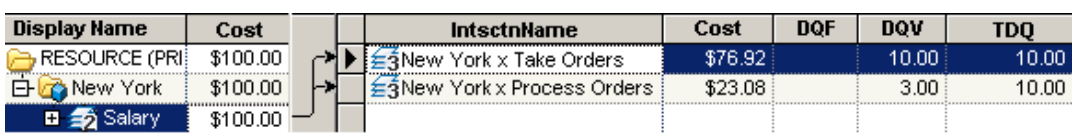
See Also

- Chapter 41, "Evenly Assigned Driver," on page 359
- "Percentage Driver" on page 363
- "Example 1: Entered Unit Cost with Fixed Driver Quantities" on page 371

Variable Driver Quantities

For variable driver quantities, the cost that flows to a destination account from a source account depends on the destination account's total driver quantity. In other words, the cost is determined by demand. It is "pulled" by the quantity required at the final destination. The source account's driver rate is multiplied by the variable driver quantity (DQV) and by the Destination.TDQ.

For example, suppose that $100.00 in the resource account Salary flows into two activity accounts, Take Orders and Process Orders. The costs that flow through a Standard, variable-quantity driver are shown in the following figure:



The total driver quantity (TDQ) for Salary is 130.00, which is calculated as follows:

Account	DQV x Test.TDQ	Total
Take Orders	10.00 x 10.00	100.00
Process Orders	3.00 x 10.00	30.00
		130.00

The driver rate that flows from Salary is calculated by dividing the total drivable cost of Salary by the total driver quantity (TDQ) of Salary: $100.00/130.00 = $0.7692.

So, the cost for each account is calculated as follows:

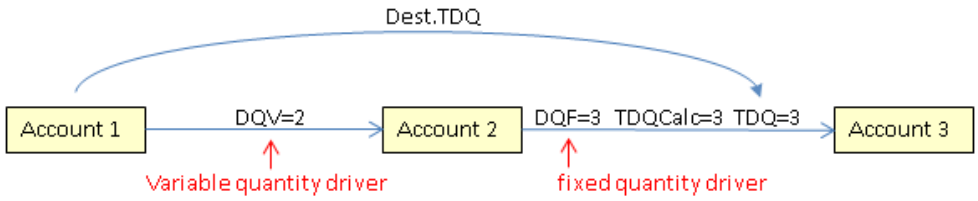
Account	DrvRate x DQV xTest.TDQ	Cost
Take Orders	$00.7692 x 10 x 10.00	$76.92
Process Orders	$00.7692 x 3 x 10.00	$23.08

Note: All values are rounded.

As for TDQ, it is determined in general by two quantities:

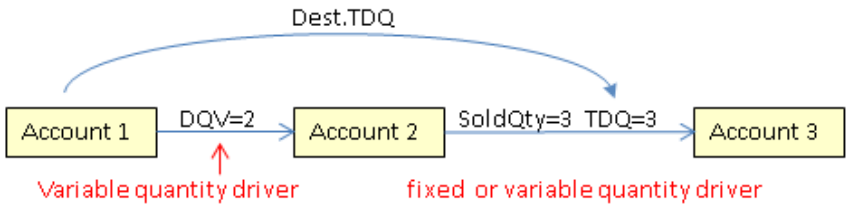
TDQCalc

For an account that uses a variable quantity driver (Account 1 in the picture below), if at least one of its destination accounts (Account 2) assigns costs using a fixed quantity driver, then the DQF for that assignment makes for a non-zero TDQCalc (TDQCalc=3 in the picture below) for that destination account, and hence for a non-zero TDQ (TDQ=3). And, this is the Dest.TDQ for the account (Account 1) that uses the variable quantity driver.

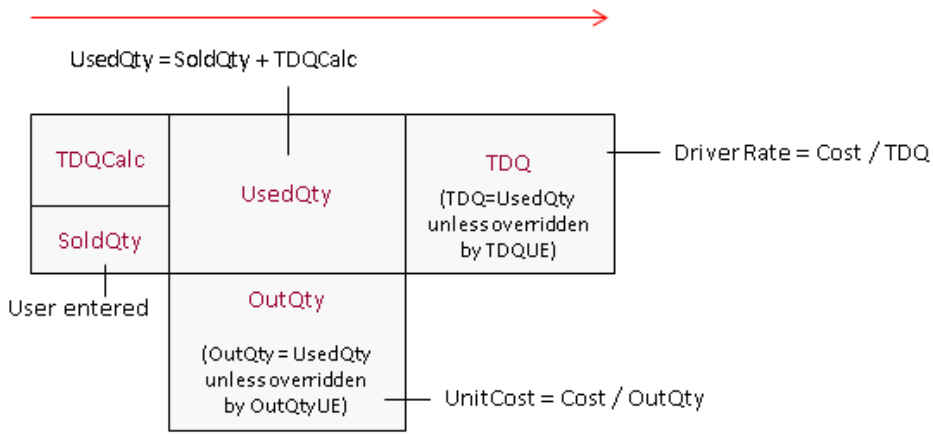


SoldQty

You can determine the quantity demanded for a product by specifying SoldQty for some destination account of a source account (Account 1 in the picture below) that uses a variable quantity driver.



The following picture (which is reproduced from "Diagram 3: Account Driver Properties " on page 529) shows the relationship between TDQ and TDQCalc and SoldQty (and UsedQty and OutQty)).



See Also

"Demand Flow" on page 374

Variable Driver Quantities and Fixed Driver Quantities

You can use both variable quantities and fixed quantities in a single driver.

For example, suppose that $100.00 in the resource account Salary flows into two activity accounts, Take Orders and Process Orders. The costs that flow through a Standard, fixed-quantity and variable-quantity driver are shown in the following figure:

Display Name	Cost
RESOURCE (PRI	$100.00
New York	$100.00
Salary	$100.00

IntsctnName	Cost	DQF	DQV	TDQ
New York x Take Orders	$76.69	2.00	10.00	10.00
New York x Process Orders	$23.31	1.00	3.00	10.00

The total driver quantity (TDQ) for Salary is 133.00, which is calculated as follows:

Account	(DQF) +	(DQV x Dest.TDQ)	Total
Take Orders	2.00	10.00 x 10.00	102.00
Process Orders	1.00	3.00 x 10.00	31.00
			133.00

The driver rate that flows from Salary is calculated by dividing the total drivable cost of Salary by the total driver quantity (TDQ) of Salary: $100.00/133.00 = $00.7519.

So, the cost for each account is calculated as follows:

Account	(DrvRate x DQF) +	(DrvRate x DQV x Dest.TDQ)	Total
Take Orders	$00.7519 x 2.00	$00.7519 x 10.00 x 10.00	$76.69

Account	(DrvRate x DQF) +	(DrvRate x DQV x Dest.TDQ)	Total
Process Orders	$00.7519 x 1.00	$00.7519 x 3.00 x 10.00	$23.31

Note: All values are rounded.

See Also

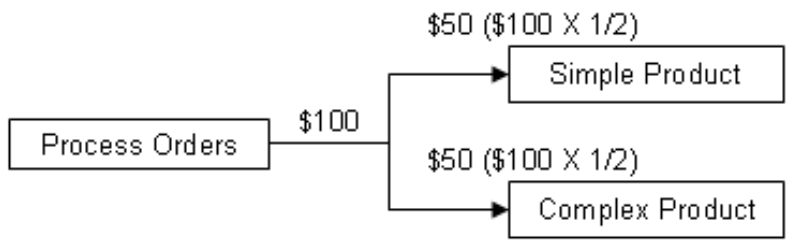
"Example 6: Weighted Driver with Fixed and Variable Driver Quantities" on page 391

Driver Weights

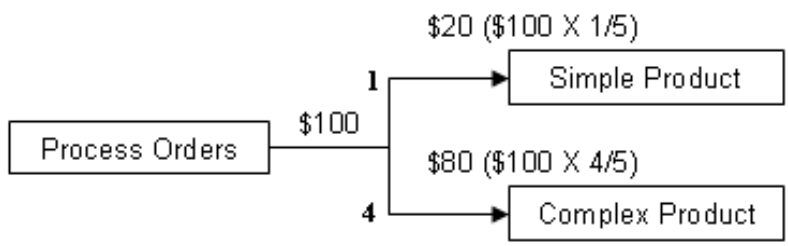
Overview

A weighted driver contains two factors that represent the relationship between accounts. The first factor measures volume (or frequency), and the second factor measures intensity (or complexity). The two factors are multiplied to produce a calculated value that is used for distributing costs. This calculated value, based on weight, is necessary because if the cost of a source account is merely passed to each destination account, the cost incurred by the complexity of each destination account is not considered. Therefore, the cost of each destination account is not accurate. To compensate for the intensity or complexity of each destination account, a relative weight is specified for each destination account.

For example, in the following figure, the cost ($100) for the activity Process Orders is equally assigned to the cost objects Simple Product and Complex Product. But, this cost is not accurate because of the difference in complexity between the products.



Suppose that Complex Product is four times more complex than Simple Product. To consider the different complexities, assign a weighted driver to Process Orders. Then, specify the driver weight of 1 to Simple Product and 4 to Complex Product. After calculating, the costs for the products are significantly different and more accurate.



Specify the driver weights using the properties Driver Weight Fixed and Driver Weight Variable, depending on whether the driver supports fixed quantities, variable quantities, or both. (Examples of weighted drivers)

Example: Processing Customer Orders

The following examples illustrate how weighted drivers can be used in a model. These equations often include numeric properties and numeric attributes.

Suppose that a company sells products with varying numbers of options. Some products offer no options and some products offer many options. The cost of processing customer orders is equal to the number of orders that are processed (frequency), multiplied by the average number of lines per order (complexity, which is the number of options for each product). The products that offer the most options have greater weights than the products that offer fewer (or no) options.

Example: Supporting Products

Suppose that a company sells products of varying complexity. Some products are simple and yield customer telephone calls that require little time. Some products are complex and yield customer calls that require significant time. Each product is assigned a weight based on the amount of time needed for an average customer call.

See Also

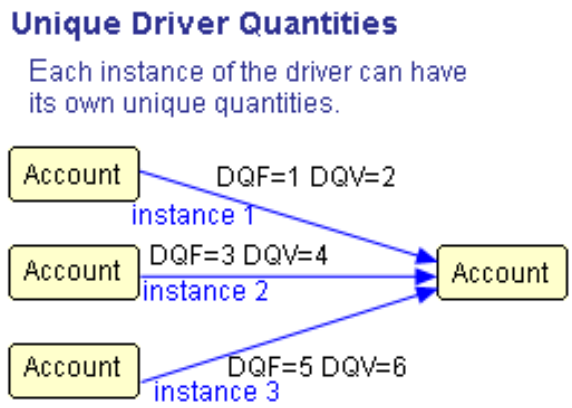
- "Driver Weight Fixed (DWF)" on page 768
- "Driver Weight Variable (DWV)" on page 768
- "Specify the Weights for a Weighted Driver" on page 424
- "Fixed Driver Quantities" on page 400
- "Variable Driver Quantities" on page 401

Unique and Non-Unique (Shared) Driver Quantities

Unique Driver Quantities

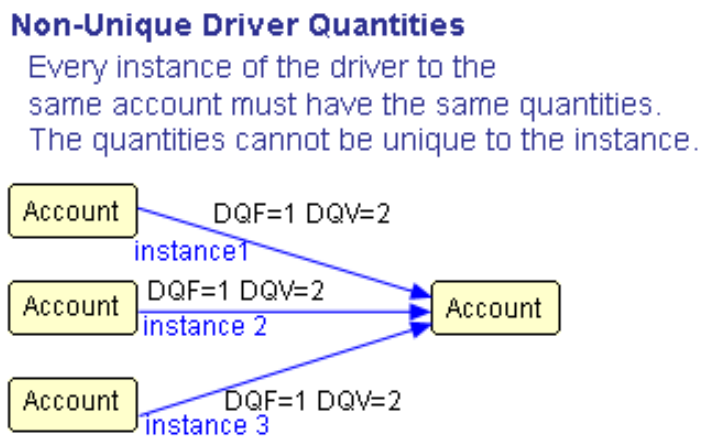
Driver quantities determine the amount of a cost to distribute from a source account to destination accounts. A driver quantity can be unique or non-unique.

A unique driver quantity is a quantity for a destination account that can differ for every assignment to that destination account through the same driver. For each account that receives costs, you must specify the driver quantity.



Non-Unique (Shared) Driver Quantities

A non-unique driver quantity is a quantity for a destination account that is identical for every assignment to that destination account through the same driver. You specify the driver quantity once for the destination account, and that driver quantity is applied to all assignments. If you determine that you must put the same driver quantity on multiple destination accounts using the same driver, then that driver probably needs to be non-unique.



Changing Driver Quantities between Unique and Non-Unique

You can change a driver quantity from unique to non-unique, or vice versa, at any time, even after driver quantities have been calculated. If you change a non-unique driver quantity to a unique driver quantity, the driver quantity is copied to each destination account. If you change a unique driver quantity to a non-unique driver quantity, the first driver quantity that is encountered is copied to each destination account.

Suppose that the activities Take Orders and Process Orders flow costs to the cost objects Simple Product and Complex Product, as shown in the following figure:

Display Name	Cost	DrvName
ACTIVITY (PRIMARY PA	$100.00	
New York	$100.00	
Take Orders	$76.34	Basic Non-Unique
Process Orders	$23.66	Basic Unique

IntsctnName	DQF
No<Customers> x Simple Product	7.00
No<Customers> x Complex Product	10.00

Display Name	Cost	DrvName
ACTIVITY (PRIMARY PA	$100.00	
New York	$100.00	
Take Orders	$76.34	Basic Non-Unique
Process Orders	$23.66	Basic Unique

IntsctnName	DQF
No<Customers> x Simple Product	1.00
No<Customers> x Complex Product	2.00

The driver for Take Orders is a basic, non-unique driver (named Basic Non-Unique). The driver for Process Orders is a basic, unique driver (named Basic Unique). Even though both activities flow costs to the same cost objects, the driver quantities for each cost object are different (the column DQF).

Now, suppose that the driver Basic Unique for the activity Process Orders is changed to the driver Basic Non-Unique. Because the driver quantities are now non-unique, the driver quantities specified for the cost objects in the assignment to Take Orders are copied to the assignment to activity Process Orders. This is shown in the following figure:

Display Name	Cost	DrvName
ACTIVITY (PRIMARY PA	$100.00	
New York	$100.00	
Take Orders	$76.34	Basic Non-Unique
Process Orders	$23.66	Basic Unique

IntsctnName	DQF
No<Customers> x Simple Product	7.00
No<Customers> x Complex Product	10.00

Display Name	Cost	DrvName
ACTIVITY (PRIMARY PA	$100.00	
New York	$100.00	
Take Orders	$76.34	Basic Non-Unique
Process Orders	$23.66	Basic Non-Unique

IntsctnName	DQF
No<Customers> x Simple Product	7.00
No<Customers> x Complex Product	10.00

In the previous example, a non-unique driver might be the best type because the number of orders processed for each product is probably the same number of orders taken for each product.

Driver Sequencing

Overview

Driver sequencing allows you define multiple calculation passes through assignments such that the driver quantity (DQF or DQV) of a sequenced driver (in a subsequent pass) is based on a cost resulting from drivers that execute in a previous pass.

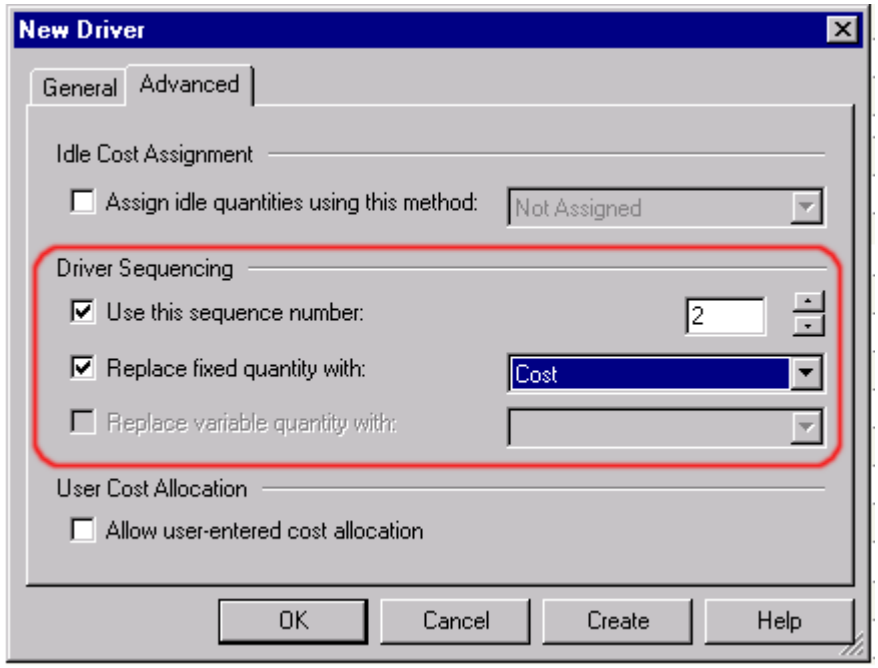
To create a sequenced driver:

1. Click the **Advanced** tab of the Driver Properties dialog and select Use this sequence number.
2. Select a sequence number greater than 1.

   By default, non-sequenced drivers have a sequence number of 1.

3. Select **Replace fixed quantity with** or **Replace variable quantity with** (or both) and select a cost from the drop down list.

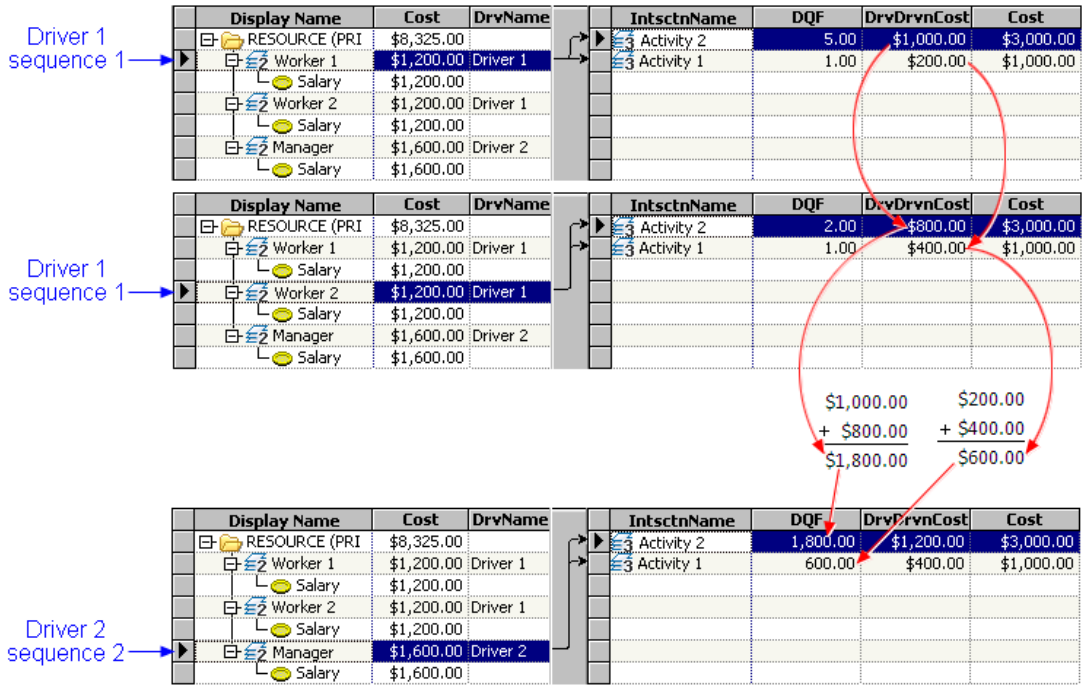
   The cost that you select is used as the Driver Quantify Fixed (DQF) or Driver Quantity Variable (DQV) for the driver.



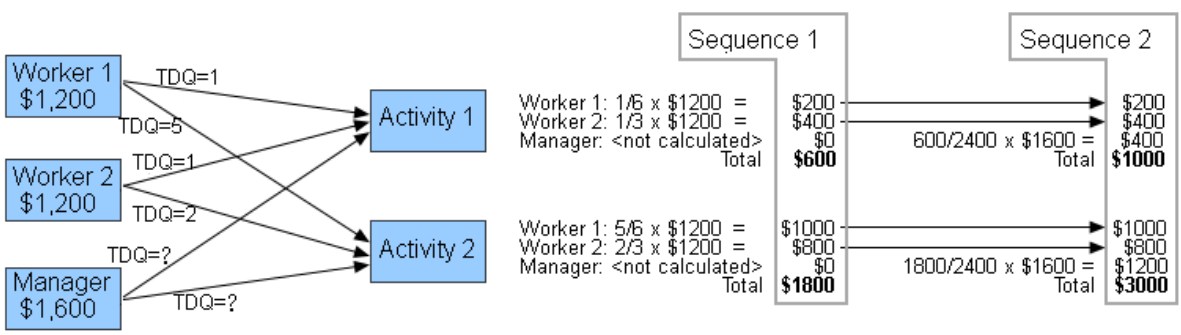
In the following example there are two workers and a manager. The two workers contribute a fixed portion (DQF) of their salary using a Basic driver to two activities, Activity 1 and Activity 2, as shown in the following table. By using a sequenced driver for the manager, you can base the manager's cost contribution to the two activities on the relative contributions of the two people that the manager manages.

	DQF Activity 1	DQF Activity 2
Worker 1	1	5
Worker 2	1	2
Manager	to be determined	to be determined

As the following picture shows, the driver quantities that are used for Driver 2 (the driver that executes in sequence 2 for the manager) are based on costs from the drivers that execute in sequence 1.



The following picture perhaps shows more clearly the processing sequence in the example under consideration. In a first pass, all drivers of sequence 1 are calculated. In this first pass the Manager's costs for Activity 1 and Activity 2 are calculated as $0. In a second pass, the Manager's costs are calculated based on the costs resulting from the first pass.



The following table summarizes the values shown in the preceding picture:

	Src. Cost (Salary)	DQF Activity 1	DQF Activity 2	Dst.Cost Activity 1	Dst.Cost Activity 2
Worker 1	$1200	1	5	$200	$1000
Worker 2	$1200	1	2	$400	$800
Manager	$1600	200+400=600	1000+800=1800	$400	$1200

Note: These overwritten values remain, even when you remove driver sequencing for the driver. To change the driver quantities, you must manually change them. Because of this potentially extensive impact on model data, SAS recommends that you define

a new driver for use with driver sequencing. You should not use an existing driver to experiment with driver sequencing.

Misconceptions about Driver Sequencing

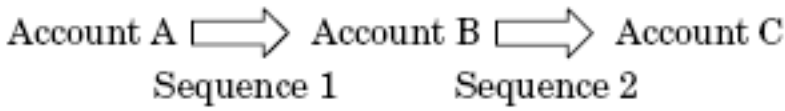
The following list presents common misconceptions about driver sequencing:

- Every driver needs a sequence number.

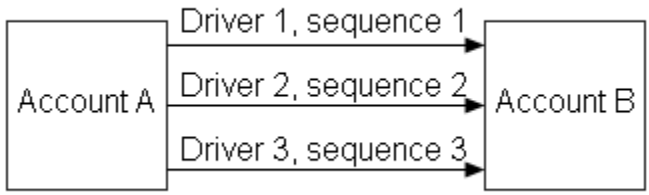
  Specify a sequence number for a driver only when the driver quantities on the destination accounts need to be replaced. This replacement is the only function of driver sequencing.

- Every driver in an assignment path needs a sequence number.

  Do not specify a driver sequence in an attempt to indicate the calculation sequence between accounts in an assignment path. For example in the following picture specifying a sequence is unnecessary. The driver between Account A and Account B will automatically run before the driver between Account B and Account C.



- A sequenced driver is not paired with any particular previous driver. The cost that replaces the DQF or DQV for a sequenced driver is the cost for an account at the time the sequenced driver runs, regardless of what drivers have contributed to that cost. For example in the following picture, the DQF for Driver 3 is not just the cost resulting from Driver 2 - it is the cost resulting both from Driver 1 and Driver 2. More generally, the cost of a driver in a sequence is the cost resulting from all drivers in a previous sequence.



Idle Quantities

Overview

Idle quantities are unused resources or unused time in a model. For example, if a machine can run for 10 hours a day, but is used for only eight hours a day, the idle quantity for that machine is two hours. You can assign the idle quantity to the destination accounts, as follows:

Display Name	IdlQty	TDQUE
ACTIVITY (PRIMARY P		
Production		
Stamping Parts	2.00	10.00

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00			$0.00
North America x Rear Fender	3.00			$0.00

The 10 hours that the machine can run is specified in the user-entered total driver quantity (the column TDQUE). The eight hours that the machine is used is specified in the fixed-driver quantities (the column DQF; five hours for the Front Fender and three hours for the Rear Fender). The two hours of idle quantity is calculated by the system (the column IdlQty; 10 TDQUE on the source account minus 8 DQF on the destination accounts).

You can specify how idle quantities are distributed for a driver. If you do not activate this feature, idle quantities are not displayed in an account's hierarchy, and idle quantities are not used when an account's cost is calculated.

To specify how idle quantities are distributed in an assignment, you can select one of the choices described below. The effect of each choice on the previous example is shown, based on the hourly cost of the machine (a press, in this example), which is $100.00.

User-Entered Quantities

If you choose to distribute idle quantities using user-entered quantities, you specify the idle quantities to be assigned to the idle components (the property idle quantity user-entered; the column IdlQtyUE). For example, you enter two hours for the Rear Fender.

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00			$0.00
North America x Rear Fender	3.00	2.00	2.00	$200.00

The system-generated value for the idle driver quantity (the column IdlDrvQty) corresponds to your entry. After calculating the costs, the driver idle cost (the column DrvIdlCost) is $200.00 (2.00 x $100.00).

User Proportion Quantities

If you choose to distribute idle quantities using user-proportioned quantities, you specify the proportions for the idle quantities (the property idle quantity user-entered; the column IdlQtyUE). For example, you enter 1 for the Front Fender and 3 for the Rear Fender.

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00	1.00	0.50	$50.00
North America x Rear Fender	3.00	3.00	1.50	$150.00

The total user-entered idle quantities is 4.00: 1.00 + 3.00. After calculating the costs, the idle driver quantity (the column IdlDrvQty) for the Front Fender is 0.50 (1/4 x 2.00), and the idle driver quantity for the Rear Fender is 1.50 (3/4 x 2.00).

Therefore, the driver idle cost (the column DrvIdlCost) for the Front Fender is $50.00 (0.50 x $100.00). The driver idle cost for the Rear Fender is $150.00 (1.50 x $100.00).

Use Driver Quantities

If you choose to distribute idle quantities with driver quantities, the ratio of the driver quantities is used for assigning the idle quantities. For example, the ratio of the driver quantity for the Front Fender is 5/8 (5.00/(5.00+3.00)). The ratio of the driver quantity for the Rear Fender is 3/8 (3.00/(5.00+3.00)).

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00		1.25	$125.00
North America x Rear Fender	3.00		0.75	$75.00

After calculating the costs, the idle driver quantity (the column IdlDrvQty) for the Front Fender is 1.25 (5/8 x 2.00), and the idle driver quantity for the Rear Fender is 0.75 (3/8 x 2.00).

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00		1.00	$100.00
North America x Rear Fender	3.00		1.00	$100.00

Therefore, the driver idle cost (column DrvIdlCost) for the Front Fender is $125.00 (1.25 x $100.00). The driver idle cost for the Rear Fender is $75.00 (0.75 x $100.00).

Evenly Assigned Quantities

If you choose to distribute idle quantities with evenly assigned quantities, the idle quantities are divided evenly among the destination accounts. For example, each of the destination account's idle quantity is 1.00, which is half of the source account's idle quantity of 2.00.

IntsctnName	DQF	IdlQtyUE	IdlDrvQty	DrvIdlCost
North America x Front Fender	5.00		1.00	$100.00
North America x Rear Fender	3.00		1.00	$100.00

After calculating the costs, the driver idle cost (the column DrvIdlCost) for both the Front Fender and the Rear Fender is $100.00 (1.00 × $100.00).

Example of Idle Flow

The following picture shows the four different methods of flowing an Idle Quantity from a source account to destination accounts. The four different methods are:

User Entered
The idle quantity is distributed as specified by the value of IdleQtyUE for each assignment path. Any idle quantity that is not distributed is assigned to the property Unassigned.

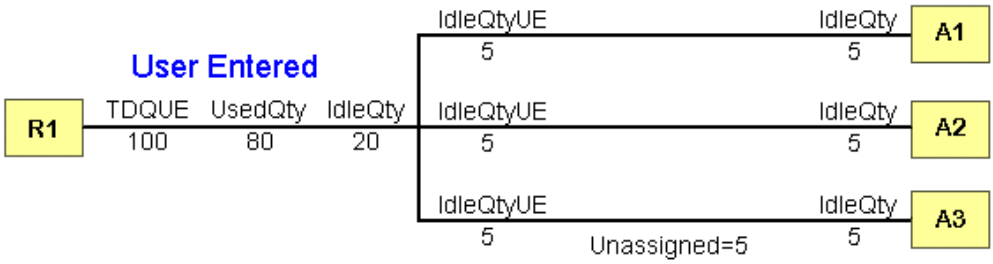
User Proportion
The idle quantity is distributed in the relative proportion of the IdleQtyUE of each assignment path.

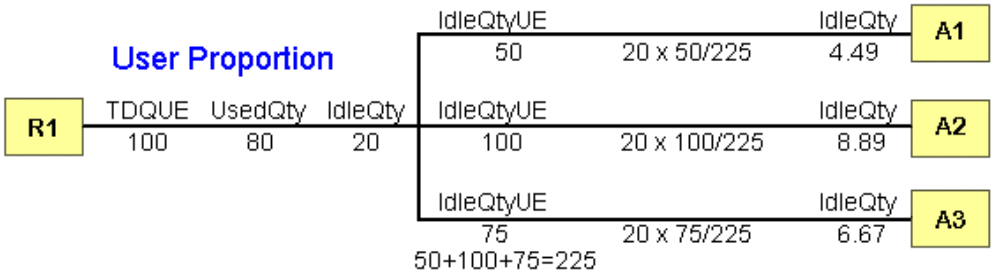
Driver Quantity
The idle quantity is distributed in the relative proportion of the DQF of each assignment path.

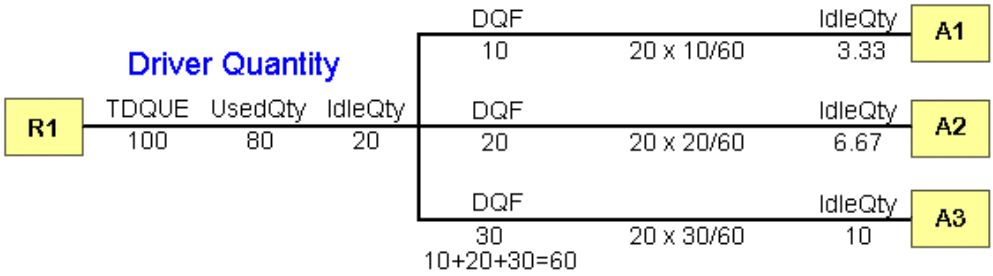
Evenly Assigned

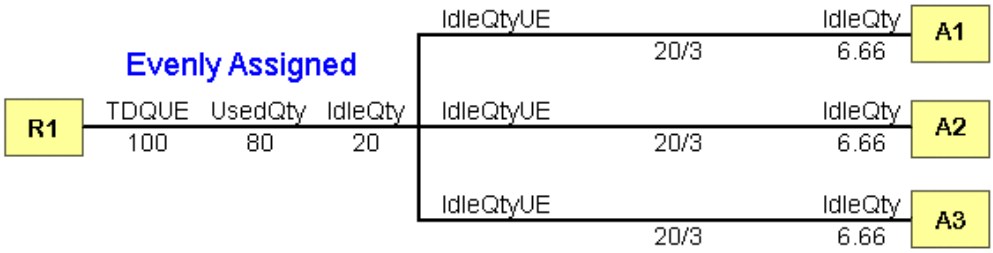
The idle quantity is distributed equally among the destination accounts.

Note: IdleQty = TDQUE - UsedQuantity









See Also

"Idle Quantities" on page 410

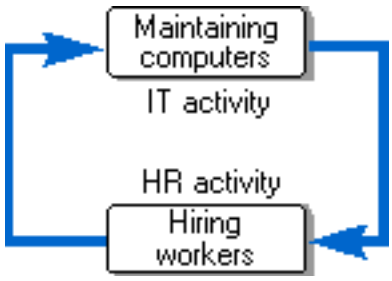
Reciprocal Costing

Overview

Reciprocal costing is a technique that enables costs to be shared between two or more accounts in the same module. With reciprocal costing, part of an account's cost flows to another account, and part of that account's cost flows back to the original account.

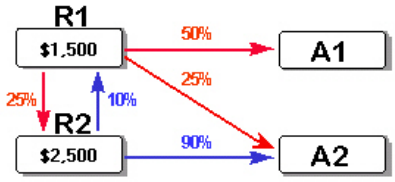
Note: In the Model Summary window, reciprocal cost assignments are referred to as cycles.

For example, suppose that part of the cost of the Information Technology (IT) department is based on how much time IT personnel spend maintaining the computers in the Human Resources (HR) department. Part of the cost of the HR department is based on how much time HR personnel spend hiring IT workers. These two activities are reciprocal accounts because they share costs.



Reciprocal Costs Calculation

Reciprocal costs are calculated using simultaneous equations. This means that the costs that are shared between accounts A and B are calculated at the same time, as shown below:



The equations:

R1 = 1500+.10B
R2 = 2500+.25A
A1 = .5A
A2 = .25A+.9B

The calculations:

R1 = 1500+.10(2500+.25A)
= 1500+250+.025A
.975A = 1750
R1 = 1750/.975 = 1794.87

R2 = 2500+.25(1794.87) = 2948.72

A1 = .5A = .5(1794.87) = 897.44

A2 = .25(1794.87)+.9(2948.72) = 3102.56

The results:

R1 = $1,794.87
R2 = $2,948.72
A1 = $897.44
A2 = $3,102.56

See Also

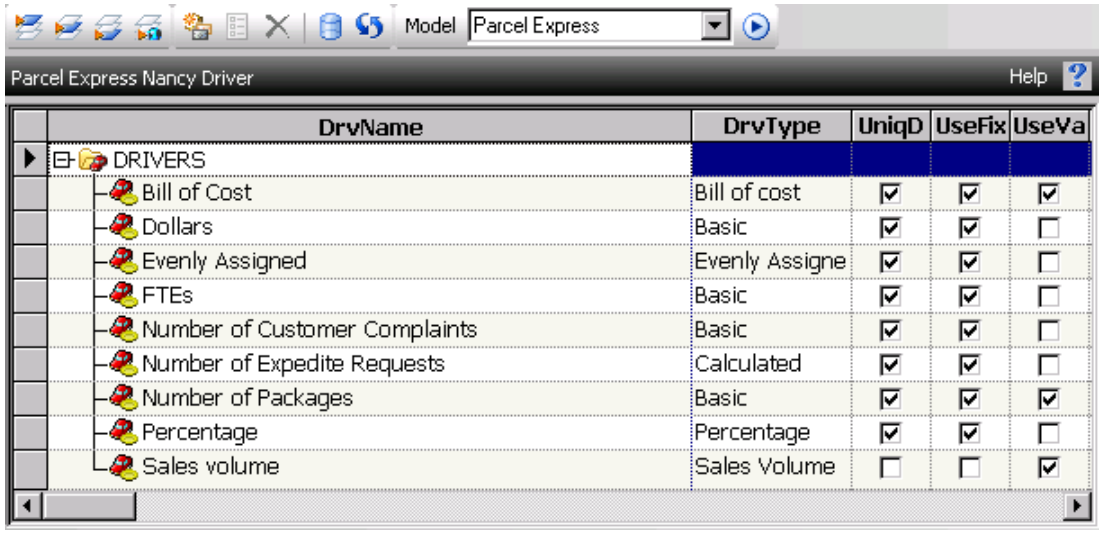
- "IsReciprocal (IsRecip)" on page 780
- "Reciprocal Id (RecipId)" on page 788

Chapter 46

Windows for Drivers

Drivers View

About the Drivers View



On the Drivers view, you can manage a model's drivers, the types of drivers , and the **driver quantities**.

Note: You cannot directly edit the information on the Drivers view.

How to Access the Drivers View

Open a model and select **Model** ⇨ **Drivers**.

See Also

- Chapter 40, "Introducing Drivers," on page 355
- "Create a Driver" on page 423
- "New Driver Dialog Box" on page 416
- "Driver Properties Dialog Box" on page 419

New Driver Dialog Box

About the New Driver Dialog Box

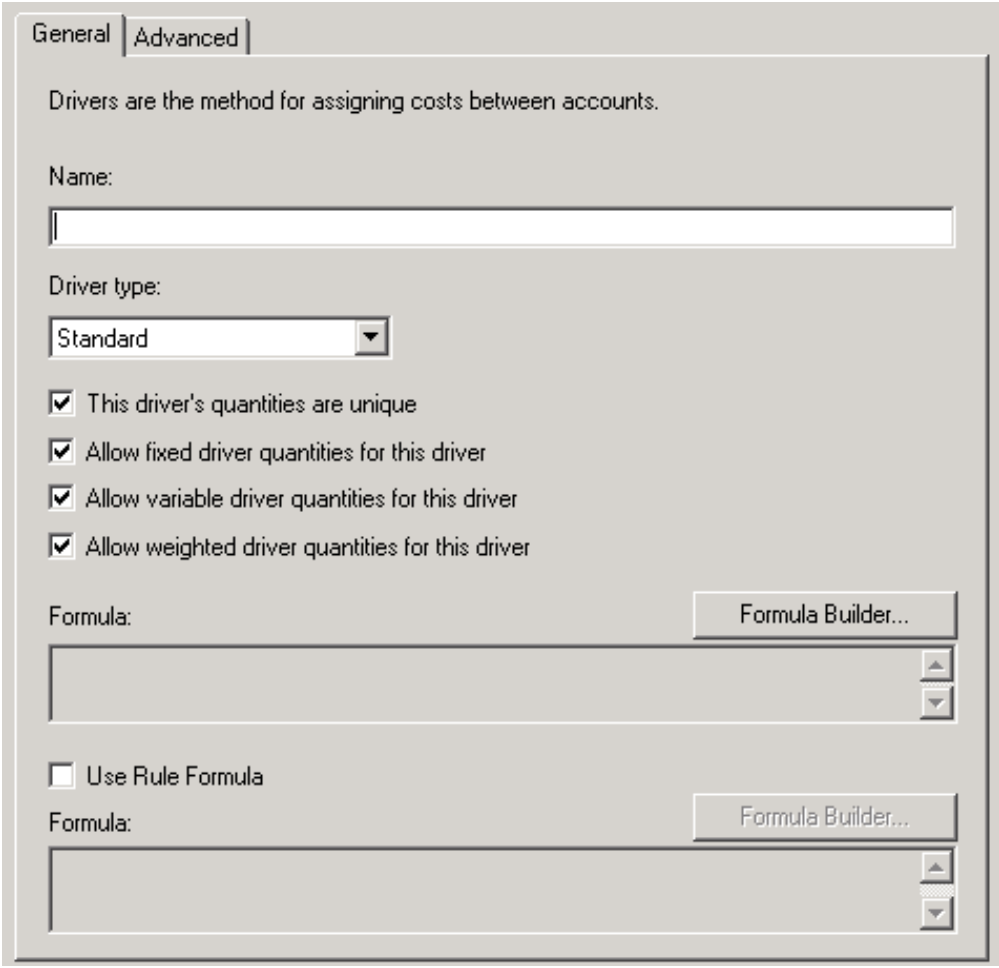
In the New Driver dialog box, you can name a new driver and you can specify other information about the driver.

Note: The availability of these features depends on your permissions.

How to Access the New Driver Dialog Box

On the Drivers view, select **Edit** ⇨ **New Driver**.

General Tab



Specify the following information:

Name
See "Driver naming conventions" on page 72.

Driver type
See Chapter 40, "Introducing Drivers," on page 355.

This driver's quantities are unique
Unique means that when this driver goes from multiple source accounts to the same destination account, then DQF, DQV, DWF, and DWV can have different values on the assignments to that destination account.

Non-Unique (also called "Shared" and "Destination-Specific") means that when this driver goes from multiple source accounts to the same destination account, then DQF, DQV, DWF, and DWV have the same value on all assignments to that destination account.

See "Unique and Non-Unique (Shared) Driver Quantities" on page 405.

Allow fixed driver quantities
DQF is user-editable on an assignment.

See "Fixed Driver Quantities" on page 400.

Allow variable driver quantities

DQV is user-editable on an assignment.

See "Variable Driver Quantities" on page 401.

Allow weighted quantities

DWF and DWV are user-editable on an assignment.

See "Driver Weights" on page 404.

Formula

This option is available for calculated drivers. See Chapter 44, "Calculated Driver," on page 393.

Click **Formula Builder** to define a formula. See "Formula Builder Dialog Box" on page 471.

The Formula Builder dialog box appears.

TIP You can also modify the formula in the Formula box.

Use Rule Formula

Select this option if you want the driver to be a rule-based driver. See "Make Assignments Using a Rule" on page 436.

Then, click **Formula Builder**. See "Formula Builder Dialog Box" on page 471.

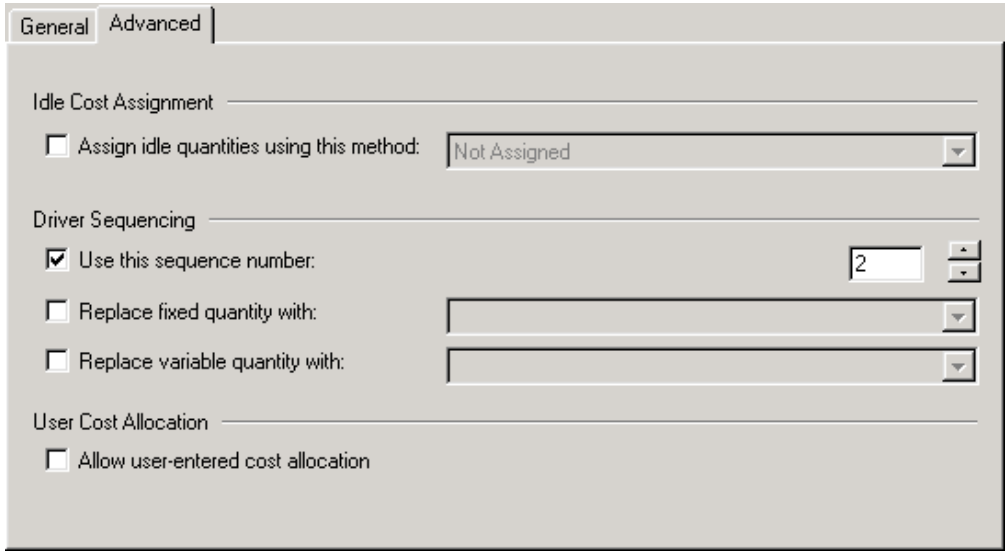
Note: If you create a formula for the rule and run calculate to create assignments, then you can subsequently uncheck **Use Rule Formula** to cause calculate not to create assignments based on the rule formula. The rule formula is retained but calculate ignores it. This is useful in allowing you to run calculate once to create the assignments, and then remove some of those assignments or add additiional ones. If you have done this, then you don't want calculate to restore the assignments that you removed or remove the assignments that you added. You can also turn off all rule formulae by selecting **Disable Driver Rules** during calculate.

Note: Selecting **Use Rule Formula** does not by itself cause assignments to be generated. It only allows you to define a rule formula. Assignments are generated only during calculate.

See Also

- "Make Assignments Using a Rule" on page 436
- Chapter 44, "Calculated Driver," on page 393
- "Formula Builder Dialog Box" on page 471

Advanced Tab



These choices are optional. The available options depend on the type of driver that is being created.

Assign idle quantities using this method
See "Idle Quantities" on page 410.

Use this sequence number
If you select this option, then specify a number in the box on the right. See "Driver Sequencing" on page 407.

Replace fixed quantity with
If you select this option, then select a numeric property from the menu on the right. See "Fixed Driver Quantity Override" on page 770. Also see "Fixed Driver Quantities" on page 400.

Note: This option is only available for Basic, Bill of Costs, and Weighted drivers.

Replace variable quantity with
If you select this option, then select a numeric property from the menu on the right. See "Variable Driver Quantity Override" on page 801.

Note: This option is only available for Basic, Bill of Costs, and Weighted drivers.

Allow user-entered cost allocation
See "Allocated Cost (AllocCost)" on page 747.

Driver Properties Dialog Box

About the Driver Properties Dialog Box

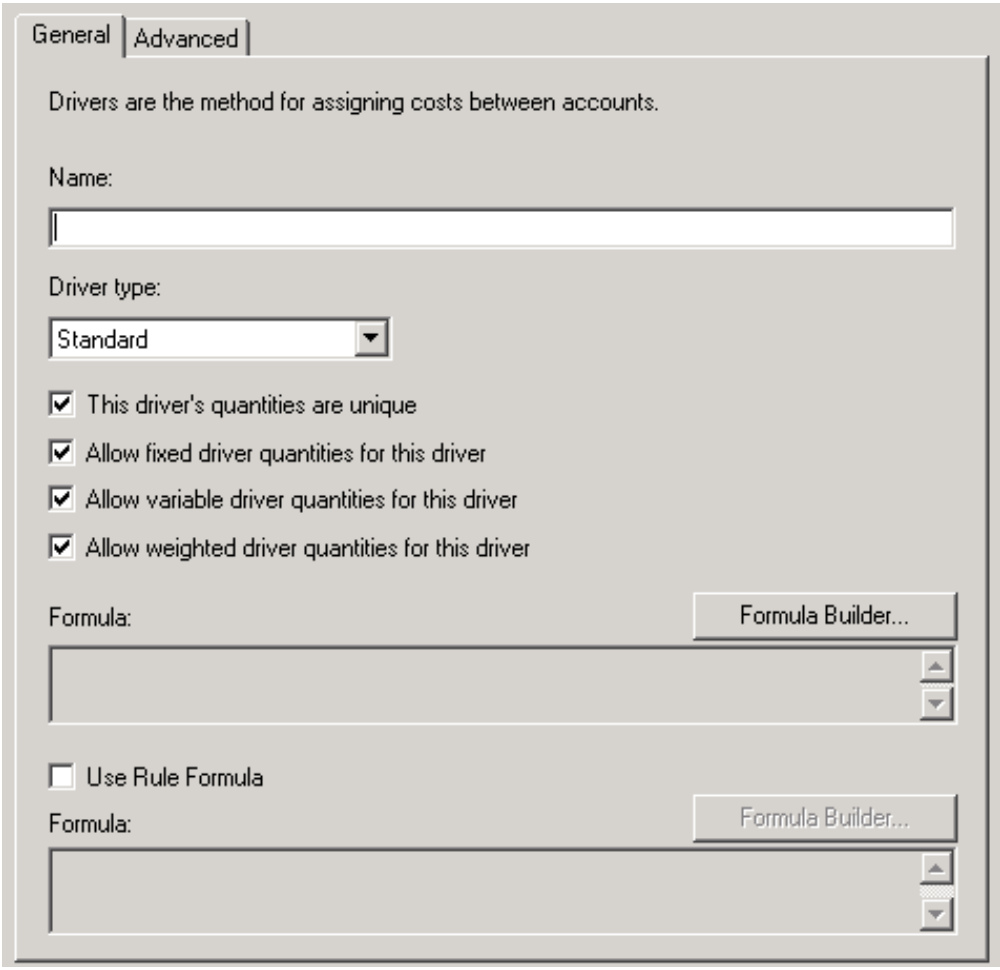
The availability of these features depends on your permissions.

In the Driver Properties dialog box, you can review or change the properties of a driver.

How to Access the Driver Properties Dialog Box.

On the Drivers view, select a driver and select **Edit** ⇨ **Item Properties**.

General Tab



Specify the following information:

Name
See "Driver naming conventions" on page 72.

Driver type
See Chapter 40, "Introducing Drivers," on page 355.

This driver's quantities are unique
Unique means that when this driver goes from multiple source accounts to the same destination account, then DQF, DQV, DWF, and DWV can have different values on the assignments to that destination account.

Non-Unique (also called "Shared" and "Destination-Specific") means that when this driver goes from multiple source accounts to the same destination account, then DQF, DQV, DWF, and DWV have the same value on all assignments to that destination account.

See "Unique and Non-Unique (Shared) Driver Quantities" on page 405.

Allow fixed driver quantities

DQF is user-editable on an assignment.

See "Fixed Driver Quantities" on page 400.

Allow variable driver quantities

DQV is user-editable on an assignment.

See "Variable Driver Quantities" on page 401.

Allow weighted quantities

DWF and DWV are user-editable on an assignment.

See "Driver Weights" on page 404.

Formula

This option is available for calculated drivers. See Chapter 44, "Calculated Driver," on page 393.

Click **Formula Builder** to define a formula. See "Formula Builder Dialog Box" on page 471.

The Formula Builder dialog box appears.

TIP You can also modify the formula in the Formula box.

Use Rule Formula

Select this option if you want the driver to be a rule-based driver. See "Make Assignments Using a Rule" on page 436.

Then, click **Formula Builder**. See "Formula Builder Dialog Box" on page 471.

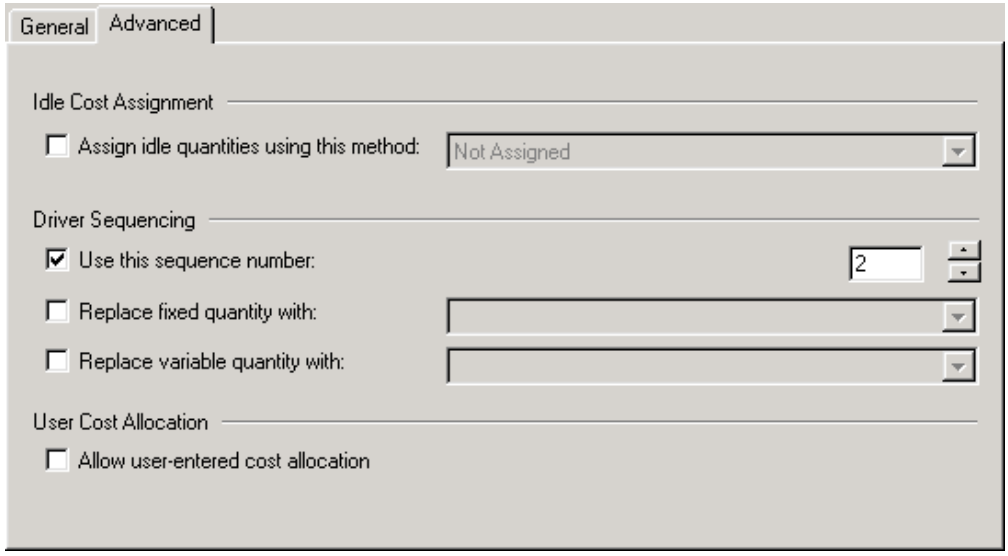
Note: If you create a formula for the rule and run calculate to create assignments, then you can subsequently uncheck **Use Rule Formula** to cause calculate not to create assignments based on the rule formula. The rule formula is retained but calculate ignores it. This is useful in allowing you to run calculate once to create the assignments, and then remove some of those assignments or add additiional ones. If you have done this, then you don't want calculate to restore the assignments that you removed or remove the assignments that you added. You can also turn off all rule formulae by selecting **Disable Driver Rules** during calculate.

Note: Selecting **Use Rule Formula** does not by itself cause assignments to be generated. It only allows you to define a rule formula. Assignments are generated only during calculate.

See Also

- "Make Assignments Using a Rule" on page 436
- Chapter 44, "Calculated Driver," on page 393
- "Formula Builder Dialog Box" on page 471

Advanced Tab



These choices are optional. The available options depend on the type of driver that is being created.

Assign idle quantities using this method
See "Idle Quantities" on page 410.

Use this sequence number
If you select this option, then specify a number in the box on the right. See "Driver Sequencing" on page 407.

Replace fixed quantity with
If you select this option, then select a numeric property from the menu on the right. See "Fixed Driver Quantity Override" on page 770. Also see "Fixed Driver Quantities" on page 400.

Note: This option is only available for Basic, Bill of Costs, and Weighted drivers.

Replace variable quantity with
If you select this option, then select a numeric property from the menu on the right. See "Variable Driver Quantity Override" on page 801.

Note: This option is only available for Basic, Bill of Costs, and Weighted drivers.

Allow user-entered cost allocation
See "Allocated Cost (AllocCost)" on page 747.

Chapter 47

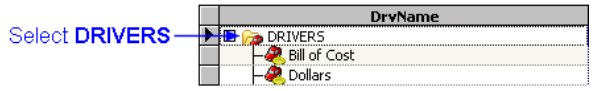
How To: Drivers

Create a Driver

1. Open a model and select **Model** ⇨ **Drivers Page**.

   The Drivers page appears.

2. Select the **DRIVERS** folder.



3. Select **Edit** ⇨ **New Driver**.

   The New Driver dialog opens

4. Select options on the **General** tab of the New Driver dialog. See "General Tab" on page 420.

5. Select options on the **Advanced** tab of the New Driver dialog. See "Advanced Tab" on page 422.

See Also

Chapter 40, "Introducing Drivers," on page 355

Create a Rule-Based Driver

See "Make Assignments Using a Rule" on page 436.

Specify the Weights for a Weighted Driver

1. Open the module that contains the destination accounts for the source account that uses a weighted driver.
2. For a weighted driver that uses fixed driver quantities, add a column to contain the property Driver Weight Fixed.
3. For a weighted driver that uses variable driver quantities, add a column to contain the property Driver Weight Variable.
4. Click in the Driver Weight Fixed (**DWF**) column or the Driver Weight Variable (**DWV**) column for a destination account, and type a value.

Display Name	Cost	DWF	DWV
ACTIVITY (PRIMARY PANE)	$9,000.00		
No<Organization>	$9,000.00		
Stamping Parts	$7,333.33	1.00	2.50
Chroming Parts	$1,000.00	3.00	5.00
Attaching Bumper	$666.67	2.00	4.00

If you are prevented from entering these values, verify that the following are correct:

- The source account's driver is a weighted driver.
- The weighted driver allows fixed driver quantities, variable driver quantities, or both.

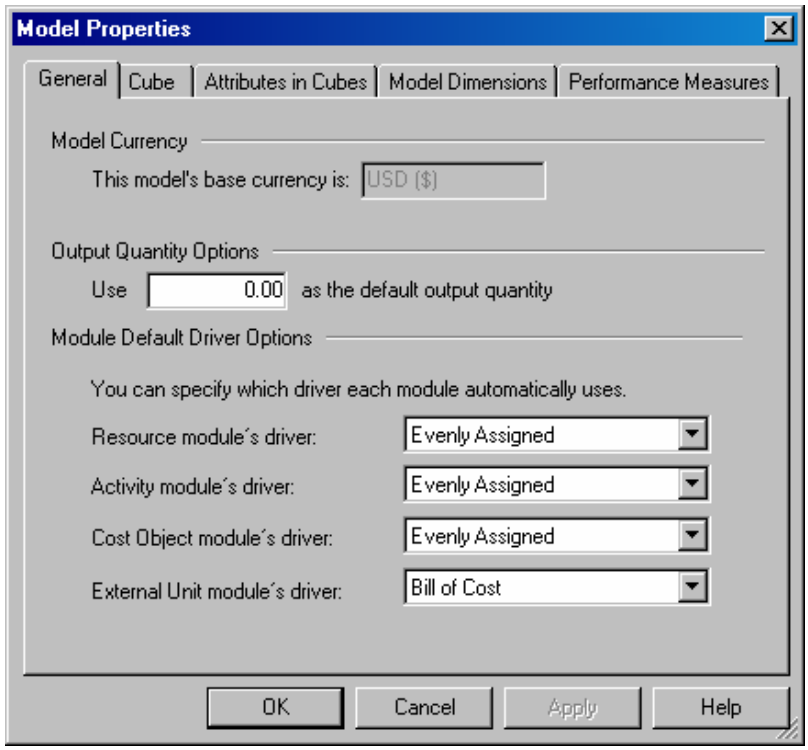
See Also

"Driver Weights" on page 404

Specify the Default Driver

1. Select **Model** ⇨ **Properties**.

   The Model Properties dialog box appears.



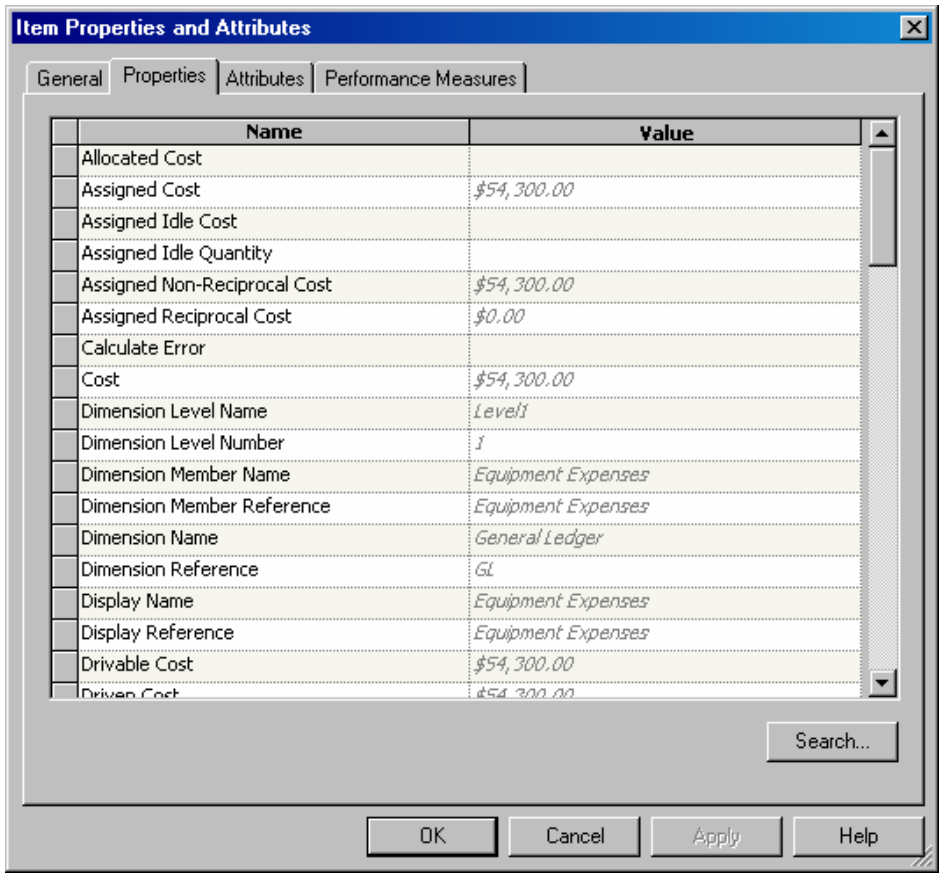
2. Click the **General** tab.
3. In the **Module Default Driver Options** section, select a default driver for each module.

   Note: If you delete a driver that is used, all accounts using the driver are flagged as using the module's default driver. If there is no default driver, then the account is flagged as having no driver (i.e. undefined). If you delete the default driver, then the default driver is set to "undefined".

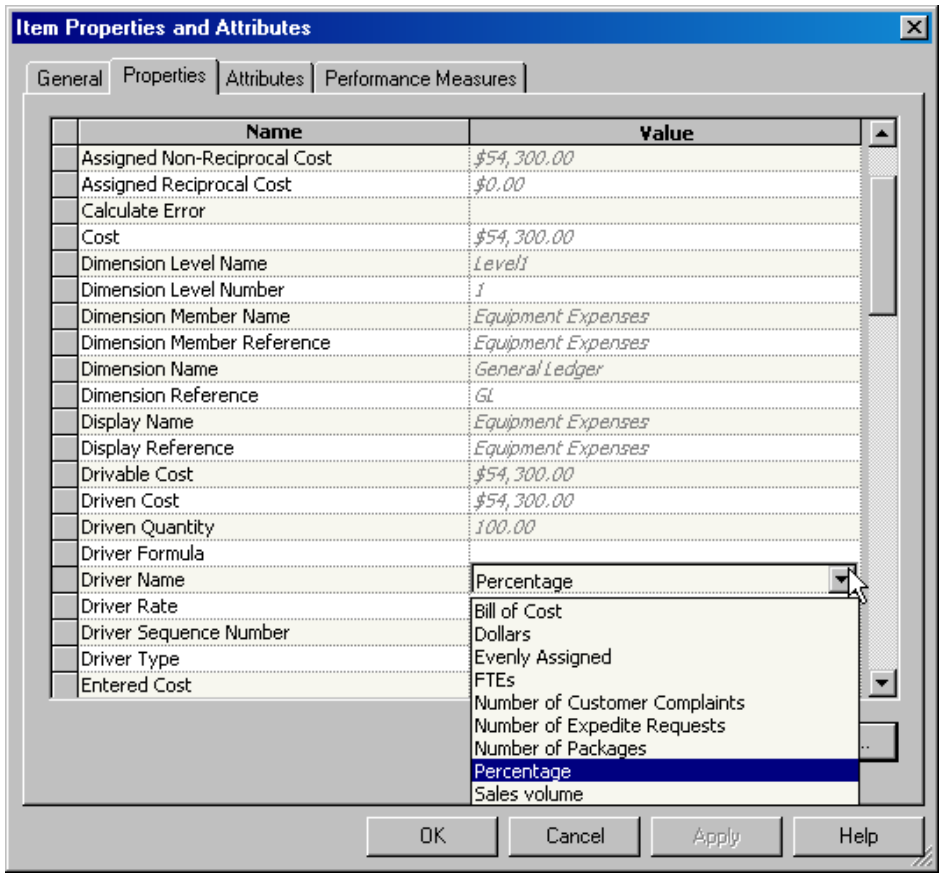
Associate a Driver Using an Account's Item Properties

1. In a module view, select an account.
2. Select **Edit** ⇨ **Item Properties**.

   The Item Properties dialog box appears.

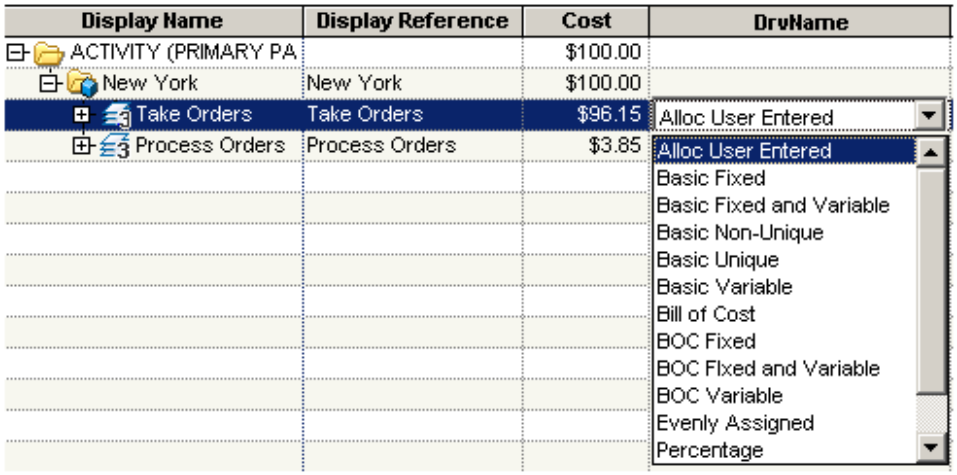


3. Click the **Properties** tab.
4. From the **Properties** list, locate the **Driver Name** property.
5. Click in the **Value** column and select a driver.



Associate a Driver Using a Module's Grid

1. In a module view, add a column to display the **Driver Name** property.
2. Click in the **DrvName** column and select a driver.



Change the Properties of a Driver

1. Open a model and select **Model** ⇨ **Drivers**
2. Select a driver.
3. Select **Edit** ⇨ **Item Properties**.

   The Driver Properties dialog box appears. See "Driver Properties Dialog Box" on page 419.

Part 14

Assignments

Chapter 48

Making Assignments

What Is an Assignment?

Overview

An assignment distributes costs between accounts using a driver. The amount of cost that flows is controlled by the driver. A single source account can make assignments to any number of destination accounts, but each source account can have only one driver.

Costs that are assigned between accounts in the same module are called inner-module assignments. Costs that are assigned between accounts in different modules are cross-module assignments.

Before creating an assignment, you must select accounts to add to the assignment.

Assignments are periodic elements. An assignment between two accounts in one period/scenario may or may not exist between the same accounts in another period/scenario. And, when an assignment exists between the same accounts in multiple periods/scenarios, its values in one period/scenario are independent of its values in another period/scenario.

See Also

- Chapter 40, “Introducing Drivers,” on page 355
- “Make an Assignment” on page 450

- "Make Assignments Using a Rule" on page 436

Assignment Paths

The sequence of accounts through which costs flow is called the assignment path. For example, suppose that costs are assigned from account A to account B. Then costs are assigned from account B to account C. This process creates an assignment path from account A to account C.

The following picture illustrates the rules for making assignments:

1. An account can receive assignments from any number of accounts (including zero) and can make assignments to any number of accounts (including zero).

   A rollup account can not make or receive assignments. See "Rollup Accounts and Module Rollups" on page 228.

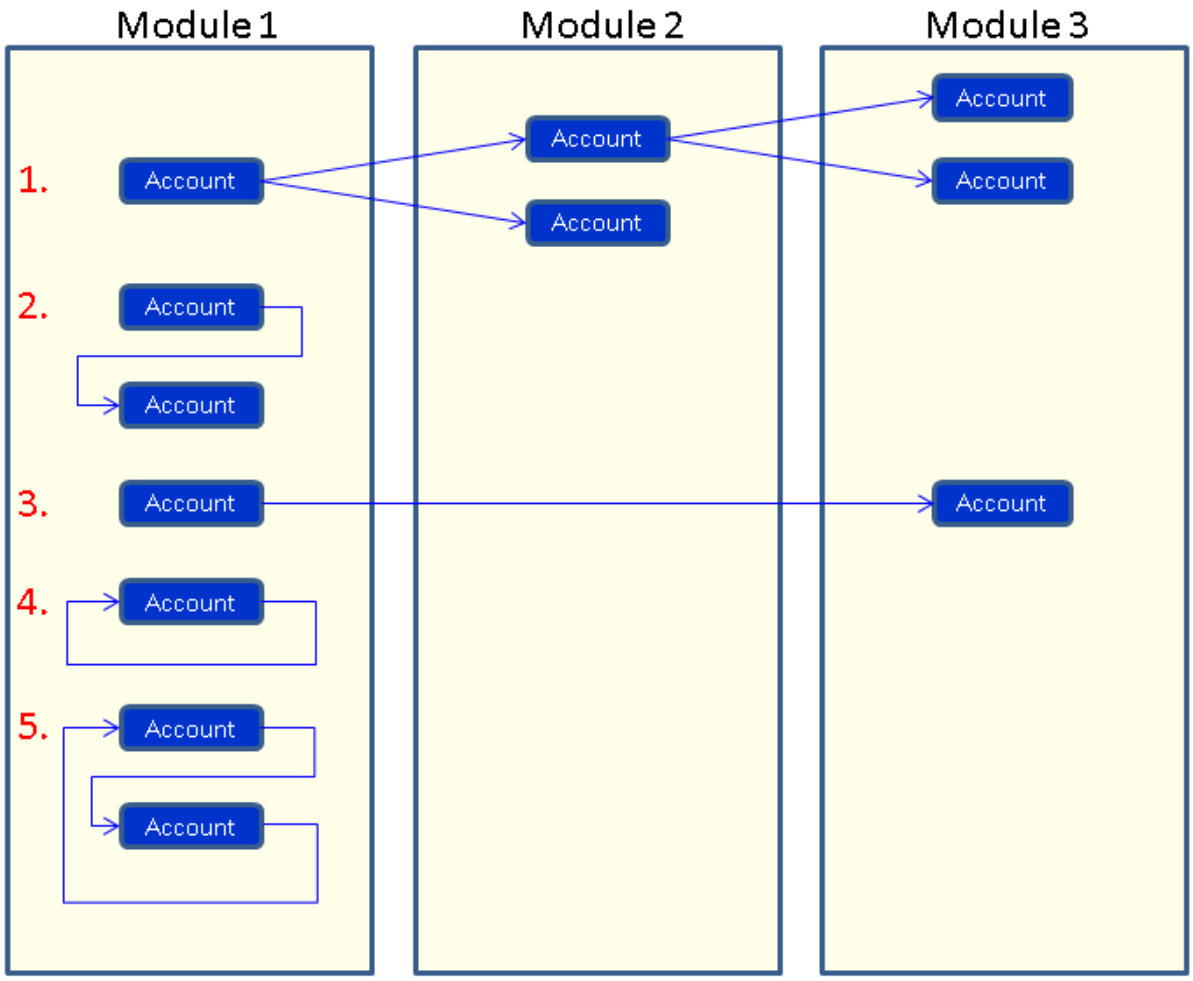
   Try to assign 100 percent of an account's costs to other accounts. If you do not assign all costs, the system issues warnings when you calculate costs.

2. An account can make an assignment to another account in the same module. However, an assignment can not go backwards—from an account in one module to an account in a previous module.

3. Assignments can skip modules—for example from an account in module 1 to an account in module 3.

4. An account can make an assignment to itself. See "Reciprocal Costing" on page 414.

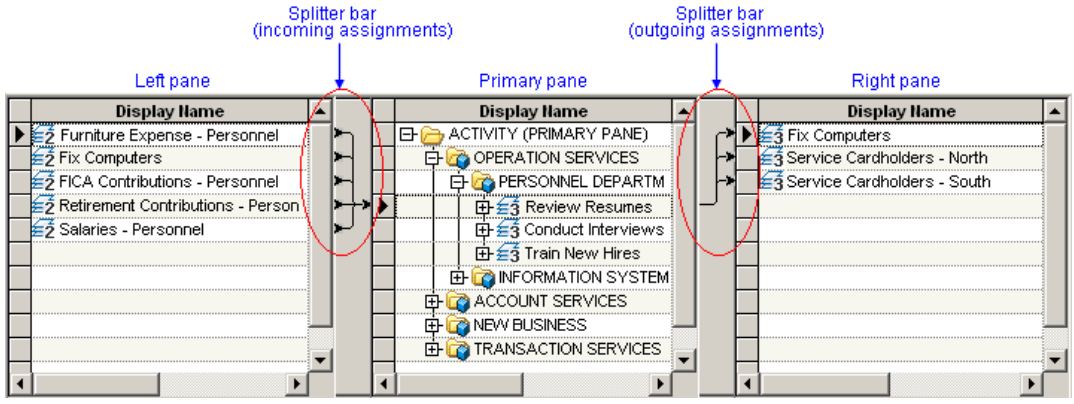
5. One account can indirectly make an assignment to itself—by assigning, directly or indirectly, to an account that makes an assignment to the one account.



Assignments Panes

Overview

You can view the accounts in a module using up to three window panes. This enables you to create assignments and to see the assignments between accounts.



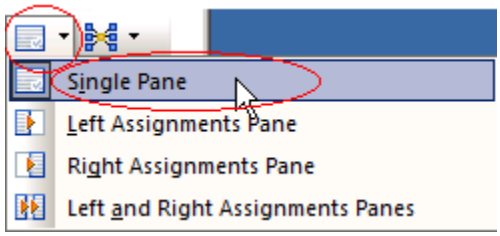
In both the two-pane view and the three-pane view, costs flow from the left to the right.

Primary Pane

The primary pane shows you one module at a time. The primary pane displays the hierarchy of a module—its accounts and rollup accounts (See Chapter 27, "What are Accounts?," on page 223.)

To show the primary pane, do one of the following:

- Select **Model** ⇨ **Assignments** ⇨ **Show Single Pane**.
- Click the View Assignments Pane icon and select **Single Pane**.



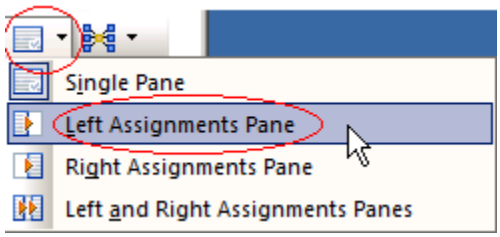
Left Assignments Pane

The left assignments pane shows all and only those accounts that make assignments to accounts in the primary pane..

Note: The accounts displayed in the left assignments pane are all and only those accounts that make assignments to accounts in the primary pane—regardless of which module the accounts in the left assignments pane belong to. For example, because an account can assign costs to another account in the same module, accounts displayed in the left assignments pane can belong to the same module as that displayed in the primary pane. Similarly, because an account can skip modules in assigning costs to another account, the accounts displayed in the left assignments pane can belong to modules considerably to the left of the module displayed in the primary pane.

To show the left assignments pane, do one of the following:

- Select **Model** ⇨ **Assignments** ⇨ **Left Assignments Pane**.
- Click the View Assignments Pane icon and select **Left Assignments Pane**.



Right Assignments Pane

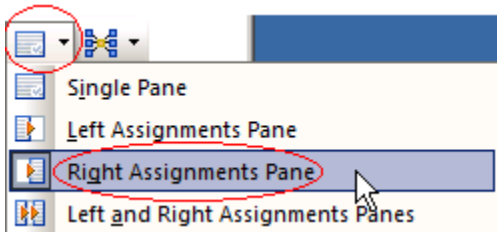
The right assignments pane shows all and only those accounts that receive assignments from accounts in the primary pane.

Note: The accounts displayed in the right assignments pane are all and only those accounts that receive assignments from accounts in the primary pane—regardless of which module the accounts in the right assignments pane belong to. For example, because an account can assign costs to another account in the same module, accounts displayed in the right assignments pane can belong to the same module as that displayed in the primary pane. Similarly, because an account can skip modules in assigning costs to another account, the accounts displayed in the right assignments

pane can belong to modules considerably to the right of the module displayed in the primary pane.

To show the right assignments pane, do one of the following:

- Select **Model** ⇨ **Assignments** ⇨ **Right Assignments Pane**.
- Click the View Assignments Pane icon and select **Right Assignments Pane**.

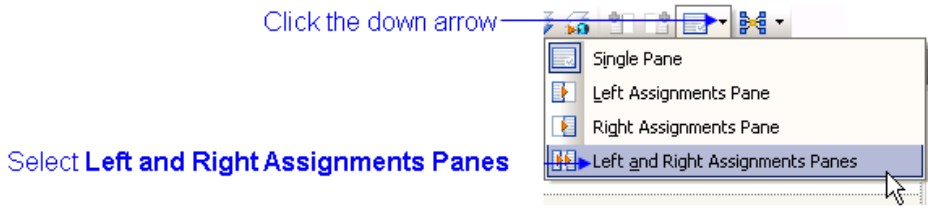


Splitter Bar

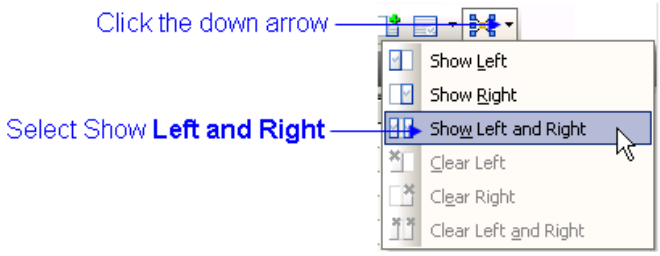
The space between the panes is called the splitter bar. Within the splitter bar, lines and arrows indicate the assignments between accounts. Costs flow from the left to the right.

To display both incoming and outgoing assignments in the splitter bar, do the following:

1. Select **Left and Right Assignments Panes** from the View Assignments Panes icon.



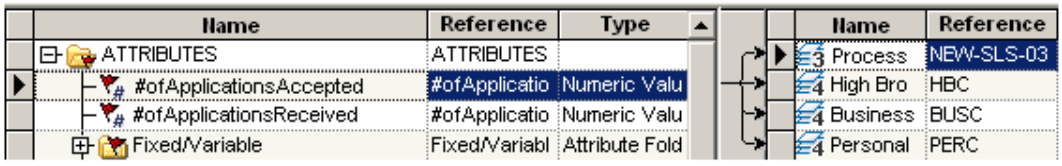
2. Select **Show Left and Right** from the Show Assignments icon.



TIP You can split the assignments panes simply by clicking the Show Left and Right Assignments Panes button  It is not necessary to pull down the menu.

If there are no assigned costs, you see nothing additional when you show assignments.

TIP You can also divide the Attributes view to see the accounts to which an attribute has been assigned. See "Attributes View" on page 296.



See Also

"Understanding Properties in a Column Layout" on page 324

Guidelines for Making Assignments

Before you create assignments, consider the following:

- You can assign costs from one account to another account, but you cannot assign costs to a roll-up account.
- Try to assign 100 percent of an account's costs to other accounts.

  If you do not assign 100 percent, SAS Cost and Profitability Management issues warnings when you calculate costs.

See Also

"What Is an Assignment?" on page 431

Make Assignments Using a Rule

About Rule-based Drivers

Rule-based drivers allow you to generate cost assignments automatically. A rule-based driver is an ordinary driver with an associated rule formula. The formula determines for which destination accounts an assignment is automatically generated from every source account to which the driver is attached.

Creating and using rule-based drivers is a four-step process:

1. **Attach a formula to a driver**.

   To attach a formula, on the Driver Properties dialog select **Use Rule Formula** and click **Formula Builder**.

   - Any sort of driver (evenly assigned, percentage, standard, calculated) can have a formula (referred to as a rule formula).
   - Use the Formula Builder to build a rule formula. The following is a sample formula:

     ```
     Module="CostObject" AND Destination.HasAttribute("SuppHours") AND SuppHours > 5
     ```

     This sample formula returns true for every destination account in the CostObject module that has an attribute "SuppHours" whose value is greater than 5.

2. **Attach the driver to one or more source accounts**.

   The source accounts can be in any module.

3. **Calculate the model**.

During calculation every source account that has a rule-based driver (a driver with an associated rule formula) attached to it starts the following process:

- Every destination account that is downstream of the source account is evaluated for the rule formula. (A destination account is downstream of a source account if it is in the same module as the source account or if it is in a later module when the modules are considered in the order in which you defined the modules, or in the following order in the case of a traditional activity-based costing model: External Unit, Resource, Activity, Cost Object).
- If the rule formula tests true for the destination account, then an assignment is made between the source account being considered and that destination account.
- If the rule formula tests false for the destination account and there is currently an assignment between the source account being considered and that destination account, then the assignment is deleted.
- If the rule formula tests false for the destination account and there is currently no assignment between the source account being considered and that destination account, then no assignment is made.

4. **`Clean up the assignments made`**.

   This step is optional and is recommended only for rapid prototyping.

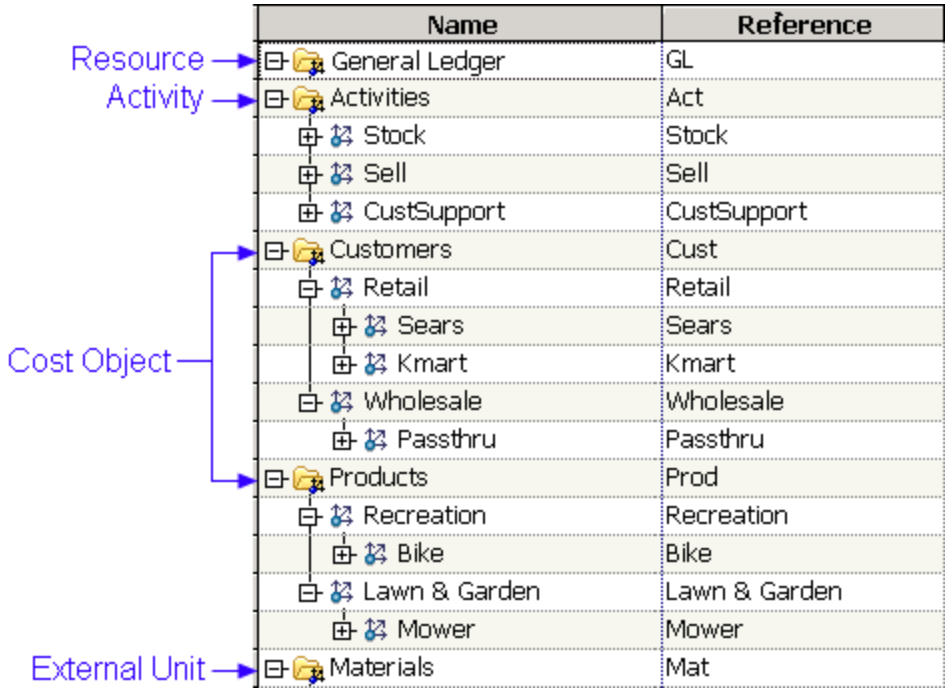
   - Manually delete any assignments that were created that you don't want.
   - Manually create any additional assignments that you do want between a source account with a rule-based driver and a destination account for which the rule formula tested false.

   Note: If you do this sort of manual clean-up, then you should be aware of the fact that running a subsequent calculation will restore the assignments that you deleted and delete the additional assignments that you made. You can avoid this in either of two ways:

   - Selecting Disable Driver Rules during calculation to disable all rule-based drivers.
   - Deselecting Use Rule Formula in the Driver Properties dialog box to disable the rule formula for an individual driver.

Sample Rule-based Drivers

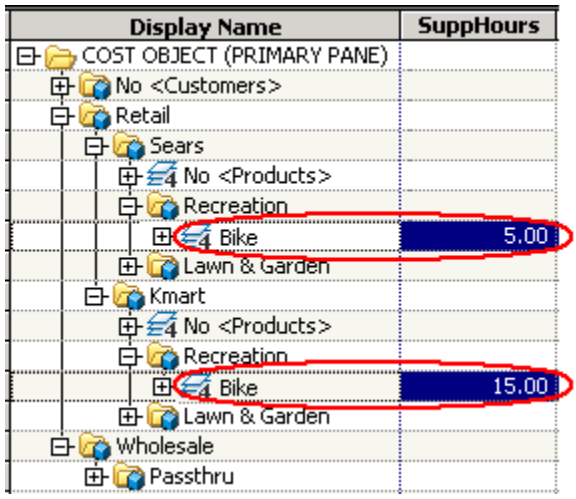
Suppose we have a model with the following dimensions:



Suppose also that we have a numeric attribute named SuppHours:

Name	Reference	Type	UoM
ATTRIBUTES	ATTRIBUTES		
SuppHours	SuppHours	Numeric	hours

The attribute is attached to two Cost Object accounts:

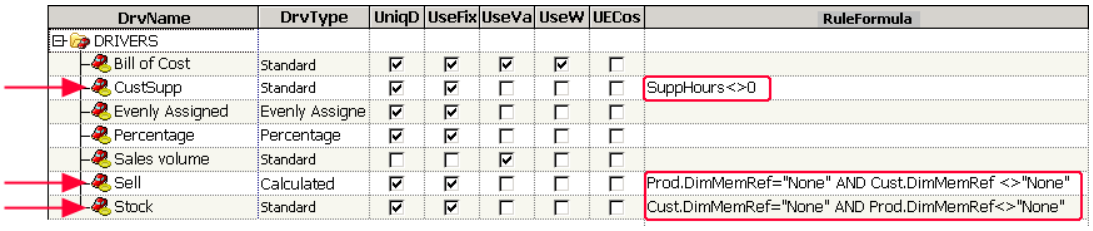


Suppose, finally, that we have defined three rule-based drivers with the rule formulas shown in the following table:

Driver Name	Driver Type	Rule Formula
CustSupp	Standard	`SuppHours<>0`

Driver Name	Driver Type	Rule Formula
Sell	Calculated	`Prod.DimMemRef="None" AND Cust.DimMemRef<>"N` *Note:* The coding of formulas is explained more fully in "Formula Context" on page 459.
Stock	Standard	`Cust.DimMemRef="None" AND Prod.DimMemRef<>"N` *Note:* The coding of formulas is explained more fully in "Formula Context" on page 459.

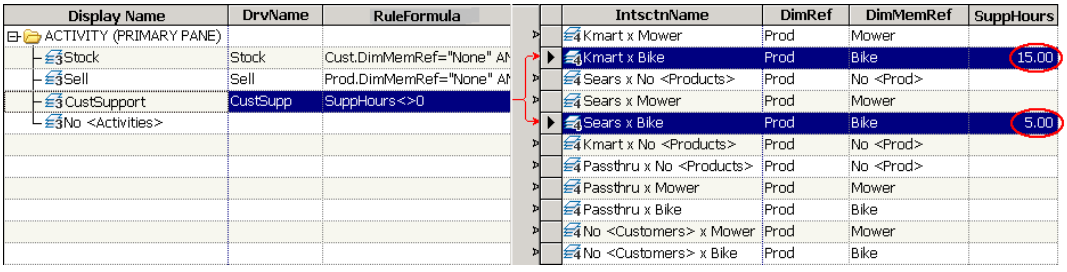
Notice that a rule-based driver can be of any type (evenly assigned, percentage, standard, calculated). So, a calculated rule-based driver has two formulas: a rule formula to determine assignments and a driver formula to determine the calculation for each assignment. The following picture shows drivers in our sample model:



DrvName	DrvType	UniqD	UseFix	UseVa	UseW	UECos	RuleFormula
DRIVERS							
Bill of Cost	Standard	☑	☑	☑	☑	☐	
CustSupp	Standard	☑	☑	☐	☐	☐	SuppHours<>0
Evenly Assigned	Evenly Assigne	☑	☑	☐	☐	☐	
Percentage	Percentage	☑	☑	☐	☐	☐	
Sales volume	Standard	☐	☐	☑	☐	☐	
Sell	Calculated	☑	☑	☐	☐	☐	Prod.DimMemRef="None" AND Cust.DimMemRef <>"None"
Stock	Standard	☑	☑	☐	☐	☐	Cust.DimMemRef="None" AND Prod.DimMemRef<>"None"

The following picture shows the two assignments that are generated by the driver CustSupp whose rule formula is: **`SuppHours<>0`**:

- Kmart x Bike (for which account SuppHours=15)
- Sears x Bike (for which account SuppHours=5)



Display Name	DrvName	RuleFormula
ACTIVITY (PRIMARY PANE)		
Stock	Stock	Cust.DimMemRef="None" AN
Sell	Sell	Prod.DimMemRef="None" AN
CustSupport	CustSupp	SuppHours<>0
No <Activities>		

IntsctnName	DimRef	DimMemRef	SuppHours
Kmart x Mower	Prod	Mower	
Kmart x Bike	Prod	Bike	15.00
Sears x No <Products>	Prod	No <Prod>	
Sears x Mower	Prod	Mower	
Sears x Bike	Prod	Bike	5.00
Kmart x No <Products>	Prod	No <Prod>	
Passthru x No <Products>	Prod	No <Prod>	
Passthru x Mower	Prod	Mower	
Passthru x Bike	Prod	Bike	
No <Customers> x Mower	Prod	Mower	
No <Customers> x Bike	Prod	Bike	

The following picture shows the two assignments that are generated by the driver Stock, whose rule formula is: **`Cust.DimMemRef="None" AND Prod.DimMemRef<>"None"`**

- No <Customers> x Mower
- No <Customers> x Bike

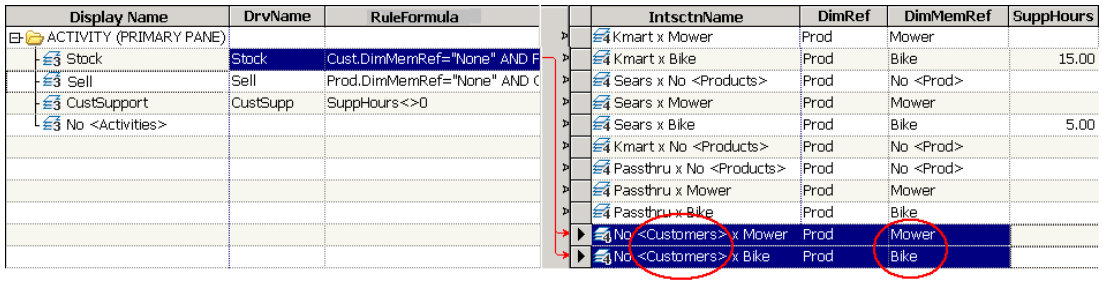
It is important to note that even though the column for the intersection name shows "No <Customers>" in the following picture, you must use the internal name "None" in your formulas.

CAUTION!

Because “None” is localizable, you must use the value of “None” for your locale. Following is a list of possible values according to locale:

German	Kein
English	None
Spanish	Ninguno
French	Néant
Italian	Nessuno
Japanese	なし
Korean	없음
Dutch	Geen
Portuguese (Brazil)	Nenhum
Chinese Simplified	无
Chinese Traditional (Hong Kong SAR)	無
Chinese Traditional (Taiwan)	無

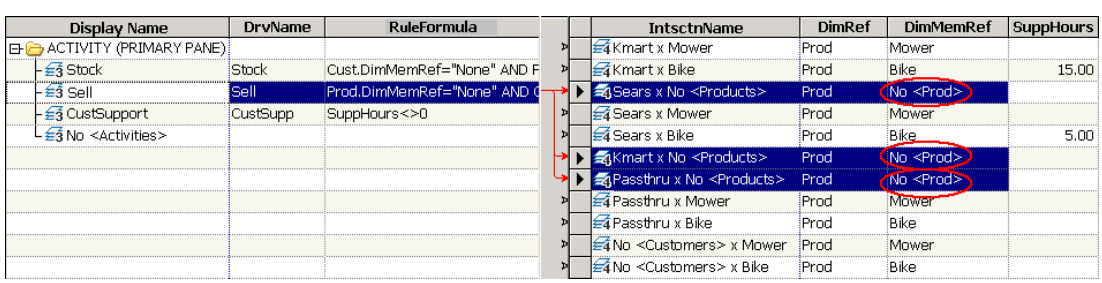
Note: Instead of using a localizable constant, you can use the IsNone() function, which will work in any locale. See “IsNone function” on page 481.



The following picture shows the three assignments that are generated by the driver Sell whose rule formula is: **`Prod.DimMemRef="None" AND Cust.DimMemRef<>"None"`**

- Sears x No <Products>
- Kmart x No <Products>
- Passthru x No <Products>

Note: Again, note that the formula uses "None“ for the null intersection even though the column heading in the assignments pane shows "No <Prod>".



Two More Rule-Based Drivers

Now let's introduce two more rule-based drivers to create assignments from Cost Object accounts to other Cost Object accounts:

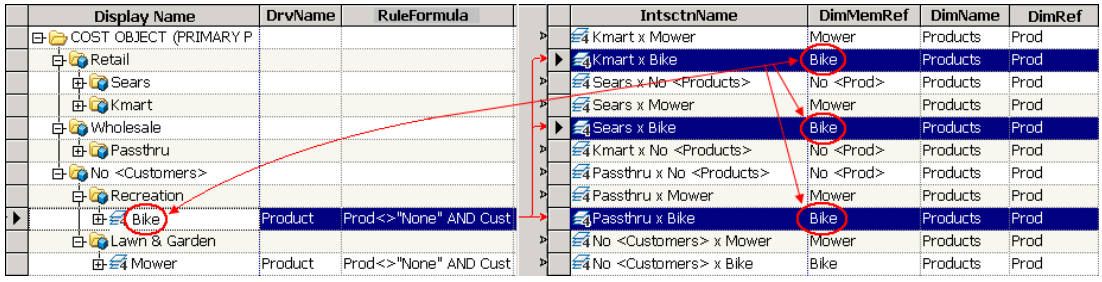
Driver Name	Driver Formula

Product	`Prod.DimMemRef<>"None"`
	`AND Cust.DimMemRef<>"None"`
	`AND Source.Prod.DimMemRef=Destination.Prod.DimMemRef`
	`AND Source.Cust.DimMemRef="None"`

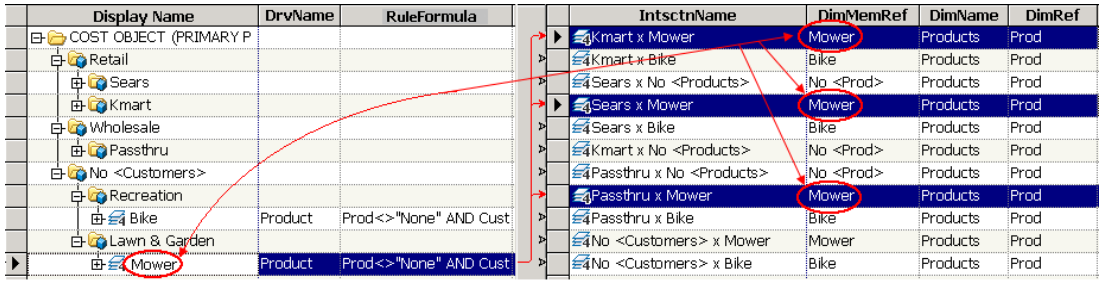
Customer	`Prod.DimMemRef<>"None"`
	`AND Cust.DimMemRef<>"None"`
	`AND Source.Prod.DimMemRef="None"`
	`AND Source.Cust.DimMemRef=Destination.Cust.DimMemRef`

The following picture shows three of the assignments that are generated by the driver Product, whose rule formula is the following:

```
Prod.DimMemRef<>"None"
AND Cust.DimMemRef<>"None"
AND Source.Prod.DimMemRef=Destination.Prod.DimMemRef
AND Source.Cust.DimMemRef="None"
```



And, the following picture shows three more assignments that are generated by the driver Product:



The following picture shows two of the assignments that are generated by the driver Customer, whose rule formula is the following:

```
Prod.DimMemRef<>"None"
AND Cust.DimMemRef<>"None"
AND Source.Prod.DimMemRef="None"
AND Source.Cust.DimMemRef=Destination.Cust.DimMemRef
```

Display Name	DrvName	RuleFormula
COST OBJECT (PRIMARY P		
Retail		
Sears		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
Kmart		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
Wholesale		
Passthru		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
No <Customers>		

Customer

IntsctnName	DimMemRef	DimName	DimRef
Kmart x Mower	Mower	Products	Prod
Kmart x Bike	Bike	Products	Prod
Sears x No <Products>	No <Prod>	Products	Prod
Sears x Mower	Mower	Products	Prod
Sears x Bike	Bike	Products	Prod
Kmart x No <Products>	No <Prod>	Products	Prod
Passthru x No <Products>	No <Prod>	Products	Prod
Passthru x Mower	Mower	Products	Prod
Passthru x Bike	Bike	Products	Prod
No <Customers> x Mower	Mower	Products	Prod
No <Customers> x Bike	Bike	Products	Prod

The following picture shows two more assignments that are generated by the driver Customer, whose rule formula is the following:

```
Prod.DimMemRef<>"None"
AND Cust.DimMemRef<>"None"
AND Source.Prod.DimMemRef="None"
AND Source.Cust.DimMemRef=Destination.Cust.DimMemRef
```

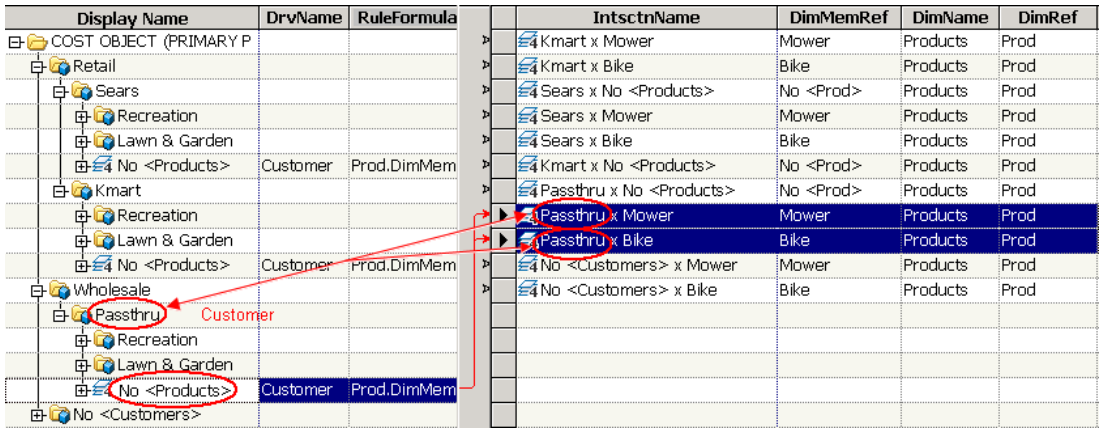
Display Name	DrvName	RuleFormula
COST OBJECT (PRIMARY P		
Retail		
Sears		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
Kmart		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
Wholesale		
Passthru		
Recreation		
Lawn & Garden		
No <Products>	Customer	Prod.DimMem
No <Customers>		

Customer

IntsctnName	DimMemRef	DimName	DimRef
Kmart x Mower	Mower	Products	Prod
Kmart x Bike	Bike	Products	Prod
Sears x No <Products>	No <Prod>	Products	Prod
Sears x Mower	Mower	Products	Prod
Sears x Bike	Bike	Products	Prod
Kmart x No <Products>	No <Prod>	Products	Prod
Passthru x No <Products>	No <Prod>	Products	Prod
Passthru x Mower	Mower	Products	Prod
Passthru x Bike	Bike	Products	Prod
No <Customers> x Mower	Mower	Products	Prod
No <Customers> x Bike	Bike	Products	Prod

And, the following picture shows the last two assignments that are generated by the driver Customer, whose rule formula is the following:

```
Prod.DimMemRef<>"None"
AND Cust.DimMemRef<>"None"
AND Source.Prod.DimMemRef="None"
AND Source.Cust.DimMemRef=Destination.Cust.DimMemRef
```



Additional Rule Formulas

The following are some examples of formulas that you can use in a rule-based driver. For a full list of functions, see Functions.

Rule Formula:	Explanation:
`Match( Name, "*Salary" )`	The name of the destination account matches "*Salary" e.g., "Wages & Salary".
`TDQ > 0 and TDQ <= 100`	The TDQ of the destination account is between 0 and 100.
`Cost > 0 and Source.Cost < 0`	The cost of the destination account is greater than 0 and the cost of the source account is less than 0.
`(NumberOfCalls > 0) OR (NumberOfReturns >`	The value of the attribute, NumberOfCalls, on the destination account is greater than 0, or the value of the attribute NumerOfReturns, also on the destination account, is greater than 0.
`if(not IsNull(SoldQuantity), SoldQuantity, OutputQuantity) > 0`	The condition is evaluated as follows: 1. If SoldQuantity is not null, then SoldQuantity is returned. 2. If SoldQuantity is null, then OutputQuantity is returned. 3. If whichever quantity is returned (SoldQuantity or OutputQuantity) is greater than 0, then the condition is true and an assignment is made.

Limitations

- Properties on the source account related to outgoing assignments will be accessible but are not recommended for use in rule formulas. In all cases these will be zero because the outgoing assignments haven't been created yet. These properties are

TotalDriverQuantity, IdleQuantity, AssignedIdleQuantity, OutputQuantity, UsedQuantity, TotalDriverQuantityCalculated, TotalDriverQuantityBasic, DrivenQuantity, DrivableCost, IdleCost, AssignedIdleCost, UsedCost, AllocatedCost, DrivenCost, UnassignedCost, AssignedCost, AssignedReciprocalCost, AssignedNonReciprocalCost.

- A syntax error occurs if a rule formula uses an assignment property since they do not exist in the context of a rule evaluation. These properties are: AllocatedCost, Cost, DrivenCost, DriverQuantityCalculated, DriverQuantityFixed, DriverQuantityVariable, DriverRate, DriverWeightFixed, DriverWeightVariable, IdleCost, IdleDriverQuantityUE, UsedCost.
- If the **Use Rule Formula** checkbox is checked for a driver but no rule formula is defined, then all outgoing assignments are deleted from accounts that use that driver. If you want to retain existing assignments make sure the **Use Rule Formula** checkbox is unchecked when the rule formula is empty.
- No warning is issued if your formula refers to dimensions that exist but in an order that does not exist. For example, not warning is issued if your formula attempts to assign from an Activity module account to a Resource module account. Such an assignment does not exist because assignments cannot go in a backwards direction (e.g., from Activity module to Resource module).
- Assignments are not generated when you define a rule formula or when you import a model with rule-based drivers. Assignments are generated only when you calculate a model.
- When a field contains space between two words, always enclose the field in double quotation marks. For example, write `Intersection.Match("Product Type", Product, Cust)`.

Recommendations

Please be aware of the following considerations and recommendations in using rule-based drivers:

Do Not Create Unnecessary Assignments

The ease of creating assignments with rule-based drivers can lead to generating millions of unnecessary assignments. Unnecessary assignments both use up memory and increase calculate time. The more accounts there are that use rules-based drivers the more work there is for calculation to do beyond flowing costs.

One way to avoid unnecessary assignments is to include a check in the rule formula to eliminate assignments with zero cost flow. For example, suppose you have a calculated driver that uses the attribute NumberOfCalls to calculate cost. You could include in a rule formula the condition NumberOfCalls > 0 to avoid making assignments with zero cost flow.

Numeric Functions Are Faster Than String Functions

Formulas that use string concatenation and functions that return strings are slower to evaluate than numeric or boolean expressions. Plan accordingly when using them in rule formulas or in calculated attributes that are referenced by rule formulas.

Be Wary of Creating Reciprocal Assignments

Remember that rule-based drivers can result in the creation of assignments from accounts in one module to destination accounts in the same module. As a result you can end up with unintended reciprocal costs. You can avoid reciprocal costs by specifying the destination module with the condition Module.DimMemRef=. For example:

- "Module".DimMemRef="ExternalUnit"
- "Module".DimMemRef="Resource"
- "Module".DimMemRef="Activity"
- "Module".DimMemRef="CostObject"

Apart from these recommendations, always enclose a field (that is present in the formula) in double quotation marks when it contains a space between two words. For example, write **`Intersection.Match("Product Type", Product, Cust)`**.

See Also

- “Enhanced Formula Capabilities” on page 465
- “Formula” on page 458
- “Formula Context” on page 459
- “Troubleshooting Formulas” on page 469
- “Using Numeric Attributes in a Formula” on page 467
- Chapter 52, “Boolean Functions,” on page 475
- Chapter 53, “Numeric Functions,” on page 485
- Chapter 54, “String Functions,” on page 491

Chapter 49

Windows for Assignments

Add Accounts for Assignments Dialog Box

Overview

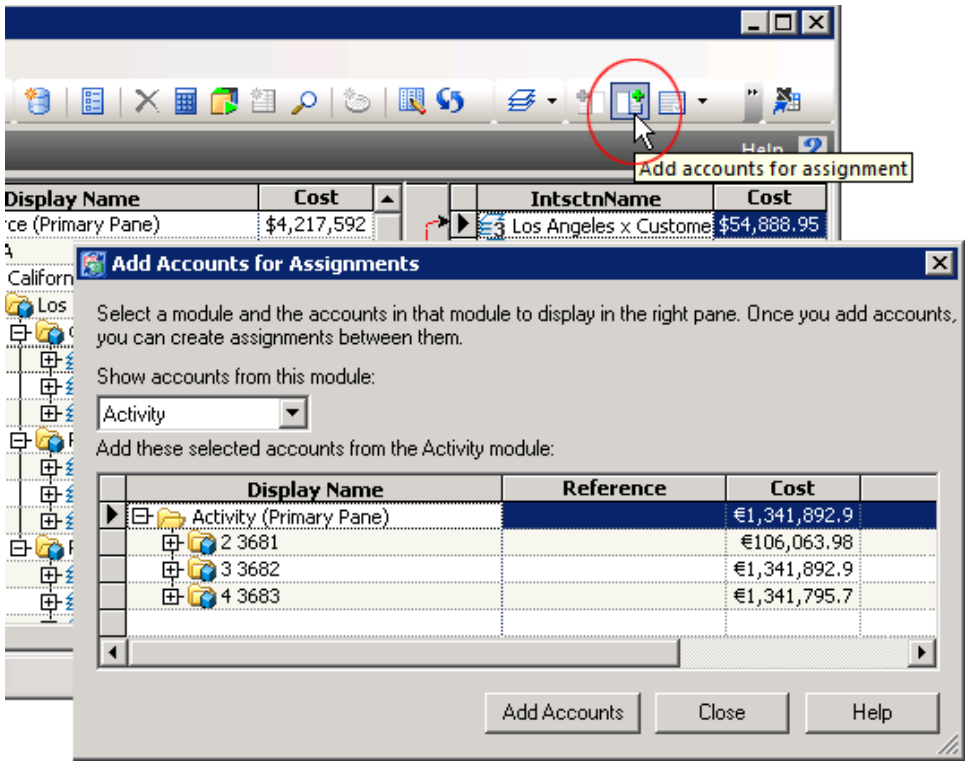
In the Add Accounts for Assignments dialog box, you can select accounts to display in an assignments pane in preparation for creating an assignment.

Note: Some functionality is not available depending upon your permissions

How to Access the Add Accounts for Assignments Dialog Box

Do one of the following:

- In any module view, open the left or right assignments pane and select **Model** ⇨ **Assignments** ⇨ **Add Accounts in Left Pane** (or **Add Accounts in Right Pane**).
- Click an **Add accounts for assignment** button on the toolbar.



See Also

- "Add an Account for Making an Assignment" on page 449
- "What Is an Assignment?" on page 431

Chapter 50

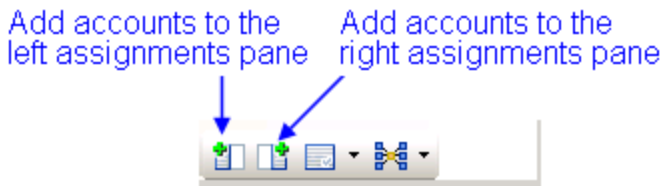
How To: Assignments

Add an Account for Making an Assignment

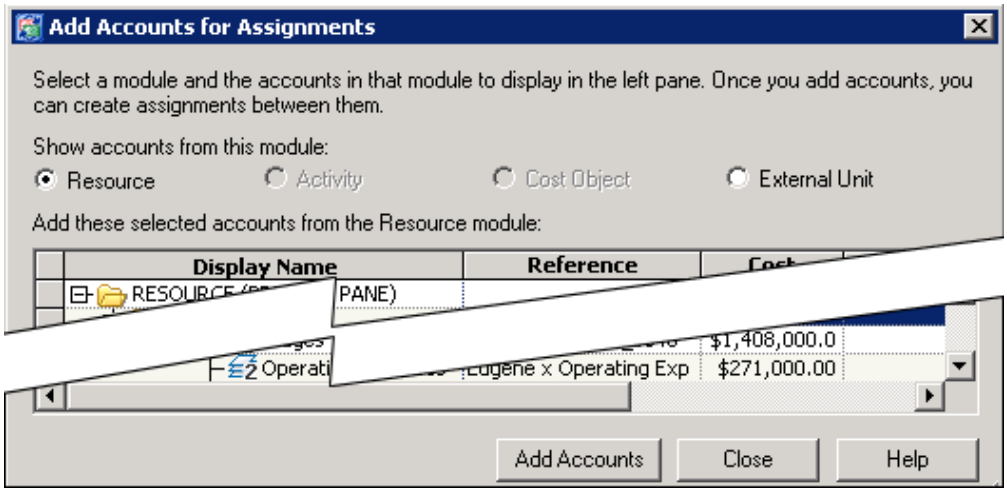
1. Select **Model** ⇨ **Assignments** to display an additional assignment pane.
   - If the source account is in the primary pane, then select **Show Right Assignments Pane** to select the destination accounts. See "Right Assignments Pane" on page 434.
   - If the source account is in the left assignments pane, then select **Show Left Assignments Pane**.

     From the left assignments pane, you can make assignments only to the primary pane. If the desired source account is not visible in the left assignments pane, you can add it to that pane.

   See "Show Assignments Panes" on page 453.
2. Add accounts for assignments.
   a. Do one of the following:
      - Select **Model** ⇨ **Assignments** ⇨ **Add Accounts in Right Pane**.
      - Select **Model** ⇨ **Assignments** ⇨ **Add Accounts in Left Pane**.
      - Click one of the Add Accounts for Assignment icon.



The Add Accounts for Assignment dialog box appears. See "Add Accounts for Assignments Dialog Box" on page 447.



b. Select the accounts that you want to add to the left or right assignments pane and click **Add Accounts**.

The selected accounts are added to the appropriate assignments pane.

If you select a roll-up account, all accounts within that roll-up account are added.

Note: You cannot select the module roll-up (e.g., Cost Object) and add all accounts within it.

Make an Assignment

To make an assignment from a source account to a destination account:

1. Add the **DrvName** property to the column layout of the assignment pane that contains the source account.

   The source account can be in either the left assignments pane or the primary assignments pane. You can not make an assignment from an account in the right assignments pane because there is no pane visible to the right of the right assignments pane.

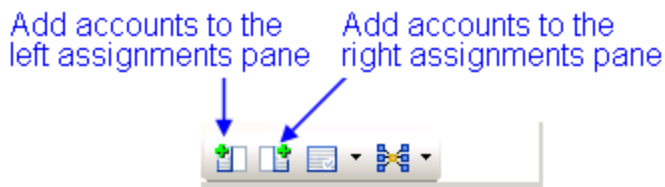
   For information on adding columns to a column layout, see "Create a Column Layout" on page 339.

   Note: If you want to make an assignment from a source account in the left assignments pane to an account in the primary pane, and if the source account is not visible in the left assignments pane, then you can add that account to the left assignments pane. See "Add an Account for Making an Assignment" on page 449.

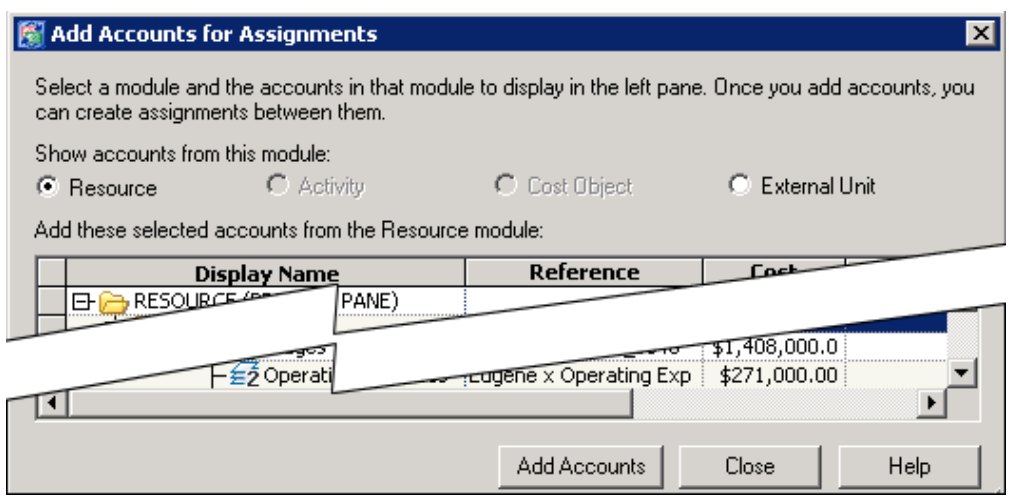
2. Select **Model** ⇨ **Assignments** to display an additional assignment pane.
   - If the source account is in the primary pane, then select **Show Right Assignments Pane** to select the destination accounts. See "Right Assignments Pane" on page 434.
   - If the source account is in the left assignments pane, then select **Show Left Assignments Pane**.

     From the left assignments pane, you can make assignments only to the primary pane. If the desired source account is not visible in the left assignments pane, you can add it to that pane.

   See "Show Assignments Panes" on page 453.
3. Add accounts for assignments.
   a. Do one of the following:
      - Select **Model** ⇨ **Assignments** ⇨ **Add Accounts in Right Pane**.
      - Select **Model** ⇨ **Assignments** ⇨ **Add Accounts in Left Pane**.
      - Click one of the Add Accounts for Assignment icon.



The Add Accounts for Assignment dialog box appears. See "Add Accounts for Assignments Dialog Box" on page 447.

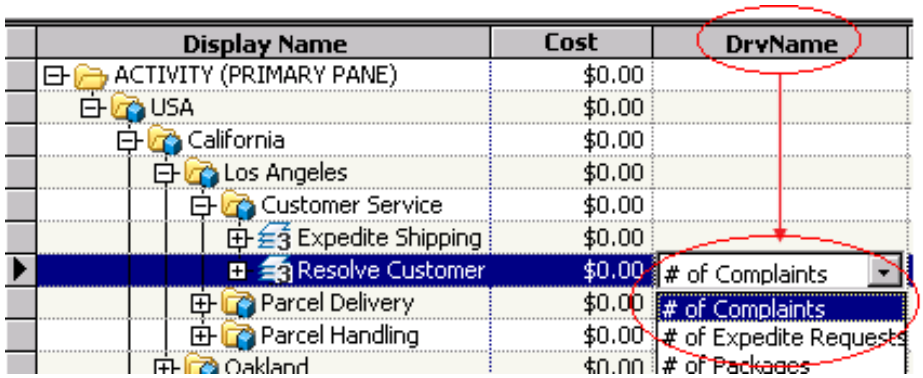


   b. Select the accounts that you want to add to the left or right assignments pane and click **Add Accounts**.

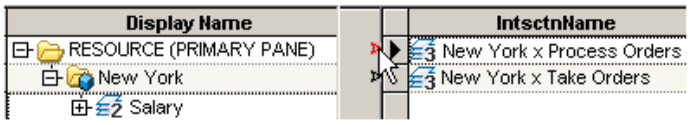
      The selected accounts are added to the appropriate assignments pane.

      If you select a roll-up account, all accounts within that roll-up account are added.

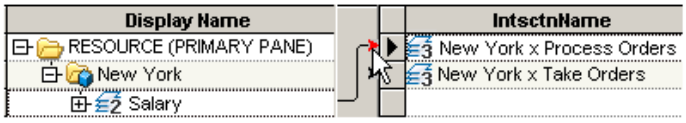
      Note: You cannot select the module roll-up (e.g., Cost Object) and add all accounts within it.
4. In the **DrvName** column for the row of the source account, select the driver that you want to use for making assignments.



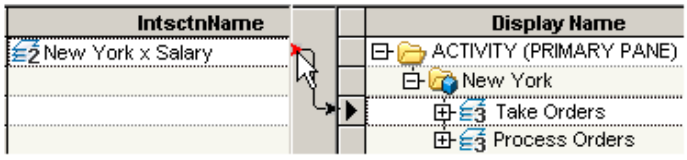
5. To assign the cost to an account in the right assignments pane, click the arrowhead to the left of the account.



An arrow connects the two accounts.

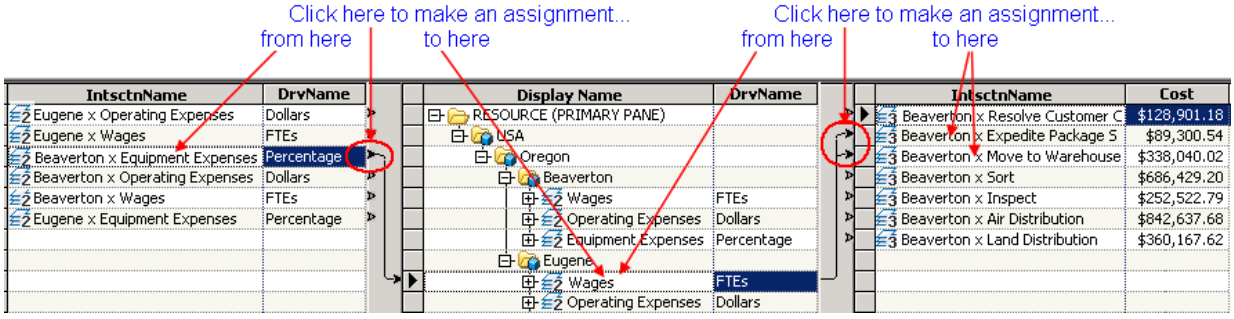


To assign the cost from an account in the left assignments pane, click the arrowhead to the right of the account.



To summarize, as the following picture shows, you click an arrowhead in the splitter bar to make an assignment either:

- from the selected source account in the primary pane to an account in the right assignments pane
- from a source account in the left assignments pane to the selected account in the primary pane.



TIP To quickly create assignments to many accounts, select **Model** ⇨ **Assignments** ⇨ **Assign All Left**, **Assign All Right**, or **Assign All Left and Right**.

TIP An even quicker method is to click F7.

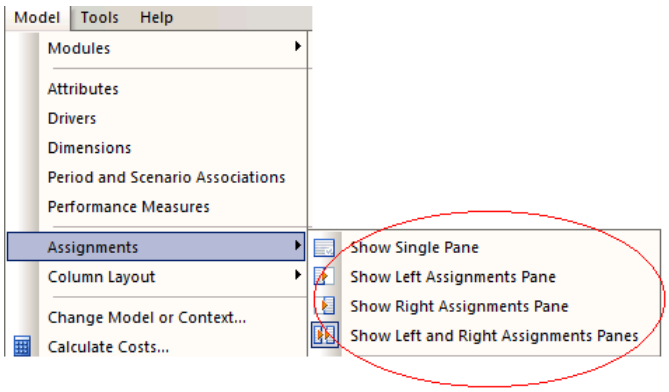
Make Assignments Using a Rule

See "Make Assignments Using a Rule" on page 436.

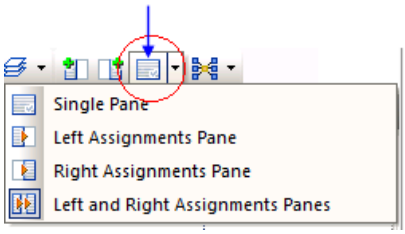
Show Assignments Panes

To show Assignments Panes, do one of the following:

- Select **Model** ⇨ **Assignments**



- Click the Assignments icon on the tool bar and select from the resulting drop-down box.



.

Show Assignments To and From an Account

Show Assignments to an Account

1. Select **Model** ⇨ **Assignments** ⇨ **Show Left Assignments Pane**.

   The view is split to include an empty left **assignments pane**. The selected module is displayed in the primary pane on the right.

2. Expand the module hierarchy and select an account.
3. Select **Model** ⇨ **Assignments** ⇨ **Show Left**.

Arrows indicate that the listed accounts are source accounts for the selected account.

Show Assignments from an Account

1. Select **Model** ⇨ **Assignments** ⇨ **Show Right Assignments Pane**.

   The view is split to include an empty right assignments pane. The selected module is displayed in the primary pane on the left.

2. Expand the module hierarchy and select an account.
3. Select **Model** ⇨ **Assignments** ⇨ **Show Right**.

   Arrows indicate that the listed accounts are destination accounts for the selected account.

Show Assignments to and from an Account

1. Select **Model** ⇨ **Assignments** ⇨ **Show Left and Right Assignments Pane**.

   The view is split to include empty left and right **assignments panes**. The selected module is displayed in the primary pane in the center.

2. Expand the module hierarchy and select an **account**.
3. Select **Model** ⇨ **Assignments** ⇨ **Show Left and Right**.

   Arrows indicate the accounts that are source accounts for the selected account, as well as the accounts that are destination accounts for the selected account.

Delete Assignments

1. Decide which assignments panes to show.
2. To delete an assignment, click the arrowhead to the right (or to the left) of the source account (or of the destination account).

   An arrow connects the two accounts.

TIP To quickly delete assignments to many accounts, select **Model** ⇨ **Assignments** ⇨ **Delete All Assignments Left**, **Delete All Assignments Right**, or **Delete All Assignments Left and Right**.

TIP To delete all assignments, left and right, you can also click **Shift** + **F7**. All assignments are deleted—not just those that are displayed, in case there are some assignments not currently displayed.

Part 15

Using Formulas with Drivers and Calculated Attributes

Chapter 51

Formulas

Formula

Overview

A formula consists of expressions made up of the following elements that are evaluated in a context:

- strings and the string concatenation operator &
- Boolean operators: NOT, AND, OR
- numbers and numerical operators +, -, *, /,**,%, (,)
- relational operators <, <=, =, >=, > <>
- functions
- system-defined properties
- user-defined attributes
- user-defined dimensions and dimension members
- user-defined dimension attributes and dimension member attributes
- system-defined dimensions: Module, Period, Scenario

Note:

Use the Formula Builder dialog box to create a formula for use with a:

- calculated driver (See Chapter 44, "Calculated Driver," on page 393.)
- rule-based driver (See "Make Assignments Using a Rule" on page 436.)
- calculated attribute (See "Calculated Attributes" on page 274.)

Using attributes in a formula

To include user-defined attributes, you must create these attributes before you define a formula.

If you use an attribute in a calculated driver formula, you must add that attribute to all destination accounts of the assignment before running any calculations. When you add the attribute, enter the appropriate values for both the numeric attribute and the driver quantity. The numeric attribute and the driver quantity must contain values; they cannot be blank. You can add the attribute either before defining the calculated driver, or after. The numeric attribute must be added before you run a calculation. Otherwise, you might encounter errors, or costs might be calculated as zero.

Source account and destination accounts

a formula can use the numeric properties and numeric attributes of a source account or a destination account. In the SAS Cost and Profitability Management interface, the list of numeric properties includes the keywords **`[Source]`** and **`[Destination]`**. When you select either of the keywords to include in a formula, you see the notation **`Source`**. or **`Destination.`**, followed by a property or attribute. For example, **`Source.Cost`** or **`Destination.UnitCost`**.

Note: By default in the formula for a calculated driver and for a rule-based driver, properties and attributes refer to destination accounts.

Property names in formulas

The property names in formulas are the same as the property names that you see elsewhere in SAS Cost and Profitability Management, except that the spaces are removed. For example, the property Allocated Cost is **`AllocatedCost`** in a formula.

Occasionally, the property name that is used in a formula contains an abbreviation, such as **`DimLevelName`** for the property Dimension Level Name, or **`TDQ`** for the property Total Driver Quantity.

Properties for a source account and a destination account

You can access the values of properties for a source account or a destination account. To access the properties for a source account, use the keyword **`[Source]`**, a period (**`.`**), and a property name. For example, **`Source.Cost`**. To access the properties for a destination account, use the keyword **`[Destination]`**. Both of the keywords are available when you define a formula.

Testing formulas

Testing a formula validates the operators and ensures that there are no spaces in the formula. However, numeric properties and numeric attributes are not validated. Testing does not guarantee that the formula is valid or that the formula will yield the intended value.

During calculation, if a formula references an item that does not exist, a warning is displayed. If you use an invalid formula, SAS Cost and Profitability Management halts the calculation as soon as it encounters the invalid formula. Subsequent values in the model are not calculated.

Using attribute names

To ensure that SAS Cost and Profitability Management correctly interprets any special characters or spaces in a numeric attribute name, enclose the name in double quotation marks **`("")`**. You do not need to use quotation marks for names of numeric properties.

For example, in the following formula, **`OutputQuantity`** does not require quotation marks because it is a system-defined numeric property. But, **`Units Per Pallet`** requires quotation marks because that name contains spaces.

```
OutputQuantity/"Units Per Pallet"
```

Formula Context

Overview

A formula consists of expressions made up of the following elements:

- strings and the string concatenation operator &

- Boolean operators: NOT, AND, OR
- numbers and numerical operators +, -, *, /,**,%, (,)
- relational operators <, <=, =, >=, > <>
- functions
- system-defined properties
- user-defined attributes
- user-defined dimensions and dimension members
- user-defined dimension attributes and dimension member attributes
- system-defined dimensions: Module, Period, Scenario

Note: .See "Operator Precedence" on page 470.

Some expressions are not dependent on context. For example the string ABC and the number 2010 refer to the same thing regardless of context. Other expressions, such as attributes and properties depend on context. The property Cost, for example, only has a value relative to an account or an assignment. Hence, you should be awareof the following principles concerning formulas:

calculated driver
The formula for a calculated driver is calculated in the context of an assignment from each source account to which the driver is attached and each destination account, in turn, from that source account.

By default, properties and attributes in the formula refer to the destination account.

rule-based driver
The formula for a rule-based driver is calculated in the context of an assignment from each source account to which the driver is attached and each potential destination account, in turn, from that source account.

By default, properties and attributes in the formula refer to the destination account.

calculated attribute
The formula for a calculated attribute is evaluated in the context of an account.

Qualifying Properties

When you include a property in a formula, you can qualify the name of the property to specify unambiguously which property you are referring to. Although qualifiers may not be necessary in a particular case, you are always allowed to use up to two levels of qualifiers:

Following are some examples:

Example

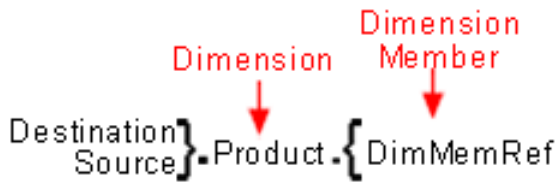
Products.DimMemRef

Source.Products.DimMemRef

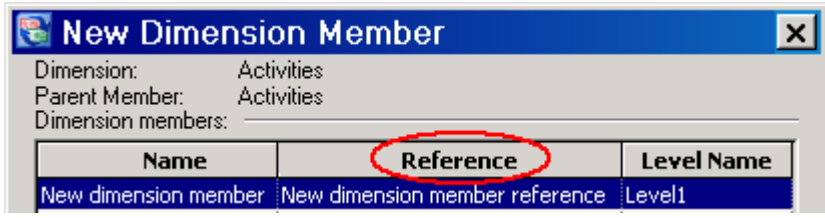
Destination.Products.DimMemRef

Explanation

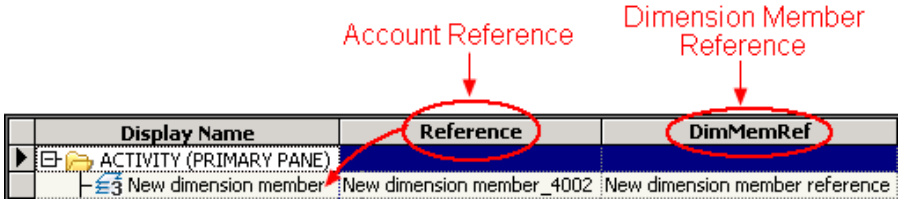
In this example, "Products" is a dimension reference. When unqualified with either Source or Destination, Destination is assumed.



When you create a dimension member you must specify its reference.



When the dimension member is displayed in an assignments pane, its column header is labelled DimMemRef to distinguish it from the column for an account reference.



Example

Products.DrivableCost

Source.Products.DrivableCost

Destination.Products.DrivableCost

Explanation

In this example, "Products" is a dimension reference. When unqualified with either Source or Destination, Destination is assumed.

Example

Module.DimMemRef

Destination.Module.DimMemRef

Source.Module.DimMemRef

Explanation

Possible values for Module.DimMemRef are:

- "ExternalUnit"
- "Resource"
- "Activity"
- "CostObject"

You can use Module.DimMemRef in expressions such as Module.DimMemRef="CostObject".

Example

Period.2009

Explanation

Quotation marks are not required around the name of the period.

Example

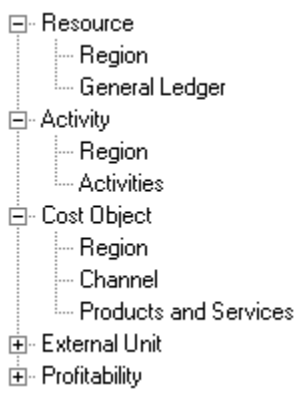
Scenario.Budget

Explanation

Quotation marks are not required around the name of the scenario.

An Example

Let's consider an extended example using the Parcel Express Tutorial. The tutorial model has the following module dimensions:



The dimensions have the following dimension members.

Name	Reference
Region	Region
USA	USA
Oregon	Oregon
Beaverton	Beaverton
Eugene	Eugene
General Ledger	GL
Activities	Act
Personnel Intensive Activities	Personnel Intensive Activities
Resolve Customer Complaints	Resolve Customer Complaints
Expedite Package Shipments	Expedite Package Shipments
Local Collection	Local Collection
Move to Warehouse	Move to Warehouse
Local Processing	Local Processing
Sort	Sort
Inspect	Inspect
Regional Distribution	Regional Distribution
Air Distribution	Air Distribution
Land Distribution	Land Distribution
Channel	Chnnl
Drop Box	Drop Box
Walk In	Walk In
Commercial Pick-up	Commercial Pick-up
Products and Services	Prod_Serv
2nd Day Guaranteed	2nd Day Guaranteed
Overnight Express	Overnight Express
Standard Ground	Standard Ground
Materials	Mat
Fixed_Variable	FV

Now let's consider an assignment, shown in the picture below, from the following source account in the Activity module:

Region Dimension Member	Activity Dimension Member, Level 1	Activity Dimension Member, Level 2
Beaverton	Personnel Intensive Activities	Expedite Package Shipments

to the following destination account in the Cost Object module:

Region Dimension Member	Channel Dimension Member	Products and Services Dimension Member
Beaverton	None	2nd Day Guaranteed

Now let's assume that the destination account is the context account for this example. As the picture below shows, the following expressions that might appear in a formula have the value shown in the following table:

Formula Expression	Value
Destination.Region.DimMemRef	Beaverton
Destination.Prod_Serv.DimMemRef	2nd Day Guaranteed

Formula Expression	Value
Destination.Chnnl.DimMemRef	None
Source.Region.DimMemRef	Beaverton
Source.Act.DimMemRef	Expedite Package Shipments

Note: It is very important to note that you refer to a null intersection with the constant None. The destination account is the intersection of the following dimension members:

Region Dimension Member	Channel Dimension Member	Products and Services Dimension Member
Beaverton	None	2nd Day Guaranteed

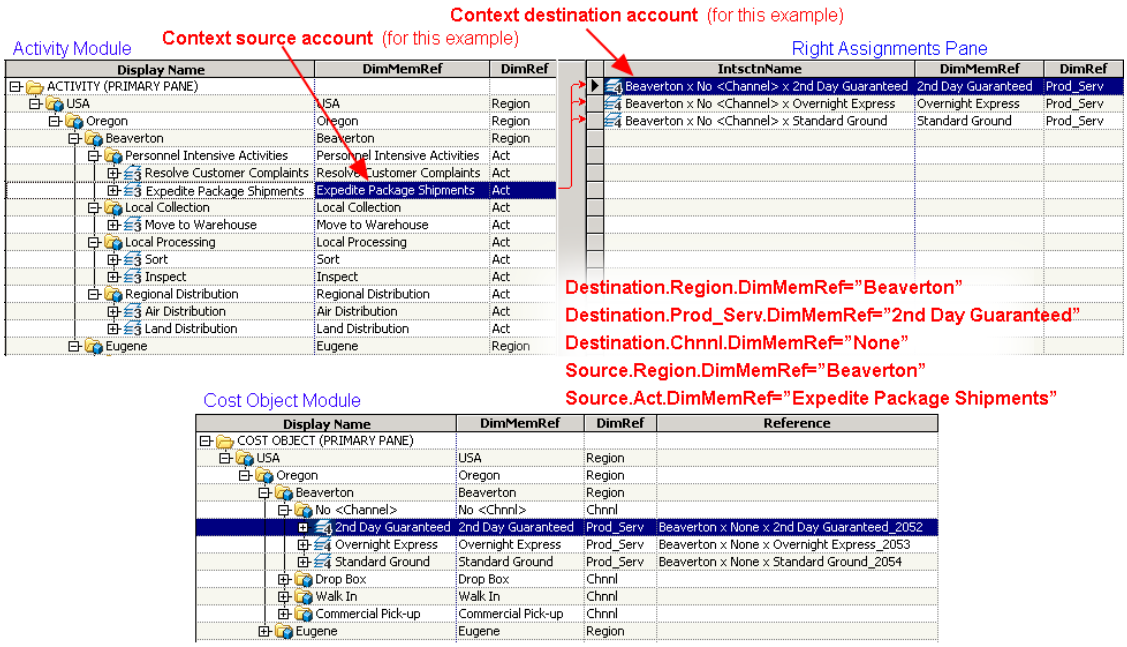
Even though the column for the intersection name shows No <Channel> for display purposes, you should use the internal name None in your formulas.

CAUTION!

Because "None" is localizable, you must use the value of "None" for your locale. Following is a list of possible values according to locale:

German	Kein
English	None
Spanish	Ninguno
French	Néant
Italian	Nessuno
Japanese	なし
Korean	없음
Dutch	Geen
Portuguese (Brazil)	Nenhum
Chinese Simplified	无
Chinese Traditional (Hong Kong SAR)	無
Chinese Traditional (Taiwan)	無

Note: Instead of using a localizable constant, you can use the IsNone() function, which will work in any locale. See "IsNone function" on page 481.



Enhanced Formula Capabilities

Overview

Significant new capabilities have been added to SAS Cost and Profitability Management for creating the formula (rule formula) of a rule-based driver. You can use these same enhanced capabilities to create the driver formula for a calculated driver or the formula for a calculated attribute. These enhanced capabilities include:

New Features

- String support
  - String (text) values are fully supported inside formulas, including the string values of text attributes.
  - You can use the following string properties of an account in any formula: Reference, Name, and DriverName.
  - You can use the following string properties of an account's intersection dimension members in any formula: Reference (or DimMemRef), Name (or DimMemName), Level, and LevelNumber.
- Access to properties
  - You can access the properties of an account's intersection dimension members as well as the properties of attribute dimension members.
  - You can reference the system-defined dimensions Period, Scenario, and Module in any formula.
- Functionality for calculated drivers

- The driver formulas for Calculated Drivers can access cost and revenue properties on accounts.

  Note: If cost properties are used, you should use driver sequencing or the cost properties will be zero.

New Functions

Boolean

Function	Description
HasAttribute(s)	Determines if an attribute is associated with an account *Note:* You can use the new HasAttribute() function with Tag (Boolean) attributes as well as with Text and Numeric attributes.
IsChildOf(s)	Determines if a dimension member is a child of (self, immediate child, or descendent) of a given dimension member
Match(s1,"s2")	Determines if string s1 matches a wildcard pattern s2.
isClose(n1,n2,digits)	Determines if two numbers are close, to accommodate floating point round-off issues

String

Function	Description
Left(s,n)	Returns the leftmost characters of a string
Mid(s,pos,len)	Returns a substring of a source string
Right(s,len)	Returns the rightmost characters of a string
Str(s,size,dec)	Converts a number to a string
Trim(s)	Trims leading and trailing spaces

Numeric

Function	Description
Find(s1,s2)	Returns the position of string s1 within string s2.
Len(s)	Returns the length of a string

For a full list of functions, see Functions.

Changed Functionality

Expressions with Null values evaluate to false in comparison (<, <=, =, >=, >, <>) and Null for all other operators. Note that "Null=Null" is false so the IsNull() function should be used when it is important to determine whether a value (or expression) is Null or not. Null values occur when:

- an attribute is not attached to any account and does not have a default value
- for some properties such as TDQUE and OutputQuantityUE.

Limitations

1. Default values and formulas are not supported for Text attributes.
2. The following properties are not valid for use in the formula of a calculated driver and, therefore, do not appear in the Formula Builder drop-down list:
   - DriverQuantityBasic
   - IdleDriverQuantity

   The reason is that in an assignment these quantities depend on DrvQtyCalc. Consequently, a vicious circle would be created if the formula were to make DrvQtyCalc depend on these quantities. That is, DrvQtyCalc could not be determined without first determining these quantities, and these quantities could not be determined without first determining DrvQtyCalc.

Using Numeric Attributes in a Formula

Overview

You can use both calculated numeric attributes and ordinary numeric attributes in a formula. All that you need to do is to refer to the attribute by its reference just as you would any property.

Note: If an attribute reference contains spaces or special characters, then you must enclose it in quotation marks.

Using a Calculated Numeric Attribute

In order to use a calculated attribute in a formula it is not necessary to attach the attribute to an account. It is sufficient that you refer to the attribute for the attribute's formula to be evaluated during calculation. This allows you to use calculated attributes to define common sub-expressions that are used by other formulas. This avoids cut-and-paste errors and allows for quick revisions without having to update multiple formulas. It also allows for the reusability of calculated attributes to define sub-expressions that can be reused by different drivers.

Example

Suppose you have the following three calculated drivers each with its driver formula as shown in the following table:

Calculated Driver	Driver Formula
Drvr1	(A / B) * (B / C)
Drvr2	(A / B) * 2
Drvr3	(B / C) * 3

Now, suppose you define the following calculated numeric attributes:

Calculated Attribute	Formula
Rate1	(A / B)
Rate2	(B / C)

then you can modify the driver formula for each of the three calculated drivers to use a calculated attribute as shown in the following table:

Calculated Driver	Driver Formula
Drvr1	Rate1 * Rate2
Drvr2	Rate1 * 2
Drvr3	Rate2 * 3

Note: The cascading evaluation of attribute formulas is fully supported. For example, suppose calculated attribute CA1 uses calculated attribute CA2, which in turn uses calculated attribute CA3, etc. There is no restriction on the depth of references, but it will be more difficult to troubleshoot with increasing depth. However, cyclic reference is not supported. For example, CA1 CA2 CA3 CA1 is a cyclic reference (back to C1) and an error is returned when cyclic reference is encountered.

Of course, whenever a calculated attribute is attached to an account its calculated value is written to the model by the calculation. Even for sub-expression use, where you wouldnt normally attach the calculated attribute to an account, it can still be useful to attach it to one account when debugging complex formulas. When it is working you can then remove it from the account.

Using a Calculated Attribute in the Driver Formula of a Calculated Driver

In the driver formula for a calculated driver you can use only properties that represent a quantity. Consequently, if such a formula contains a calculated attribute, then that calculated attribute should also use only properties that represent a quantity.

Determining the Value of a Numeric Attribute

During calculation, when the system encounters a numeric attribute in a formula it uses the following rules of precedence for determining the attribute value. As soon as the system is able to determine the attribute value according to a rule it uses that value as the final calculation and does not consult any subsequent rule.

1. If it is a calculated attribute, then evaluate its formula and use the resulting value.
2. If the attribute is attached to the account, use the value of the attribute for that account.
3. If the attribute has a default value for the current period/scenario, then use the default value.
4. Use null as the attribute value.

Troubleshooting Formulas

Using Cost in a Formula

This section explains why using cost properties in a formula (driver formula for a calculated driver, rule formula for a rule-based driver, or formula for a calculated attribute) can make it difficult to troubleshoot problems, describes some troubleshooting techniques, and provides guidelines for their use.

Whenever cost is used to determine flow, driver sequencing is required. When you use Cost in a formula, the actual value used comes from the previous sequence. So if a driver uses sequence 3, then it takes Cost from the destination account after sequence 2 is evaluated. In the case where there is a single Cost property, the value is written to DQCalc and it is easy to determine what the value that was used.

More complicated formulas become more difficult. For instance, using a formula of "Cost / NumCalls" means that you would have to multiply DQCalc * NumCalls to determine the value of Cost that was evaluated in sequence 2 and used in sequence 3.

An even more complicated situation is where the formula uses a calculated attribute that in turn uses Cost. A common way to troubleshoot a calculated attribute is to attach it to an account, calculate through the sequence in question, and look at its value. This works well, but when Cost (or any cost property) is used in the attribute formula, a final evaluation is made before it is committed to the database to ensure that it uses the proper cost value. To see how this is significant you must understand how calculation orders its work:

```
Sequence Loop
   1. Clear calculated attribute values
   2. Create rule-based assignments (using rule formulas)
   3. Consumption (eval DQCalc, using driver formulas)
   4. Cost flow
 End Loop
 Clear calculated attribute values
 Eval calculated attributes
```

In the steps above, calculated attributes can get evaluated during **`2. Create rule-based assignments`** as a result of evaluating rule formulas and during **`3.`**

Consumption as a result of evaluating driver formulas. Notice that this occurs before **4. Cost flow**. Numeric attributes need a final evaluation so that any use of cost properties is in sync with the final cost values. So the cost values used in calculated attributes could be different from those used by rule formulas or driver formulas. This could easily cause confusion when troubleshooting the actual values used. To find out what values are used, stop calculation on the sequence step before they are actually used by a rule formula or driver formula to see what their value is.

To avoid confusion consider the following guidelines

1. If all you need is to use a cost property for DQF or DQV, then use FixedDQOverride or VariableDQOverride respectively. This keeps it simple.
2. If you need a more complicated driver formula or rule formula that uses cost properties, avoid using calculated attributes that have cost properties in their formulas (directly or indirectly).
3. If you must use cost properties in the formula for a calculated attribute, keep their use to a minimum.

Troubleshooting String Expressions

A typical way to troubleshoot formulas is to put the formula in a calculated numeric attribute and attach the attribute to an account. This works as long as the formula has a numeric result. But for string expressions this is not an option because calculated text attributes do not exist in SAS Cost and Profitability Management. So, how can you troubleshoot string expressions when you aren’t getting the results you want?

A technique is to use theif(condition,then,else) function in a numeric formula, as in the following example:

```
If( Attr1 & Attr2 = "ABCD", 1, 0 )
```

This concatenates two attributes to see if they are a specific value and if they are returns a value of 1, and 0 if they aren't.

You aren't limited to concatenation—you can use any string expression and compare its results to something. For instance, you could use something like the following:

```
If( HasAttribute(Attr2) and len(Attr2) > 2, 1, 0)
```

Operator Precedence

The following table shows the priority given to operators when the system evaluates a formula. Priority goes from top to bottom of the table, and from left to right in a given row.

So, for example, NOT x+y is evaluated as (NOT x)+y rather than NOT(x+ y) because NOT takes priority over +.

And, for example, x*y/z is evaluated as (x*y)/z rather than x*(y/z) because multiplication takes priority over division.

Note: It is a good idea to use parentheses. Notice that parentheses have the highest priority and, thus, remove all ambiguity in a formula.

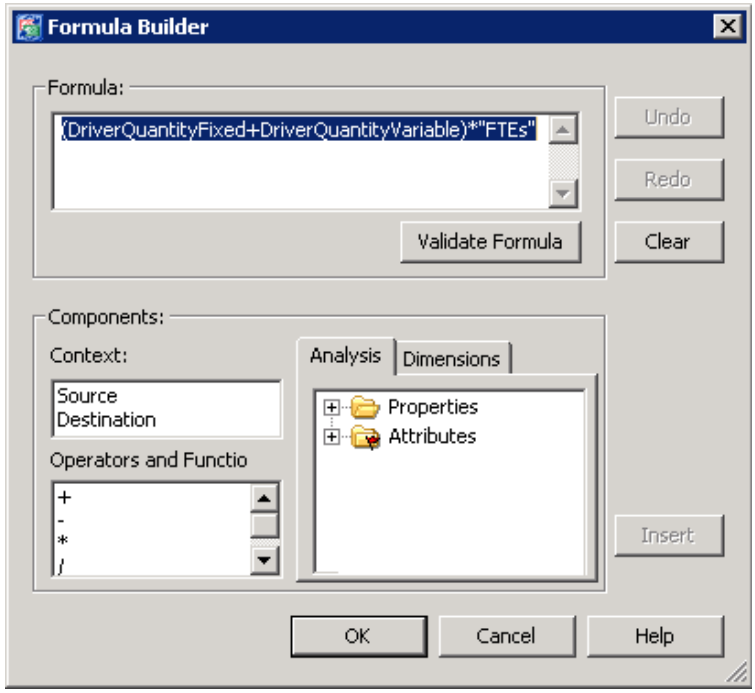
Priority	Operator	Explanation
1	(x)	parentheses
2	x**y	exponent
3	NOT x	unary boolean operator
4	+x, -x	unary numeric operators (positive, negative)
5	x*y, x/y, x%y	binary numeric operators: multiply, divide, percent
6	x+y, x-y, x&y;	binary operators: plus, minus, string concatenation
7	x<y, x<=y, x>=y, x>y	numeric or string comparison
8	x=y, x<>y	numeric or string comparison
9	x AND y	boolean AND
10	x OR y	boolean OR

Formula Builder Dialog Box

Overview

In the Formula Builder dialog box, you can create a formula for a:

- calculated driver (See Chapter 44, “Calculated Driver,” on page 393.)
- rule-based driver (See “Make Assignments Using a Rule” on page 436.)
- calculated attribute (See “Calculated Attributes” on page 274.)



Note: Every formula is evaluated in a context. (See "Formula Context" on page 459.)

How to Access the Formula Builder Dialog Box

To create a formula for a calculated driver
In the New Driver dialog box, click **Formula Builder**.

To create a formula for a rule-based driver
In the New Driver dialog box, select **Use Rule Formula** and click **Formula Builder**.

To create a calculated attribute
In the New Attribute dialog box, click **Formula Builder**.

Context

Use the keywords Source or Destination to specify whether a property or attribute applies to a source or destination account. For example, Source.Cost or Destination.UnitCost.

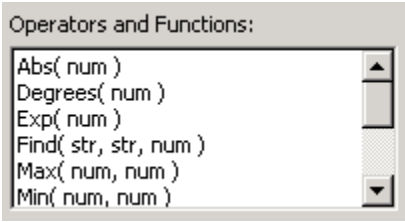
Note: By default in the formula for a calculated driver, properties and attributes refer to destination accounts.

Note: By default in the formula for a rule-based driver, properties and attributes refer to destination accounts.

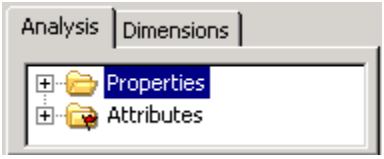
Operators and Functions

For functions, see:

- Chapter 52, "Boolean Functions," on page 475
- Chapter 53, "Numeric Functions," on page 485
- Chapter 54, "String Functions," on page 491



Analysis



This tab contains a list of system-defined properties and user-defined attributes.

To ensure that SAS Cost and Profitability Management correctly interprets any special characters or spaces in a numeric attribute name, enclose the name in double quotation marks (" "). You do not need to use quotation marks for names of numeric properties.

For example, in the following formula, OutputQuantity does not require quotation marks because it is a system-defined numeric property. But, Units Per Pallet requires quotation marks because that name contains spaces.

```
OutputQuantity/"Units Per Pallet"
```

Note: The names of the properties shown in the Formula Builder dialog box are different from the names of the properties in other parts of the interface. See: About Property names in formulas.

Dimensions

This tab contains a list of dimensions, dimension members, and dimension attributes.

Define a formula

1. Select an item from the available components.
2. Click **Insert**.

   The item is added to the formula.

   TIP You can also type the formula.
3. Continue to insert items until you have built the formula.

   Note: The formula can be at most 16383 characters long.
4. To remove the last item that you inserted, click **Undo**.
5. To delete the entire formula, click **Clear**.
6. (Optional) Click **Test**.

   If no errors are displayed, the syntax of the formula is valid.

Chapter 52

Boolean Functions

Introduction

A Boolean function, named after the English logician/mathematician George Boole, returns either a TRUE or a FALSE.

Note: Null values are considered to be false when evaluating a formula for a rule-based driver.

The following are Boolean functions:

- HasAttribute
- if
- Intersection.Match
- IsChildOf
- IsClose
- IsNull
- Match

HasAttribute function

Returns true if an account has an attribute attached to it for the attribute reference passed, and false otherwise. You can use the new HasAttribute() function with Tag (Boolean) attributes as well as with Text and Numeric attributes.

Syntax

```
HasAttribute( string )
```

where *string* is an attribute reference.

Examples

```
HasAttribute( "Quality" )
Destination.HasAttribute( "Quality" )
Source.HasAttribute( "Quality" )
```

Further explanation

If an attribute association does not exist but the attribute has a default value, false is returned. This works only for the numeric, text and tag value attributes—not for dimension attributes.

This function must be evaluated in the context of an account. Appropriate "Source." or "Destination." prefixes may be needed to select the correct account.

Note: When used in the assignment rule for a rule-based driver, the attribute is assumed to refer to the *destination* account unless Source is specified.

if function

The if function returns one value if a test evaluates to TRUE and a different value if the test evaluates to FALSE.

Syntax

```
if(test , true_value , false_value )
```

test is a value or expression that evaluates to TRUE or FALSE.

true_value is the value that is returned when test is TRUE.

false_value is the value that is returned when test is FALSE.

Examples

```
if( not IsNull(SoldQuantity), SoldQuantity, OutputQuantity )
```

Intersection.Match function

The Intersection.Match function returns TRUE if for each dimension listed:

- the source account and destination account contain the same dimension member
- or, the source account contains the No-dimension member for that dimension.

Use the Intersection.Match function in a rule formula to identify the destination accounts of a rule-based driver. See "Make Assignments Using a Rule" on page 436.

Syntax

```
Intersection.Match( dim-ref1[.dim-member-ref1][,dim-ref2[.dim-member-ref2]]* )
```

Examples

`Intersection.Match(Product, Customer)`

This example is equivalent to:

```
  (Source."Product".isNone()
  OR Source."Product".DimMemRef=Destination."Product".DimMemRef)
AND
  (Source."Customer".isNone()
  OR Source."Customer".DimMemRef=Destination."Customer".DimMemRef)
```

This example ensures that the dimension members of the Product and Customer dimensions are the same in both source and destination accounts.

Note: The No-dimension account, "None", acts like a wild-card. It matches every destination account.

`Intersection.Match(Product, Customer, Channel)`

This example is equivalent to:

```
  (Source."Product".isNone()
  OR Source."Product".DimMemRef=Destination."Product".DimMemRef)
AND
  (Source."Customer".isNone()
  OR Source."Customer".DimMemRef=Destination."Customer".DimMemRef)
AND
  (Source."Channel".isNone()
  OR Source."Channel".DimMemRef=Destination."Channel".DimMemRef)
```

This example ensures that the dimension members of the Product, Customer, and Channel dimensions are the same in both source and destination accounts.

Note: The No-dimension account, "None", acts like a wild-card. It matches every destination account.

`Intersection.Match(Module.CostObject, Product, Customer)`

This example is equivalent to:

```
  Destination."Module".DimMemRef="CostObject"
AND
  (Source."Product".isNone()
  OR Source."Product".DimMemRef=Destination."Product".DimMemRef)
AND
  (Source."Customer".isNone() )
  OR Source.Customer.DimMemRef=Destination.Customer.DimMemRef)
```

This example ensures that destination accounts are in the CostObject module. It also ensures that the dimension members of the Product and Customer dimensions are the same in both source and destination accounts.

Note: If there is a specific dimension-member-reference present (for example, Module.CostObject) then the condition applies only to destination accounts.

Note: Module is a system-defined dimension. CostObject is a dimension member.

Note: The No-dimension account, "None", acts like a wild-card. It matches every destination account.

Intersection.Match(Product.Frisbee)

This example is equivalent to:

```
  Source."Product".isNone()
OR
  Source."Product".DimMemRef=Destination."Product".DimMemRef
```

This example ensures that the destination account has dimension member named "Frisbee" in the Product dimension.

Note: If there is a specific dimension-member-reference present (for example, Product.Frisbee) then the condition applies only to destination accounts.

Note: The No-dimension account, "None", acts like a wild-card. It matches every destination account.

Intersection.Match(Module.CostObject, Product.Frisbee, Customer)

This example is equivalent to:

```
  Destination."Module".DimMemRef="CostObject"
AND
  Destination."Product".DimMemRef="Frisbee"
AND
  (Source."Customer".isNone()
  OR Source."Customer".DimMemRef=Destination."Customer".DimMemRef)
```

This example ensures that destination accounts are in the CostObject module. It also ensures that the destination account has dimension member named "Frisbee" in the Product dimension. And, it ensures that the dimension members of the Customer dimension are the same in both source and destination accounts.

Note: If there is a specific dimension-member-reference present (for example, Module.CostObject and Product.Frisbee) then the condition applies only to destination accounts.

Note: Module is a system-defined dimension. CostObject is a dimension member.

Note: The No-dimension account, "None", acts like a wild-card. It matches every destination account.

Notes:

- The dimension-references can refer to either structural dimensions or attribute dimensions.
- It is not required that all the defining dimensions of an account be included in MatchIntersection for it to test true..
- You can use system-defined dimensions , including Module, Period, Scenario and Driver.
- The dimensions can be in any order. For example, Intersection.Match(Product,Customer) and Intersection.Match(Customer, Product) give the same results.
- You can not use Intersection.Match in the formula of a calculated attribute. You can use it in a driver formula; however, as a Boolean function, it is unlikely to be useful there

You can use the Intersection.Match function to accomplish what the Sales Volume driver accomplished in previous versions of SAS Cost and Profitability Management. See "Sales Volume Driver" on page 8.

For example, suppose there is a model with two dimensions, Product and Customer, in the Cost Object module. The rule formula that was used for a Sales Volume driver was the following:

```
  (Source.Product.dimMemRef="None"
  OR Source.Product.DimMemRef=Destination.Product.DimMemRef)
AND
  (Source.Customer.dimMemRef="None"
  OR Source.Customer.DimMemRef=Destination.Customer.DimMemRef)
AND
  SoldQuantity <> 0
```

You can replace the entire formula above with the following equivalent formula using Intersection.Match

```
Intersection.Match(Product, Customer) and SoldQuantity<>0
```

This will match intersections in the following way

```
[Bike,None] matches [Bike,Target] and [Bike,WalMart]
[None,Target] matches [Bike,Target] and [Shirt,Target]
```

You can think of "None" as a type of wildcard. When an account is "matched" by the rule formula an assignment is created, such as [Bike,None] –> [Bike,Target] and [Bike,None] –> [Bike,WalMart]. In ABM 7.2 the accounts are often in the same module but starting from 8.1 the non-None accounts (e.g. [Bike,Target]) may be in a different module. In other words, a match will be made in the current or later modules and takes advantage of the new ModuleOrder property to define the order.

While it isn't shown, [Bike,Target] matches [Bike,Target] as long as the account is in a different module, preventing a meaningless cycle-to-self.

Notice that Module is treated as a dimension and CostObj is just a member of the dimension. This can also be extended to any dimension so the user can force it to match a specific member on the destination account. For instance you could do the following:

```
Intersection.Match(Module.CostObj, Product.Ugly, Customer) and SoldQuantity<>0
```

This replaces the condition

```
(Source.Product.isNone()OR(Source.Product.DimMemRef=Destination.Product.DimMemRef))
```

with

```
(Destination.Product.DimMemRef="Ugly").
```

The source account's Product is not used in this case.

IsChildOf function

Returns true if the dimension member is a child of (self, immediate child, or descendent) of a given dimension.

Syntax

dimension.reference.IsChildOf(*dim-member-reference*)

Examples

```
Customer.IsChildOf(Retail)
Source.Dept.IsChildOf(Sales)
Source.Dept.IsChildOf( "Sales" & "and Marketing" )
Product.IsChildOf(Source.Chnnl.Reference)
```

```
Product.IsChildOf(trim(Source.Chnnl.Reference))
```

Note: The dim-member-reference does not have to be the immediate child of the dimension.reference. For example, suppose you have the following hierarchy:

```
Products
   Wholesale
      ProductLineA
         Product1
         Product2
      ProductLineB
         Product3
         Product4
   Retail
      ProductLineC
         Product5
         Product6
      ProductLineD
         Product5
          Product6
```

In this case, the formula **`Products.IsChildOf(ProductLineB)`** tests true for Product3 and Product4.

Further explanation

If the account's intersection dimension member reference matches the string passed, true is returned. For instance, if the intersection has member Region.Raleigh and the formula has "Region.IsChildOf(Raleigh)" then a true is returned. IsChildOf() must evaluate in the context of a dimension member. To accomplish that, use compound references as shown in the examples to select a dimension member on an account's intersection. If an integer is passed then it is taken as-is without converting it to a number. This special case allows for syntax such as Period.2009 instead of requiring Period."2009".

Note: When used in the assignment rule for a rule-based driver, the member-reference is assumed to refer to the destination account unless Source is specified.

IsClose function

Returns true or false depending on whether the leading significant digits match.

Syntax

```
IsClose( number1, number2 [, digits] )
```

Examples

```
IsClose( 0, 0 ) → true
IsClose( 0, 1e-4 ) → false
IsClose( 0, 1e-9 ) → true
IsClose( 0, 1e-4, 3 ) → true
IsClose( 123.4567, 123.4568, 6 ) → true
IsClose( 0.00001234567, 0.00001234568, 6 ) → true
IsClose( 1.234567e-5, 1.234568e-5, 6 ) → true (same as previous example)
IsClose( 1, 1.00000001 ) → false
IsClose( 1, 1.000000001 ) → true
IsClose( 1, 0.999999999 ) → true
```

Note: In the last example the digits don't really match, but the number is just as far from 1 as in the previous example, so it is also considered close.

Further explanation

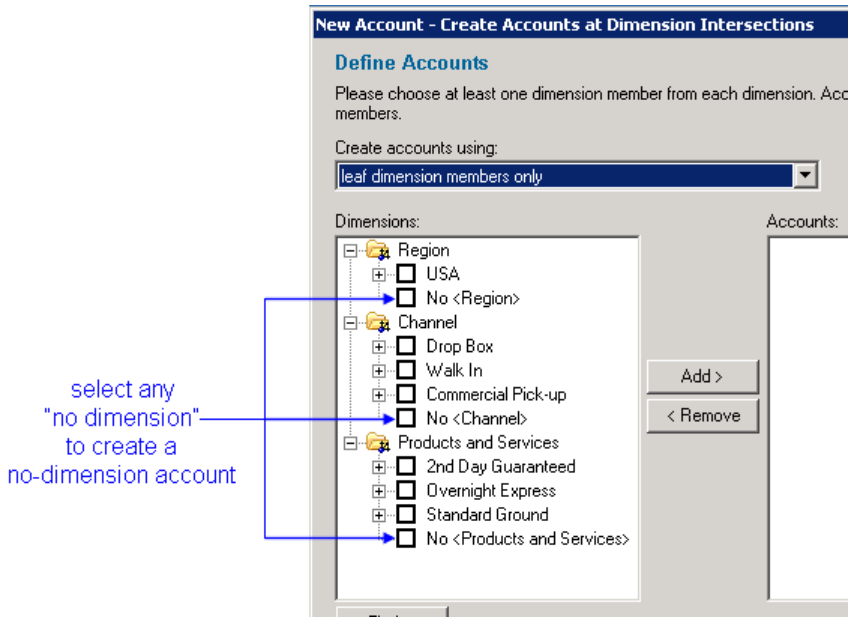
This function can be used when floating point round-off occurs. The number of digits to compare is determined by the digits parameter, which defaults to 8 if not passed.

IsNone function

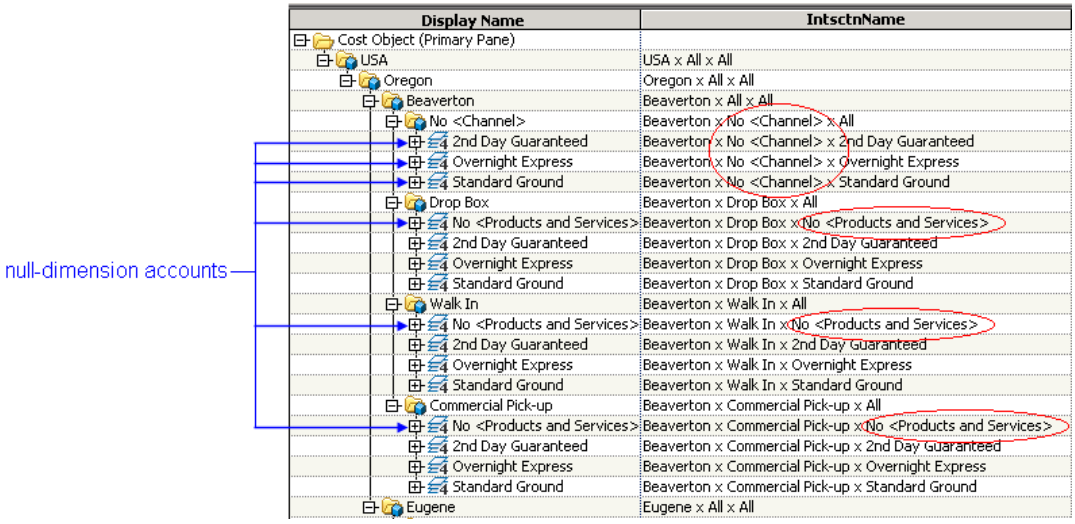
Returns true or false depending on whether an account is a No-dimension account.

A No-dimension account is one in which one or more of the dimension members in its dimension signature is None. For example, Beaverton x None x Overnight Express. None indicates that the is no dimension member from a particular dimension in the dimension signature.

You can create a No-dimension account with the New Account dialog by selecting No <*dimension-member*>, as the following picture shows:



The following picture shows a number of No-dimension accounts:



Display Name	IntsctnName
Cost Object (Primary Pane)	
USA	USA x All x All
Oregon	Oregon x All x All
Beaverton	Beaverton x All x All
No <Channel>	Beaverton x No <Channel> x All
2nd Day Guaranteed	Beaverton x No <Channel> x 2nd Day Guaranteed
Overnight Express	Beaverton x No <Channel> x Overnight Express
Standard Ground	Beaverton x No <Channel> x Standard Ground
Drop Box	Beaverton x Drop Box x All
No <Products and Services>	Beaverton x Drop Box x No <Products and Services>
2nd Day Guaranteed	Beaverton x Drop Box x 2nd Day Guaranteed
Overnight Express	Beaverton x Drop Box x Overnight Express
Standard Ground	Beaverton x Drop Box x Standard Ground
Walk In	Beaverton x Walk In x All
No <Products and Services>	Beaverton x Walk In x No <Products and Services>
2nd Day Guaranteed	Beaverton x Walk In x 2nd Day Guaranteed
Overnight Express	Beaverton x Walk In x Overnight Express
Standard Ground	Beaverton x Walk In x Standard Ground
Commercial Pick-up	Beaverton x Commercial Pick-up x All
No <Products and Services>	Beaverton x Commercial Pick-up x No <Products and Services>
2nd Day Guaranteed	Beaverton x Commercial Pick-up x 2nd Day Guaranteed
Overnight Express	Beaverton x Commercial Pick-up x Overnight Express
Standard Ground	Beaverton x Commercial Pick-up x Standard Ground
Eugene	Eugene x All x All

Syntax

dimension-ref`.IsNone()`

Examples

```
"Product".IsNone()
"Channel".IsNone()
```

IsNull function

Returns true or false depending on whether the leading significant digits match.

Syntax

`IsNull(` *expression* `)`

Examples

```
IsNull(SoldQuantity)
```

Further explanation

Null values are considered to be false when evaluating a formula for a rule-based driver.

Match function

Returns true when a *string* matches a given *pattern*. All comparisons are case *insensitive*.

Syntax

`Match(` *string*, "*pattern*" `)`

Examples

```
Match( Product.Reference, "*or*" ) matches or, ord, facilitator, but not asteroid
Match( Product.DimMemRef, "abc*" ) matches abc, abcd, but not asteroid, a, ab or bc
Match( Product.Reference, "*xyz" ) matches xyz, wxyz, but not asteroid, x, y, z, or yz
Match( Product.DimMemRef, "*bil*ion" ) matches billion, abillion, but not billions
Match( Product.Reference, "???DEF" ) matches abcDEF, ABCDEF, but not DEF, aDEF, abDEF
Match( Product.DimMemRef, "\*bike\*" ) matches *bike*, *Bike*, but not *bikes*
```

Further explanation

Special characters:

*	matches zero or more characters
?	matches any single character
\	used as an escape character to allow matching a literal '*', '?' or '\' in the string. The escape character says that the immediately following character in the pattern (*, ?, or \) is being used as itself and not as a special character. So, for example, Match(Product, Name\?) matches Name? and does not match either Name or Names

Note: When used in the assignment rule for a rule-based driver, properties are assumed to refer to the destination account unless Source is specified.

Chapter 53

Numeric Functions

abs function

The abs function returns the absolute value of a number.

Syntax

`abs(x)`

where *x* is the number.

degrees function

The degrees function converts radians into degrees.

Syntax

`degrees(x)`

where *x* is the radians.

Examples

```
degrees(1.57079633) = 90
```

exp function

The exp function returns e raised to the power of the number. The constant e equals 2.71828182845904, the base of the natural logarithm.

Syntax

```
exp(x)
```

where *x* is the number.

Examples

```
exp(2) = 7.389056099
```

max function

The max function returns the larger number of two numbers, or the greater of two strings.

Note: One string is greater than another if it comes after the other in UTF-16 sorting sequence (more simply stated, if it comes later in alphabetical order).

Syntax

```
max(x , y)
```

where *x* and *y* are two numbers or two strings.

min function

The min function returns the smaller number of two numbers, or the smaller of two strings.

Syntax

```
min(x , y)
```

where *x* and *y* are two numbers or two strings.

pi function

The pi function returns the number 3.14159265358979, the mathematical constant pi, and is accurate to 15 digits.

Syntax

```
pi( )
```

There are no parameters.

Examples

```
pi( ) = 3.14159265358979
```

power function

The power function returns the result of a number raised to a power.

Syntax

```
power(x , y)
```

x raised to the power of *y*

Examples

```
power(3,2) = 9
```

quotient function

The quotient function returns the integer portion of a division. Use this function when you want to discard the remainder of a division.

Syntax

```
quotient(x , y)
```

Examples

```
quotient(17,3) = 5
```

radians function

The radians function converts degrees to radians.

Syntax

```
radians(x)
```

x is the degrees.

Examples

```
radians(90) = 1.57079633
```

round function

The round function returns a number that has been rounded to a specified number of decimal places.

Syntax

```
round(x , y)
```

x is the number.

y is the number of decimal places. *y* can be any of the following:

Greater than zero	x is rounded to the specified number of decimal places.
Zero	x is rounded to the nearest integer.
Less than zero	x is rounded to the left of the decimal point.

sign function

The sign function returns a number that indicates the sign of a number.

Syntax

```
sign(x)
```

x is the number.

Return values

1	indicates that the number is positive.
Zero	indicates that the number is zero.
-1	indicates that the number is negative.

sqrt function

The sqrt function returns the positive square root of a number.

Syntax

```
sqrt(x)
```

x is the positive number. A negative number produces an error message.

trunc function

The trunc function truncates a number to an integer by removing the fractional part of the number.

Syntax

```
trunc(x)
```

x is the number to truncate.

Examples

```
trunc(5.46) = 5
```

Chapter 54

String Functions

& function (string concatenation)

Returns the concatenation of two strings.

Syntax

```
string & string
```

Examples

```
"Las Vegas" & "," & "Nevada"
Product.DimMemRef & " & " & Source.DimMemRef
```

find function

Returns the position of *find_string* in *within_string*, starting at *pos* (*pos* starts at 1).

Syntax

```
find( find_string, within_string, pos )
```

Examples

```
find(Source.DimMemRef, "x")
```

Further explanation

If *find_string* isn't found in *within_string*, a Null is returned.

If *pos* is less than 1 or larger than the length of *within_string,* a Null is returned.

If *find_string* is an empty string, then 1 is returned.

All comparisons are case sensitive.

Note: Null values are considered to be false when evaluating a formula for a rule-based driver.

left function

Returns a string with the leftmost length characters of string.

Syntax

```
left( string, length )
```

Examples

```
left(Source.Name, 5)
left(Source.Name, find(Source.Name, " and")-1)
```

Further explanation

If *length* is greater than the length of *string*, then *string* is returned. .

len function

Returns the number of characters in a string.

Syntax

```
len( string )
```

Examples

```
len(Source.Channel.Reference)
```

mid function

Returns a substring starting at pos with a length of length (pos starts at 1).

Syntax

```
mid( string, pos [, length] )
```

Examples

```
mid(Source.Channel.Reference,3)
mid(Channel.Reference, find(Channel.Reference, " and"),
    len(Channel.Reference)-find(Channel.Reference)-1)
```

Further explanation

If *length* is not included then all characters starting from *pos* to the end of the string are returned. .

right function

Returns a string with the rightmost length characters of string.

Syntax

```
right( string, length )
```

Examples

```
right(Source.Channel.Reference,3)
right(Channel.Reference, find(Channel.Reference, " and"),
   len(Channel.Reference)-find(Channel.Reference)-1)
```

Further explanation

If *length* is greater than the length of *string*, then *string* is returned. .

str function

Converts a number to a string.

Syntax

```
str( number [, size [, dec]] )
```

Examples

```
str(Destination.Cost)
str(len(Source.IdleQuantity))
```

Further explanation

The default parameter values are size=16 (includes the decimal point when dec > 0) and dec=6 for the number of digits past the decimal. If size is passed but dec is not, then *dec* defaults to a value of zero instead. If the *number* is larger than the *size* passed then more space will be created for it, so the string returned may be longer than the size passed. Passing a size of "1" removes all leading spaces from the number. .

trim function

Returns the string with all spaces removed except for single spaces between words.

Syntax

```
trim ( string )
```

Examples

```
trim(Destination.Name)
```

value function

Converts a string to a number.

Syntax

`value( string )`

Examples

```
value(Destination.DimLevel)
```

Further explanation

If a non-numeric character is encountered the conversion stops. An empty string evaluates to zero. .

Chapter 55

Properties That Can Be In Formulas

Properties That Can Be in Formulas

The following table identifies all the properties that can be used in formulas. The Assignment column applies to assigned, internal and external cost elements.

Property	Type	Account	Assign-ment	Dim Member	Comment
[Value Attributes]	Num or Char	✓			Text and Tag attributes can be used in addition to numeric attributes.
AllocatedCost	Num	✓	✓		
AssignedCost	Num	✓			
AssignedIdleCost	Num	✓	✓		
AssignedIdleQuantity	Num	✓			Valid property for assignment, but not accessible from formula because of circular reference.
AssignedNonReciprocalCost	Num	✓			
AssignedReciprocalCost	Num	✓			
Cost	Num	✓	✓		
DimLevelName	Char			✓	
DimLevelNumber	Num			✓	
DimMemberName	Char			✓	Synonym of Name for Dim Member in an intersection
DimMemberRef	Char			✓	Synonym of Reference for Dim Member in an intersection

DrivableCost	Num	✓			
DrivenCost	Num	✓	✓		
DrivenQuantity	Num	✓			Valid property for assignment, but not accessible from formula because of circular reference.
DriverName	Char	✓			
DriverQuantityCalculated	Num		✓		
DriverQuantityFixed	Num		✓		
DriverQuantityVariable	Num		✓		
DriverRate	Num	✓	✓		
DriverWeightFixed	Num		✓		
DriverWeightVariable	Num		✓		
EnteredCost	Num	✓			
IdleCost	Num	✓	✓		
IdleDriverQuantityUE	Num		✓		Was ”IdleDriverQuantity_UE” in 6.4
IdleQuantity	Num	✓			
Name	Char	✓		✓	
OutputQuantity	Num	✓			Unavailable as Source property from assignment
OutputQuantityUE	Num	✓			
Profit	Num	✓			
ReceivedAllocatedCost	Num	✓			
ReceivedCost	Num	✓			
ReceivedDrivenCost	Num	✓			
ReceivedNonReciprocalCost	Num	✓			
ReceivedReciprocalCost	Num	✓			
Reference	Char	✓		✓	
Revenue	Num	✓			
SoldCost	Num	✓	✓		
SoldQuantity	Num	✓			

TDQ	Num	✓			Unavailable as Source property from assignment
TDQCalculated	Num	✓			Unavailable as Source property from assignment
TDQUE	Num	✓			
UnassignedCost	Num	✓			
UnitCost	Num	✓			
UnitProfit	Num	✓			
UnitRevenue	Num	✓			
UsedCost	Num	✓	✓		
UsedQuantity	Num	✓			

Part 16

Cube Configurations

Chapter 56

What is a Cube Configuration?

Overview

A cube configuration holds the options that you select for generating cubes and fact tables. After creating a cube configuration, you use it to generate cubes. Creating a cube configuration is preliminary to generating cubes—you don't generate cubes until later.

For information, see "Create a Cube Configuration" on page 507.

Chapter 57

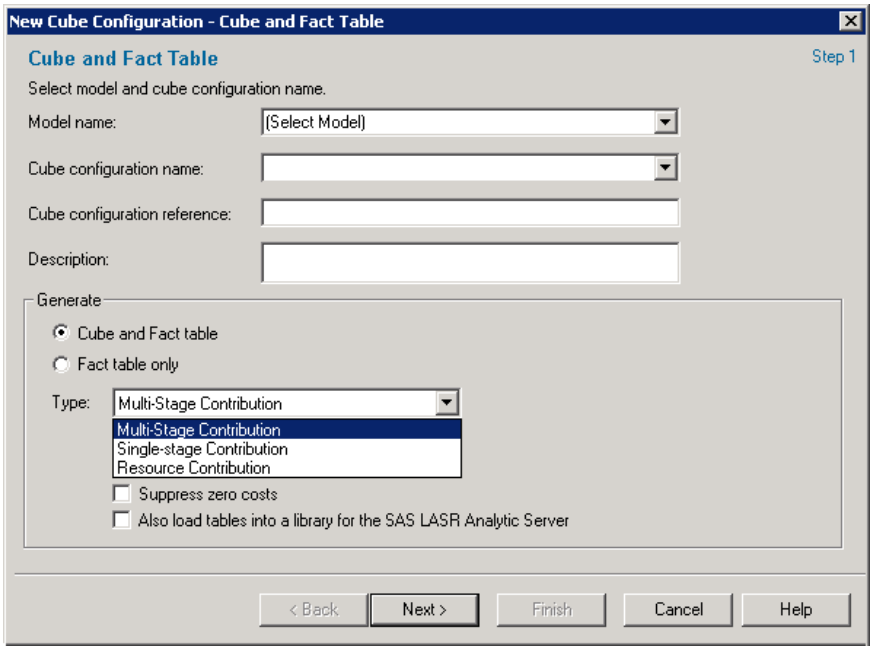
Windows for Cube Configurations

New Cube Configuration

Overview

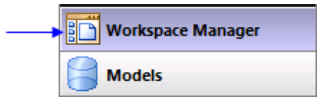
Use the New Cube Configuration wizard to create a cube configuration that remembers your options for creating a cube and fact table from a particular model.

For information, see "Create a Cube Configuration" on page 507.



How to Access the New Cube Configuration Dialog Box

1. Go to the Workspace Manager.



2. Select **File** ⇨ **New** ⇨ **Cube Configuration**.

   The New Cube Configuration wizard opens.

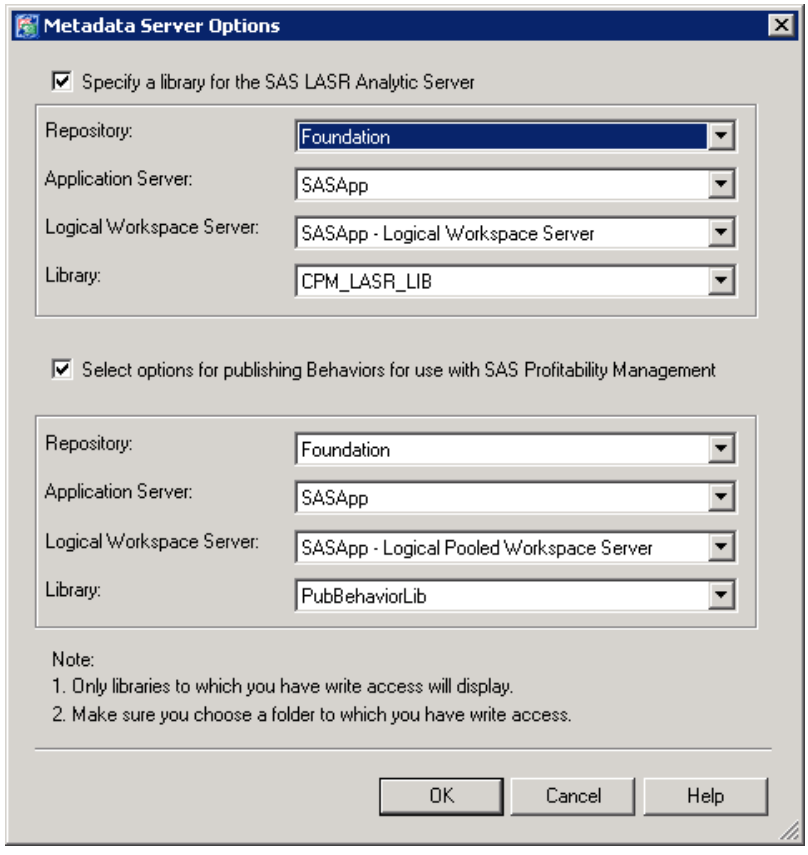
Metadata Server Options

About Metadata Server Options

Use the Metadata Server Options dialog box to specify a metadata library:

- for use with the SAS LASR Analytic Server.
- for publishing data to use as behaviors with SAS Profitability Management

Note: The settings displayed in the Metadata Server Options dialog box are determined during installation of SAS Cost and Profitability Management. However, you can use the Metadata Server Options dialog box to change these settings after installation.



After specifying a metadata library:

for the SAS LASR Analytic Server

- Generate a cube for a model whose cube configuration specifies output for the SAS LASR Analytic Server. See "Cube Configuration: Select a Model and General Options" on page 508. The output of cube generation goes to the metadata library that you have specify here in Metadata Server Options..
- You can export other tables that are registered in a SAS metadata folder that is enabled for the SAS LASR Analysis Server. See Chapter 82, "Export Registered Tables," on page 695.

For information on creating a metadata library that is enabled for the SAS LASR Analytic Server, see Chapter 23, "Working with SAS LASR Analytic Server," in *SAS Cost and Profitability Management: Data Administration 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" in Chapter 23 of *SAS Cost and Profitability Management: Data Administration Guide*.

for SAS Profitabililty Management

To publish behaviors to SAS Profitabililty Management, select **File** ⇨ **Publish** ⇨ **Behaviors**. The output goes to the metadata library that you have specified. See "Publish Behaviors to SAS Profitability Management" on page 731.

How to Access the Metadata Server Options Dialog Box

Select **Tools** ⇨ **Metadata Server Options**.

Chapter 58

How To: Cube Configurations

Create a Cube Configuration

Overview

A cube configuration holds the options that you select for generating cubes and fact tables. After creating a cube configuration, you use it to generate cubes. Creating a cube configuration is preliminary to generating cubes—you don't generate cubes until later.

1. Go to the Workspace Manager or to the Resource, Activity, or Cost Object module for a model.
2. Select **File** ⇨ **New** ⇨ **Cube Configuration**.

   The New Cube Configuration wizard opens.
3. Depending on the type of cube that you want to generate, complete the steps in the following table:

Type of Contributions Cube	Steps to Create the Cube Configuration
Single-stage and Resource	• Step 1: "Cube Configuration: Select a Model and General Options" on page 508 • Step 2: "Cube Configuration: Cube Options" on page 514 • Step 3: "Cube Configuration: Select Numeric Attributes" on page 517 • Step 4: "Cube Configuration: Finish" on page 517

Type of Contributions Cube	Steps to Create the Cube Configuration
Multi-Stage	• Step 1: "Cube Configuration: Select a Model and General Options" on page 508 • Step 2: "Cube Configuration: Cube Options" on page 514 • Step 3: "Cube Configuration: Options for a Multi-Stage Contribution Cube" on page 513 • Step 4: "Cube Configuration: Select Numeric Attributes" on page 517 • Step 5: "Cube Configuration: Finish" on page 517

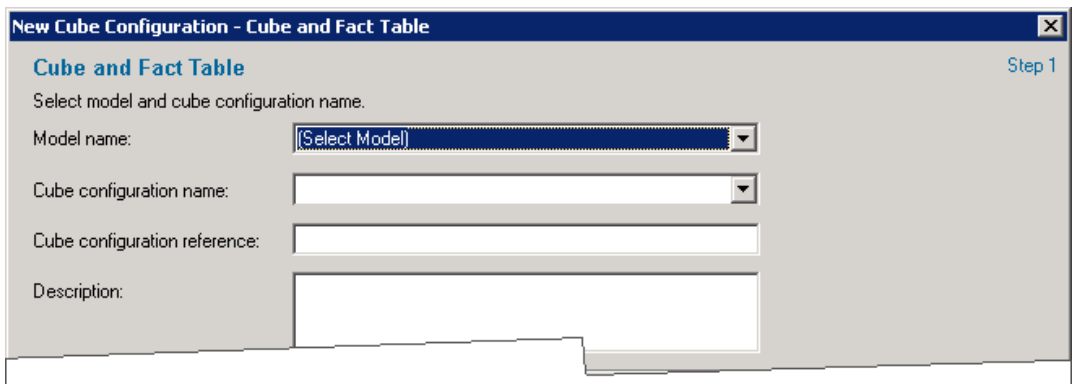
See Also

"Generate Cubes" on page 573

Cube Configuration: Select a Model and General Options

In this step of creating a cube configuration you select the model, name the cube configuration, and select other options for cube generation.

Select a Model and Name the Cube Configuration



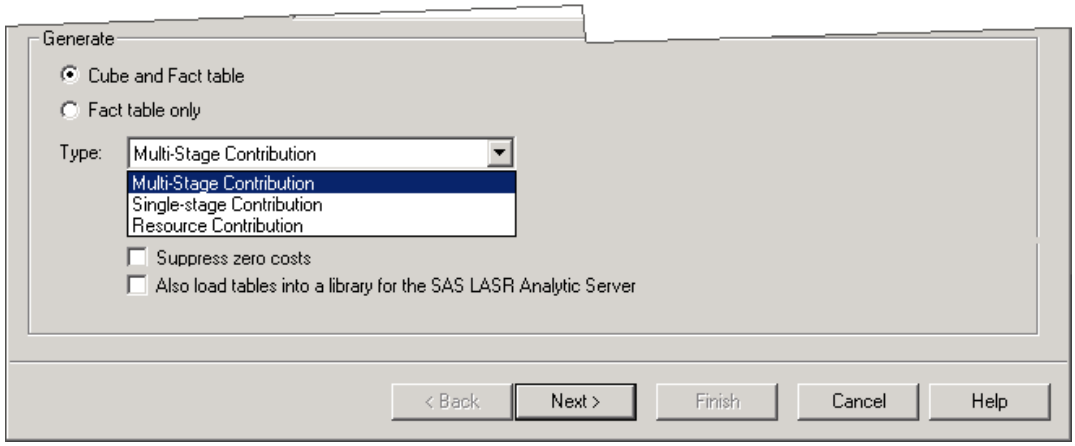
Model name
: Select the model whose cube and/or fact table is to be generated. A model can have more than one cube configuration because different cubes can be generated from the same model.

Cube configuration name
: You will use the cube configuration later to generate cubes. In this step of the wizard, you assign a name that describes the cube to be generated.

You cannot use the following characters in the cube configuration name: ' (single quotation mark), " (quotation mark), / (slash), \ (backslash), or | (pipe).

Cube configuration reference
: The cube configuration reference is used in public views as an unchangeable identifier for the configuration.

Select the Type of Cube and Whether to Generate a Cube and Fact Table or a Fact Table Only



Cube and Fact table or Fact table only
: You can choose to generate both a cube and fact table, or only a fact table. You can use the fact table to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft SQL Server Analysis Services .

Type
: Multi-Stage Contribution (MSC)
: Analyzes the cost contributions to or from accounts that have stages attributes.

You can generate different multi-stage contribution cubes from the same model. A cube configuration remembers your selections for a particular cube and for a particular model.

Note: You can copy an existing cube configuration and change only the model with which the cube configuration is associated. See Copy a Cube Configuration to a Different Model.

Single-Stage Contribution (SSC)
: Analyzes the cost contributions from one assignment level back.

Resource Contribution (RC)
: Analyzes the costs contributions from original accounts where costs were entered for final accounts that do not assign costs to other accounts.

Suppress zero cost
: Suppress items that have no associated costs to save generation time and reduce cube size.

Note: This option applies only to a Multi-Stage Contribution cube.

Also load tables into a library for the SAS LASR Analytic Server
: The following tables are published to the database whenever you generate. If you select this option, then these tables are, in addition, published to and registered in the SAS metadata folder that you specified with Metadata Server Options. See "Metadata Server Options" on page 504.

View Name	Description
*ModelRef*_PD_COSTELEMENT<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> 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 “modelRef_PD_COSTELEMENT<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_*dimShortRef*<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_REGIONRC ABC_PD_REGIONSSC ABC_PD_REGIONC1001	Single dimension-member details: level by level, noting ID, Reference, and Name. See “modelRef_PD_dimShortRef<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_DRIVER<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_DRIVERRC ABC_PD_DRIVERSSC ABC_PD_DRIVERC1001	Driver ID and corresponding Driver Name. See “modelRef_PD_DRIVER<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_MODULE<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_MODULERC ABC_PD_MODULESSC ABC_PD_MODULEC1001	Details defining the types of modules. See “modelRef_PD_MODULE<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

View Name	Description
*ModelRef*_PD_PERIOD<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_PERIODRC ABC_PD_PERIODSSC ABC_PD_PERIODC1001	Details defining the periodic hierarchy. See “modelRef_PD_PERIOD<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_SCENARIO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_SCENARIORC ABC_PD_SCENARIOSSC ABC_PD_SCENARIOC1001	Details defining the types of available scenarios. See “modelRef_PD_SCENARIO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_YESNO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_YESNORC ABC_PD_YESNOSSC ABC_PD_YESNOC1001	Dimensional definition for Boolean values: Text strings. See “modelRef_PD_YESNO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

View Name	Description
*ModelRef*_PF_<**suffix**> where <**suffix**> is RC, SSC, or MSC_C<cubeID> for a multi-stage contributions cube. For example: ABC_PF_RC ABC_PF_SSC ABC_PF_MSC_C1001 If you have checked **Also load tables into a library for the SAS LASR Analytic Server**, 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: *ModelRef*_PF_<**suffix**> where <**suffix**> is RCSTAR, SSCSTAR, or MSC_C<cubeID>STAR for a multi-stage contributions cube. For example: ABC_PF_RCSTAR ABC_PF_SSCSTAR ABC_PF_MSC_C1001STAR	Fact Table: Stages and member IDs for each step through contribution. Source table for cube generation. See “modelRef_PF_<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide* .

Note: Suppose that you selected the **Also load tables into a library for the SAS LASR Analytic Server** check box, the star-schema and all associated tables are loaded into a library for the SAS LASR Anlaytic Server, and the SAS LASR Analytic Server goes down. When the SAS LASR Analytic Server is running again, all the tables (including the star-schema) that are present in the library for the SAS LASR Anlaytic Server are unloaded. To reload the star-schema and associated table into the library for the SAS LASR Anlaytic Server, perform the following steps:

1. Ensure that the SAS LASR Analytic Server is running.
2. On the **Tools** menu, select **Operations Summary**.
3. In the **Operations Summary** dialog box, select the required operation that contains the information about the cube generation and click **View**. The **Summary** dialog box displays a summary of the cubes that are generated.
4. Copy the SAS code that is under the heading **CODE FOR PUSH TO LASR** for the required cube from the **Summary** dialog box into a text file.
5. Format the SAS code in the text file so that it can run on a SAS environment. Each SAS statement must end with a semicolon and the subsequent SAS statement must start on a new line.
6. Add the following statement at the beginning of the SAS code and then run the formatted SAS code in the SAS environment to load the cube-related tables into the library for the SAS LASR Analytic Server: *options metaserver="<<server_name>>" metauser="<<username>>"*

port=<<port_number>> metapass="<<password>>"
metarepository="<<FOUNDATION>>" ;

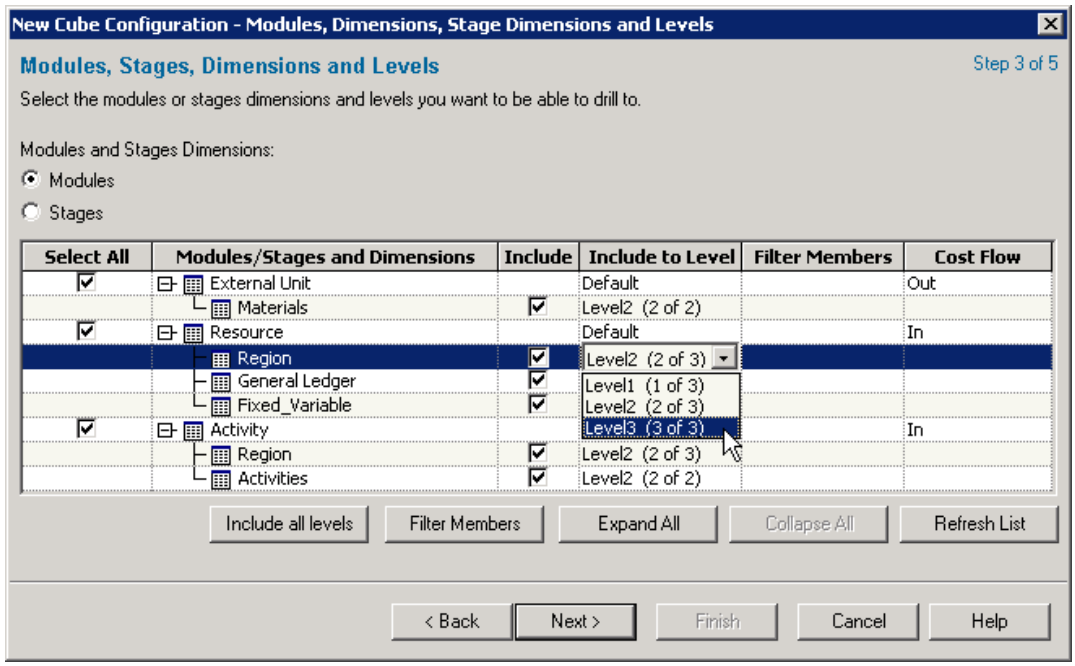
In addition to these tables, you can export other tables that are registered in a SAS metadata folder that is enabled for the SAS LASR Analytic Server. See Chapter 82, "Export Registered Tables," on page 695.

See Also

Chapter 66, "How To: Generating and Managing Cubes," on page 575

Cube Configuration: Options for a Multi-Stage Contribution Cube

The following dialog box appears only for a multi-stage contribution cube.



Modules and Stages

Modules

Each module defines a single stage (one stage per module) in the order in which you defined your modules, or in the following order for a traditional activity-based costing model:

1. External Units
2. Resource
3. Activity
4. Cost Object

Stages

Each stage is defined by a dimension member attribute in a dimension attribute that is named Stages. For more information, see Stage attributes and Adding Stage Attributes to Accounts .

Select All
Check this box to select all dimensions in a module or stage for inclusion in the cube. Selecting this check box turns on the Include flag for each individual dimension.

Include
Check this box to include the selected dimension in the cube.

Cost Flow
Select whether to include cost flows into or out of the selected module or stage.

With some models, it makes a difference in generating a cube whether you select to show costs flowing into a module/stage or out of a module/stage. It makes a difference in case the model has assignments from accounts in a module/stage to accounts within the *same* module/stage. You cannot choose to show both assignments because that would result in double-counting final costs.

If a model has no cost assignments from accounts within a module/stage to accounts within the same module/stage, then the choice of showing cost flows into or out of a module/stage makes no difference (every cost into a module/stage is also a cost out of the module/stage). By defining multiple stages such that there are no assignments within a single stage, you avoid having to make the choice of showing cost flows in or out.

Include to Level
For dimensions with multiple levels, specify how many levels you want included in the cube for drill down. The default is to include up to and including level 2 of a dimension.

Note: Click **Include All Levels** to select all levels for inclusion in the cube. After clicking this button you can override the depth for particular dimensions so as to include fewer levels.

Filter Members
Click this button to select the dimension members to be included in the generated cube. You can select dimension members in any level of the dimension hierarchy up to and including the level specified in **Include to Level**. See "Select Dimension Members for Inclusion in Cubes" on page 519.

Include All Levels
Click this button to select all levels for inclusion in the cube. Clicking the button has the same effect as manually selecting the lowest level of each dimension for inclusion. After clicking the button you can override the depth for particular dimensions so as to include fewer levels.

See Also

- "Create a Cube Configuration" on page 507
- Chapter 66, "How To: Generating and Managing Cubes," on page 575

Cube Configuration: Cube Options

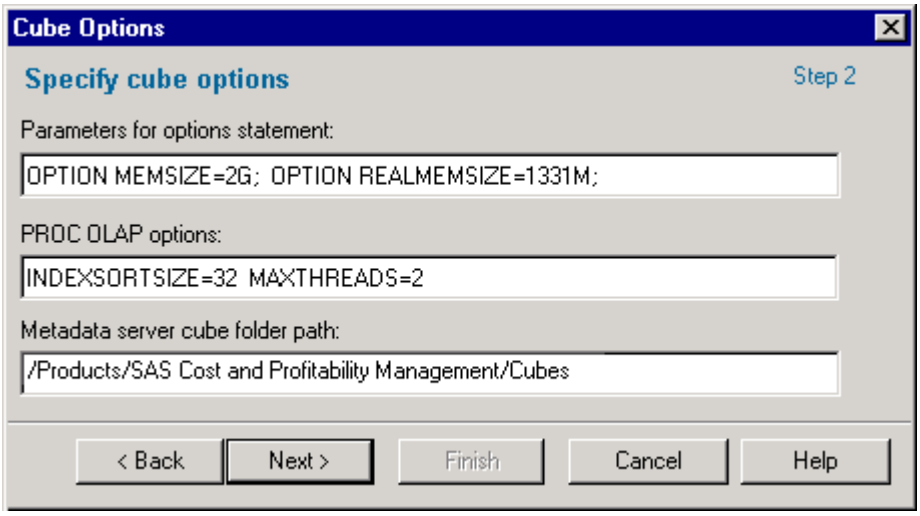
Overview

Your options for creating a cube depend on whether you are using:

- SAS OLAP on page 515
- Microsoft SQL Server Analysis Services on page 516

Cube Options for SAS OLAP

Note: The following picture is only an example. By default you do not have to specify any options to generate a cube. Indeed, you should not specify any options unless you have experience with SAS programming.



Parameters for options statement

By default you do not have to specify any options for cube generation. However, if you want to override the default SAS system options, you can specify them here. The following are some sample options:

```
OPTION NOSYNTAXCHECK;
OPTION SORTPGM=SAS;
OPTION MEMSIZE=2G;
OPTION REALMEMSIZE=1331M;
OPTION SUMSIZE=1331M;
OPTION BUFSIZE=64K;
OPTION IBUFSIZE=32767;
OPTION UBUFSIZE=64K;
OPTION SORTSIZE=512M;
OPTION CPUCOUNT=2;
OPTION NOOVP;
```

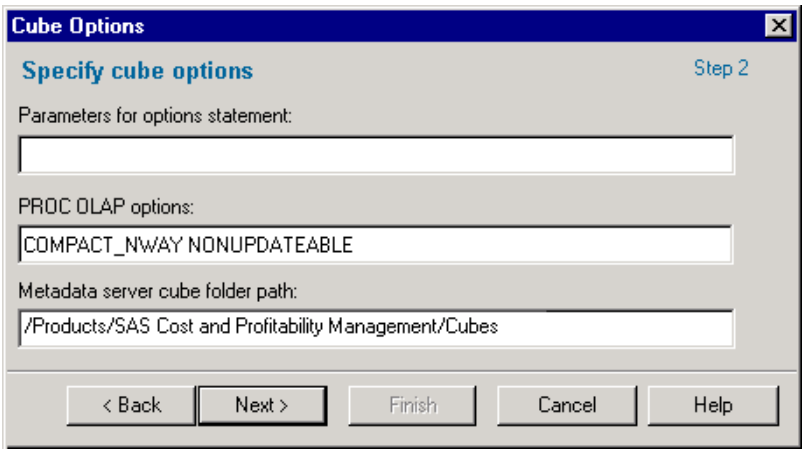
For all system options, see SAS System Options in the SAS 9.2 Language Reference: Dictionary.

PROC OLAP options

SAS Cost and Profitability Management uses the SAS OLAP procedure to generate cubes. By default you do not have to specify any PROC OLAP options for cube generation. However, if you want to override the default OLAP options, you can specify them here. The following are two sample options:

```
INDEXSORTSIZE=32 MAXTHREADS=2
```

For all PROC OLAP options, see The OLAP Procedure in the SAS OLAP Server: User's Guide (**`http://support.sas.com/documentation/onlinedoc/olap/index.html`**). The following two options can possibly save disk storage space by reducing cube size: COMPACT_NWAY and NONUPDATEABLE.



COMPACT_NWAY
: specifies that the cube build will include an additional summarization step that is designed to decrease the size of the NWAY aggregation and improve viewing performance. The amount of improvement depends on the nature of the data. The cubes that improve the most are those that have the largest

NONUPDATEABLE
: specifies that the dimension(s) should be built with the minimum amount of disk space to represent the members available when the cube is created. By default, new dimensions are built to allow for new members to be added in future updates. NONUPDATEABLE is valid only when the cube is first created.

Note: The following options are not supported: DRILLTHROUGH_TABLE=, WORKPATH=, DESCRIPTION=, IGNORE_MISSING_DIMKEYS=.

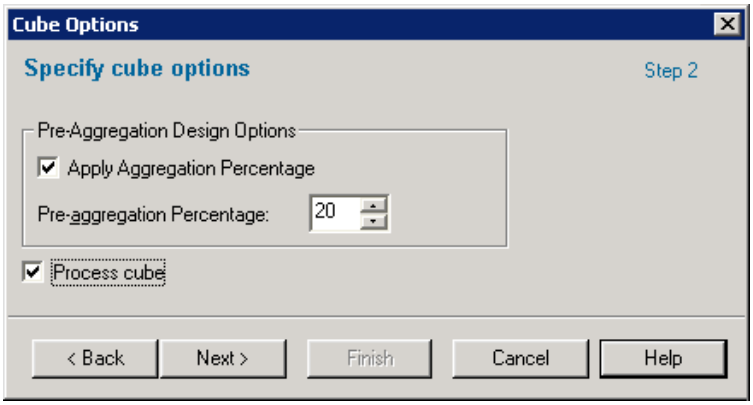
Metadata server cube folder path

If you want to override the default location for the generated cube, you can specify a path on the SAS Metadata Server.

By default, the path is: **`/Products/SAS Cost and Profitability Management/Cubes/`**. However, the default path could have been changed during installation. In either case you can override the path here.

Note: The folder that you specify must already exist when cube generation is started and you must have write access to the folder for cube generation to be successful.

Cube Options for Microsoft SQL Server Analysis Services



Pre-aggregation Percentage

Pre-aggregating numeric data results in faster performance when you work with a cube in the OLAP analyzer. But, for large models pre-aggregating data can take a

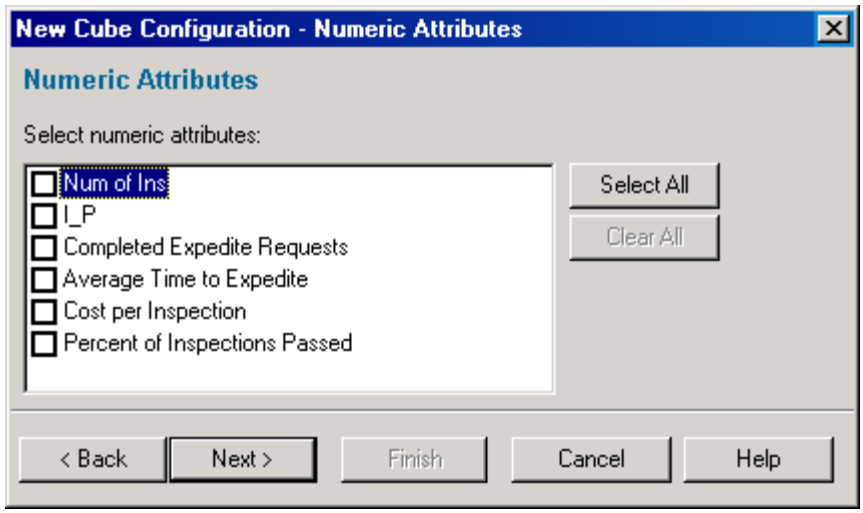
long time which increases cube-generation time. So, the more pre-aggregation you do, the faster you can navigate a cube, but the longer it takes to generate it. This option allows you to trade generation time for execution speed.

Process cube

Processing a cube causes processing of all the measure groups within the cube and the constituent dimensions that are currently in an unprocessed state. When you process a cube, an SQL query is issued to retrieve values from the fact table to populate each member in the cube with appropriate measure values. For any specific path to a node in the cube, there is a value or a calculable value. Processing a cube creates machine-readable files that store relevant fact data. If there are aggregations created, they are stored in aggregation data files.

Cube Configuration: Select Numeric Attributes

Select the numeric attributes to be included in the generated cube.



Note: You can select the numeric attributes that are to be checked by default in a new cube configuration by doing the following:

1. Open a model and select **Model** ⇨ **Properties**.
2. Select the **Attributes in Cube** tab.
3. Select the attributes that are to be checked by default in a new cube configuration for that model.

   Attributes that you select in Model Properties are automatically checked in a new cube configuration to be included in the generated cube. However, you can uncheck the attributes in the cube configuration and select others.

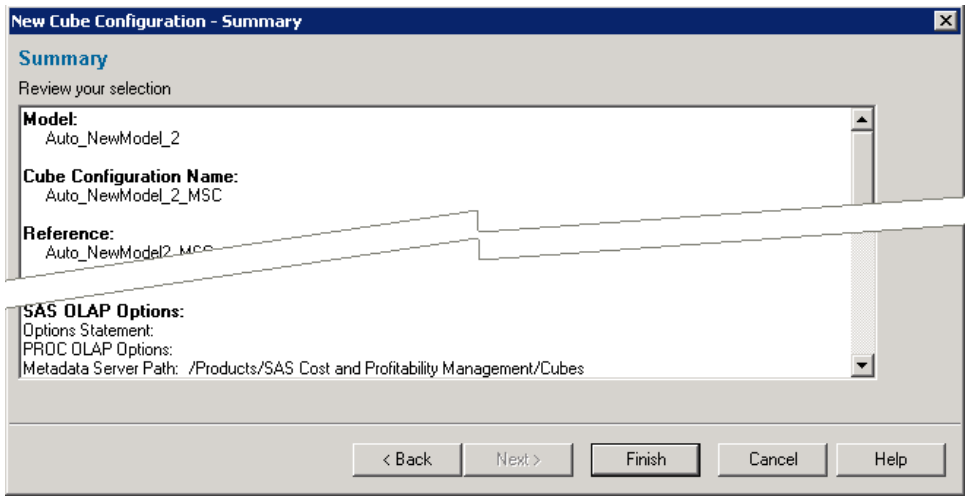
See Also

- "Create a Cube Configuration" on page 507
- "Include Numeric Attributes in a Cube" on page 577
- Chapter 66, "How To: Generating and Managing Cubes," on page 575

Cube Configuration: Finish

Review your selections; then click **Finish**.

The cube configuration is created. You can use it to generate cubes.



See Also

- "Create a Cube Configuration" on page 507
- Chapter 66, "How To: Generating and Managing Cubes," on page 575

Copy a Cube Configuration to a Different Model

A cube configuration is associated with one and only one model. You can copy a cube configuration, preserving all its specifications except the model with which it is associated. By doing this, you create a new cube configuration with all the specifications of the old one except for the model with which it is associated.

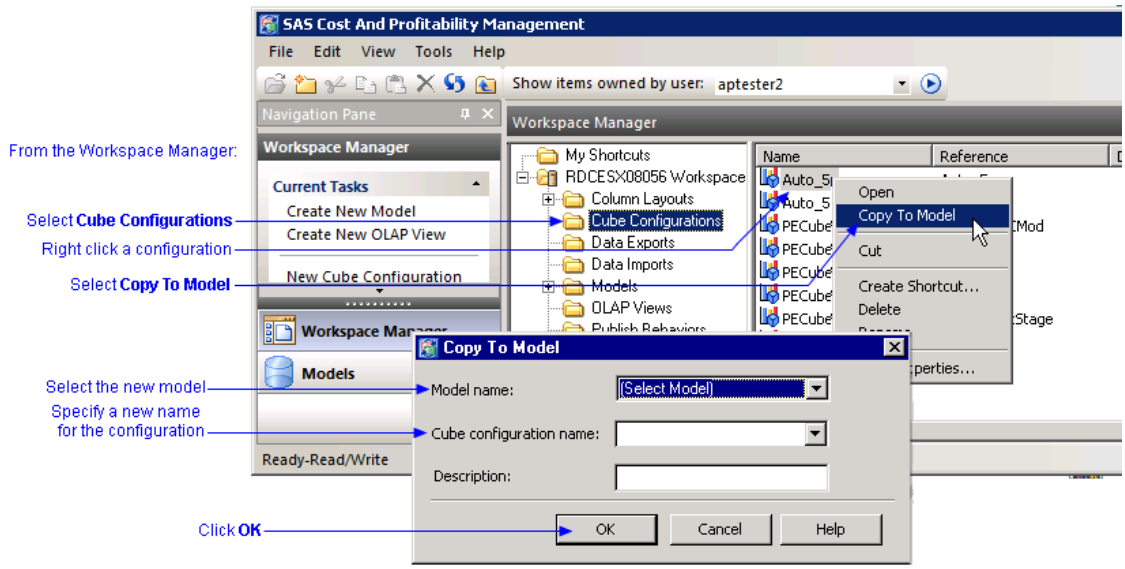
To make a copy of a cube configuration that is associated with a different model, do the following:

1. From the Navigation Pane, go to the Workspace Manager.
2. Select **Cube Configurations**.
3. Select **Copy To Model** from the pop up menu.

   The **Copy To Model** dialog box opens.
4. Select the new model to which the cube configuration is to be assigned.

   Note: The new model must have the same dimensions as the old one.
5. Specify the name of the new cube configuration.
6. Click **OK**.

   A new cube configuration is created with all the attributes of the old one, but associated with a different model.

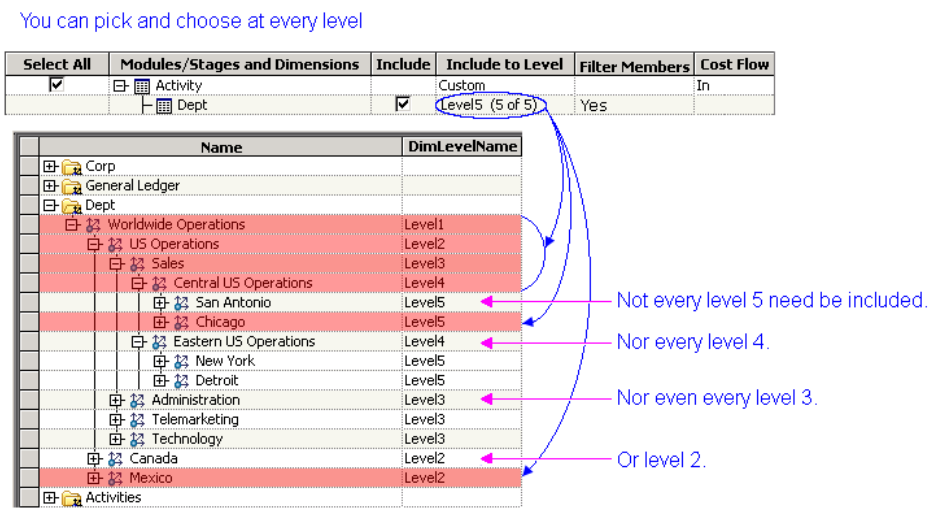


Select Dimension Members for Inclusion in Cubes

Overview

Not only can you choose what dimensions to include and up to what level to include their dimensions members. But, you can also choose at any level what individual dimension members to include. This ability allows you to create cubes that go deep into the dimension hierarchy and yet remain small.

The following picture shows selecting dimension members at a deep level, and yet creating a small cube because it contains fewer dimension members.



Selecting Dimension Members

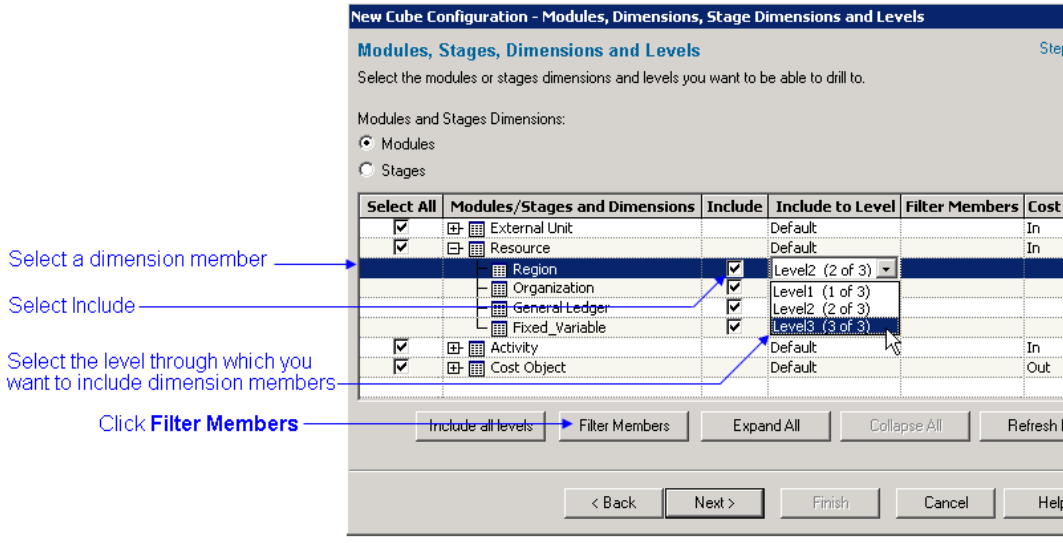
Note: If, instead of selecting individual dimension members, you want to include all dimension members at all levels, you can click **Include all levels**. Include all levels

To select individual dimension members to include in a generated cube, do the following:

1. Go to the Workspace Manager or to the module view for a model.
2. Select **File** ⇨ **New** ⇨ **Cube Configuration** or open an existing cube configuration.
3. For Step 1 of the Cube Configuration wizard, specify the following:
   - Model name
   - Cube configuration name
   - Cube configuration reference
   - Specify **Create cube and fact table**.
   - Specify **Multi-Stage contribution** as the type of cube to generate.

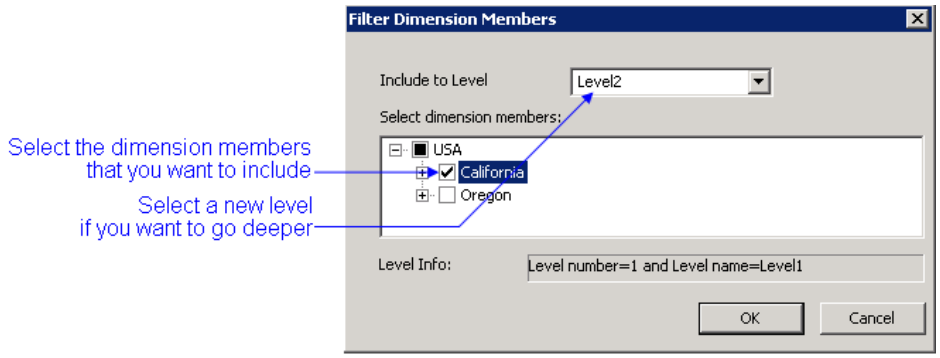
   Click **OK** to proceed to Step 2 of the Cube Configuration wizard.
4. For Step 2 of the Cube Configuration wizard, specify cube options.

   Click **OK** to proceed to Step 3 of the Cube Configuration wizard.
5. For Step 3 of the Cube Configuration wizard:
   a. Select a dimension member
   b. Select **Include** to include this dimension member in the generated cube.
   c. For **Include to Level**, select the level of the dimension hierarchy through which you want to include dimension members.
   d. Click **Filter Members**.



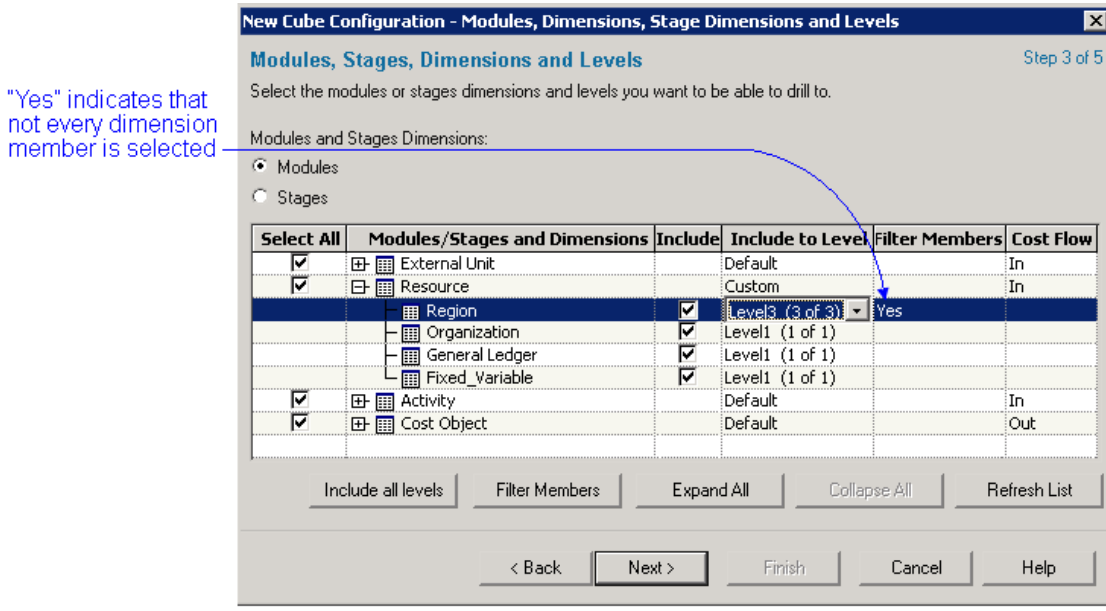
   The Filter Dimension Members dialog opens.
6. In the Filter Dimension Members dialog, select the dimension members that you want to include in the cube.

   Note: If you want to go deeper (or shallower) in the dimension hierarchy, use the Include to Level drop-down list to select a level. When you leave this dialog, the level is reflected in the previous Modules, Stages, Dimensions, and Levels window.



7. Click **OK**.

   The Filter Dimension Member dialog closes, and a "Yes" in the Filter Members column indicates that not every dimension member is selected for inclusion in the cube.



Note: Although dimension members that you deselect are not included in a cube, their costs are included in the None category.

Part 17

Calculation

Chapter 59

Calculating Costs

Overview

As you begin entering data from the paper model into the model in SAS Cost and Profitability Management, you should notice how the costs are conforming to expectations. At any time during the development of a model, you can assign the cost of each account according to the account's driver.

You can enter costs interactively. However, interactively entering data can be tedious and prone to errors. Importing data from an existing data source can minimize tedium and error. For example, if the costs from the general ledger are available, and if you have built the model so that the models resource account references match the general ledgers account numbers, it is more efficient to add costs to the model by importing them.

After you calculate costs, you might want to republish your data to SAS Strategy Management because your costs are not automatically updated.

To calculate the costs for a model, select **Model** ⇨ **Calculate Costs**.

The Calculate Costs dialog box appears.

Changing the Model Structure after Calculating Costs

After you have calculated costs for an entire model, any of the following actions can invalidate some of the model costs:

- Adding, deleting, or changing the cost of an entered cost element
- Changing an accounts driver, driver quantities, attribute quantities, or output quantities

- Adding or deleting a rollup account, an account, or a cost element
- Creating new assignments or deleting existing ones

In all of these cases, costs are updated if you calculate costs. However, it might not be necessary to calculate costs for a minor change.

Period/Scenario Association

When you calculate costs, you can choose a specific period/scenario association, or you can choose all period/scenario associations.

Errors

You should choose to report all errors at least once while you are calculating costs and generating cubes. Fix any problems that are reported. If you determine that the remaining errors are insignificant, you can choose to not report all errors. Then, you can limit the number of errors that are reported. If you limit the number of errors, performance might improve when costs are calculated and when cubes are generated.

Chapter 60

Properties in Calculation

Types of Properties

Introduction

Model properties that are involved in calculation fall into catagories that are represented by the following diagrams:

- "Diagram 1: Cost Flowing Into and Out of an Account" on page 527
- "Diagram 2: Properties of an Assignment" on page 528
- "Diagram 3: Account Driver Properties " on page 529

This chapter begins by presenting the diagram for each category of properties. Then, in the subsequent sections, the interrelationships among the diagrams are examined in detail.

See Also

"How to Read these Diagrams" on page 810

Diagram 1: Cost Flowing Into and Out of an Account

The following diagram shows costs flowing into and out of a single account. This is the same diagram as shown in "Cost Flowing Into and Out of an Account (Non-Reciprocal)" on page 804 with the addition of numbering.

- The diagram is for non-reciprocal cost flow.

- The column in the center shows the total cost in the account. Cells to the left show cost flowing into the account. Cells to the right show cost flowing out.
- The numbers in the cells show the logical sequence for the calculation of properties. The calculation of a property with a higher number depends upon the prior calculation of certain properties with lower numbers. These numbers are used later in this chapter to label descriptions of the individual calculations. Letters are used (for example, 2a and 2b) to indicate that either can be calculated first.

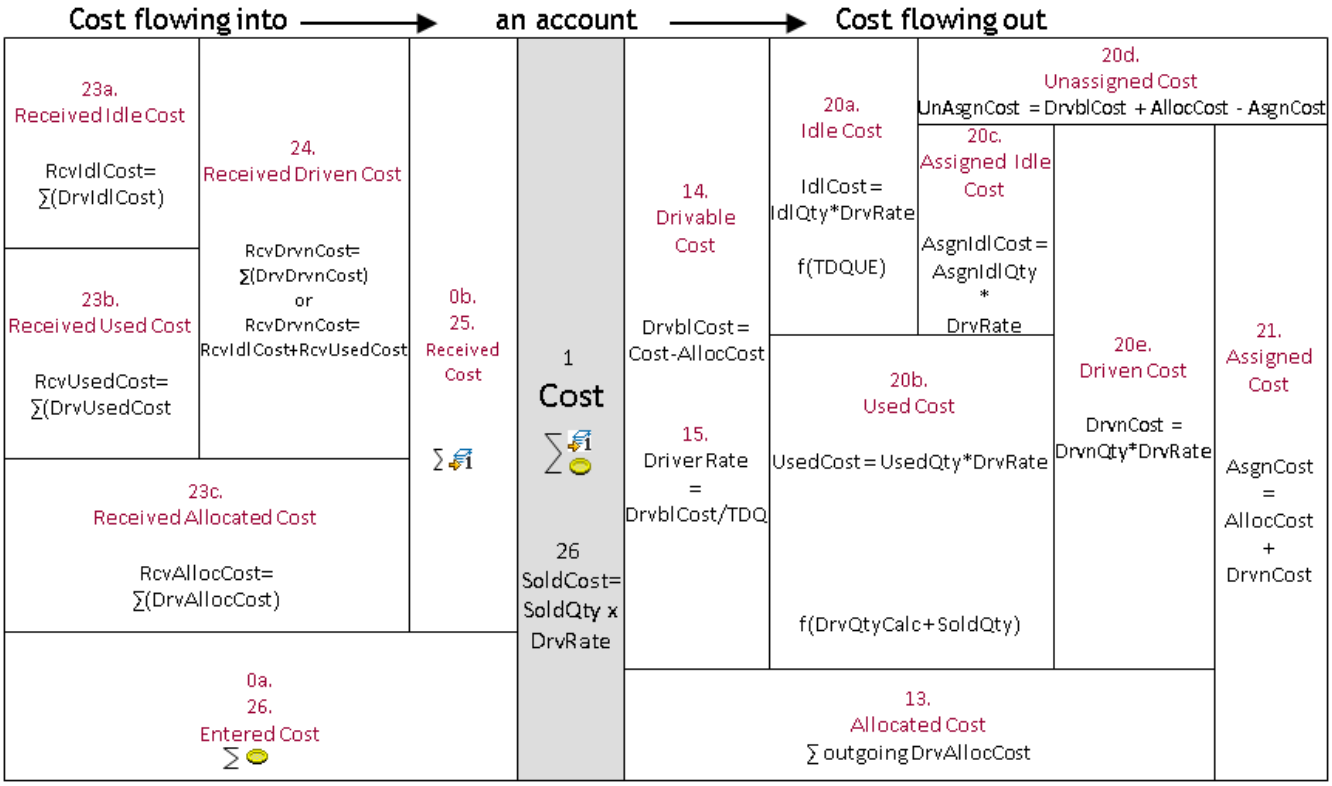


Diagram 2: Properties of an Assignment

The following diagram shows the properties associated with an individual assignment. This is the same diagram as shown in "Assignment Properties " on page 807 with the addition of numbering.

Note: This diagram does not apply to assignments made with a calculated driver. In the case of a calculated driver, DrvQtyCalc is derived from a user-specified formula.

Note: Although the properties are prefixed with the word "Driver" (or include it, as in the case of IdlDrvQty), they are all properties of an individual assignment.

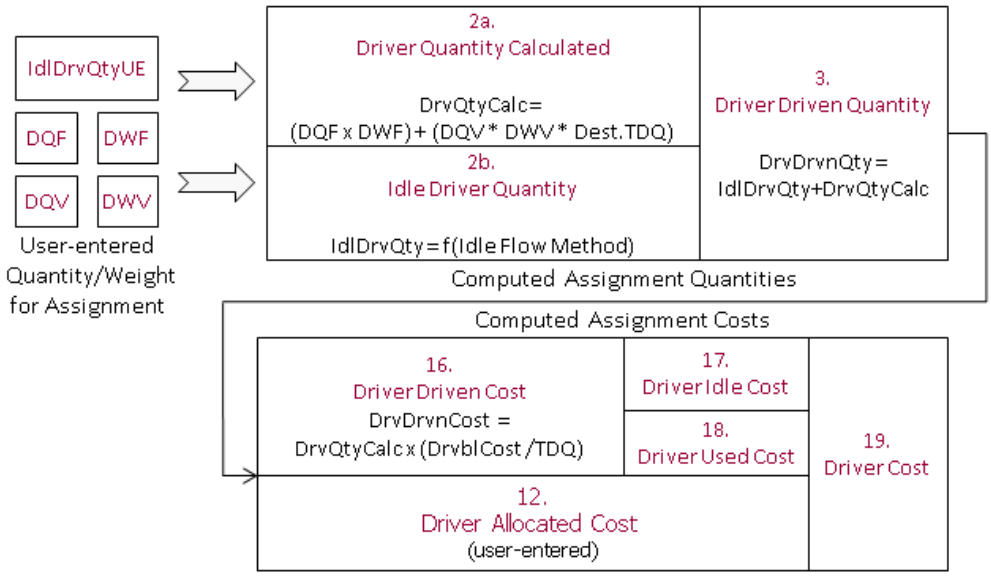
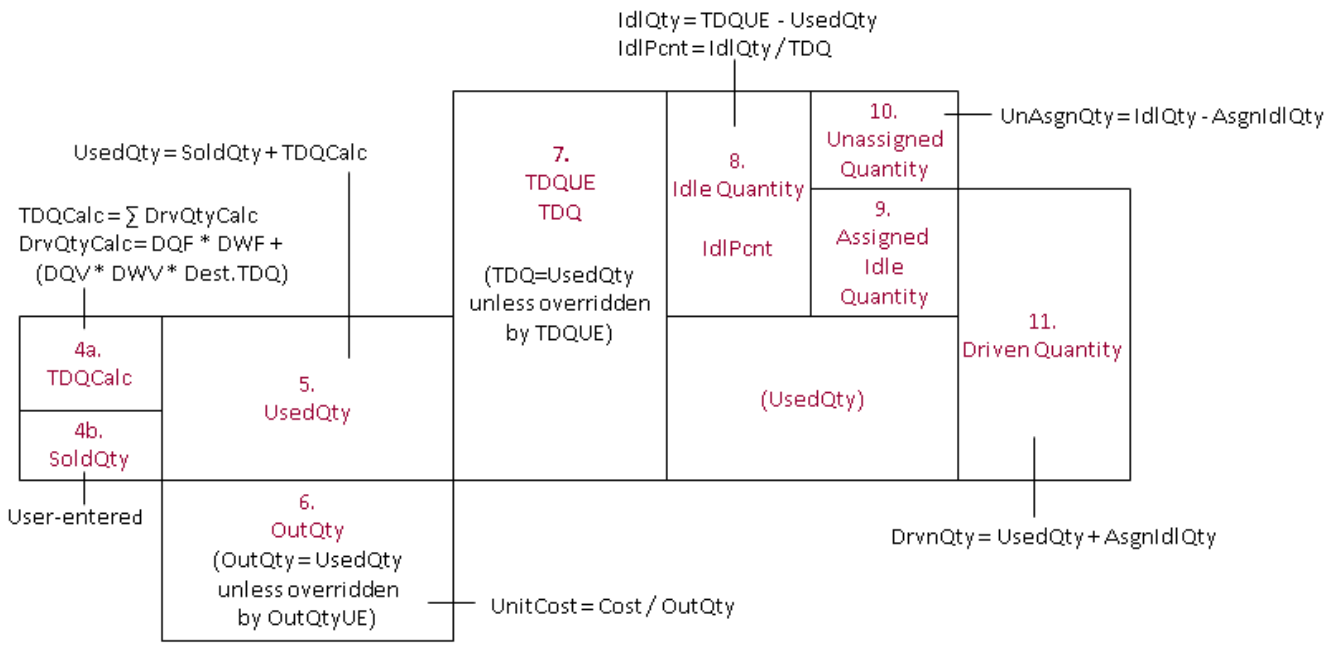


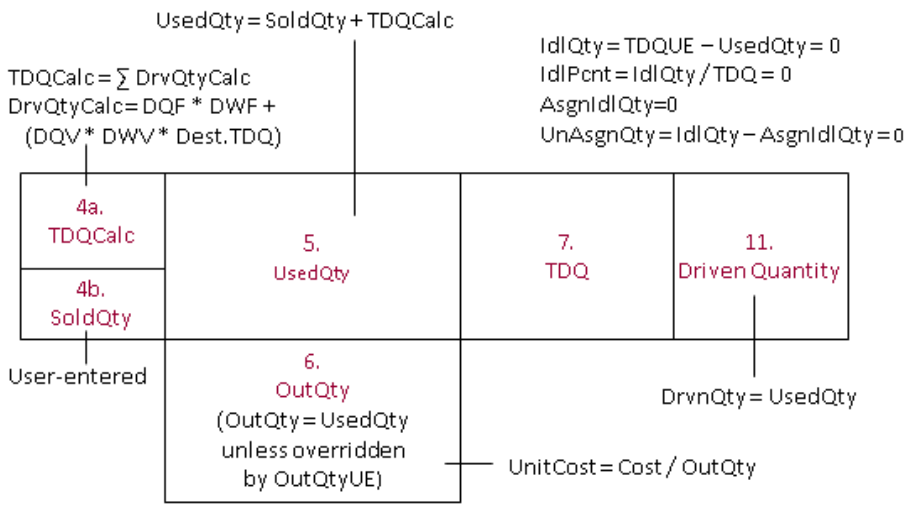
Diagram 3: Account Driver Properties

The following diagram shows the properties associated with the driver for an account. For example, TDQCalc for an account is the sum of DrvQtyCalc for the individual outgoing assignments that make up the driver for that account. Assignments are to one or more destination accounts.

This diagram is the same one as in "Account Driver Properties" on page 807 with the addition of numbering.



When TDQUE is not specified, then there are no idle quantities (**`IdlQty=0`**, **`AsgnIdlQty=0`**, and **`UnAsgnQty=0`**) and, therefore, **`TDQ=UsedQty=DrivenQty`**. When TDQUE is not specified, the diagram reduces to the following:



Assignment and Account Driver Properties

This section examines the interrelations among the following diagrams:

- "Diagram 1: Cost Flowing Into and Out of an Account" on page 527
- "Diagram 2: Properties of an Assignment" on page 528
- "Diagram 3: Account Driver Properties " on page 529

In the following diagram, the long blue arrows denote a dependency relationship between properties. The property pointed to by the arrow depends on the property that points to it. For example, the arrow that points to

```
4a. TDQCalc
```

from

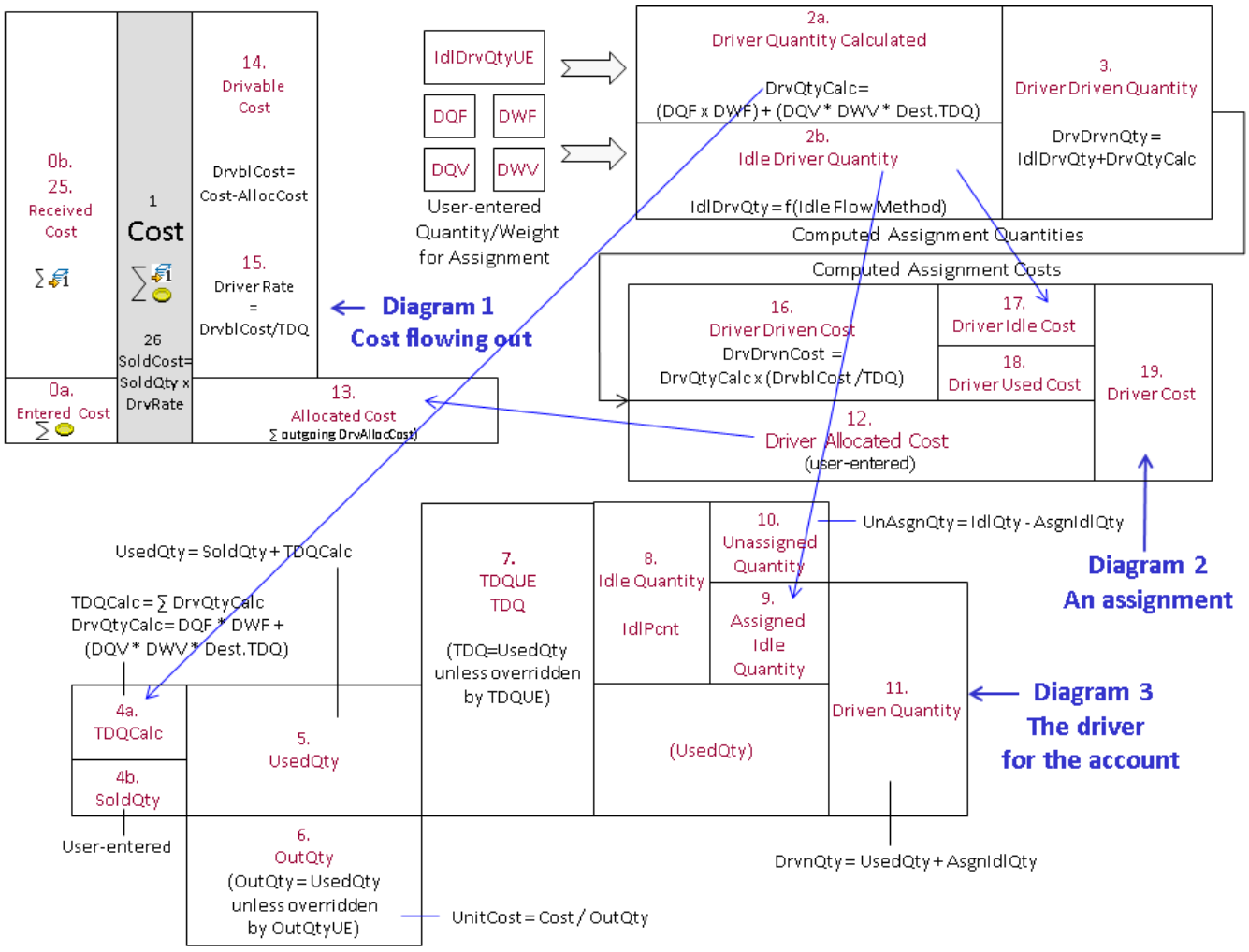
```
2a. Driver Quantity Calculated
```

indicates that TDQCalc (4a) depends on Driver Quantity Calculated (2a). In the table that follows the diagram, look for the row number that matches the number of the property pointed to. This row describes the dependency relationship between the two properties. For the current example, the row number is 4a and its text includes the following formula:

```
TDQCalc = ΣDrvQtyCalc (2a)
```

Note that (2a) is the number of the cell in the diagram for the property that TDQCalc is dependent on (DrvQtyCalc). DrvQtyCalc is described by the table row numbered 2a.

The portion of the diagram that is labeled "Diagram 2. An assignment" stands for one or more assignments to different destination accounts. So, for example, TDQCalc (which is in the portion of the diagram labeled "Diagram 3") is the sum of DrvQtyCalc (in the portion of the diagram labeled "Diagram 2") for all outgoing assignments.



Step #	Property	Property Type	Calculation
0a.	Entered Cost	Incoming Diagram 1	User-entered. Entered Cost is the sum of entered cost elements for the account (a single account can have multiple entered cost elements).
0b.	Received Cost	Incoming Diagram 1	RcvCost = EnteredCost (0a) + RcvDrvnCost (24) + RcvAllocCost (23c) All cost in a model originates with entered cost elements: either with EnteredCost or with EnteredUnitCost (see "Entered Cost Elements" on page 260).
1.	Cost	Cost in an account Diagram 1	Cost = Entered Cost (0a) + Received Cost (0b) The cost in an account is the sum of its own Entered Cost (its own entered cost elements) plus its Received Cost.

Now calculation looks at each individual outgoing assignment in turn.

Step #	Property	Property Type	Calculation
2a.	Driver Quantity Calculated	Assignment Diagram 2	DrvQtyCalc = (DQF x DWF)+(DQV x DWV x Dest.TDQ) *Note:* This formula does not apply to a Calculated driver—where DrvQtyCalc is determined by a user-defined formula. There are two kinds of quantities: fixed vs. variable. Fixed works by "push" (how much is supplied) whereas variable works by "pull" (how much is demanded). Variable requires looking forward until finding Dest.TDQ ne 0 (that is until Dest.SoldQty ne 0 or Dest.TDQUE ne 0 or Dest.TDQCalc ne 0). A single assignment can have both fixed and variable quantities, (it can work both by push and by pull). See "Fixed Driver Quantities" on page 400.Also see "Variable Driver Quantities" on page 401.
2b.	Idle Driver Quantity	Assignment Diagram 2	IdlDrvQty for an assignment is determined by its idle flow method (user-entered, user proportion, driver quantity, evenly assigned). For information, see "Idle Quantities" on page 410.
3.	Driver Driven Quantity	Assignment Diagram 2	DrvDrvnQty =DrvQtyCalc (2a) + IdlDrvQty (2b)

Now that the system has looked at all the outgoing assignments from an account, it can calculate the quantities associated with an account's driver.

Step #	Property	Property Type	Calculation
4a.	TDQCalc	Account driver Diagram 3	TDQCalc = ∑DrvQtyCalc (2a) TDQCalc for a driver is the sum of DrvQtyCalc for the individual outgoing assignments of that driver.
4b.	Sold Quantity	Account driver Diagram 3	User-entered.
5.	Used Quantity	Account driver Diagram 3	UsedQty = TDQCalc (4a) + Sold Quantity (4b)
6.	Output Quantity	Account driver Diagram 3	Output Quantity = Used Quantity (5) unless overridden by OutQtyUE. Output Quantity is used to determine Unit Cost according to the following formula: UnitCost = Cost/OutQty Also, UnitRevenue = Revenue / OutQty Also, UnitProfit = Profit / OutQty
7.	TDQ	Account driver Diagram 3	TDQ = Used Quantity (6) unless overridden by TDQUE

Step #	Property	Property Type	Calculation
8.	Idle Quantity Idle Percent	Account driver Diagram 3	IdlQty = TDQ (7) – UsedQty (5) Because TDQ equals UsedQty unless overridden by TDQUE, the only way that IdlQty can be non-zero is if TDQUE is greater than UsedQty. *Note:* If TDQUE is less than UsedQty, then the following error message appears: "Overdriven source account. The value entered for TDQUE is less than the Driven Quantity." IdlPcnt = 100 * IdlQty / TDQ (7)
9.	Assigned Idle Quantity	Account driver Diagram 3	AsgnIdlQty = ∑ IdlDrvQty (2b) AsgnIdlQty for a driver is the sum of IdlDrvQty (2b) for the individual assignments.
10.	Unassigned Quantity	Account driver Diagram 3	UnAsgnQty = IdlQty (8) – AsgnIdlQty (9)
11.	Driven Quantity	Account driver Diagram 3	Driven Quantity = UsedQty (5) + AsgnIdlQty (9)
12.	Driver Allocated Cost	Assignment Diagram 2	User-entered. Driver Allocated Cost (DrvAllocCost) for an individual assignment is user-entered.
13.	Allocated Cost	Outgoing Diagram 1	AllocCost=∑ DrvAllocCost (12) Allocated Cost is the sum of DrvAllocCost (12) on assignments going out of the account. DrvAllocCost is user-entered.
14.	Drivable Cost	Outgoing Diagram 1	DrvblCost = Cost (1) – Allocated Cost (13) Also, DrvblCost = IdleCost (20b) + UsedCost (20a)
15.	Driver Rate	Account driver Diagram 3	Driver Rate = DrvblCost (14) / TDQ (7) *Note:* Driver Rate can be different from Unit Cost if any of the following is true: • TDQUE ne 0 (which overrides TDQ in the formula DrvRate = DrvblCost/TDQ) • AllocCost ne 0 (which affects DrvblCost in the formula DrvRate = DrvblCost/TDQ) • OutQtyUE ne 0 (which overrides OutQty in the formula UnitCost = Cost/OutQty)

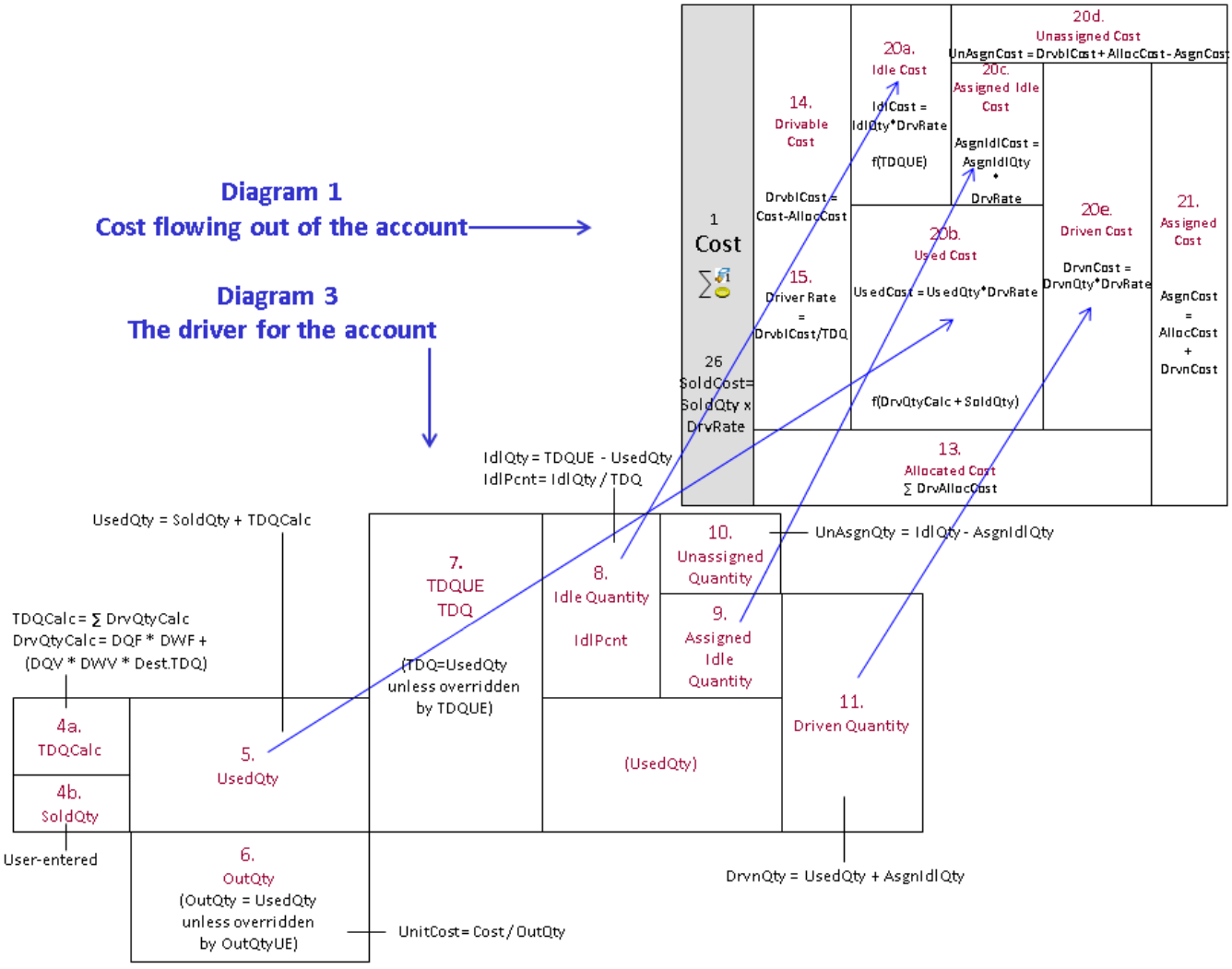
Once the system has determined Driver Rate, it can calculate the costs flowing out on each outgoing assignment.

Step #	Property	Property Type	Calculation
16.	Driver Driven Cost	Assignment Diagram 2	Driver Driven Cost = DrvQtyCalc (2a) x Driver Rate (15) So, Driver Driven Cost = DrvQtyCalc (2a) x (DrvblCost/TDQ) Also, DrvDrvnCost = DrvIdlCost (17) + DrvUsedCost (18)
17.	Driver Idle Cost	Assignment Diagram 2	DrvIdlCost = IdlDrvQty (2b) x DrvRate (15)
18.	Driver Used Cost	Assignment Diagram 2	DrvUsedCost = DrvDrvnCost (16) – DrvIdlCost (17) Also, DrvUsedCost = DrvQtyCalc (2a) x DrvRate (15)
19.	Driver Cost	Assignment Diagram 2	Driver Cost = DrvAllocCost (12) + DrvDrvnCost (16) That is, the outgoing cost for an assignment is the sum of the cost that is allocated plus the sum that is driven by the driver.

Outgoing Cost

This section examines the interrelations among the following diagrams:

- “Diagram 1: Cost Flowing Into and Out of an Account” on page 527
- “Diagram 3: Account Driver Properties ” on page 529



Step #	Property	Property Type	Calculation
20a.	Used Cost	Outgoing Diagram 1	Used Cost = UsedQty (5) x DrvRate (15)
20b.	Idle Cost	Outgoing Diagram 1	Idle Cost = IdlQty (8) x DrvRate (15) Also, Idle Cost = AsgnIdlCost (20c) + UnAsgnCost (22)
20c.	Assigned Idle Cost	Outgoing Diagram 1	AsgnIdlCost = AsgnIdlQty (9) x DrvRate (15) Also, AsgnIdlCost = sum of DrvIdlCost (17) on outgoing assignments
20d.	Driven Cost	Outgoing Diagram 1	Driven Cost = DrvnQty (11) x DrvRate (15) Also, Driven Cost = Used Cost (20a) + AsgnIdlCost (20c)
21.	Assigned Cost	Outgoing Diagram 1	Assigned Cost = Allocated Cost (13) + Driven Cost (20d)

Step #	Property	Property Type	Calculation
22.	Unassigned Cost	Outgoing Diagram 1	UnAsgnCost = DrvblCost (14) + AllocCost (13) – AsgnCost (21) Also, UnAsgnCost = Cost (1) – AsgnCost (21) Also, UnAsgnCost = IdlCost (20b) – AsgnIdlCost (20c)

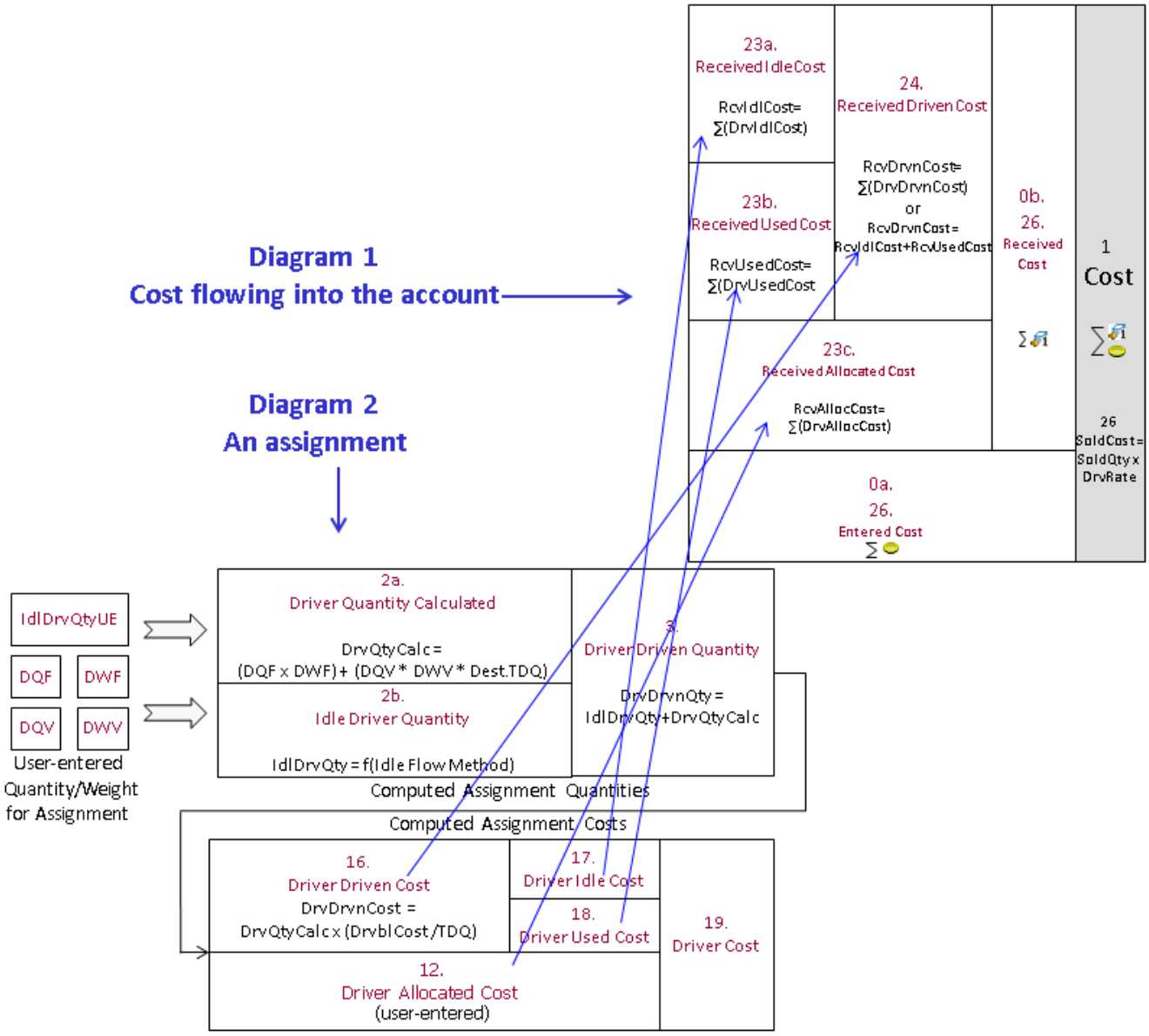
Next, the system looks at costs flowing into the destination account.

Incoming Cost

This section examines the interrelations among the following diagrams:

- "Diagram 1: Cost Flowing Into and Out of an Account" on page 527
- "Diagram 2: Properties of an Assignment" on page 528

The assignment shown in this diagram stands for one or more assignments from different source accounts. For example, RcvIdlCost, in the receiving account, is the sum of DrvIdlCost in all the incoming assignments to that account.



Step #	Property	Property Type	Calculation
23a.	Received Idle Cost	Incoming Diagram 1	RcvIdlCost = ∑ DrvIdlCost (17) RcvIdlCost is the sum of idle costs from individual assignments, DrvIdlCost (17), to the destination account.
23b.	Received Used Cost	Incoming Diagram 1	RcvUsedCost = ∑ DrvUCost (18) RcvUsedCost is the sum of used costs from individual assignments, DrvUCost (18), to the destination account.
23c.	Received Allocated Cost	Incoming Diagram 1	RcvAllocCost = ∑ DrvAllocCost (12) RcvAllocCost is the sum of costs allocated on individual assignments, DrvAllocCost (12), to the destination account.

Step #	Property	Property Type	Calculation
24.	Received Driven Cost	Incoming Diagram 1	RcvDrvnCost = ∑ DrvDrvnCost (16) RcvDrvnCost is the sum of costs driven by individual assignments, DrvDrvnCost (16), to the destination account. Also, RcvDrvnCost = RcvIdlCost (23a) + RcvUsedCost (23b)
25.	Received Cost	Incoming Diagram 1	Received Cost = RcvDrvnCost (24) + RcvAllocCost (23c) Also, Received Cost = RcvBOCCost (25a) + RcvAsgnCost (25b)
26.	Sold Cost	Account Diagram 1	SoldCost = SoldQty (4b) x DrvRate (15)

Now the destination account contains a cost that is equal to its Received Cost plus its own Entered Cost, and the process of examining outgoing assignments starts over until the calculation process has traversed all the accounts and all the assignments in the model.

Chapter 61

Detailed Example of Calculation

Introduction

This chapter describes what the system does to calculate the cost flows between accounts for the following simple example:

IntsctnName	($) Cost	DrvName
2 Resource_account	100.00	Weighted

Display Name	($) Cost	DrvName
ACTIVITY (PRIMARY PANE)	100.00	
3 Activity_account1	17.89	Standard
3 Activity_account2	82.11	Percentage

IntsctnName	($) Cost
4 CO_account1	58.21
4 CO_account2	41.79

IntsctnName	($) Cost	DrvName
2 Resource_account	100.00	Weighted

Display Name	($) Cost	DrvName
ACTIVITY (PRIMARY PANE)	100.00	
3 Activity_account1	17.89	Standard
3 Activity_account2	82.11	Percentage

IntsctnName	($) Cost
4 CO_account1	58.21
4 CO_account2	41.79

The description can help you to further understand that some system properties describe quantities flowing along an assignment path, while other system properties describe costs coming into or flowing out of an account. Although this example does not include all the properties available in a model, it does include those that are most frequently used in assignments.

To summarize the calculation process for this example:

- Resource_account assigns costs to Activity_account1 and Activity_account2 using a Standard driver, named “Weighted”.

  The Weighted driver has both fixed driver quantities (DQF and DWF) and variable driver quantities (DQV and DWV).

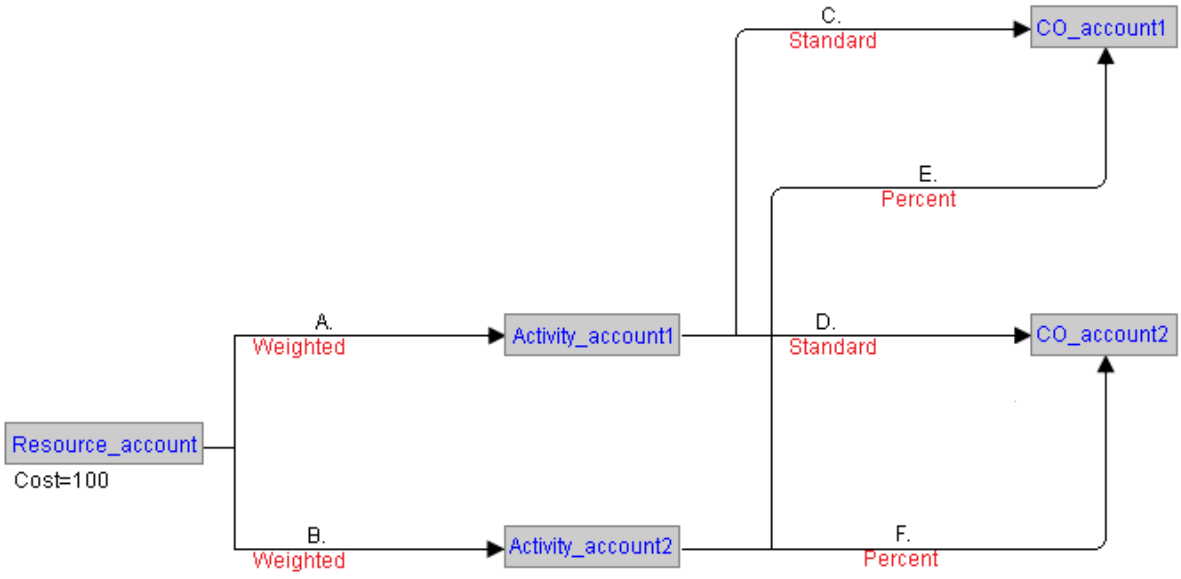
  The driver also has a user-entered cost allocation (DrvAllocCost).

- Activity_account1 assigns costs to CO_account1 and CO_account2 using a Standard driver named “Standard”.

  This Standard driver has both a fixed driver quantity (DQF) and a variable driver quantity (DQV) — but not DWF and DWV.

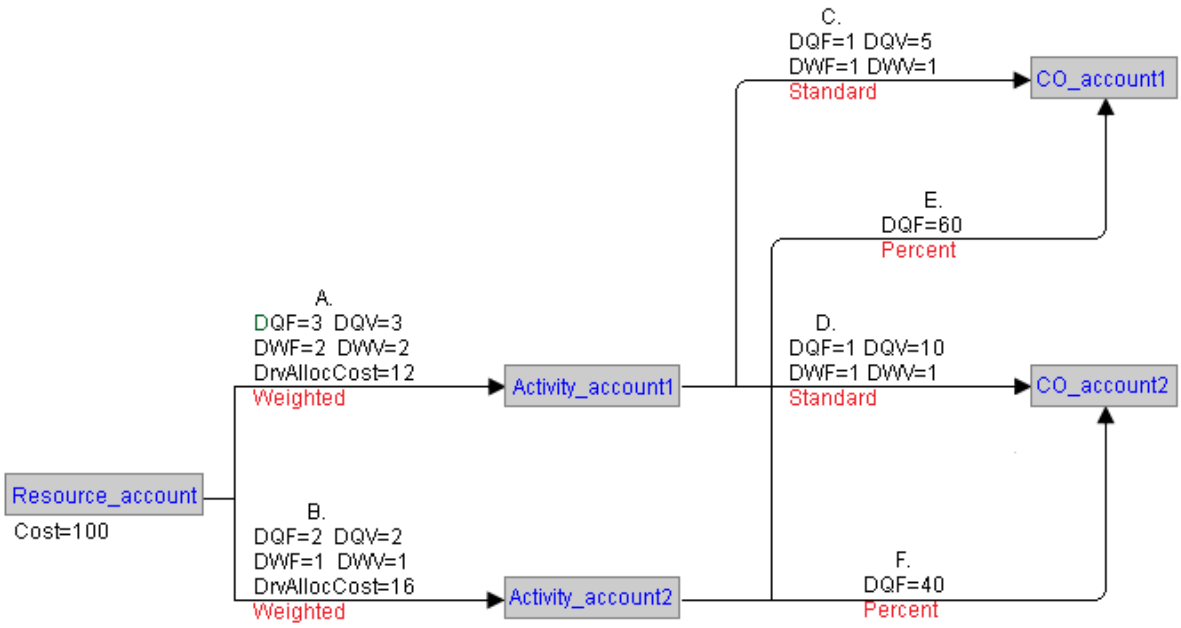
- Activity_account2 also assigns costs to CO_account1 and CO_account2 but using a Percentage driver.

The following is a schematic drawing which we will use in the rest of this chapter in discussing the calculation process.



The properties whose value is determined during calculation are the following:

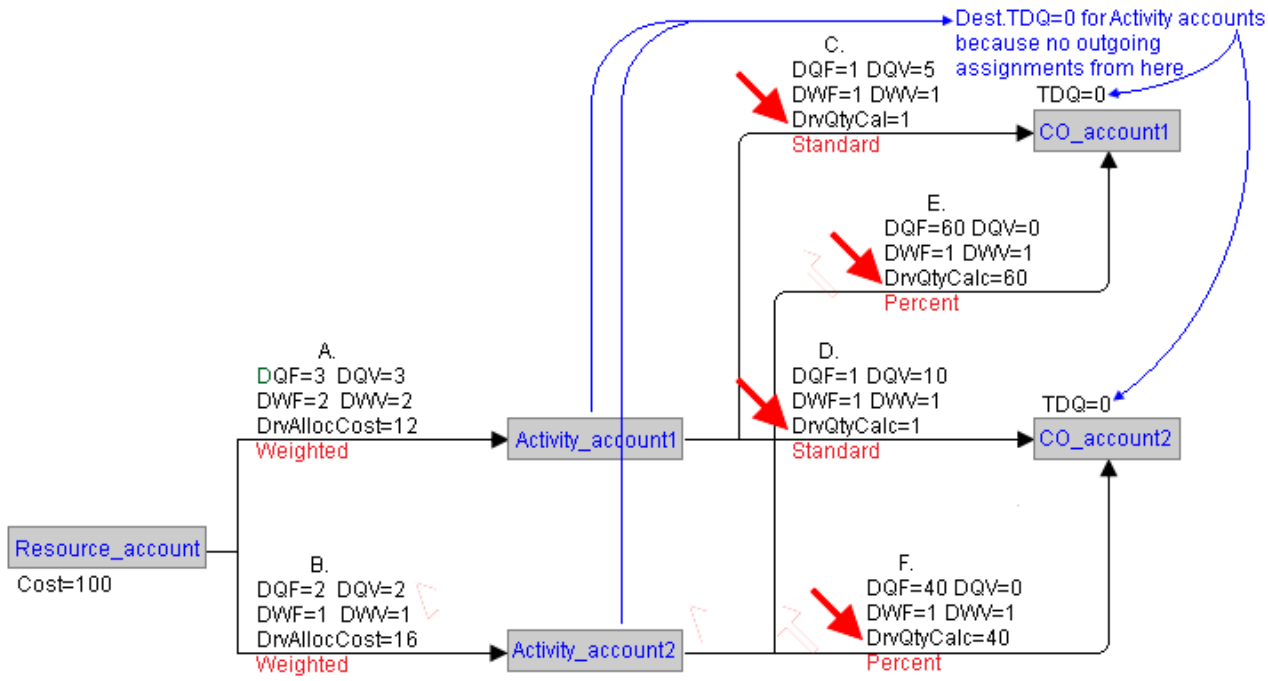
Step 1–User Enters Driver Quantities (DQF, DQV, DWF, DWV, DrvAllocCost)



The following are user-entered driver quantities:

- DQF (Driver Quantity Fixed)
- DQV (Driver Quantity Variable)
- DWF (Driver Weight Fixed)
- DWV (Driver Weight Variable)
- DrvAllocCost (Driver Allocated Cost)

Step 2–DrvQtyCalc for Assignments to Cost Object Accounts



DrvQtyCalc=(DQF x DWF) + (DQV x DWV x Dest.TDQ):

DrvQtyCalc from Activity_account1 to CO_account1 = (1 x 1) + (5 x 1 x 0) = 1

DrvQtyCalc from Activity_account1 to CO_account2 = (1 x 1) + (10 x 1 x 0) = 1

DrvQtyCalc from Activity_account2 to CO_account1 = (60 x 1) + (0 x 1 x 0) = 60

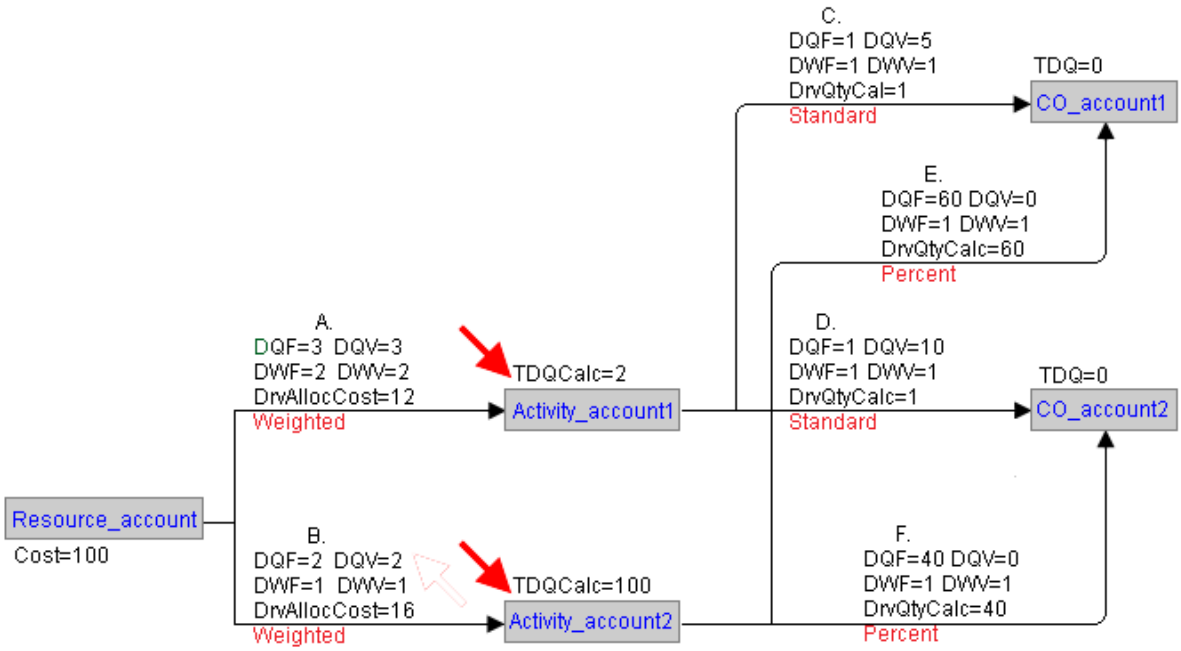
DrvQtyCalc from Activity_account2 to CO_account2 = (40 x 1) + (0 x 1 x 0) = 40

Note: TDQ=0 for CO_account1 and for CO_account2 because there are no costs flowing out of these Cost Object accounts.

Note: A Percentage driver does not allow variable driver quantities. So, its DQV is null, and for calculations a null is counted as zero (DQV=0).

Note: Both the Percentage driver and the "Standard" driver are non-weighted drivers. For non-weighted drivers, DWF=1 and DWV=1.

Step 3–TDQCalc for Activity Accounts

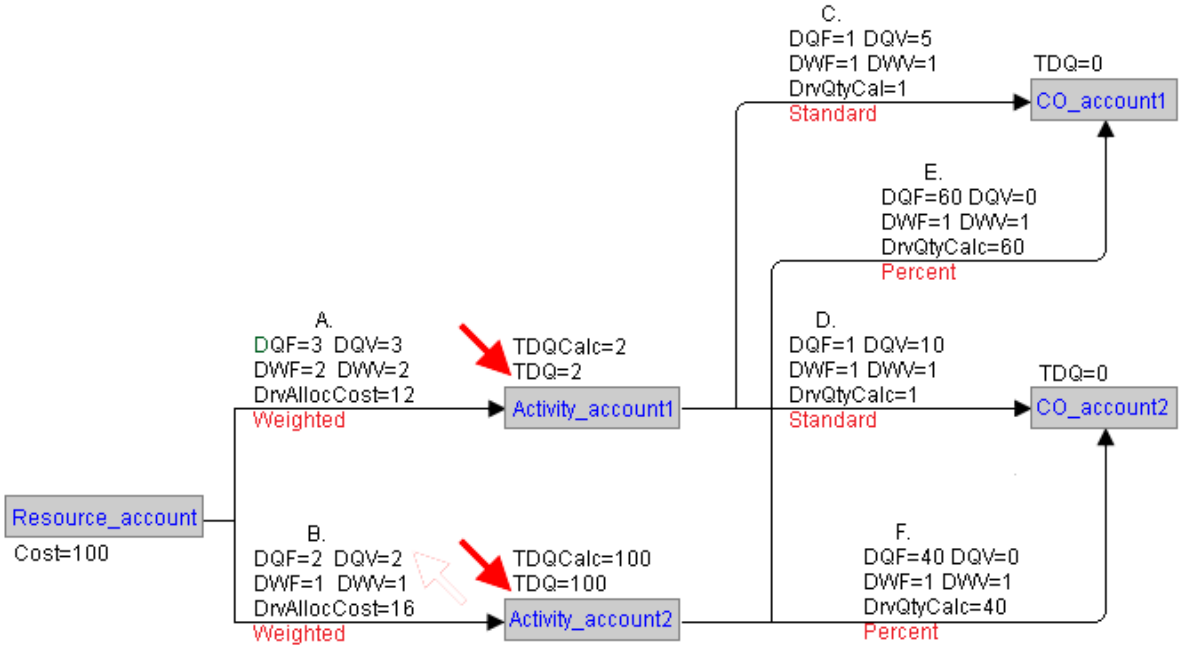


TDQCalc=∑(DrvQtyCalc)

TDQCalc for Activity_account1 = 1 + 1 = 2

TDQCalc for Activity_account2 = 60 + 40 = 100

Step 4–TDQ for Activity Accounts



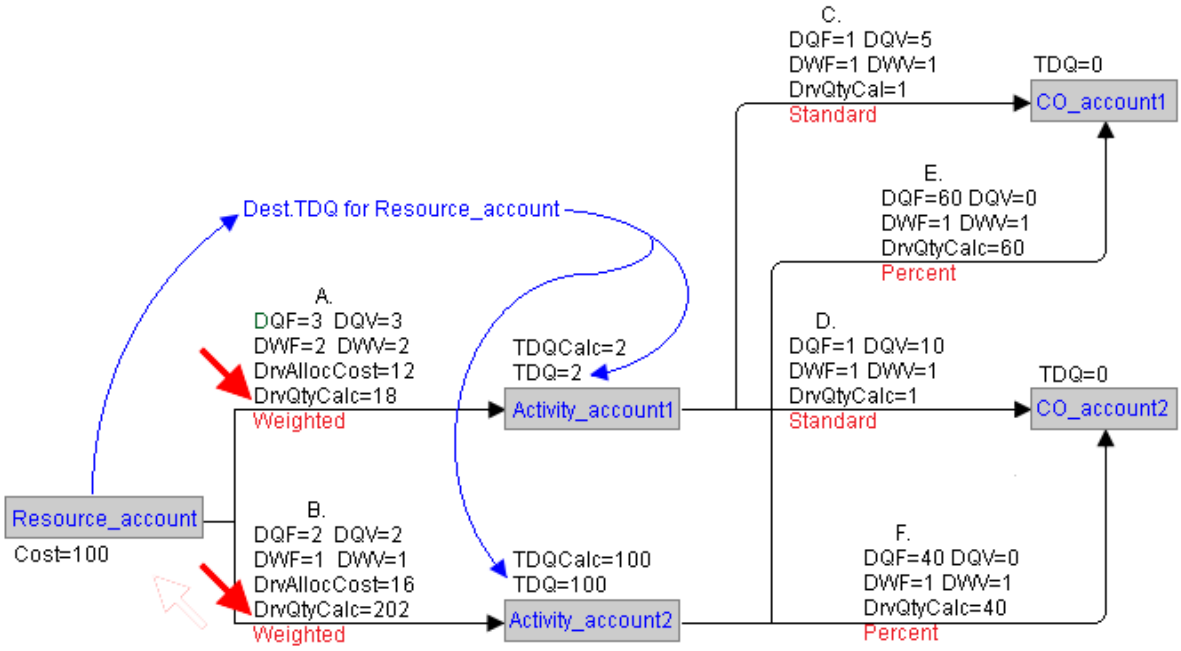
TDQ=TDQCalc + SoldQty

TDQ(Activity_account1) = 2 + 0 = 2

TDQ(Activity_account2) = 100 + 0 = 100

Note: SoldQty is a usser-entered property and is zero in this example.

Step 5–DrvQtyCalc for Assignments to the Activity Accounts

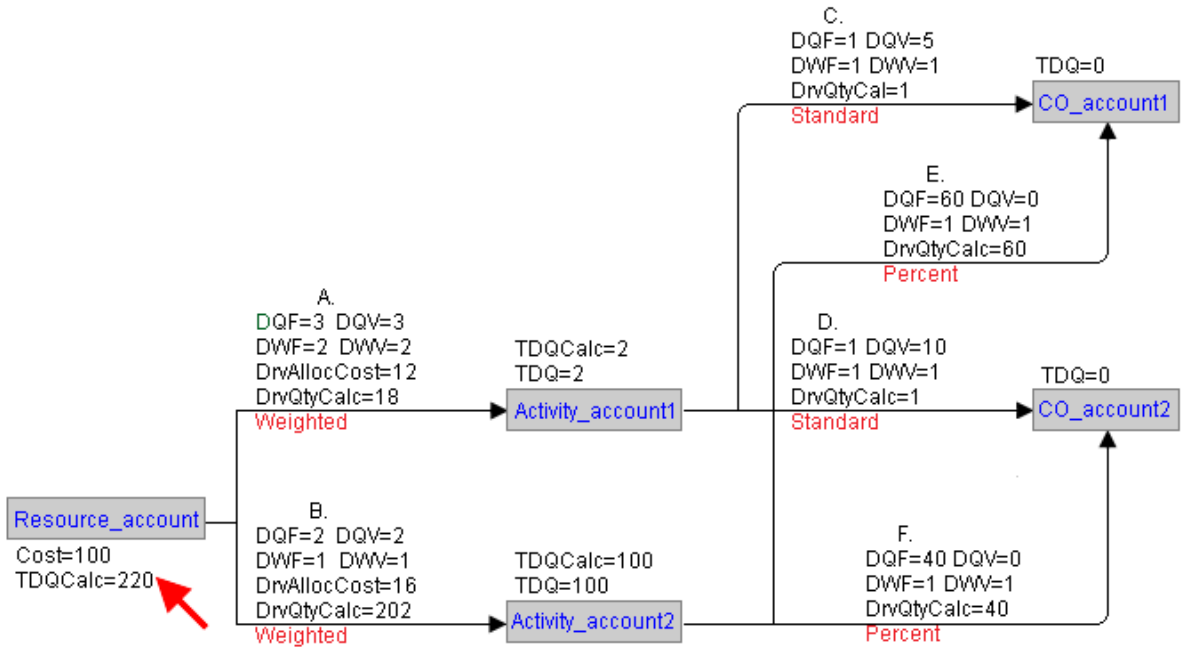


DrvQtyCalc=(DQF x DWF) + (DQV x DWV x Dest.TDQ):

DrvQtyCalc(Activity_account1) = (3 x 2) + (3 x 2 x 2) = 18

DrvQtyCalc(Activity_account2) = (2 x 1) + (2 x 1 x 100) = 202

Step 6–TDQCalc for Resource Account

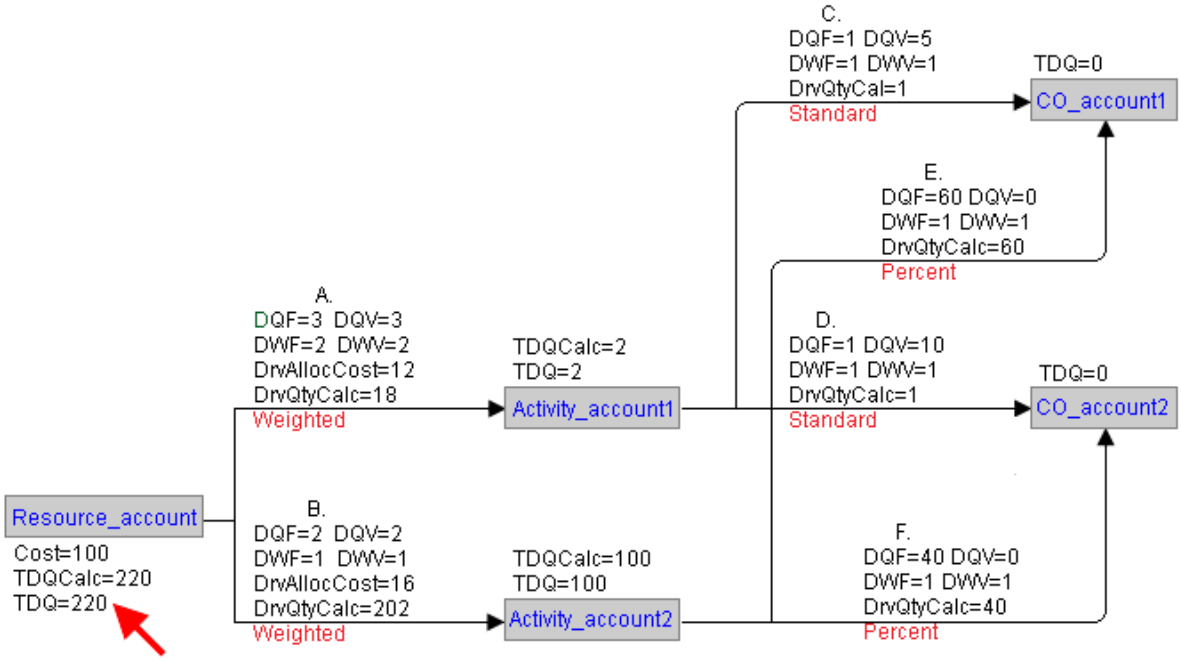


TDQCalc=∑(DrvQtyCalc)

TDQCalc = DrvQtyCalc(Activity_account1) + DrvQtyCalc(Activity_account2)

TDQCalc = 18 + 202 = 220

Step 7–TDQ for Resource Account

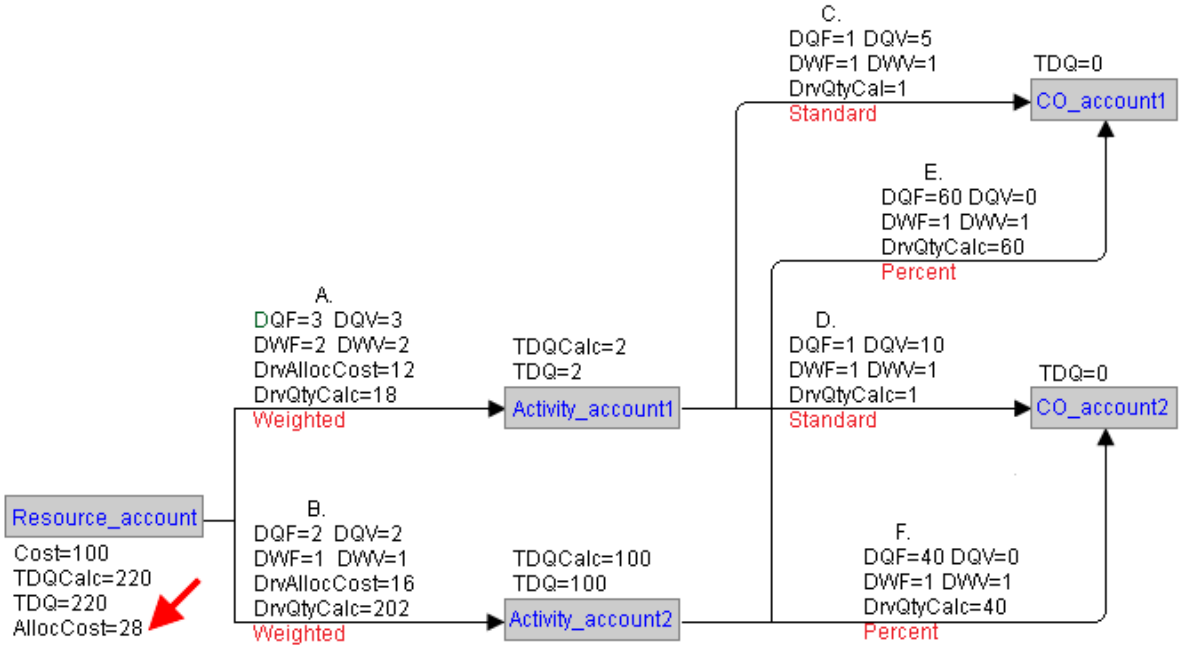


TDQ = TDQCalc + SoldQty

TDQ = 220 + 0 = 220

Note: SoldQty is a usser-entered property and is zero in this example.

Step 8–AllocCost for Resource Account



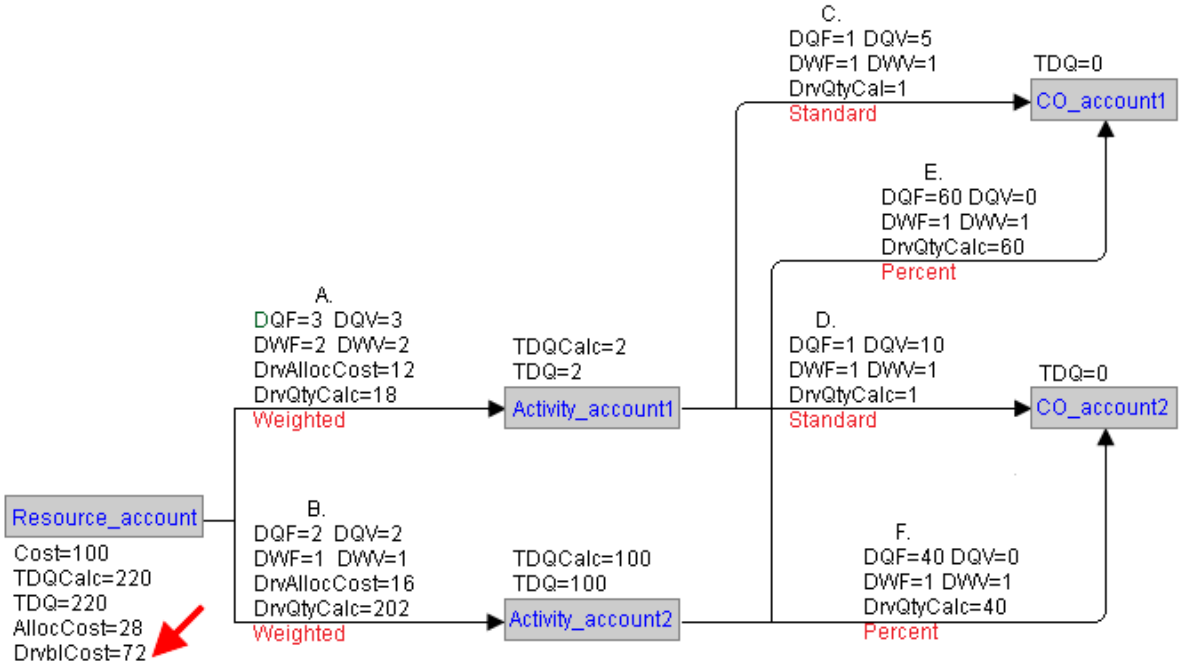
Allocated Cost = ∑(DrvAllocCost) for all outgoing assignments

DrvAllocCost(to Activity_account1) = $12

DrvAllocCost(to Activity_account2) = $16

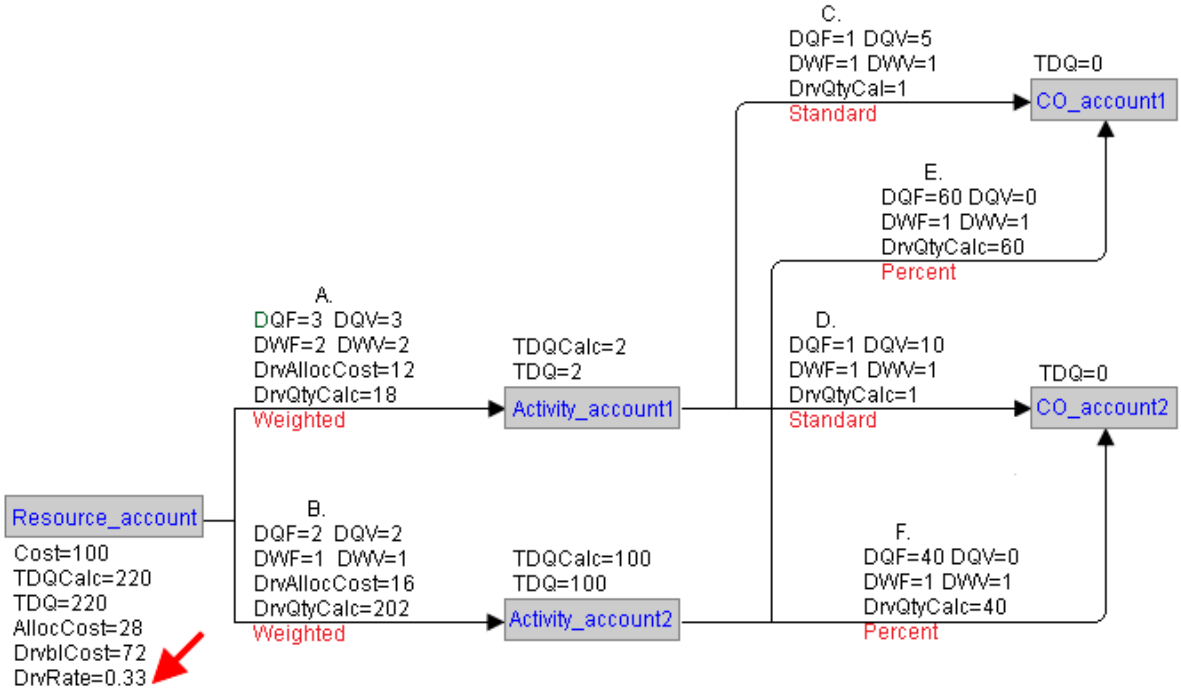
(source) Allocated Cost = $12 + $16 = $28

Step 9–DrvblCost for Resource Account



Drivable Cost = Cost - Allocated Cost:
Drivable Cost = $100 - $28 = $72

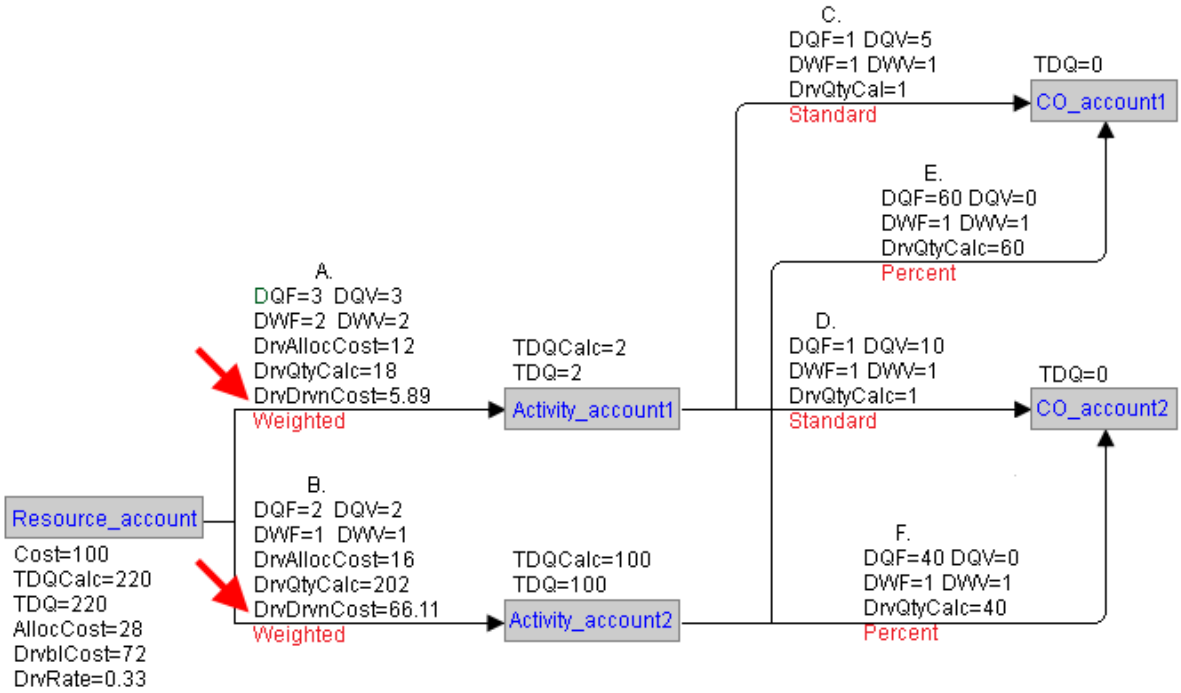
Step 10–DrvRate for Resource Account



DriverRate = DrivableCost / TDQ:

DriverRate = $72 / 220 = $0.33 (rounded up)

Step 11–DrvDrvnCost for Assignments to Activity Accounts

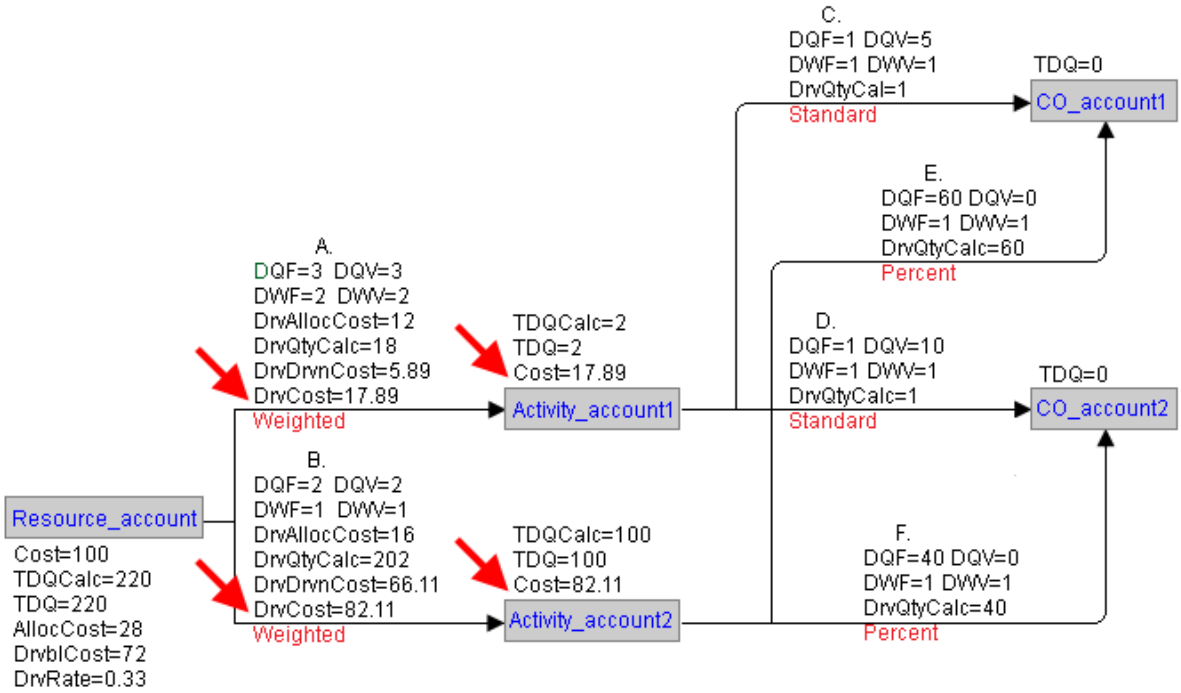


DrvDrvnCost = DrvblCost x (DrvQtyCalc / TDQCalc):

DrvDrvnCost(to Activity_account1) = $72 x (18 / 220)= $5.89

DrvDrvnCost(to Activity_account2) = $72 x (202 / 220)= $66.11

Step 12–DrvCost for Assignments to Activity Accounts



DriverCost = DrvDrvnCost + DrvAllocCost

Driver cost equals the cost that is driven by an account's driver plus cost that is allocated..

DriverCost(to Activity_account1) = $5.89 + $12 = $17.89

DriverCost(to Activity_account2) = $66.11 + $16 = $82.11

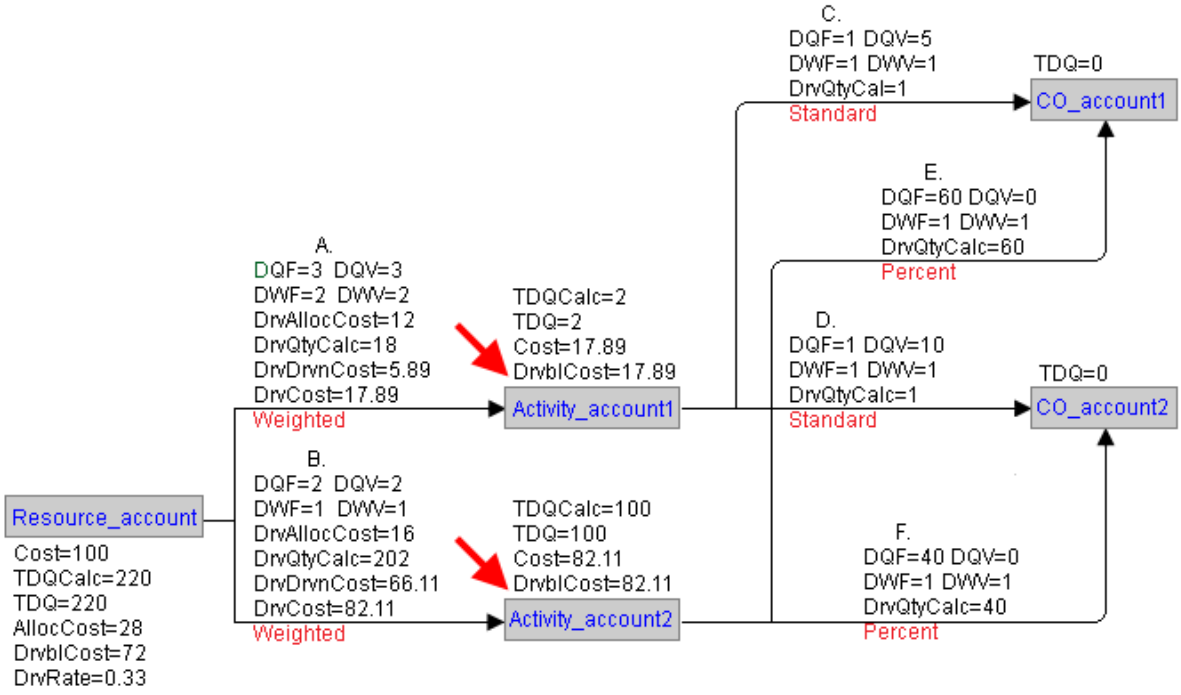
Cost = RcvCost + EntCost

The cost in an account equals the cost it receives on assignments from other accounts plus cost that is entered directly into the account.

Cost(Activity_account1) = $17.89 + $0 = $17.89

Cost(Activity_account2) = $82.11 + $0 = $82.11

Step 13–DrvblCost for Activity Accounts

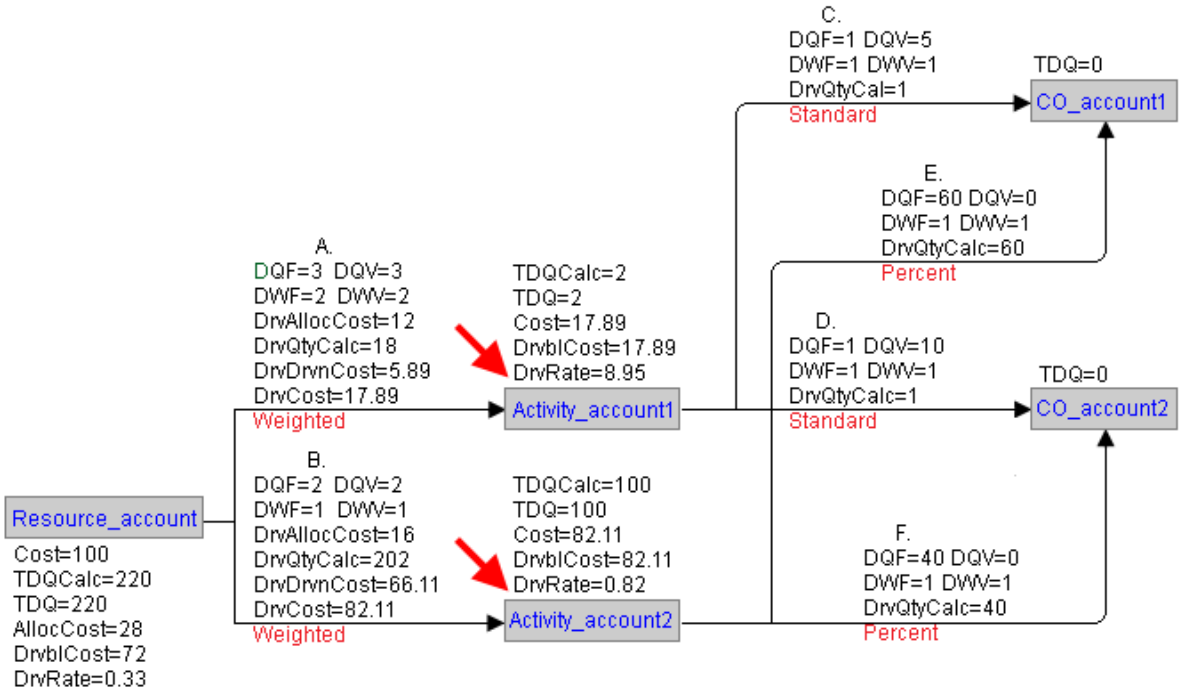


DrivableCost = Cost - Allocated Cost:

DrivableCost for Activity_account1 = $17.89 - $0 = $17.89

DrivableCost for Activity_account2 = $82.11 - $0 = $82.11

Step 14–DrvRate for Outgoing Drivers from Activity Accounts

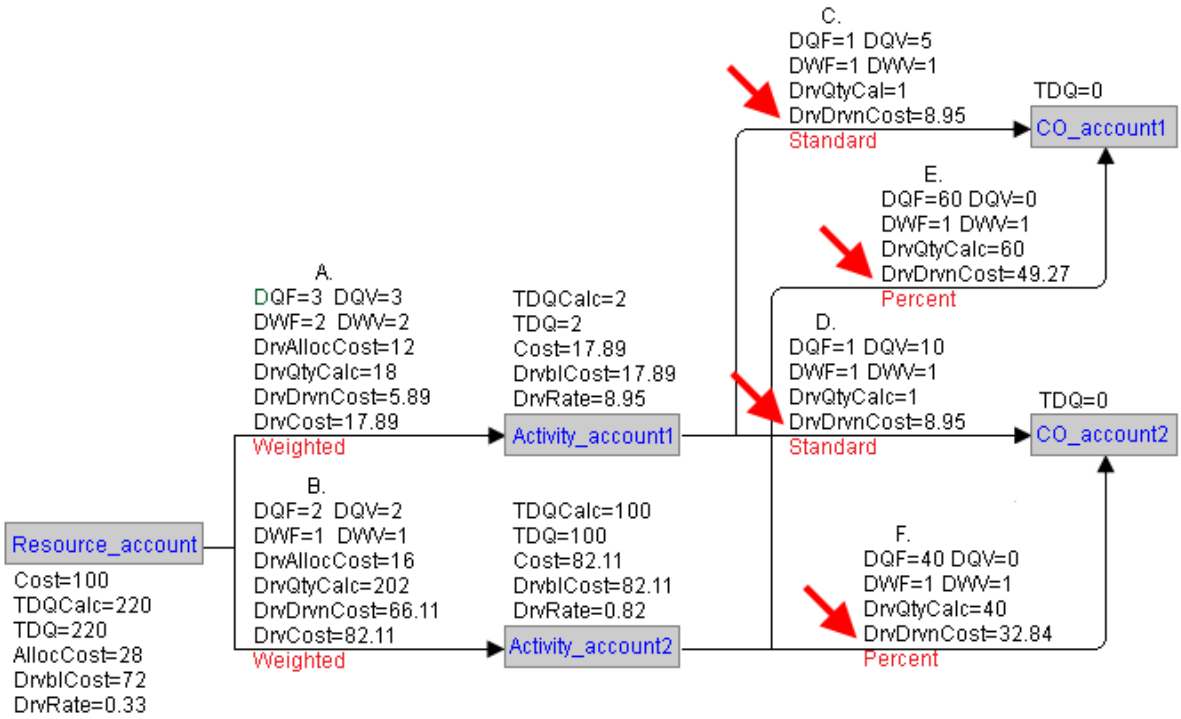


DriverRate = DrivableCost / TDQ:

DriverRate(Activity_account1)= $17.89 / 2 = $8.95 (rounded up)

DriverRate(Activity_account2) = $82.11 / 100 = $0.82 (rounded down)

Step 15–DrvDrvnCost for Assignments to Cost Object Accounts



DrvrDrvnCost = DrvblCost * (DrvQtyCal / TDQ)

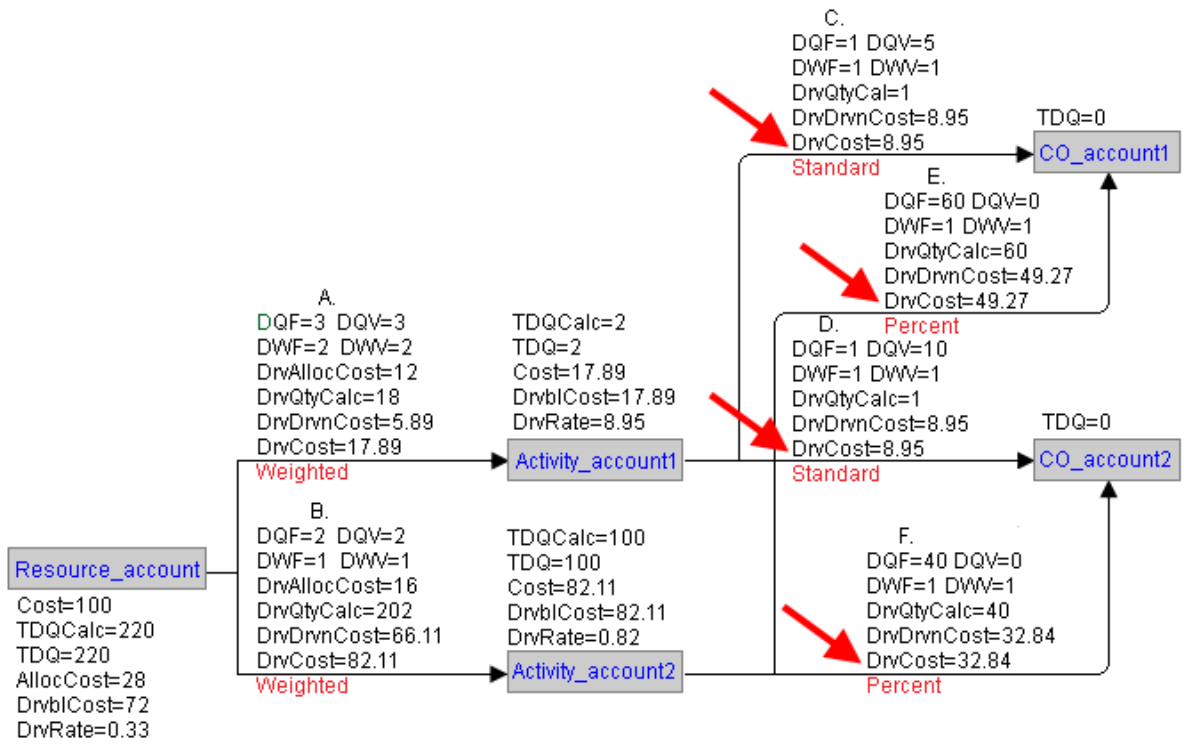
DrvrDrvnCost(**Activity_account1** ⇨ **CO_account1**) = $17.89 x (1/2) = $8.95

DrvrDrvnCost(**Activity_account1** ⇨ **CO_account2**) = $17.89 x (1/2) = $8.95

DrvrDrvnCost(**Activity_account2** ⇨ **CO_account1**) = $82.11 x (60/100) = $49.27

DrvrDrvnCost(**Activity_account2** ⇨ **CO_account2**) = $82.11 x (40/100) = $32.84

Step 16–DrvCost for Assignments to Cost Object Accounts



DriverCost = DrvDrvnCost + DrvAllocCost

Driver cost equals the cost that is driven by an account's driver plus cost that is allocated..

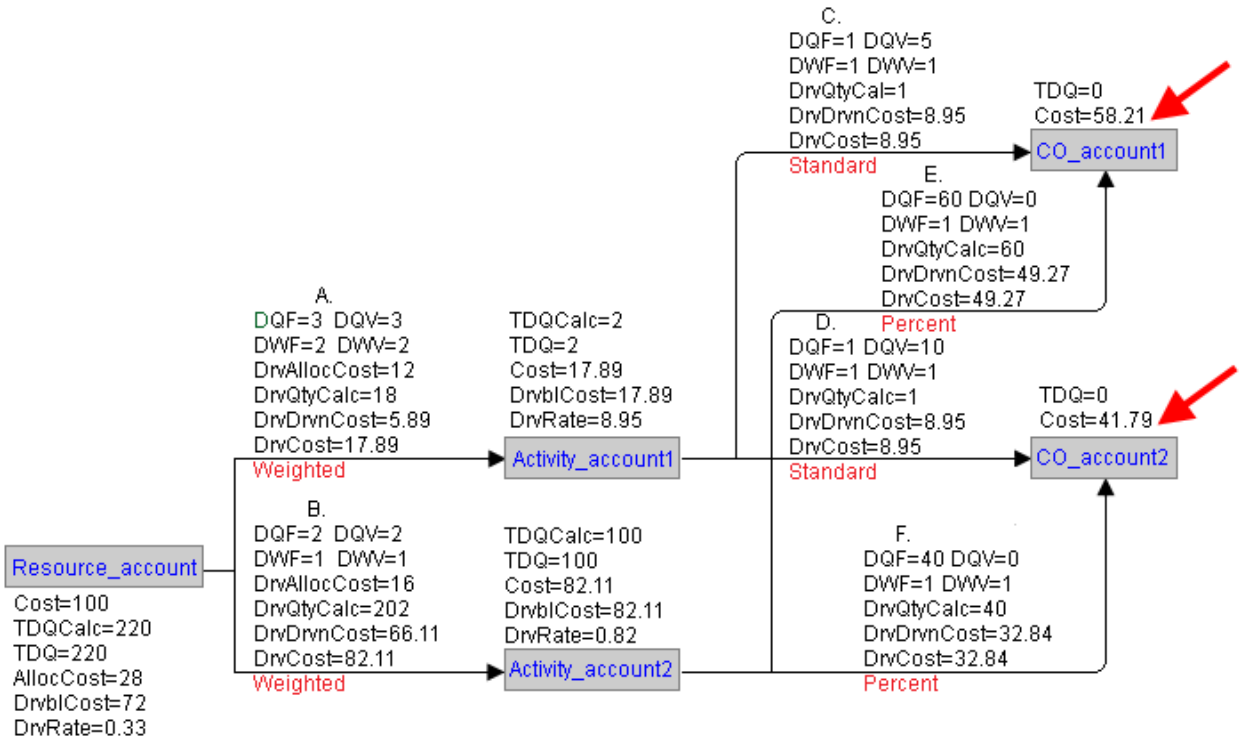
DriverCost(Activity_account1 to CO_account1) = $8.95 + $0 = $8.95

DriverCost(Activity_account1 to CO_account) = $8.95 + $0 = $8.95

DriverCost(Activity_account2 to CO_account1) = $49.27 + $0 = $49.27

DriverCost(Activity_account2 to CO_account2) = $32.84 + $0 = $32.84

Step 17–Cost for Cost Object Accounts



Cost = RcvCost + EntCost

The cost in an account equals the cost it receives on assignments from other accounts plus cost that is entered directly into the account.

Cost(CO_account1)= $8.95 + $49.27 = $58.21

Cost(CO_account2) = $8.95 + $32.84 = $41.79

Chapter 62

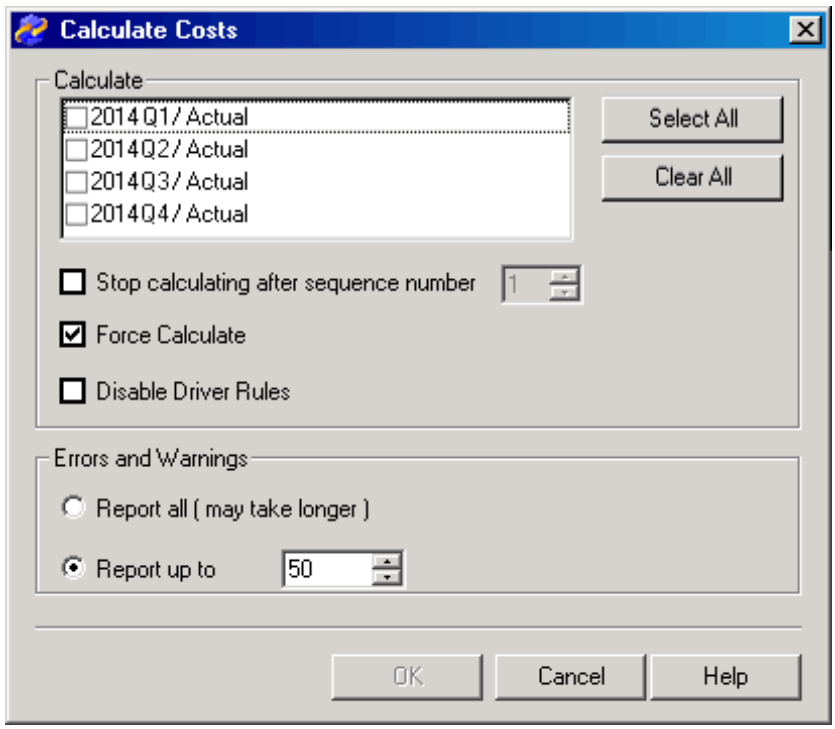
Windows for Calculation

Calculate Costs Dialog Box

Note: The availability of these features depends on your permissions.

1. Open a model and select **Model** ⇨ **Calculate Costs**.

   The Calculate Costs dialog box opens:



2. Select one or more period/scenario associations.

   Note: If there is an error calculating one period/scenario, then no others are calculated.

3. Select or clear the **Stop calculating after sequence number** option. If applicable, specify the sequence number.

For more information, see Driver sequencing.

4. Select **Force Calculate** to perform calculations even if a flag says that calculations are up to date.

   To speed processing, SAS Cost and Profitability Management skips calculating if the calculations are up to date. Use this option to perform the calculations even if they are already available.

5. Select **Disable Driver Rules** if you want calculation to ignore the rule formula attached to any rule-based driver.

   You can choose this option to disable driver rules to speed up calculation if you have previously done a calculation that has already created the driver assignments for rule-based drivers.

   If you do not select this option to disable driver rules, then calculation evaluates for which potential destination accounts the driver's formula tests true and creates an assignment from the source account to each such destination account. And, calculation evaluates for which potential destination accounts the driver's formula test false and deletes the assignment to that account if the assignment exists. This means that if you have previously allowed calculation to create assignments and subsequently you removed some of those assignments from the source account or created additional assignments to other destination accounts, then those manually created assignments will be undone and you will have to do them over again.

   Note: You can also turn off the formula for individual drivers by deselecting Use Rule Formula in the Driver Properties dialog box.

6. Select the number of error and warning messages to be displayed.

7. Click **OK**.

Part 18

Generating Cubes

Chapter 63

What is a Cube?

Introduction

Cubes

A cube is the basic unit of analysis: it represents a particular domain of inquiry in online analytical processing (OLAP). A cube contains a subset of model data, such as single-stage contributions, multi-stage contributions, or resource contributions. Each cube combines multiple dimensions and the measures that the dimensions contain into one unit. SAS Cost and Profitability Management cubes are standard OLAP cubes.

You use SAS Cost and Profitability Management to connect to the cubes on a SAS Cost and Profitability Management server and to interact with these cubes. After generating cubes, you can then manipulate these cubes in the OLAP Analyzer to interactively analyze data.

In an analysis grid, a cube's dimensions determine the columns and rows. The measures are the data in the cells of the rows and columns.

A cube in which costs are indexed by two dimensions (for example, customer and product) is a two-dimensional cube. A cube in which costs are indexed by three dimensions (for example, region, customer, and product) is a three-dimensional cube.

It does not matter how many dimensions a cube has; the storage unit is still a cube, and the cube represents n dimensions of data. A cube enables you to perform

multidimensional data analysis. You extract useful knowledge from an n-dimensional cube, and you represent the knowledge so that it can be easily understood.

In a cube, measures are aggregated within a single dimension and for all combinations of dimension members from different dimensions. These aggregations enable you to analyze measures by dimension members in different dimensions simultaneously. For example, you can analyze quarterly costs for products within a region.

Fact Tables

Each cube is based on a fact table, which stores model data for the cube. When you generate a cube, the fact table is generated first. Then, the cube is created from the fact table.

To make your data analysis capabilities more flexible, you can choose to generate a fact table without generating the associated cube. You can use the fact table to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft Analysis Services.

Note: When you generate a fact table, SAS Cost and Profitability Management determines whether any model data has changed since the last time the model was calculated. If any model data has changed, the period/scenario association is calculated to ensure that the model data is correct. You can force the costs to be calculated.

Generating Cubes

After you calculate costs, you can generate cubes to analyze a model with the SAS OLAP Analyzer.

Note: When you generate a cube, SAS Cost and Profitability Management determines whether any model data has changed since the last time the model was calculated. If any model data has changed, the period/scenario association is calculated to ensure that the model data is correct. You can force the costs to be calculated.

When SAS Cost and Profitability Management generates cubes, it performs calculations that pre-aggregate numeric data to give you faster performance when you work on the OLAP view. For large models, pre-aggregating data for cubes can take many hours to complete.

The following factors are listed in order, from those that take more time to generate a cube, to those that take less time:

- the number of stages
- the number of dimensions
- the length of assignment paths

  Assignment paths should contain fewer than 10 items.
- the number of accounts and the number of assignment paths

Note: Whenever you generate cubes, cubes that were generated previously are no longer available for viewing.

OLAP Analyzer Performance

If the SAS Cost and Profitability Management server does not finish pre-aggregating data during cube generation, some data might be aggregated while you are using the OLAP Analyzer. This can result in poor performance. Therefore, you might want to increase the server's default time limit (about 10 hours) for pre-aggregating data. To increase the default time limit, contact your regional SAS Technical Support for assistance.

Predefined Cubes

Resource Contributions Cube

The Resource Contributions cube enables you to analyze resource costs that contribute to a product, customer, service cost, and so on. Or, use this cube to analyze the products, customers, service costs, and so on, that receive costs from resources.

The Resource Contributions cube enables you to study cost contributions from original accounts where costs were entered to final accounts that do not assign costs to other accounts. Generally, these cost contributions are from resource accounts to cost object accounts, but where the original or final accounts reside does not matter.

Note: The Resource Contributions cube contains only the costs of the first account in an assignment path and the costs of the last account. It does not contain the intermediate accounts and their costs.

Use the Multi-stage Contributions cube to analyze cost contributions to or from accounts that are tagged with stages attributes.

Use the Single-stage Contributions cube to analyze the cost contributions from one assignment level back.

Single-stage Contributions Cube

The Single-stage Contributions cube enables you to answer questions such as:

- Which activity costs contribute to product, customer, service cost, and so on?
- When costs are assigned within the Cost Object module, which sub-assembly costs contribute to product costs?
- What are the costs of resources that contribute to activities?

The Single-stage Contributions cube enables you to analyze the cost contributions from one assignment level back. Where costs originate or end does not matter. Typically, cost is contributed from:

- Activities to cost objects
- Resources to activities

You do not need to add stages attributes to accounts; you can use each module as a stage.

Use the Resource Contributions cube to analyze costs contributions from original accounts where costs were entered to final accounts that do not assign costs to other accounts.

Use the Multi-stage Contributions cube to analyze cost contributions to or from accounts that are tagged with stages attributes.

Multi-stage Contributions Cube

The Multi-stage Contributions cube enables you to address issues and answer questions such as:

- Product A is not profitable. I want to trace the costs back through activities and then to resources that contribute costs to this product.
- What are the costs for Product B that originate in salary resources and are assigned through the Inspection activity to this product?

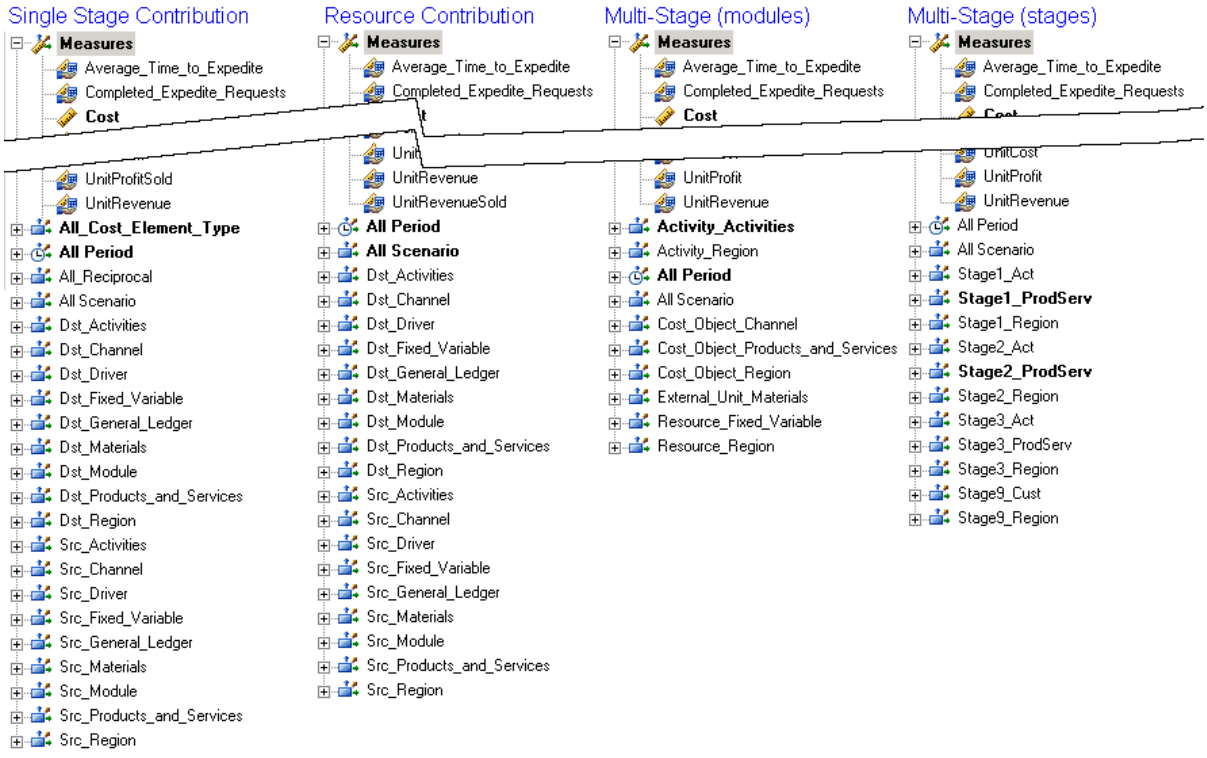
The Multi-stage Contributions cube enables you to analyze cost contributions into and out of stages defined in a model. You can define each module as a stage or you can use stages attributes. The SAS OLAP Analyzer Cube Explorer View enables you to visually trace cost contributions through all the stages.

Use the Resource Contributions cube to analyze costs contributions from original accounts where costs were entered to final accounts that do not assign costs to other accounts.

Use the Single-stage Contributions cube to analyze the cost contributions from one assignment level back.

Summary

Using the Parcel Express Tutorial model as an example, the following picture shows for each type of cube the elements that are available in the SAS OLAP Analyser for adding to rows or columns.



Using the Cubes with Other Software

Cognos PowerPlay

To open SAS Cost and Profitability Management cubes with Cognos PowerPlay, see the Cognos online document titled *OLAP Server Connection Guide*. The chapter "Connect to Microsoft SQL Server OLAP Services" describes the procedure and concepts in detail.

The SAS Cost and Profitability Management Administrator can give you the model ID for each model. Instructions are in the "SAS Cost and Profitability Management Installation Checklist."

If you require additional help after reviewing the Cognos documentation, please contact your regional SAS Technical Support.

Microsoft Excel

You can export a cube to Microsoft Excel and then modify, print, or save the data.

Chapter 64

Incremental Cube Generation

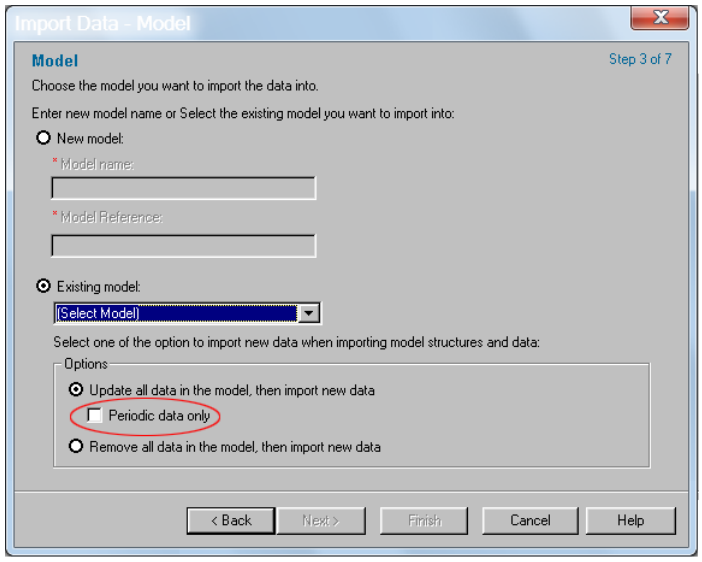
Incremental Cube Generation

Now, when you generate a cube, for every period/scenario association that is to be included in an existing cube, if

- the cube already contains that period/scenario association, and
- the period/scenario association has not been modified since the cube was last generated

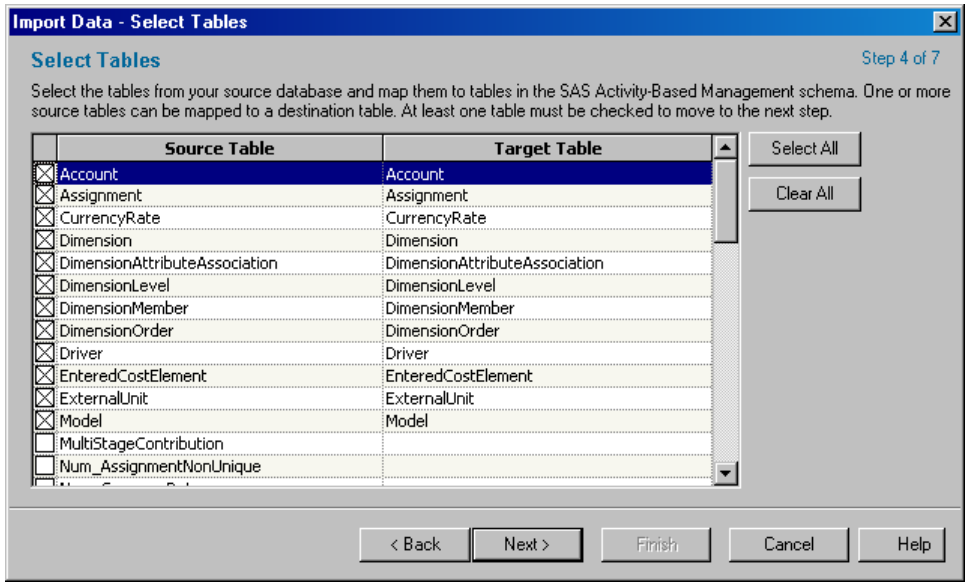
then the period/scenario association is not regenerated. This means that cube generation is faster because periods that have already been generated are not regenerated.

In order to support incremental cube generation, SAS Cost and Profitability Management provides a new **Periodic data only** option in the Import Wizard that allows you to import only the periods that have changed in a model (for example, the new periods).



Note: Incremental cube generation is not supported for a Postgres database with SAS OLAP.

If you select **Periodic data only**, then only those staging tables that contain periodic data are displayed in the Import Wizard for you to select for importing.



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

The following table lists staging tables that contain periodic data and those that contain structural data:

Periodic data	Structural data
Account	Dimension
Assignment	DimensionMember
CurrencyRate	DimensionLevel
ExternalUnit	DimensionOrder
EnteredCostElement	Driver
ValueAttributeAssociation	ValueAttributes
ValueAttributePeridicDef	Period
DimensionalAttributeAssociation	PeriodLevel
	Scenario
	Scenariolevel

When you generate a cube for a model that has been generated before, SAS Cost and Profitability Management determines whether the entire cube must be regenerated or

whether only the new or modified periods need to be generated. You do not have to specify, when you generate a cube, whether you want incremental generation or not. SAS Cost and Profitability Management makes the determination for you.

Note: If you are using Microsoft Analysis Services for cubes, then incremental cube generation is available only if you are using either Microsoft SQL Server Enterprise or Microsoft SQL Server Developer. Incremental cube generation is not available with Microsoft SQL Server Standard.

Note: If you are using SAS OLAP for cubes, then incremental cube generation is not available under certain circumstances because SAS OLAP does not allow deleting or updating a Period/Scenario inside a cube. For example, if you have generated a cube containing a Period/Scenario which you subsequently modify inside the model, then when you regenerate the same cube, adding a new Period/Scenario, the entire cube must be regenerated because the previously generated Period/Scenario can neither be deleted nor updated. When you generate a cube using SAS OLAP, SAS Cost and Profitability Management determines whether it can use incremental cube generation or whether it must regenerate the entire cube.

Note: Cubes in SAS OLAP that are generated using either the NO_NWAY option or the NONUPDATEABLE option are not eligible for incremental generation. See "Cube Options for SAS OLAP" on page 515. Also see The OLAP Procedure in the SAS OLAP Server: User's Guide for more information:: **`http://support.sas.com/documentation/onlinedoc/olap/index.html`**.

Note: With the implementation of incremental cube generation, OLAP views that were saved in a previous release of SAS Cost and Profitability Management will no longer work in SAS Cost and Profitability Management 7.2 if you are using Microsoft Analysis Services to build the cube. This is because the saved view is an MDX query, and the period names embedded in the query are no longer correct.

Chapter 65

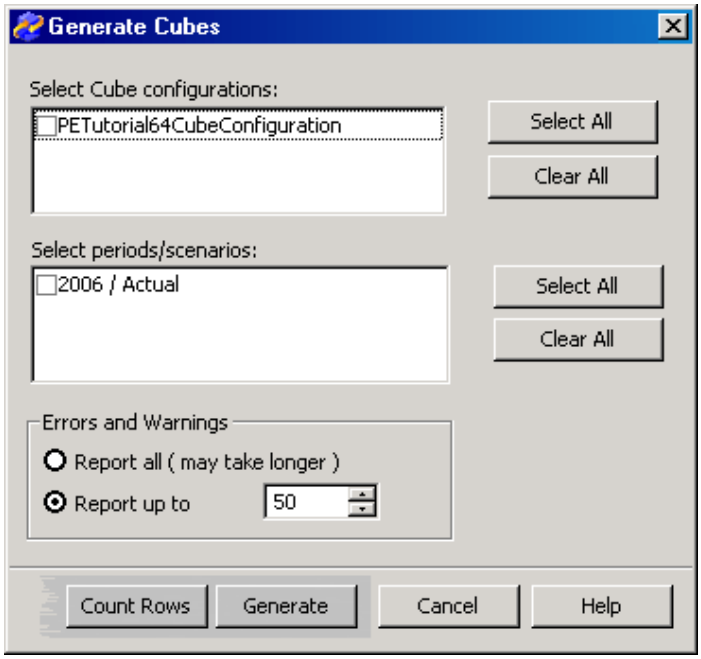
Windows for Cube Generation

Generate Cubes

How to Generate a Cube

Use the Generate Cubes dialog box to generate a cube and/or fact table for a model.

1. Open the model for which you want to generate cubes.
2. Select **Model** ⇨ **Generate Cubes**. The Generate Cubes dialog box opens.



3. Select the cube configurations to use. Each cube configuration generates a single cube.

   See "Create a Cube Configuration" on page 507.

4. Select **Force Cube Generate** to generate the cubes even though they have already been generated. See "Incremental Cube Generation" on page 569.
5. Select either a period/scenario association or **Select All**.
6. Select how many error and warning messages to display.
7. Click **Count Rows** to count the number of rows in the cube to be generated. This gives you an idea of how long it will take to generate the cube. (It is not required to count the rows before generating.)

How to Access the Generate Cubes Dialog Box

1. Open a model.
2. Select **Model** ⇨ **Generate Cube**

Note: Before you can generate a cube, you must have created a cube configuration that contains your options for generating the cube or fact table.

See Also

- "Generate Cubes" on page 573
- "Create a Cube Configuration" on page 507

Chapter 66

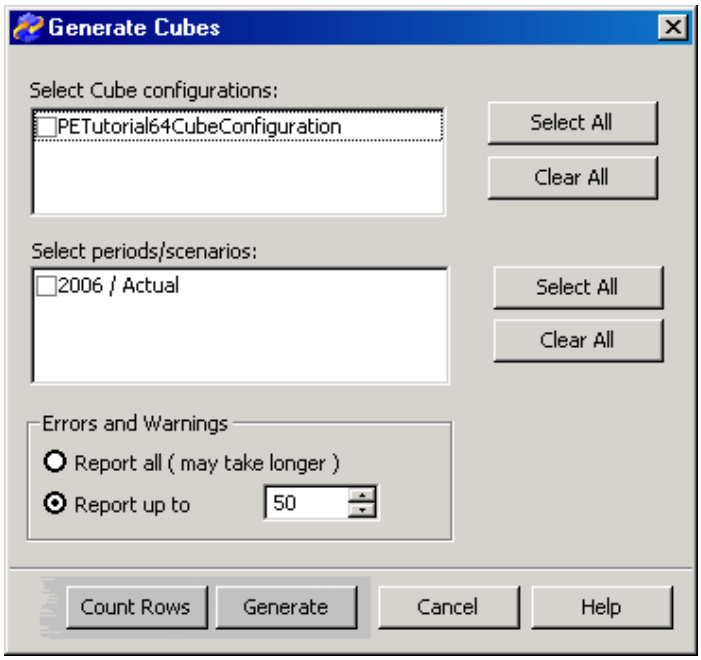
How To: Generating and Managing Cubes

Generate Cubes

How to Generate a Cube

Use the Generate Cubes dialog box to generate a cube and/or fact table for a model.

1. Open the model for which you want to generate cubes.
2. Select **Model** ⇨ **Generate Cubes**. The Generate Cubes dialog box opens.



3. Select the cube configurations to use. Each cube configuration generates a single cube.

   See "Create a Cube Configuration" on page 507.

4. Select **Force Cube Generate** to generate the cubes even though they have already been generated. See "Incremental Cube Generation" on page 569.

5. Select either a period/scenario association or **Select All**.

6. Select how many error and warning messages to display.

7. Click **Count Rows** to count the number of rows in the cube to be generated. This gives you an idea of how long it will take to generate the cube. (It is not required to count the rows before generating.)

How to Access the Generate Cubes Dialog Box

1. Open a model.

2. Select **Model** ⇨ **Generate Cube**

Note: Before you can generate a cube, you must have created a cube configuration that contains your options for generating the cube or fact table.

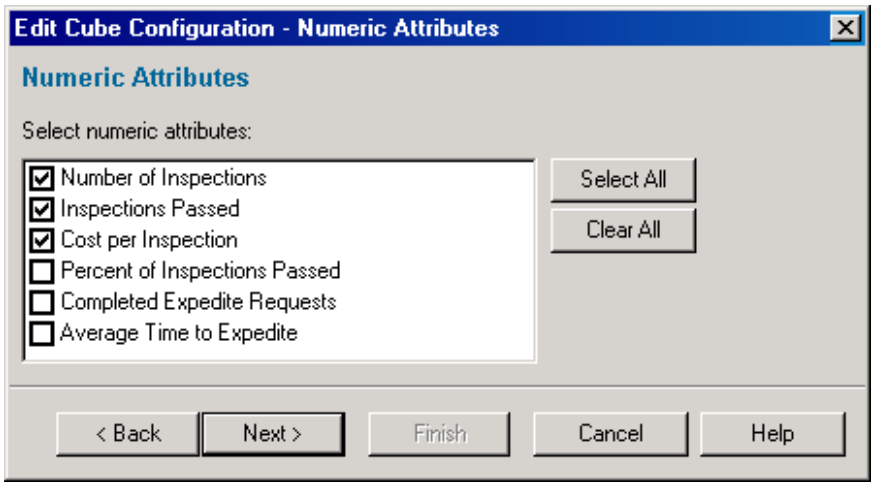
See Also

- "Generate Cubes" on page 575
- "Create a Cube Configuration" on page 507

Include Numeric Attributes in a Cube

To include numeric attributes in a cube:

1. Go to the Numeric Attributes dialog of the Cube Configuration wizard. See "Cube Configuration: Select Numeric Attributes" on page 517.



2. Select the numeric attributes to be included in the cube.

   When you generate a cube, then the numeric attributes that you select are included in the cube.

You can select the numeric attributes that are to be checked by default in a new cube configuration by doing the following:

1. Open a model and select **Model > Properties**.
2. Select the **Attributes in Cube** tab.
3. Select the numeric attributes that are to be checked by default in a new cube configuration for that model.

Attributes that you select in Model Properties are automatically checked in a new cube configuration to be included in the generated cube. However, you can uncheck the attributes in the cube configuration and select others before generating the cube.

See Also

"Cube Configuration: Select Numeric Attributes" on page 517

Display the Internal Name of a Cube

Instead of using SAS Cost and Profitability Management to view a generated cube, you can use SAS Enterprise Guide, SAS Web Report Studio, or SAS OLAP Cube Studio. Or you can use a third party tool other than SAS to view a cube and its fact table. Fact

tables are stored in the database and cubes are stored in the OLAP server. To open a fact table or cube you need to know its internal name.

Note: Using a program other than SAS Cost and Profitability Management to modify a cube that has been generated by SAS Cost and Profitability Management can affect display of the cube inside of SAS Cost and Profitability Management.

To display the internal name of a fact table or cube:

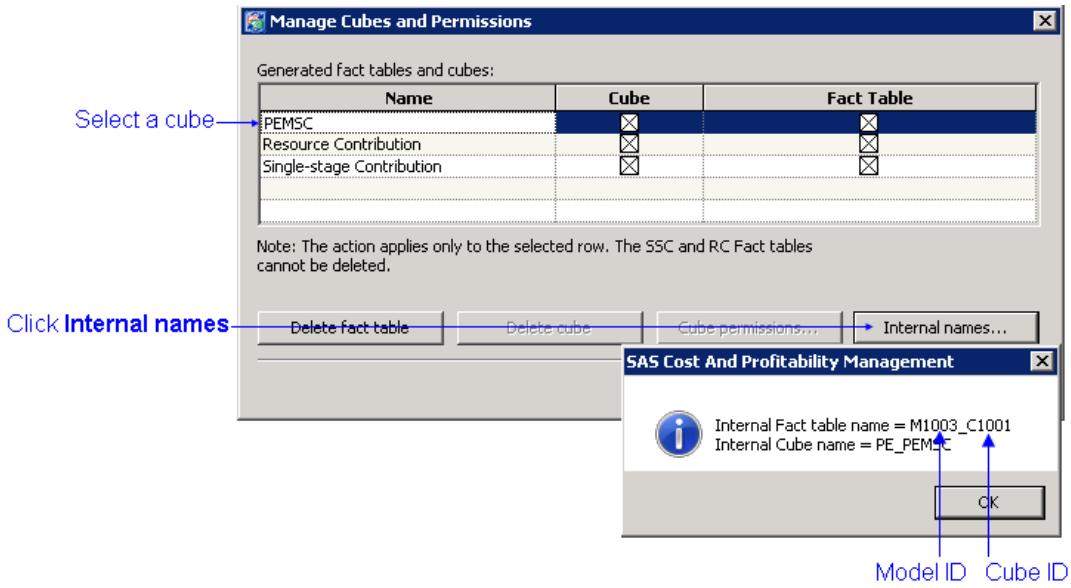
1. Open the model for whose fact tables or cubes you want to display the internal name.
2. Select **Model** ⇨ **Manage Cubes and Permission** .

   The Manage Cubes and Permissions dialog box opens.

   Note: For managing Microsoft Analysis Services cubes, you must have logged onto SAS Cost and Profitability Management having specified a domain name—for example, domain123/userxyz.
3. Select the fact table or cube whose internal name you want to display.

   Note: You can select only one row at a time.
4. Click **Internal names**. A dialog box opens displaying the internal names.

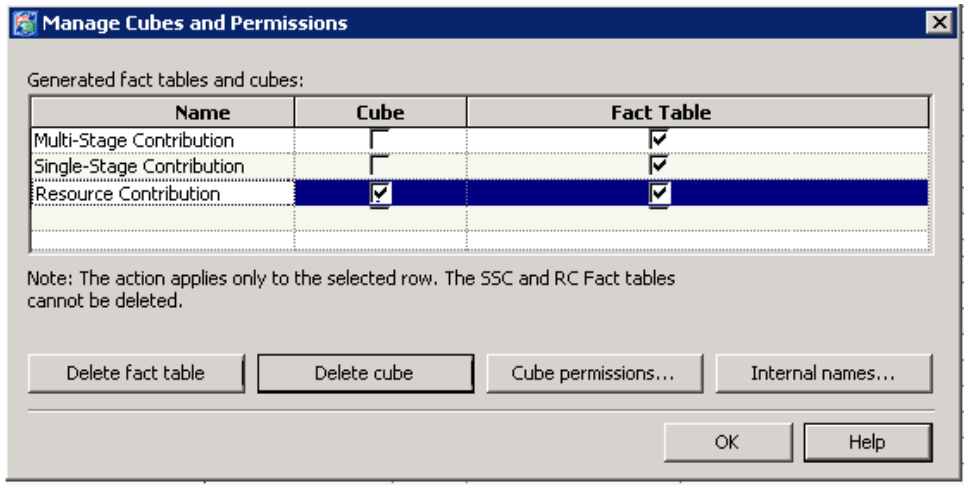
   Note: The internal name is composed by concatenating the model ID with the cube ID, separated by an underscore.



Delete a Cube or Fact Table

1. Open the model whose cubes or fact tables you want to delete.
2. Select **Model** ⇨ **Manage Cubes and Permissions**.

   The Manage Cubes and Permissions dialog box opens.



3. Select the cube or fact table that you want to delete.

   Note: You can delete only one cube or fact table at a time.

4. Click **Delete**.

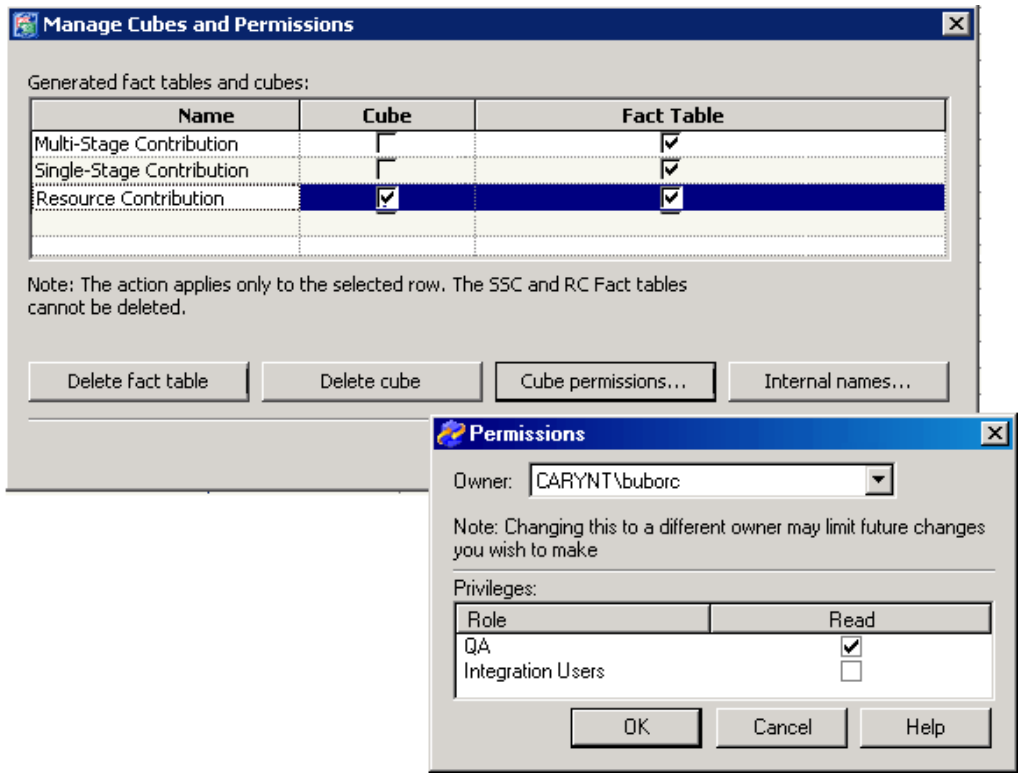
Manage Cube Permissions

After creating a cube, you can change its owner, and add or remove Read access to it.

To manage cube permissions:

1. Open the model for whose cubes you want to manage permissions.
2. Select **Model** ⇨ **Manage Cubes and Permissions**.

   The Manage Cubes and Permissions dialog box opens.



Note: For managing Microsoft Analysis Services cubes, you must have logged onto SAS Cost and Profitability Management having specified a domain name—for example, domain123/userxyz.

3. Select the cube whose permissions you want to change.

   Note: You can change only one cube at a time.

4. Click **Permissions**. The Permissions dialog box opens.

5. You can add or remove Read access to groups. All members of the group inherit the permissions that you select for that group.

See Also

"Display the Internal Name of a Cube" on page 577

Part 19

OLAP Analysis

Chapter 67

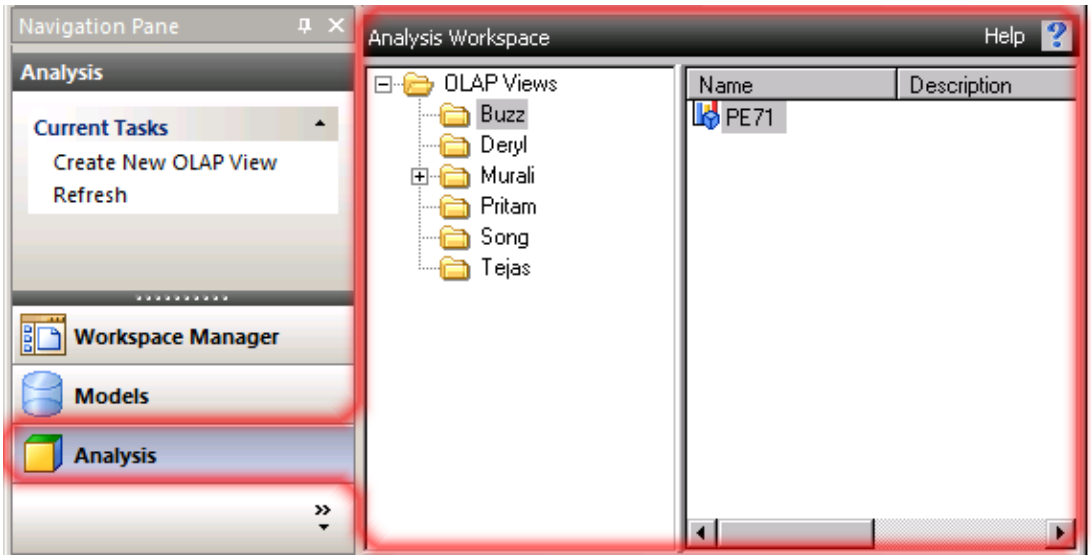
Using the Analysis Workspace

Analysis Workspace

About the Analysis Workspace

In the Analysis workspace, you can open an OLAP view.

Note: The availability of these features depends on your permissions.



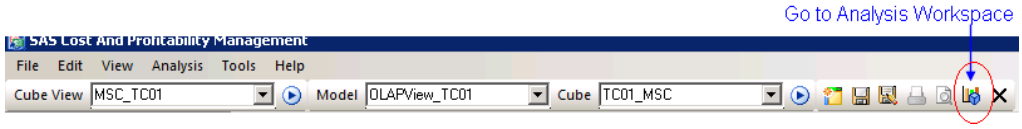
The list of Folders and the list of OLAP Views correspond to the OLAP branch of the server area in Workspace Manager.

How to Access the Analysis workspace

Do one of the following:

- If no OLAP view is open, click **Analysis** in the Navigation Pane.

- If an OLAP view is already open, click the **Go to Analysis Workspace** icon in the toolbar



See Also

- "Create an OLAP View" on page 592
- "Open an OLAP View" on page 592

OLAP Views

About OLAP Views

OLAP is a technology that is used to create decision-support software. OLAP enables users to quickly analyze data that has been summarized into multidimensional views and hierarchies. By summarizing predicted queries into multidimensional views and hierarchies before run time, SAS Cost and Profitability Management's OLAP tool provides the benefit of increased performance over traditional database access tools. Most of the resource-intensive calculation that is required to summarize the data is done before a query is submitted.

You use the OLAP Analyzer to open and view an OLAP cube.See "OLAP Analyzer" on page 587.

Cube Availability

When you try to display a cube in an OLAP view, the following situations might cause the cube to be unavailable:

- Another user is currently regenerating the cube.
- The cube on which a saved OLAP view is based has been deleted.

SAS OLAP Limitations

The following restrictions exist in the SAS OLAP Server:

- Most OLAP names can be up to 32 characters in length except for aggregation names that can be up to 256 characters. When SAS processes a name it will be uppercased.
- Level names must be unique within a cube.
- Measure names must be unique within a cube.
- Number of dimensions: Maximum 128 (Minimum 1).
- Number of levels per dimension: Maximum 19.
- Number of levels per cube: Maximum 256.
- Number of measures per cube: Maximum 1024.
- Maximum length of a unique name is 32767 characters.
- Size of the MDX string is unlimited, but extremely long strings can affect performance

Chapter 68

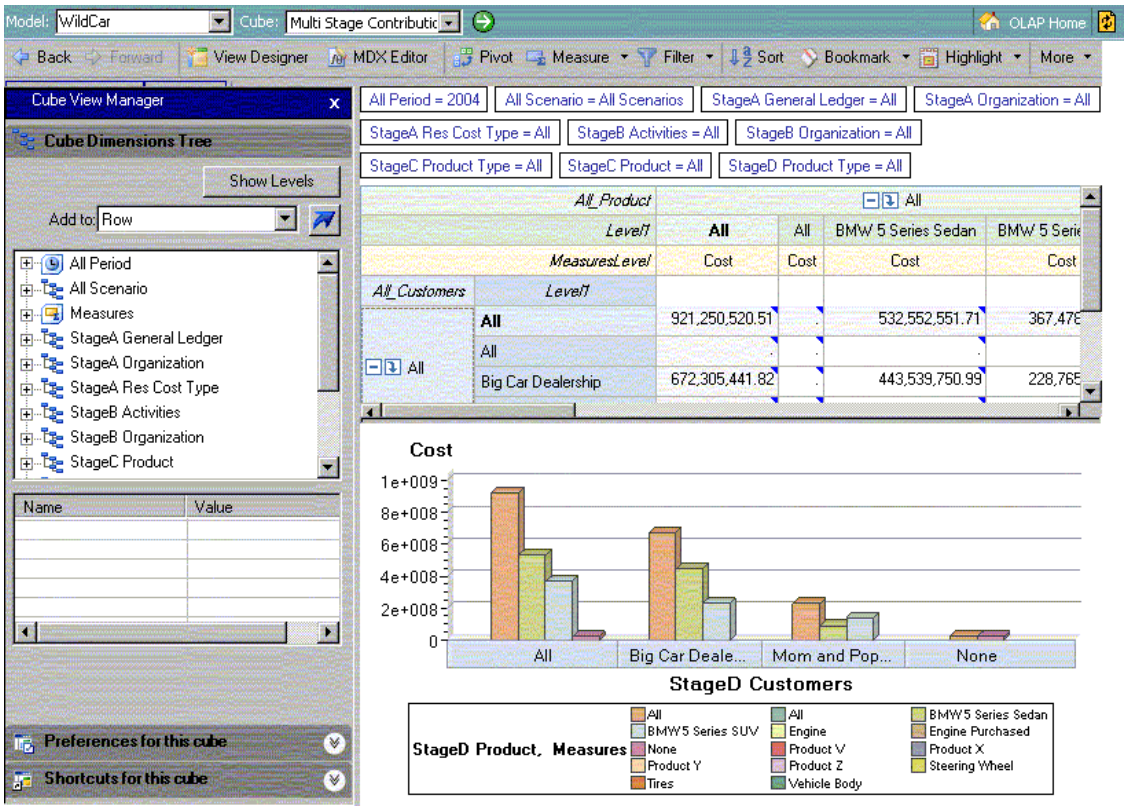
Analysis Workspace Windows

OLAP Analyzer

About the OLAP Analyzer

With the OLAP analyzer you can study measures at dimension intersections by creating an OLAP view. You can analyze dimensions and measures in Grid View, Chart View, and Cube Explorer View.

Note: For the most information on the OLAP Analyzer, Press **F1** for help while working in the analyzer, or select **Help** ⇨ **SAS OLAP Analyzer Help**.



The predefined cube that you choose determines which dimensions and measures are available to you, as well as what kinds of analyses you can perform.

Note: Models are not displayed in the **Model** menu until you generate the model's cubes.

How to Access the OLAP Analyzer

Open an OLAP view in the Analysis Workspace. See "Open an OLAP View" on page 592..

See Also

"SAS OLAP Limitations" on page 585

Save OLAP View As Dialog Box

About the Save OLAP View As Dialog Box

In the Save OLAP View As dialog box, you can save a customized OLAP view.

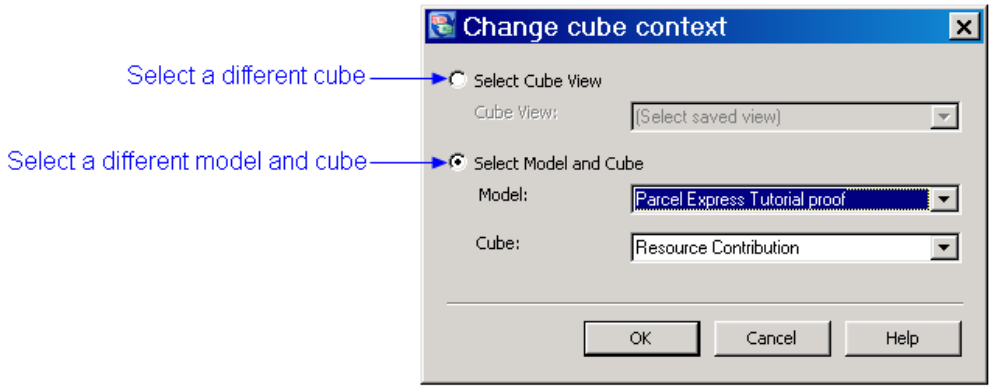
How to Access the Save OLAP View As Dialog Box

Open an OLAP view and select **Analysis** ⇨ **Save View As**.

Change Cube Context Dialog Box

About the Change Cube Context Dialog Box

Use this dialog box to select a different cube to view for the currently open model or to select a different cube of a different model.



How to Access the Change Cube Context Dialog Box

Select **Analysis** ⇨ **Change cube context**.

Note: You must be in the Analysis Workspace with a cube view currently open for this menu item to be available.

See Also

"Analysis Workspace" on page 583

Chapter 69

How To: Analysis Workspace

Use the SAS OLAP Analyzer

SAS Cost and Profitability Management uses the SAS OLAP Analyzer to view cubes. With the SAS OLAP Analyzer, you can analyze dimensions and measures in a grid view, a chart view, and a Cube Explorer View. The predefined cube that you choose determines which dimensions and measures are available to you, as well as what types of analyses you can perform. You can view data in one or two windows, depending on your needs. The OLAP toolbar buttons that are available to you depend on which window is selected and on what type of information is displayed in the window.

Note: Models are not displayed in the **Model** drop-down list of the OLAP Analyzer until you generate the model's cubes.

The SAS OLAP Analyzer is usually part of SAS Enterprise Guide, which you might not have installed on your system. (Even if you don't have SAS Enterprise Guide, you do have the SAS OLAP Analyzer as part of SAS Cost and Profitability Management.) The help for SAS OLAP Analyzer includes help for SAS Enterprise Guide.

You can open the Help for SAS OLAP Analyzer from the Help menu of SAS Cost and Profitability Management.

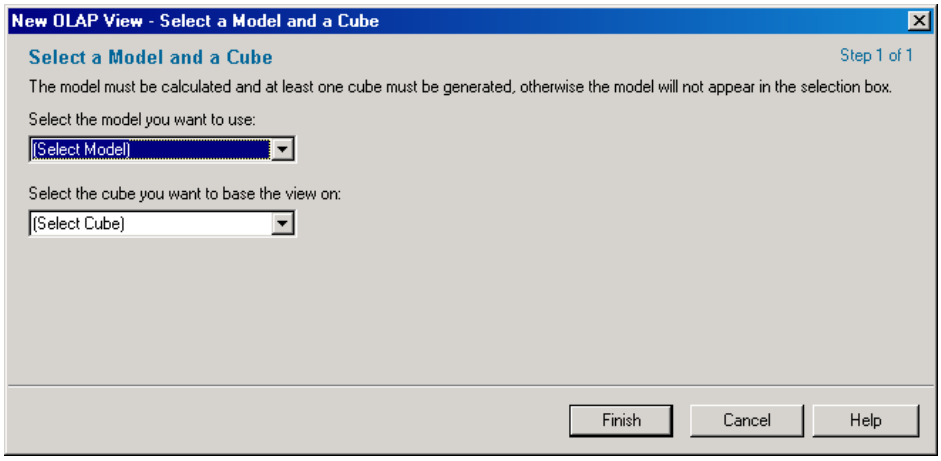
See Also

- "OLAP Analyzer" on page 587
- "SAS OLAP Limitations" on page 585

Create an OLAP View

1. Select **File** ⇨ **New** ⇨ **OLAP View**.

   The New OLAP View Wizard appears.



2. From the **Select the model you want to use** drop-down list, select a model.
3. From the **Select the cube you want to base the view on** drop-down list, select a cube.

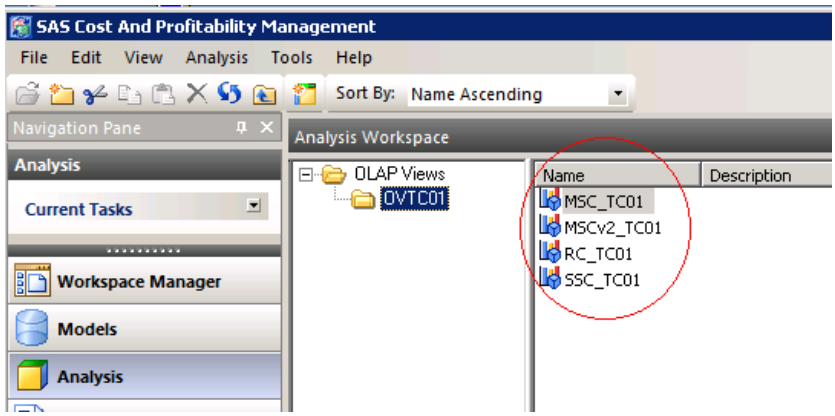
   The list contains only those cubes that have been generated.
4. If the cube contains a large amount of data and you want to quickly create the OLAP view, select the **Do not select any dimensions by default** option.

Open an OLAP View

Open an OLAP View with No OLAP View Open

If you open the Analysis workspace without an OLAP view open, you can open an OLAP view from the list of views in the workspace. The list of **Folders** and **OLAP Views** corresponds to the OLAP branch of the server area in Workspace Manager.

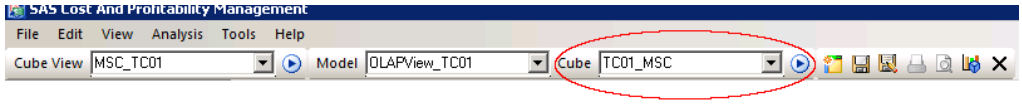
1. From the list of **Folders**, select a folder.
2. From the list of **OLAP Views**, click a view.



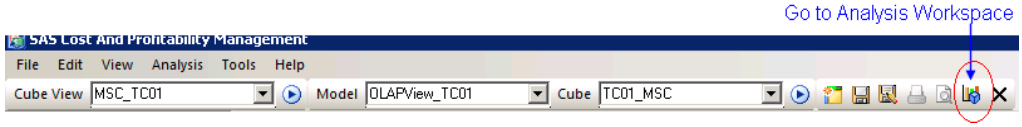
Open an OLAP View with an OLAP View Already Open

If an OLAP view is already open in the Analysis workspace and you want to open a different view, do one of the following:

- Select an OLAP view from the **Cube View** down-down list.



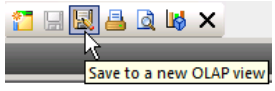
- Click the Go to Analysis Workspace button and select another OLAP view.



Save an OLAP View

To save an OLAP view, do the following:

1. Go to the Analysis workspace.
2. Open an OLAP view.
3. Click **Save the current OLAP view**, or **Save to a new OLAP view**.



Once an OLAP view has been saved, open it from the Analysis workspace.

When you save an OLAP view, all of the following are saved:

- the cube
- the model
- the OLAP view's layout
- the OLAP view's contents

During analysis, changes that you make to a grid view, a chart view, and the Cube Explorer View are retained during a session, even when you return to the OLAP view after viewing other tabs. However, your changes are lost when you close SAS Cost and Profitability Management or when you close a grid view, a chart view, or the Cube Explorer View.

If you want these changes to be available later, save the OLAP view. However, the window positions and window states are not saved.

Change the Appearance of an OLAP Window

1. Open an OLAP view on the Analysis workspace.
2. First, click a window's title bar.
   a. To display a grid, select **OLAP** ⇨ **Grid**.
   b. To display a chart, select **OLAP** ⇨ **Chart** ⇨ **<type of chart>**.
   c. To display a Decomposition Tree, select **OLAP** ⇨ **Decomposition Tree**.
   d. To display a perspective view, select **OLAP** ⇨ **Perspective**.

Show or Hide the Cube View Manager

Select **Analysis** ⇨ **View Manager**.

A check indicates that View Manager is displayed.

Open the View Editor

Select **Edit View** ⇨ **Edit with View Editor**.

The View Editor allows you to create OLAP views by dragging and dropping dimensions and measures into the view.

Open the MDX Editor

Select **Edit** ⇨ **Edit with MDX Editor**.

The Edit MDX Statement dialog box appears.

Export a Cube

Select **Analysis** ⇨ **Export To Excel**.

Microsoft Excel opens and displays the exported data.

Print an OLAP View

From the **Print** menu, select **Print Preview** or **Print**.

Part 20

What-If Analysis

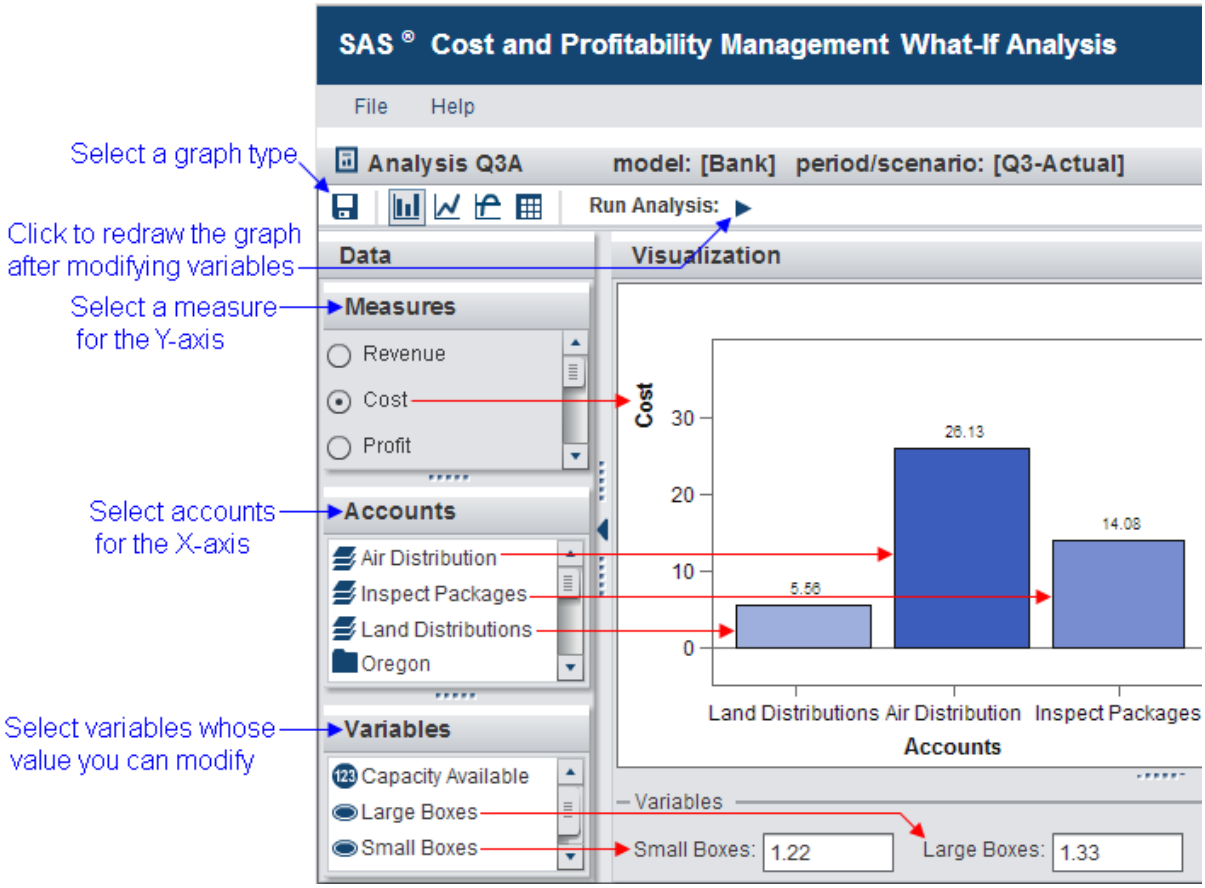
Chapter 70

What is What-If Analysis?

Introduction

SAS Cost and Profitability Management What-If Analysis allows you to see what happens to accounts when you modify selected variables. After modifying the value of selected variables, you can perform a calculation to see the effect on a measure, such as Cost or Profit, for the selected accounts.

The following picture shows a sample what-if analysis graph.



Graphs

You can produce the following charts and graphs:

- Bar chart
- Line graph
- Profit cliff
- Grid view

Measures

The selected measure whose value appears on the Y-axis of a graph can be one of the following:

- Cost
- Revenue
- Profit
- Unit Cost
- Unit Revenue
- Unit Profit

Accounts

Selected accounts appear along the X-axis of a what-if analysis graph. You can mark any account in a model for what-if analysis. Specifically, this includes both accounts:

leaf accounts 

roll-up accounts

See "Mark Accounts for the X-Axis" on page 606.

Variables

You can modify the value of selected variables and then recalculate the model to see the effect on the measures in the graph. The variables whose value you can modify are the following:

Entered cost elements

You can mark any entered cost element to appear as an independent variable in what-if analysis. See "Mark Entered Cost Elements as Independent Variables" on page 607.

For each entered cost element that you select, two independent variables are created when you use the cost element in a what-if analysis—one for Entered Cost and another for Entered Unit Cost.

Numeric attributes

You can mark any numeric attribute to appear as an independent variable in what-if analysis. See "Mark Numeric Attributes as Independent Variables" on page 608.

Note: Only the default value of a numeric attribute enters into what-if analysis. If the default value of a numeric attribute is overridden when the attribute is attached to a particular account, then changing the default value has no effect for that particular account. It is only where the default value is not overridden that changing the default value has an effect.

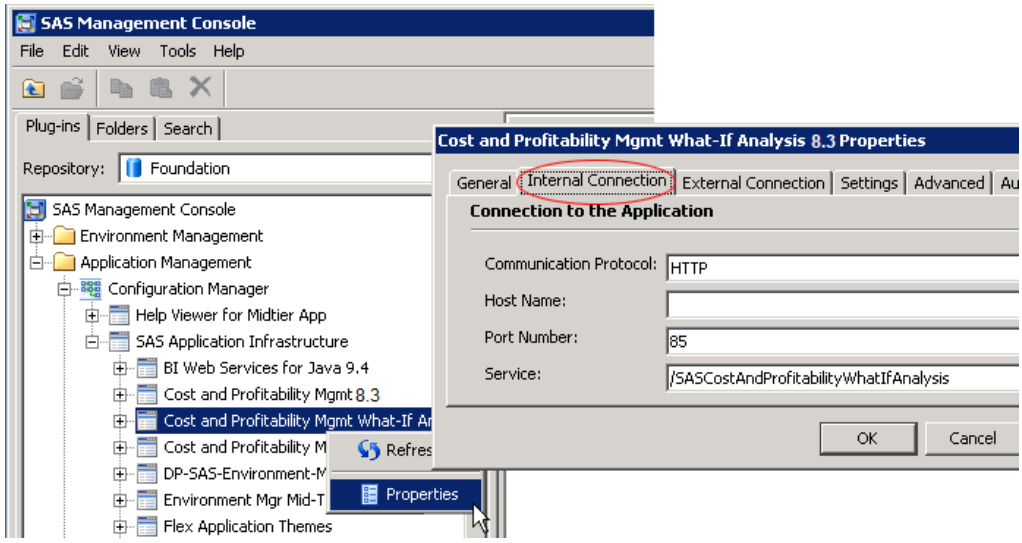
Log On

Log on to What-If application by typing the URL into the command line of a browser. To determine the URL, you can do the following:

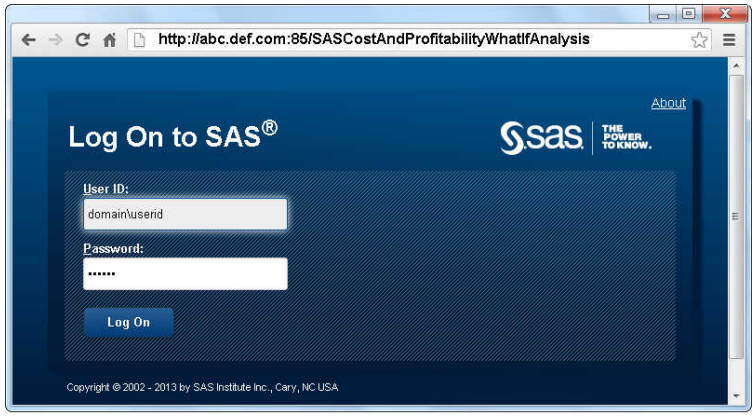
1. Log on to SAS Management Console as an administrator, and access the SAS Cost and Profitability Management middle-tier server.
2. Click the **Plug-Ins** tab.
3. Expand **Application Management**.
4. Expand **Configuration Manager**.
5. Expand **SAS Application Infrastructure**.
6. Right-click **Cost and Profitability Mgmt What-If Analysis 8.3** and select **Properties**.
7. Click the **Internal Connection** tab on the Properties window.

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

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



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



9. Log on with a user ID and password.

   Note: The user ID must exist in the SAS Metadata Server with the capability to Create What-If Analysis Usually, this means that the user is a member of a group with that capability. Alternatively, the user can have a role with that capability. See Chapter 1, "User Capabilities and Groups," in *SAS Cost and Profitability Management: Data Administration Guide*.

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#p1871e69gmwdr0n1o182krslc10p.htm.

For More Information

See *SAS Cost and Profitability Management: What-If Analysis* (available from the Help menu)

Chapter 71

Marking Items for What If Analysis

Overview

In order for anyone to do what-if analysis using the What-If Analysis application, there must be model elements that have been marked for what-if analysis.

Using the SAS Cost and Profitability Management client, you can mark the following model elements for use in what-if analysis:

Accounts
: You can mark any account, including roll-up accounts to appear as items along the X-axis of a what-if analysis. See "Mark Accounts for the X-Axis" on page 606.

Entered cost elements
: You can mark any entered cost element to appear as an independent variable in what-if analysis. See "Mark Entered Cost Elements as Independent Variables" on page 607.

  For each entered cost element that you select, two independent variables are created when you use the cost element in a what-if analysis—one for Entered Cost and another for Entered Unit Cost.

Numeric attributes
: You can mark any numeric attribute to appear as an independent variable in what-if analysis. See "Mark Numeric Attributes as Independent Variables" on page 608.

  Note: Only the default value of a numeric attribute enters into what-if analysis. If the default value of a numeric attribute is overridden when the attribute is attached to a particular account, then changing the default value has no effect for that particular account. It is only where the default value is not overridden that changing the default value has an effect.

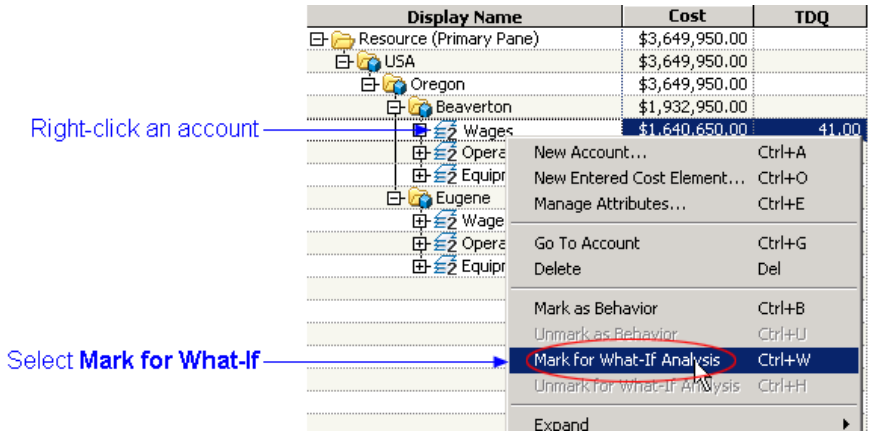
Mark Accounts for the X-Axis

To mark an account for use on the X-axis of a graph in What If Analysis:

1. Right-click an account in any module view.

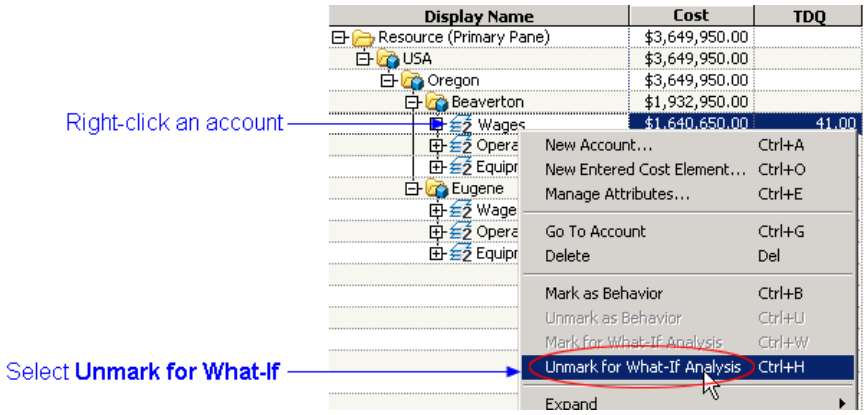
   Note: You can select any account including roll-up accounts.

2. Select **Mark for What-If Analysis**.



To Unmark an Account for What-If

1. Right-click the account.
2. Select **Unmark for What-If Analysis**.



Select a Measure for the Y-Axis

The measures that you can select to appear on the Y-axis of a what-if analysis are pre-determined. For a profit-cliff graph, only Profit can appear as the measure on the Y-axis. For any other graph, you can select one of the following measures:

- Cost

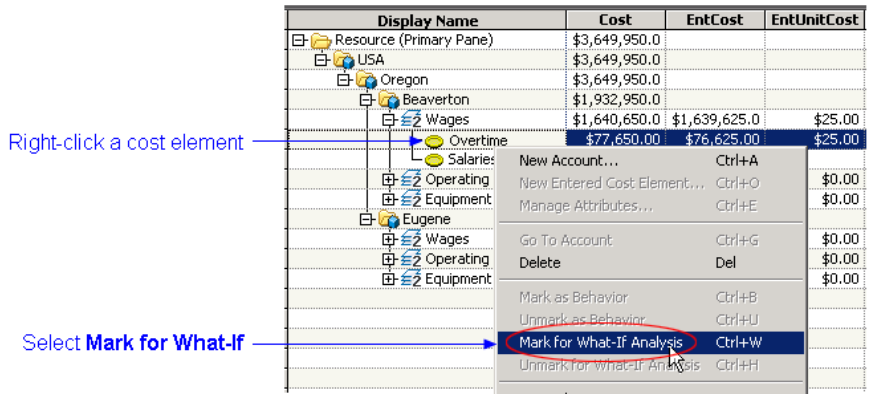
- Revenue
- Profit
- Unit Cost
- Unit Revenue
- Unit Profit

Note: You cannot change the list of available measures.

Mark Entered Cost Elements as Independent Variables

To mark an entered cost element for use as an independent variable in What If analysis:

1. Right-click the entered cost element in a module view.
2. Select **Mark for What-If Analysis**.

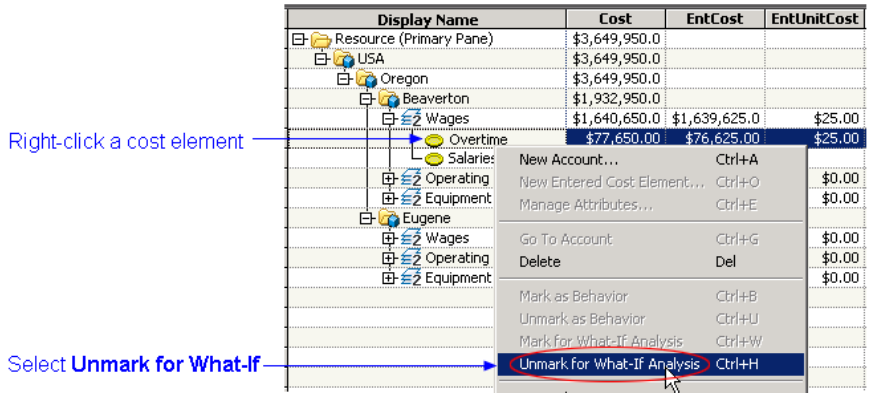


When you mark an entered cost element for what-if analysis, two independent variables are created whose value you can modify during what-if analysis:

- one for the entered cost element's EnteredCost
- one for the entered cost element's EnteredUnitCost

To Unmark a Cost Element for What-If

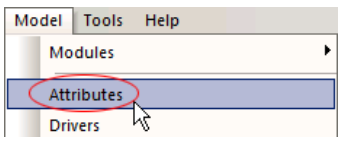
1. Right-click a cost element in a module view.
2. Select **Unmark for What-If Analysis**.



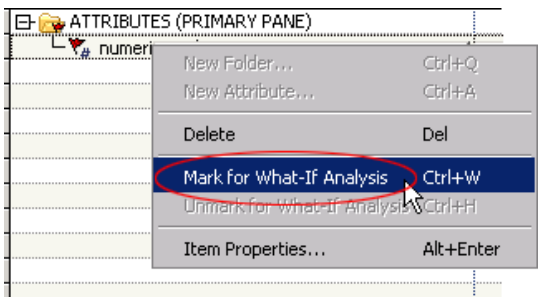
Mark Numeric Attributes as Independent Variables

To mark a numeric attribute for use as an independent variable in What If Analysis:

1. Select **Model** ⇨ **Attributes** to open the Attributes page for a model.

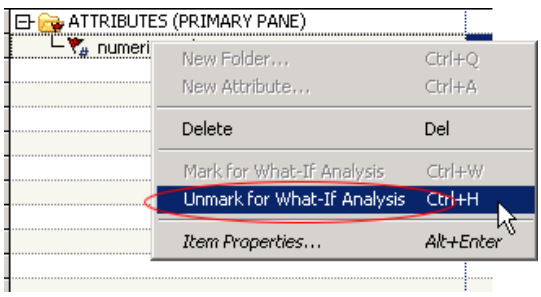


2. Right-click a numeric attribute.
3. Select **Mark for What-If Analysis**.



To Unmark an Attribute for What-If Analysis

1. Right-click a numeric attribute.
2. Select **Unmark for What-If Analysis**.



Part 21

Conducting Surveys

Chapter 72

Cost and Profitability Management Surveys

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 "Assignment table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.
Entered Cost Element	EnteredCostElement.EnteredUnitCost EnteredCostElement.EnteredCost See "EnteredCostElement table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.
Output Quantities	Account.OutputQuantityUE See "Account table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

Type of survey	Fields that can be updated
Attribute	ValueAttributeAssociation.NumericValue ValueAttributeAssociation.StringValue See "ValueAttributeAssociation table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

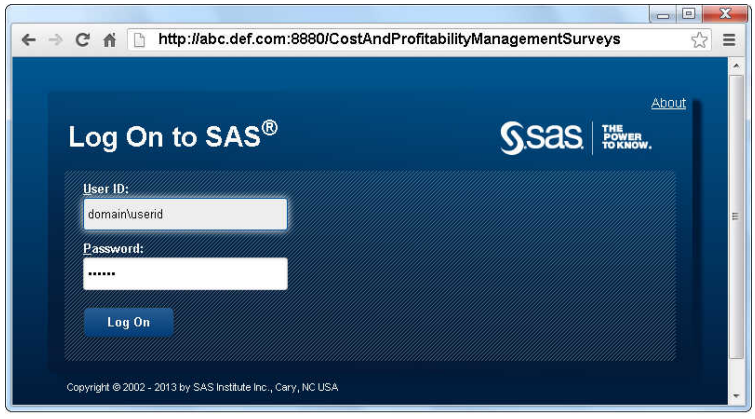
Logging On

Log on to surveys by entering 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 8.3** and select **Properties**.
7. Click the **Connection** tab on the Properties window.

   The connection information appears.
8. Use the connection information that is provided in the fields of the **Connection** tab to prepare the URL for invoking surveys.

   For example, the surveys URL can be **`http://xyz.sas.com:85/SASCostAndProfitabilityManagementSurveys`**.
9. Enter the survey URL into the command line of a browser.



10. 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," in *SAS Cost and Profitability Management: Data Administration Guide*.

For More Information

See Chapter 7, "Creating Surveys," in *SAS Cost and Profitability Management: Data Administration Guide*.

Part 22

Importing and Exporting

Chapter 73

Connecting to a Database

Overview

For importing or exporting, you can connect to a database using either a:

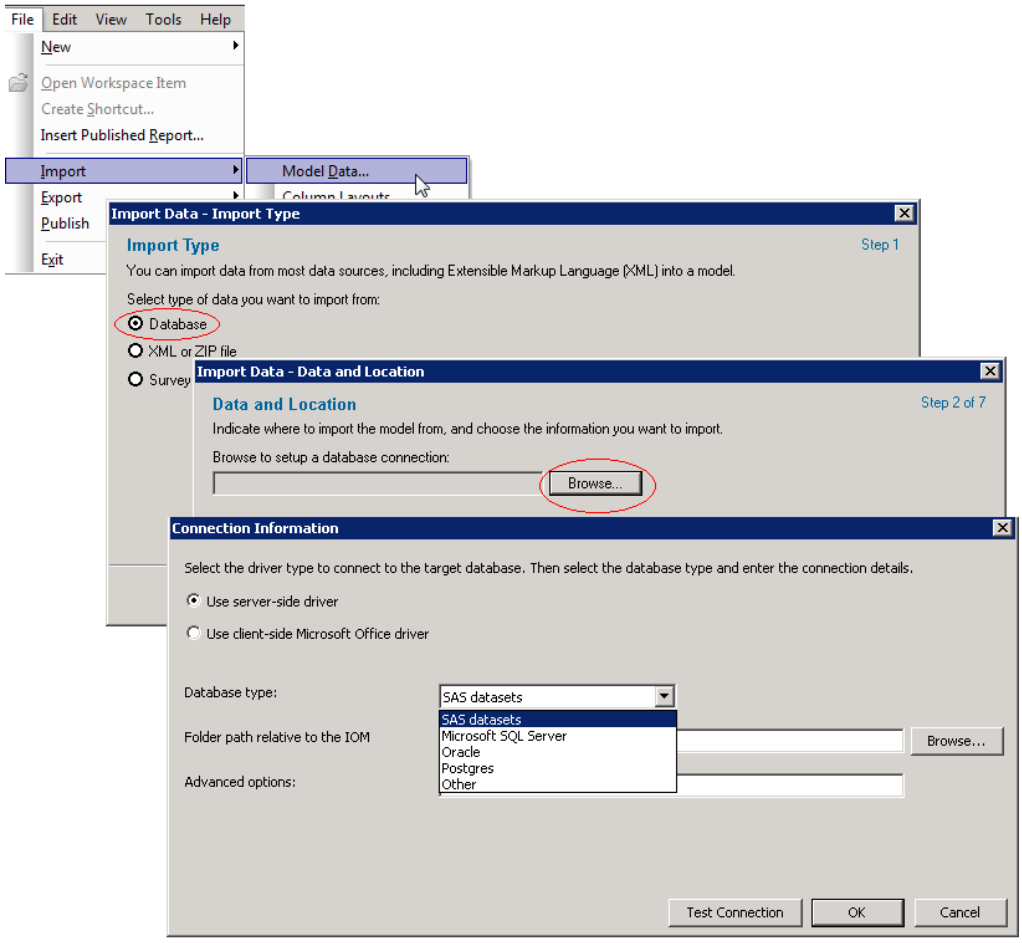
server-side driver
To use a server-side driver, you must have installed, on the IOM server machine, a 64-bit ODBC driver or the native driver for a particular database.

client-side Microsoft Office driver
To use a client-side Microsoft Office driver, you must have installed a 32–bit ODBC driver on the SAS Cost and Profitability Management client machine. See "Installing the ODBC Driver for Microsoft Access and Excel" on page 620.

To connect to a database:

1. From the **File** menu, select either to import or to export model data:
   - **File ⇨ Import ⇨ Model Data**
   - **File ⇨ Export ⇨ Model Data**
2. Select **Database**.
3. Click **Browse**.

   The Connection Information dialog opens.

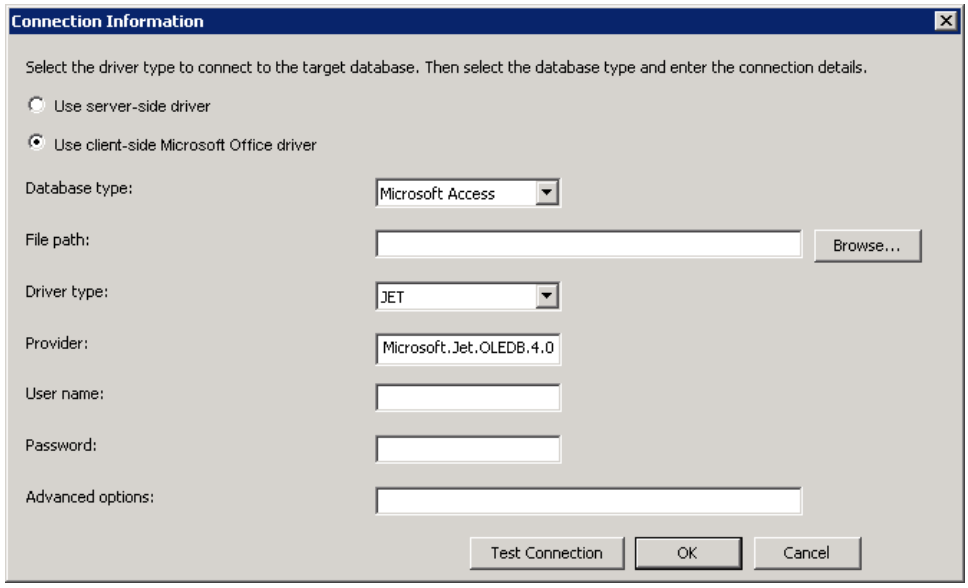


Client-Side Microsoft Office Driver

You can import from or export to Microsoft Access and Microsoft Excel using a client-side driver. This has the advantage that you can now import from or export to client installations of Microsoft Office even if your SAS Cost and Profitability Management server is on a UNIX system where the Microsoft Office drivers would not work.

Microsoft Access

The following picture shows an example of connecting to Microsoft Access. The file path that you enter should have READ/WRITE permission.

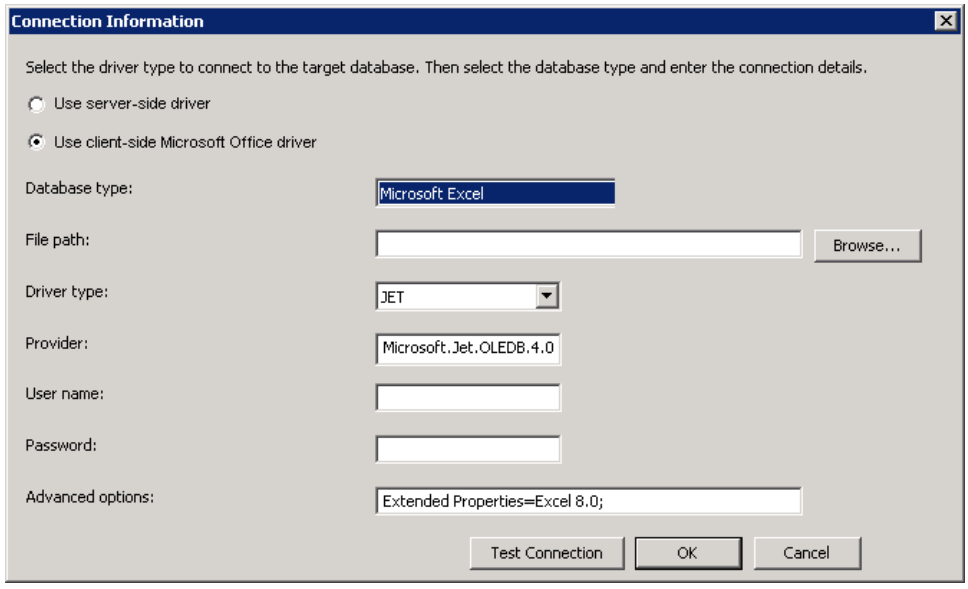


Note:

- You must have installed a 32–bit ODBC driver on the SAS Cost and Profitability Management client machine. See "Installing the ODBC Driver for Microsoft Access and Excel" on page 620.
- The file path that you specify is relative to the machine on which the SAS Cost and Profitability Management client is installed.

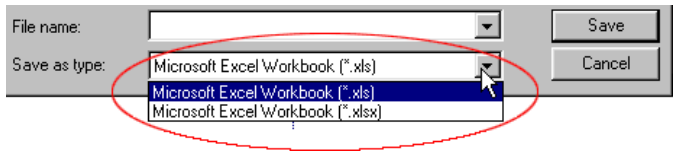
Microsoft Excel

The following picture shows an example of connecting to Microsoft Excel. The file path that you enter should have READ/WRITE permission.



Note:

- You must have installed a 32–bit ODBC driver on the SAS Cost and Profitability Management client machine. See "Installing the ODBC Driver for Microsoft Access and Excel" on page 620.
- It is recommended that you save to a new Excel file rather than replace an already-existing one. If the model being exported contains more columns than the already-existing Excel file, then (because of a limitation with Microsoft Office) export fails with the message "Too many fields defined"—even though, from the server point-of-view, the operation appears to have been successful. .
- You can save the Excel file as either *.xls(2003 format) or *.xlsx (2007 format).



- The file path that you specify is relative to the machine on which the SAS Cost and Profitability Management client is installed.

Installing the ODBC Driver for Microsoft Access and Excel

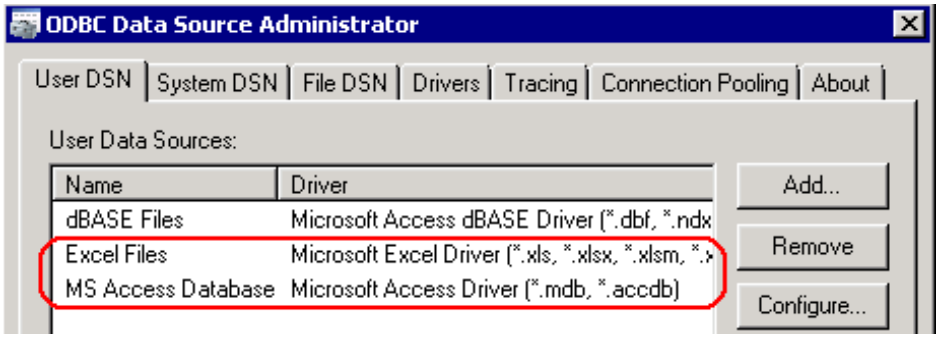
You must install 32–bit ODBC driver on the SAS Cost and Profitability Management client machine to import from or export to Access or Excel.

The following providers are supported:

- Microsoft.ACE.OLEDB.12.0 for use with either Microsoft Office 2007 or Microsoft Office 2010
- Microsoft.Jet.OLEDB.4.0 for use with JET

To display a list of ODBC drivers that are already installed, do the following:

1. Select **Start** ⇨ **Control Panel** on the SAS Cost and Profitability Management client machine.
2. Select **Administrative Tools**.
3. Select **Data Sources (ODBC)**.
4. Click the **Drivers** tab.
5. Verify that drivers for Microsoft Access and Microsoft Excel are listed as shown.



If the drivers are not installed, you can install them by running AccessDatabaseEngine.exe, which you can obtain from the Microsoft Download Center at http://www.microsoft.com/downloads/. Because the SAS Activity-Based Management

client is a 32–bit application and can work only with 32–bit drivers, there are four cases to consider when downloading the drivers:

You have not installed Microsoft Office
: You can download the 32–bit drivers for Office 2010. On the Microsoft Download Center, search for "Microsoft Access Database Engine 2010 Redistributable" and select the 32–bit verson.

You have installed Microsoft Office 2007, 32–bit
: If you do not already have the drivers, you can download them. On the Microsoft Download Center, search for "2007 Office System Driver: Data Connectivity Components". Or search for "Microsoft Access Database Engine 2010 Redistributable" and select the 32–bit version (the 32–bit driver for Office 2010 works with Office 2007).

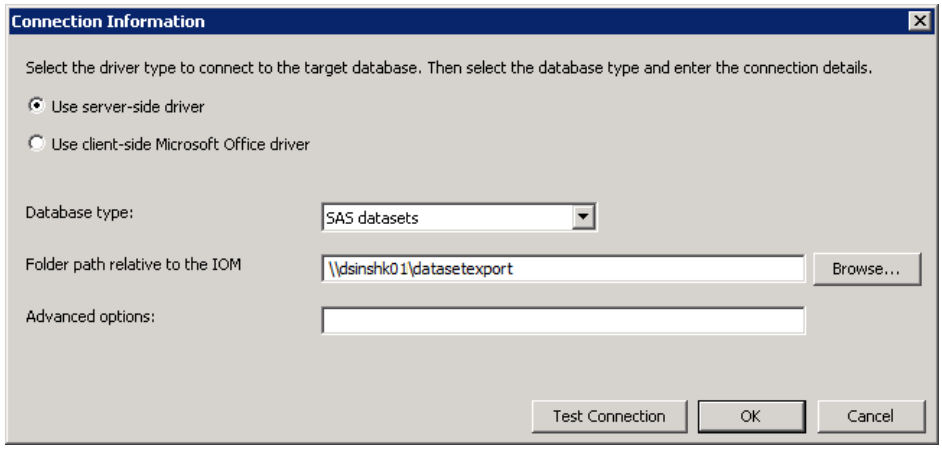
You have installed Microsoft Office 2010, 32–bit
: If you do not already have the 32–bit drivers, you can download them. On the Microsoft Download Center, search for "Microsoft Access Database Engine 2010 Redistributable" and select the 32–bit version.

You have installed Microsoft Office 2010, 64–bit
: If you have installed the 64–bit drivers, then you need to install the 32–bit drivers. However, Microsoft does not allow you to install 32–bit drivers for Office 2010 along with 64–bit drivers. Therefore, you must install the 32–bit drivers from Microsoft Office 2007 instead. On the Microsoft Download Center, search for "2007 Office System Driver: Data Connectivity Components".

Note: The 32–bit drivers for Microsoft Office 2007 do not replace the 64–bit drivers for Microsoft Office 2010, so the 64–bit drivers remain available for use with applications other than SAS Activity-Based Management.

Server-Side Driver with SAS Datasets

The following picture shows an example of using SAS datasets.



Folder path relative to the IOM server
: The folder containing SAS datasets for import or export must be accessible by the SAS IOM Server.

- If the folder is on the same machine as the SAS IOM server, then you can specify the folder using a path such as `c:\datasetexport`.

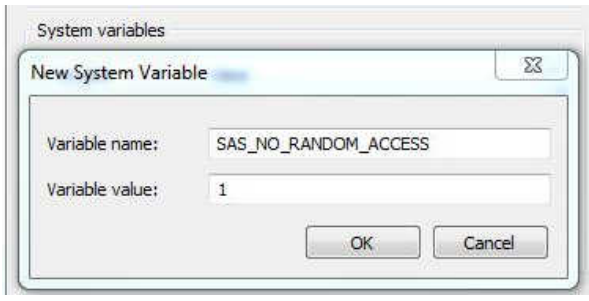
- If the folder is on a different machine than the SAS IOM server, then you must specify the machine name in specifying the path. For example, if the folder, **`datasetexport`**, is on machine **`XYZ`**, then you might specify the path as **`\\XYZ\datasetexport`**.

TIP If importing SAS datasets consumes an inordinate amount of RAM, you can sidestep the problem by setting the system variable SAS_NO_RANDOM_ACCESS. To set the system variable, do the following:

1. Stop the SAS Object Spawner service (**Start** ⇨ **All Programs** ⇨ **Administrative Tools** ⇨ **Component Services** ⇨ **Services**).
2. Right-click **My Computer** or **Computer** and select **Properties**.
3. Click **Advanced system settings** to open the System Properties dialog box
4. On the **Advanced** tab, click **Environment Variables**.
5. In the System variables pane (not the User variables pane), click **New**.

   The New System Variable dialog box appears.

   Note: Make sure that the title is **New System Variable** and not **New User Variable**.

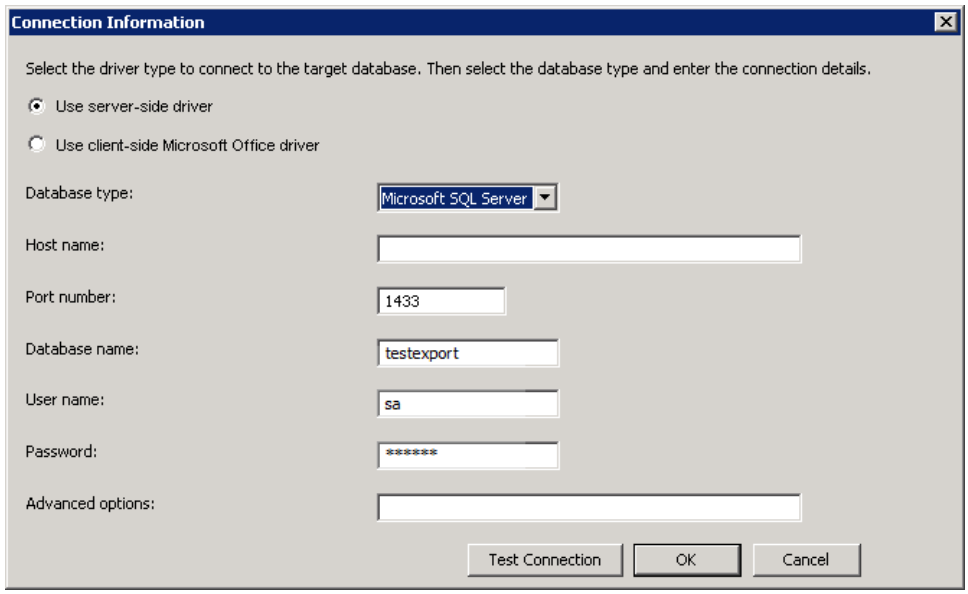


6. For **Variable name**, enter *SAS_NO_RANDOM_ACCESS*.

   For **Variable value**, enter *1*.
7. Click **OK**.
8. Start the SAS Object Spawner service.

Server-Side Driver with Microsoft SQL Server

The following picture shows an example of connecting to a Microsoft SQL Server database.



The following are some considerations to keep in mind when connecting to a Microsoft SQL Server database:

Host name

the MachineName where the target database is installed from where users want to import or export

Port

the TCP port where the target database instance is running

If the database is on a Named Instance, then go to the SQL Server Configuration manager to check the TCP port on which it is running. Named Instances run on a dynamic TCP port.

Advanced options

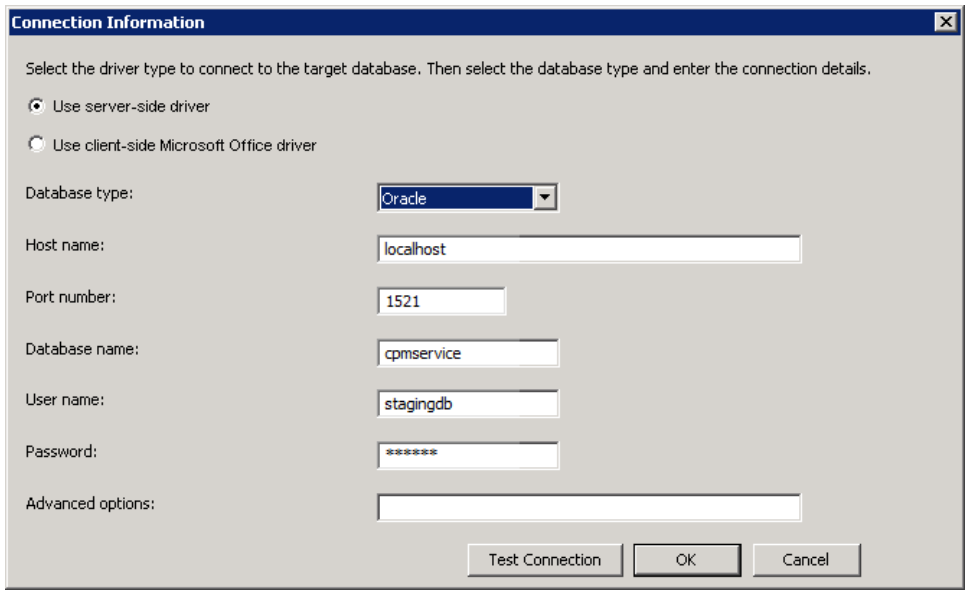
separate multiple options with a semicolon. For example:

```
driver_trace=all;driver_tracefile='c:\temp\trace_lower.log';traceflags=7;
```

Note: Use logging options only in case of failures where you want to provide internal driver logs to technical support. Otherwise, logging operations slow your operation.

Server-Side Driver with Oracle

The following picture shows an example of connecting to an Oracle database.



Following are some considerations to keep in mind when connecting to an Oracle database:

- **Host name** is the MachineName where the target database is installed from where you want to import or export.
- **Port** is the TCP port where the target database instance is running.

  Note: The Oracle client should be installed on the machine on which SAS IOM Server is running for SAS Cost and Profitability Management.
- You can find the values of **Host name** and **Service name** in the file **`tnsnames.ora`**.

```
ABMDSN2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = abmblade1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = abmdsn)
    )
  )
```

- Open the **`sqlnet.ora`** file and verify that it has EZCONNECT specified under NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT).

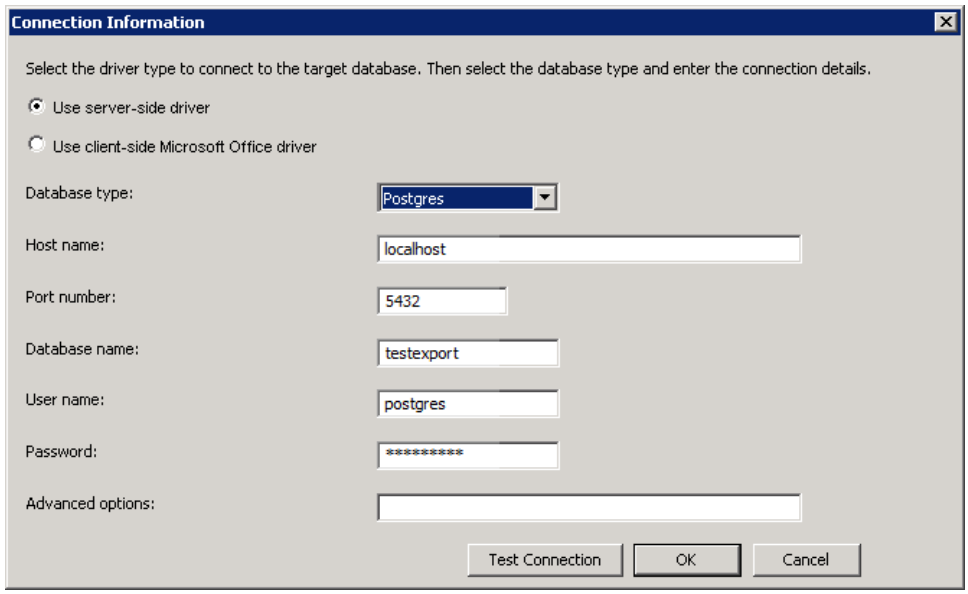
  This file is at the same location as **`tnsnames.ora`**. For example:

  `C:\Oracle\product\11.1.0\client_1\network\admin\sqlnet.ora`

  .

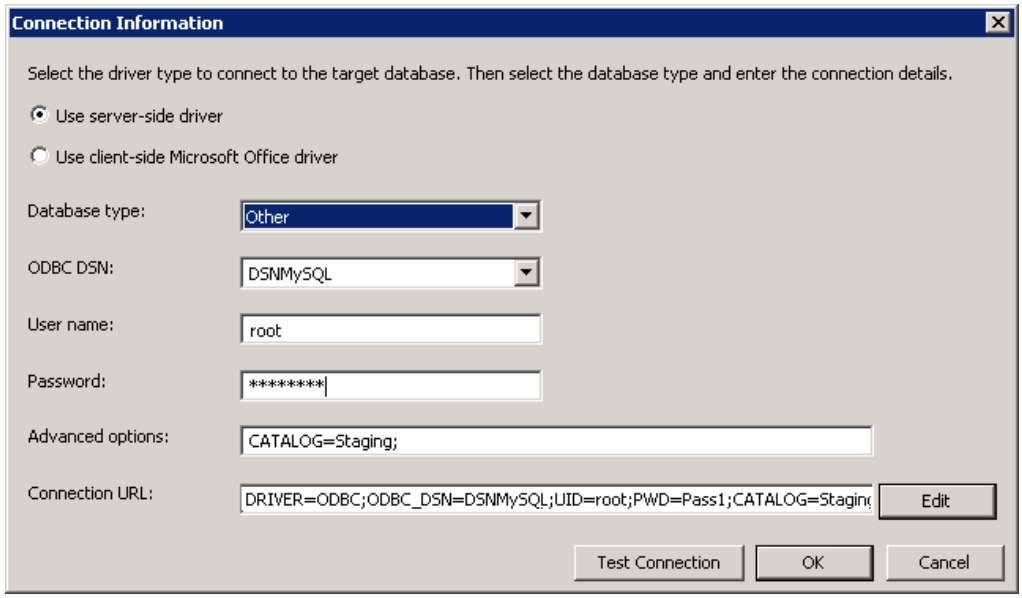
Server-Side Driver with PostgreSQL

The following picture shows an example of connecting to a PostgreSQL database



Server-Side Driver with Other Database

Use the **Database type** of `Other` to connect to a database not already mentioned, such as MySQL.



OBDC DSN

Select from the drop-down list. The list includes all the System ODBC DSN that users have created on the SAS IOM server.

Note:

- It is recommended that you create a System ODBC DSN on the SAS IOM Server machine and use it in the connection details.

- A DSN for Postgres should be created with the Unicode version of the Postgres driver (PostgreSQL Unicode(x64)).

User name
Database user

Password
Database user's password

Advanced options
By default, the catalog name is *Staging*. You can specify any arbitrary name for CATALOG= as it is a logical name that is used internally by the system. Most databases require a catalog name to be specified in the connection URL, but not all. For example, SQL Server does not require a catalog name. If you are attempting to connect to a database that does not require a catalog name, then you must remove it from the **Advanced options** field or the connection will fail. Click **Edit** to modify the connection URL.

Connection URL
When you click **Test Connection**, the system takes the information that you enter and constructs a connection URL that it displays in the **Connection URL** field. Click **Edit** to modify the connection URL.

Following is a sample connection URL for a connection based on ODBC DSN:

```
DRIVER=ODBC;ODBC_DSN=DSNMySQL;UID=root;PWD=Pass99;CATALOG=Staging;
```

If you want to use an ODBC driver to connect to the target database, then the system creates a connection URL (which you can modify) for you. If you want to use a driver other than ODBC, then click **Edit** to enter the URL yourself.

Chapter 74

Import Model Data from a Database

General steps

The following general steps describe how to import data:

1. Verify that you have permission.
2. Import the data. You can choose to import the entire data set at once, or you can import the data in groups and populate the model in the following general steps:

   Import 1
   The first import step defines the existence of the model and includes the Dimension, DimensionOrder, Period, and Scenario tables. This step is equivalent to the finishing point of the New Model Wizard when you interactively build a model.

Import 2
The second import step defines the model's content (resources, activities, and cost objects) and includes the Account, DimensionLevel, and DimensionMember tables. This step is equivalent to the finishing point of the New Account Wizard when you interactively create accounts. The Account table includes the revenue and sold quantities.

Import 3
The third import step loads costs into the model and includes the EnteredCostElement and ExternalUnit tables. This step is equivalent to the point where you create cost elements in the New Account Wizard when you interactively build a model.

Import 4
The fourth import step flows costs through the model using assignments with quantities for flow calculation. This step includes the Assignment and Driver tables and is equivalent to the point where you interactively create assignments and specify driver quantities in the New Account Wizard.

Import 5
The fifth import step creates attributes for analysis. This step includes the DimensionAttributeAssociation, ValueAttribute, and ValueAttributeAssociation tables. This step is equivalent to creating attributes and attaching attributes to accounts when you interactively build a model.

Import 6
(optional): The sixth import step includes the AssignmentNonUnique, CurrencyRate, PeriodLevel, and ScenarioLevel tables.

3. Calculate costs.
4. Generate cubes.

The following table summarizes which tables to group in each step and in what order to import tables:

Table name	Requires dimension signature?	Requires reference ?	Has multiple keys?	Import step order
Dimension			No	Import 1
DimensionOrder			No	Import 1
Period			No	Import 1
Scenario			No	Import 1
Account*	Yes	Yes	Yes	Import 2
DimensionLevel			No	Import 2
DimensionMember			No	Import 2
EnteredCostElement		Yes	Yes	Import 3

Table name	Requires dimension signature?	Requires reference ?	Has multiple keys?	Import step order
ExternalUnit	Yes	Yes	Yes	Import 3
Assignment		Yes	Yes	Import 4
Driver			No	Import 4
DimensionAttributeAssociation	Yes	Yes	Yes	Import 5
ValueAttribute			No	Import 5
ValueAttributeAssociation	Yes	Yes	Yes	Import 5
AssignmentNonUnique		Yes	Yes	Optional
CurrencyRate			No	Optional
PeriodLevel			No	Optional
ScenarioLevel			No	Optional

Note: The Account table must contain both a dimension signature and a reference number, while the other tables listed with "Yes" in both columns must have either a dimension signature or a reference number.

Database Considerations

Overview

To import from a database, you can use the wizard, or you can write a program.

For information on connecting to a database, see Chapter 73, “Connecting to a Database,” on page 617 .

Make sure to observe the restrictions on table names in a database for all staging tables.

The data schema

The database to be imported must match the data schema. For information on the data schema, see Chapter 14, “Staging-Table Schemas,” in *SAS Cost and Profitability Management: Data Administration Guide*.

All tables and fields in the data schema (except tables that are designated for export only) must exist in the database. When the database is imported, SAS Cost and Profitability Management attempts to convert all values to a reasonable format type. For example, if imported dates are in the Microsoft SQL Server varchar format, the dates are

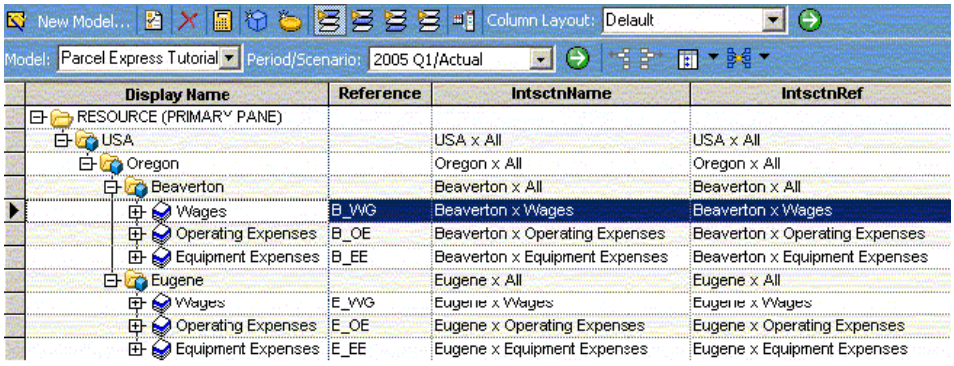
converted to binary dates that are compatible with the SAS Cost and Profitability Management database. SAS Cost and Profitability Management attempts to convert all numeric values.

Understanding keys

The SAS Cost and Profitability Management model is based on a dual-key concept. To define any account in the model, you can describe it based on its dimension signature key or its reference key.

Dimension signature key

The dimension signature consists of a dimension reference and a dimension member reference for each dimension used to define an account. In the following example, the Resource module has been defined based on two dimensions (Region and General Ledger Account). The individual accounts are defined as intersections of these two dimensions. So, the Wages account (highlighted) consists of an intersection of the Region=Beaverton and the General Ledger Account=Wages. This account definition can be displayed in the grid with the intersection name (the column IntsctnName) or the intersection reference (the column IntsctnRef) properties.



Display Name	Reference	IntsctnName	IntsctnRef
RESOURCE (PRIMARY PANE)			
USA		USA x All	USA x All
Oregon		Oregon x All	Oregon x All
Beaverton		Beaverton x All	Beaverton x All
Wages	B_WG	Beaverton x Wages	Beaverton x Wages
Operating Expenses	B_OE	Beaverton x Operating Expenses	Beaverton x Operating Expenses
Equipment Expenses	B_EE	Beaverton x Equipment Expenses	Beaverton x Equipment Expenses
Eugene		Eugene x All	Eugene x All
Wages	E_WG	Eugene x Wages	Eugene x Wages
Operating Expenses	E_OE	Eugene x Operating Expenses	Eugene x Operating Expenses
Equipment Expenses	E_EE	Eugene x Equipment Expenses	Eugene x Equipment Expenses

The dimension signature for this account is based on the dimensions used and the dimension members used, so the dimension signature for this account requires multiple values as shown here:

Property	Example
DimRef1	Region
DimMemberRef1	Beaverton
DimRef2	General Ledger Account
DimMemberRef2	Wages

Reference key

In the previous example, the reference for the account is B_WG, and this single value uniquely identifies the account. See "Guidelines for using dimension signatures and references" on page 632.

Note: Importing with reference numbers is faster than with dimension signatures. However, when importing a new model, you must use dimension signatures.

Creating sample database tables

To understand how to create database tables, you should interactively import a sample model. The sample model demonstrates the dimensions and dimension order for each module, the default period/scenario association, and anticipated periods and scenarios. Use the sample model as a source for the model export to a database for general use.

Create the sample database tables by performing the following general steps:

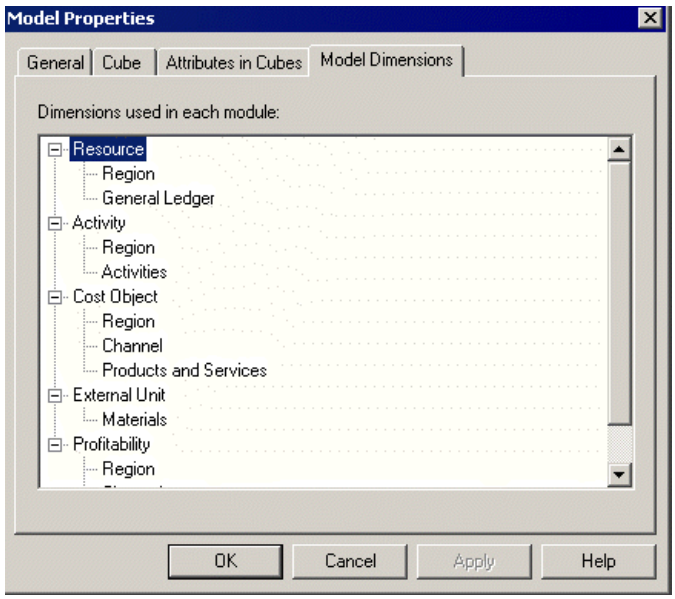
1. Import a sample model in XML format. The sample models are located in the following directory:

   ```
   C:\Program Files\<Client_Installation_Location>
   \SAS\Cost and Profitability Management Solution\Client\Samples\Models\Native
   ```

2. Create a target database to export your sample database.
3. Archive the model.

   This creates a set of database tables.

4. Review the contents of the model.



If the model that you want to import has the same number of dimensions in each module as the sample model shown, you can use the sample model directly. The External Unit module consists of one dimension; the Resource module consists of two dimensions; the Activity module consists of two dimensions; and the Cost Object module consists of three dimensions. After you export the sample database, modify the contents of the tables to reflect your own data in the appropriate dimension signature columns.

If the model that you want to import has a different number of dimensions in each module as the sample model, then the sample can still provide a good starting point for creating a staging table template. But, you must customize all of the tables that require the dimension signature (the Account, Assignment, EnteredCostElement, ExternalUnit, DimensionAttributeAssociation, and ValueAttributeAssocation tables) to include all of the dimensions that you used in the model that you want to import.

Number of dimensions in the imported model	Difference in the number of dimensions in the sample model
External Unit module: two dimensions	Add one dimension
Resource module: one dimension	Remove one dimension
Activity module: three dimensions	Add one dimension
Cost Object module: five dimensions	Add two dimensions

Exporting to a database creates all of the required tables with most of the required columns. The required columns in some of the tables depend on the number of dimensions in a model. Because each dimension signature consists of a pair of columns, any table with dimension signature columns will add columns as the number of dimensions in the model increases. Although you can manually create the tables, exporting to a database reduces the risk of omitting required tables, omitting required columns within tables, and creating typographical errors. And, it gives you a viable sample to follow in developing your own staging table content and when building a SAS Cost and Profitability Management model.

Database table relationships, dependencies, and data values

There are no database-enforced relationships between any of the SAS Cost and Profitability Management tables, and there are no dependencies between tables. You can specify any values in the tables that you need. However, while the data is being imported, SAS Cost and Profitability Management checks the validity of the values and rejects any invalid records. You will receive messages that indicate any errors.

If you import all the tables at one time, the wizard will import the tables in the correct order, so errors are minimized. However, if you import individual tables, then you must ensure that the tables are imported in the correct order, as noted in the Importing data: General steps Help section.

Following are a few examples of common errors made while importing tables:

- A record in the DimensionMember table does not correspond to a record in the Dimension table. You must import the Dimension table before you import the DimensionMember table.
- A record in the Account table does not correspond to a record in the DimensionMember table. You must import the DimensionMember table before you import the Account table.
- A record in the Assignment table (where the source account or the destination account is) is not in the Account table. You must import the Account table before you import the Assignment table. If you select the option to create a new Account through the Assignment's dimension signature, you do not need to import the Account table first.

Guidelines for using dimension signatures and references

When importing data for assignments, cost elements, dimension attribute associations, and value attribute associations, you can choose to import with either a dimension signature or reference key. As you define the extraction and transformation processes to

create the staging tables for SAS Cost and Profitability Management, you should be aware of the advantages and disadvantages of using one key over another.

If you are importing model data using references rather than dimension signatures, you must define both the reference and the dimension signature in the Account table. The Account table can be used as a mapping index to match the imported data in staging tables (Assignment, EnteredCostElement, DimensionAttributeAssociation, and the ValueAttributeAssociation tables) to their respective dimension signature as defined in the model. The advantage of using references rather than dimension signatures is that the import tables can have significantly fewer columns. For example, in a two dimension model, the reference would be a single column to import and the dimension signature would be four columns to import. The total impact of this change to the Assignment table would be six fewer columns required.

If you are importing model data using dimension signatures, you must include the dimension reference and dimension member reference for each dimension, which can be tedious to maintain. For example, in a two dimension model, the required keys include four columns for the two dimensions. However, there is one advantage of importing data using dimension signatures: You can automatically create new accounts that appear in the transaction tables (Assignment, EnteredCostElement, DimensionAttributeAssociation, and the ValueAttributeAssociation tables). The dimension signature method provides a distinct advantage over the reference key method because the reference key method skips over any new accounts found in the transaction tables and fails to import them.

Period/scenario associations

If you use the wizard to import data, you can import multiple period/scenario associations. If you write a program to import data, you can import more than one period/scenario association at a time. Period/scenario associations are imported if a model element (such as an account or a cost element) contains data for the period/scenario association.

Period/scenario associations are ignored unless there is a model element (such as an account or a cost element) that contains data for the period/scenario association.

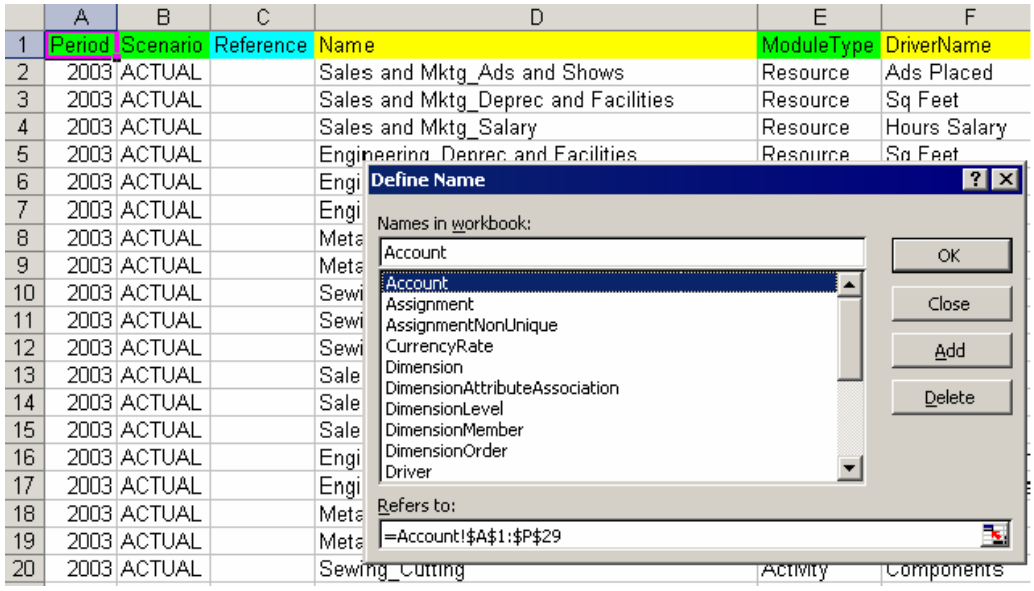
Data from multiple databases

You can import data from multiple databases by first creating a database view. The view specifies which data you want to import from multiple tables in multiple databases. Then, you can use the wizard.

Alternatively, you can use the SAS Cost and Profitability Management Web Services Integration API to write a program that uses an XML import configuration to specify each database in a separate StagingArea element.

Data from a Microsoft Excel workbook

If you want to import data from a Microsoft Excel workbook, you must create named ranges for the different sets of data. To create a named range in Microsoft Excel, highlight the data and select Insert > Name > Define. Multiple named ranges must be defined for each required stage table inside a single Microsoft Excel spreadsheet, as shown in the following example:



Periodic Import

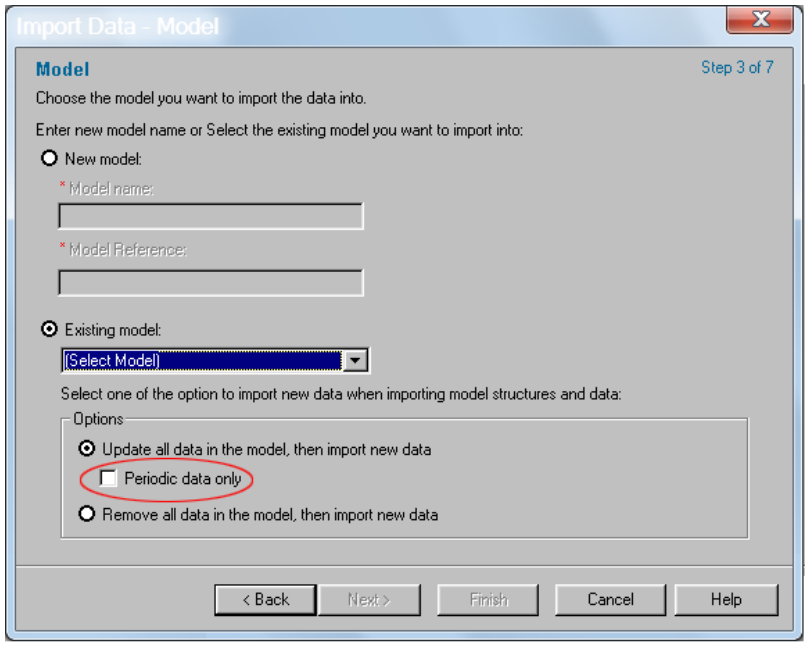
When you generate a cube, for every period (period and scenario association) that is to be included in an existing cube, if:

- the cube already contains that period, and
- the period has not been modified since the cube was last generated

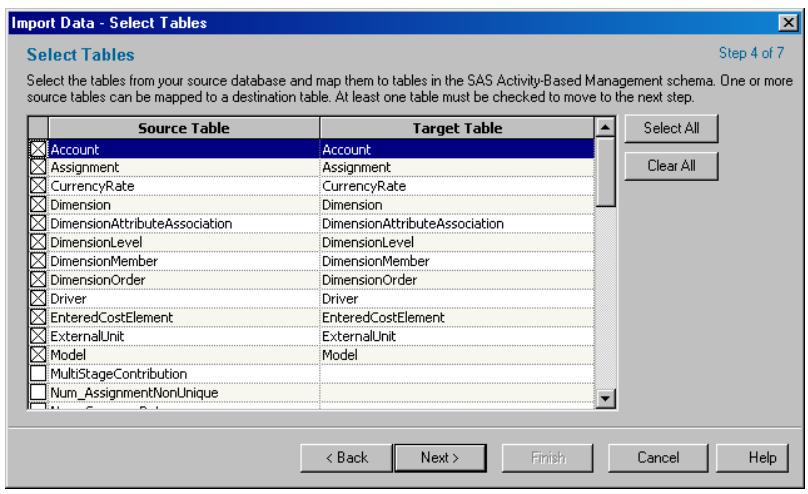
then the period is not regenerated.

This means that cube generation is faster because periods that have already been generated are not regenerated.

In order to support incremental cube generation, SAS Cost and Profitability Management provides a **Periodic data only** option in the Import Wizard that allows you to import only the periods that have changed in a model—for example, the new periods.



If you select **Periodic data only**, then only those staging tables that contain periodic data are displayed in the Import Wizard for you to select for importing.



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 model data which is independent of any period/scenario association. It is data that is common to all period/scenario associations.

The following staging tables contain **periodic data**:

- Account
- Assignment
- CurrencyRate
- ExternalUnit
- EnteredCostElement
- PerformanceMeasure
- ValueAttributeAssociation

- ValueAttributePeridicDef
- DimensionalAttributeAssociation

The following staging tables contain **structural data**:

- Dimension
- DimensionMember
- DimensionLevel
- DimensionOrder
- Driver
- Model
- ValueAttributes
- Period
- PeriodLevel
- Scenario
- Scenariolevel

When you generate a cube for a model that has been generated before, SAS Cost and Profitability Management determines whether the entire cube must be regenerated or whether only the new or modified periods need to be generated. You do not have to specify, when you generate a cube, whether you want incremental generation or not. SAS Cost and Profitability Management makes the determination for you.

Note: Cubes in SAS OLAP that are generated using either the NO_NWAY option or the NONUPDATEABLE option are not eligible for incremental generation.

Importing data with your own program

To write a program for importing data in batch, you can use Easy API. For information, see Chapter 21, "Easy API," in *SAS Cost and Profitability Management: Data Administration Guide*.

Using the Import Wizard to Import from a Database

Before Importing

Before attempting to import data from a database, see Chapter 73, "Connecting to a Database," on page 617, which provides information about connecting to the most common databases.

Note: You can import a model without first opening a model.

1. Create a database or a database view that matches the data schema.

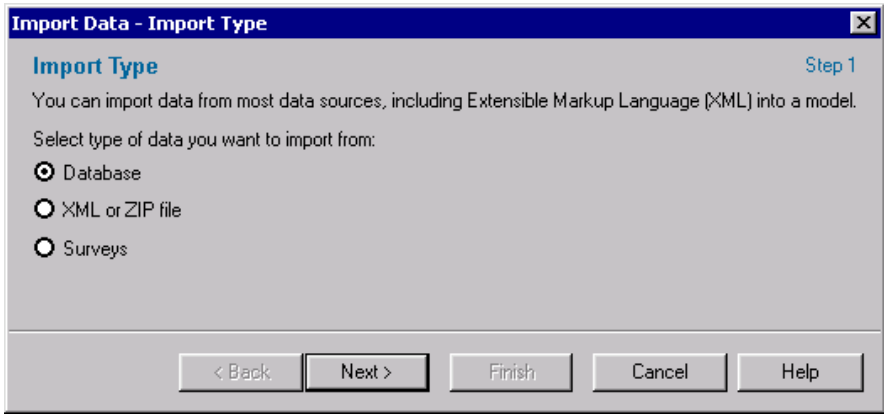
   For information about the data schema, see .

2. In SAS Cost and Profitability Management, verify or create any required period/scenario associations.
3. Select **File** ⇨ **Import** ⇨ **Model Data**.

   The Import Data Wizard opens.

Step 1 – Select Database

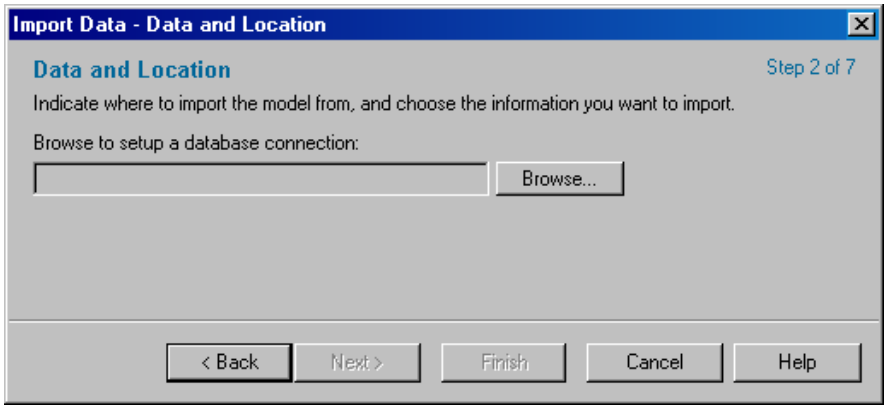
1. Select the **Database** option.



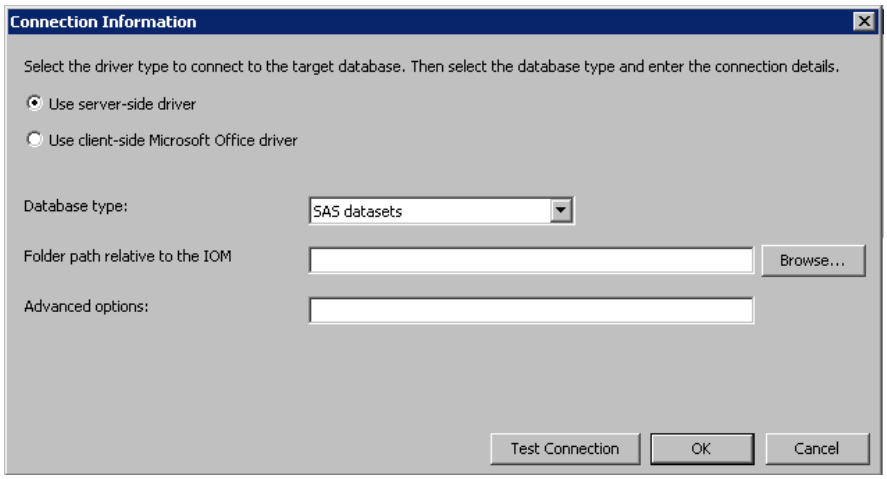
2. Click **Next**.

Step 2 – Connect to the Database

1. Click the Browse... button on the Wizard page.



   The Connection Information window opens.

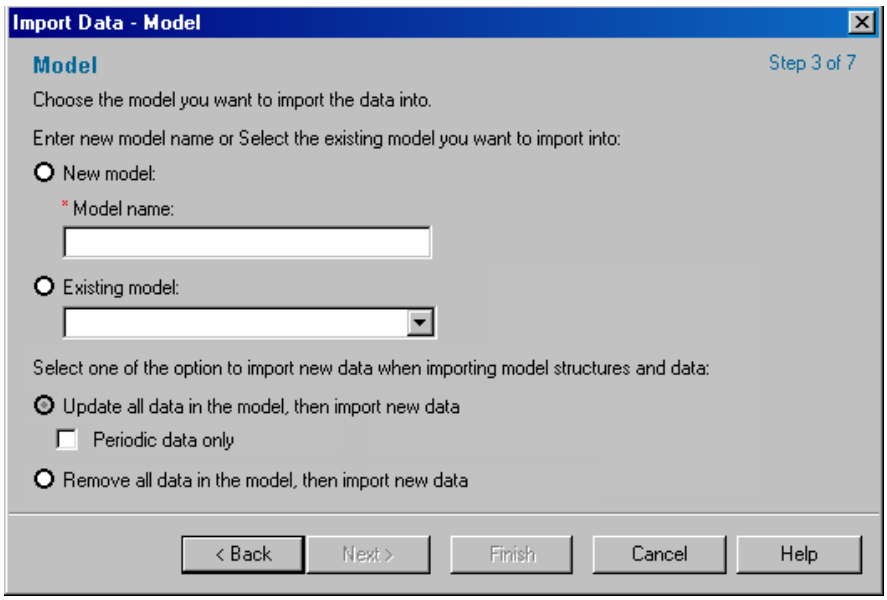


For information on using this window, see Chapter 73, "Connecting to a Database," on page 617.

2. Click **Next**.

Step 3 – Select Model

1. If you want to import the database into a new model, do the following:



   a. Select the **New model** option.

   b. Type the **Model name**.

2. If you want to import the database into an existing model, do the following:

   a. Select the **Existing model** option.

      Note: If you attempt to replace a model that is currently open by importing a model, then it is possible that an error will occur when the system attempts to update the display of the model. To avoid any possible problem it is safer to close a model before attempting to replace it by importing a model.

   b. From the drop-down list, select a model.

c. To append the imported data to the existing model data, select the **Update all data in the model, then import new data** option.

d. Select **Periodic data only** to import only the staging tables that contain periodic data. See "Periodic Import" on page 634.

e. To remove all existing data, select the **Remove all data in the model, then import new data** option.

Note: Ensure that the existing model does not contain any published period/scenario associations. The published period/scenario association might cause an error when you try to replace the existing model by importing another.

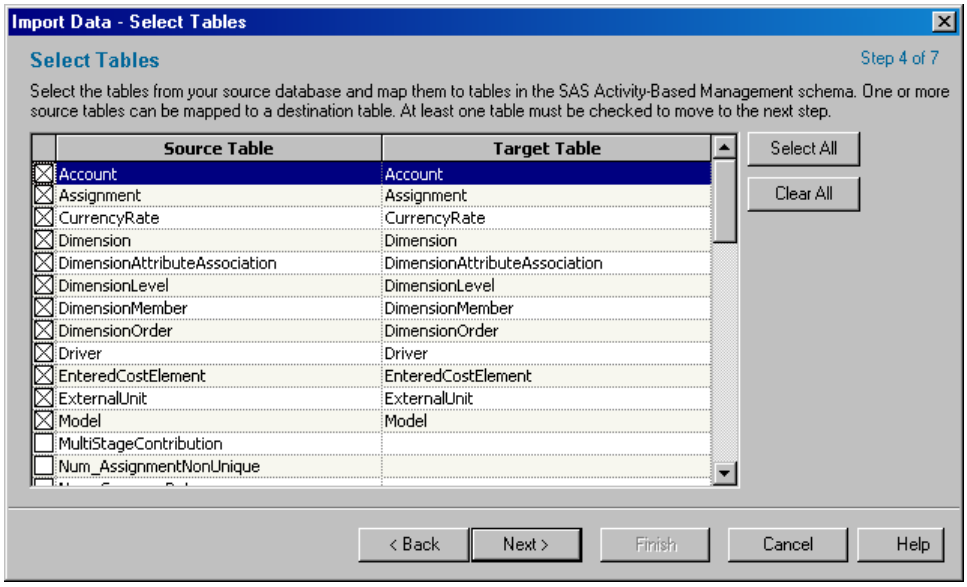
3. Click **Next**.

Step 4 – Map Source Tables to Target Tables

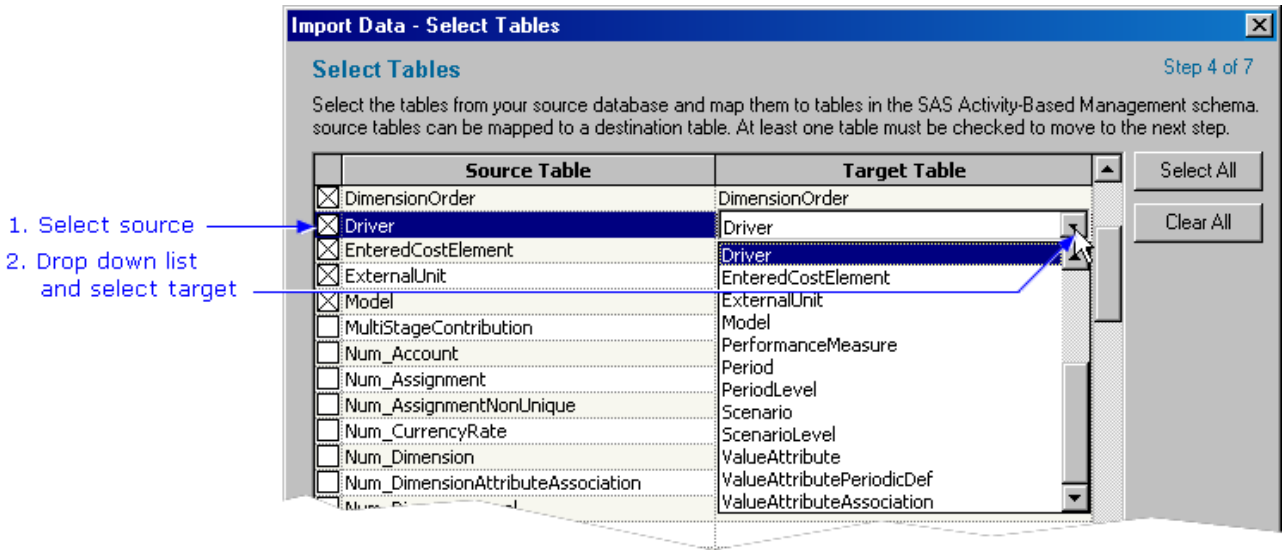
In this step, you specify which tables to import. And, for each table that you import, you specify to what SAS Cost and Profitability Management table it maps.

1. To select a table to import, select the check box to the left of the table name in the **Source Table** column.

   Note: If you selected **Periodic data only** in the previous step, then only tables that contain periodic information are displayed for selection.

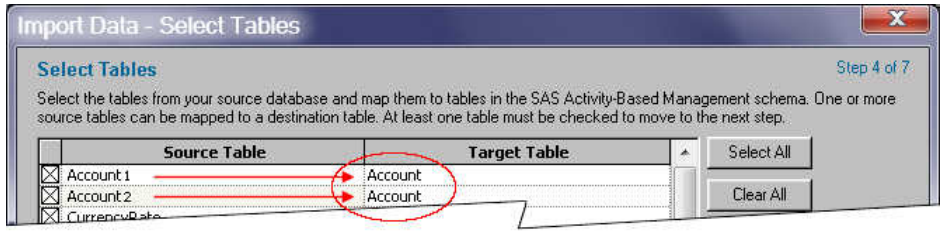


2. To map a **Source Table**, click in the **Target Table** column to the right of the source table, and select a SAS Cost and ProfitabilityManagement table from the drop-down list.



Note: If you have named the source tables with the same names that are used in SAS Cost and Profitability Management, the wizard automatically creates the mappings and selects the tables to be imported. If you have not used the same names, then you must specify which source table maps to which SAS Cost and Profitability Management table.

Note: You can map multiple source tables to the same target table. The following picture shows the source tables Account1 and Account2 both mapped to Account.



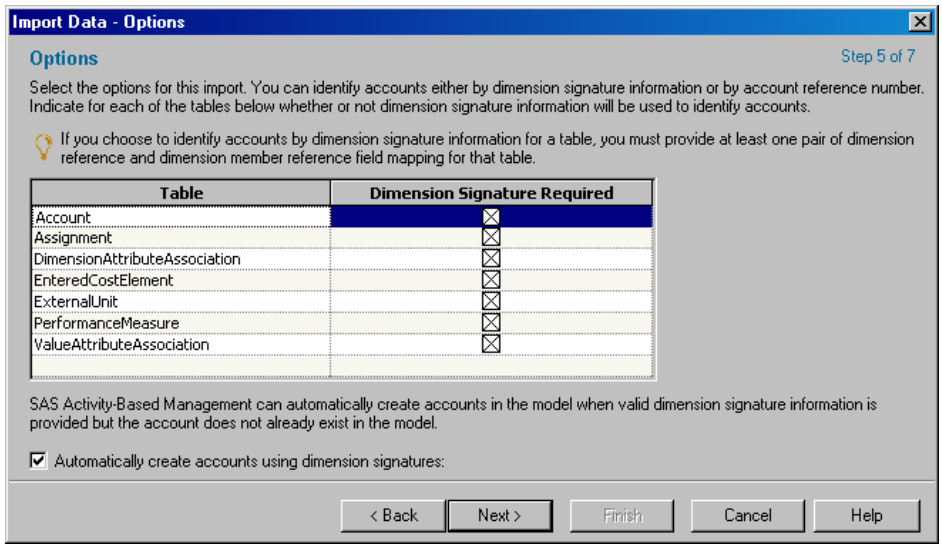
3. Repeat steps 1 and 2 to map every table that you want to import.

4. Click **Next**.

Step 5 – Specify Type of Account Keys

In this step, you specify how accounts are identified in source tables that contain accounts. You can identify each account by a dimension signature or by reference number. You can choose to have the wizard create accounts for all valid dimension signatures, even if accounts have not been created for those intersections.

1. To identify accounts by the dimension signature, select the check box in the **Dimension Signature Required** column to the right of each source **Table**.



If you clear the check box, accounts will be identified by their references.

2. To have the wizard automatically create accounts for all valid dimension signatures, select the **Automatically create accounts using dimension signatures** option.

   Note: The import process supports automatically creating accounts from the following tables only: Account, Assignment, EnteredCostElement, ValueAttributeAssociation, and Dimension AttributeAssociation.

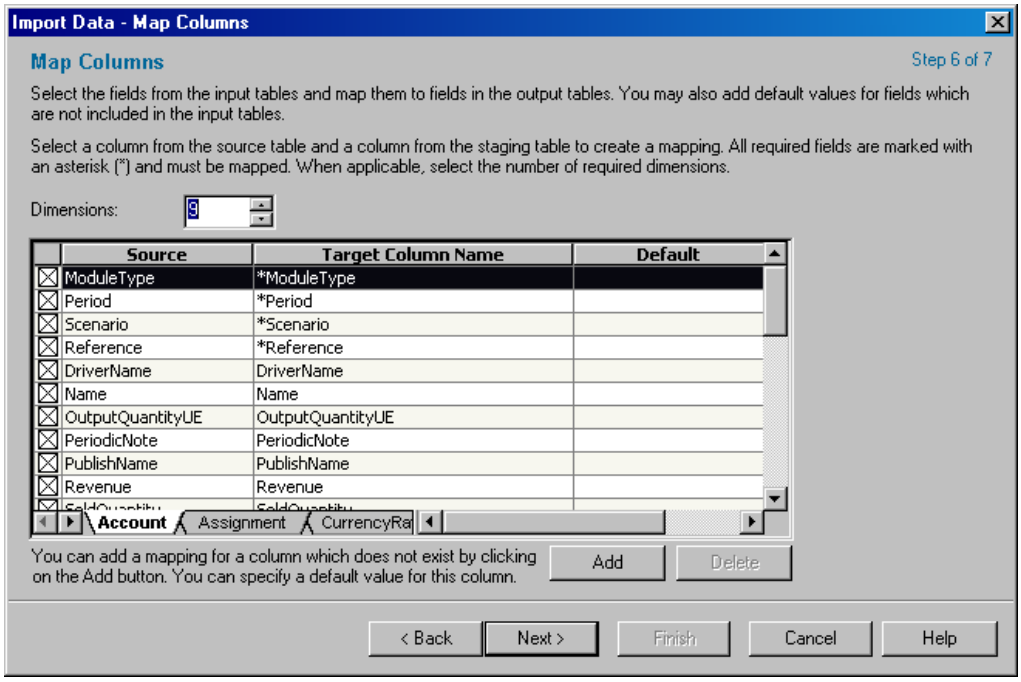
3. Click **Next**.

Step 6 – Map Input Columns to Target Columns

In this step, you specify which table columns to import. And, for each table column that you import, you specify to which table column in SAS Cost and Profitability Management it maps.

1. To select a table column to import, select the check box to the left of the table name in the **Source** column.

   You can select as many table columns as needed, but you must import those columns that are required. Required columns are denoted by an asterisk (*).



Note: If you have named the source table columns with the same column names that are used in SAS Cost and Profitability Management, the wizard automatically creates the mappings and selects the columns to be imported. If you have not used the same names, then you must specify which source table column maps to which SAS Cost and Profitability Management table column.

2. If you chose in the previous step to identify accounts by their dimension signatures, specify the number of **Dimensions** that are contained in the source table.

   The **Dimensions** option appears only when you choose to identify accounts by their dimension signatures. You must specify the number of dimensions so that the interface displays the correct column names that you must map. For each dimension in the source table, there must be two columns that can be mapped to the SAS Cost and Profitability Management table columns; these two columns are named DimRef<number> and DimMemberRef<number>. For example, a source table that contains two dimensions must contain columns that must be mapped to DimRef1, DimMemberRef1, DimRef2, and DimMemberRef2.

3. To map a **Source** table column, click in the **Target Column Name** column to the right of the **Source** table column, and select a SAS Cost and Profitability Management table column from the drop-down list.

   The drop-down list contains the values DimRef<number> and DimMemberRef<number> only if you chose to identify accounts by their dimension signatures, which is the only situation in which these values are needed.

4. It is not necessary that the source table contain every column that is required in the target table. If a source table does not contain a column that you want to create in the model, do the following:

   a. Select the target table from among the tabs at the bottom of the dialog.

   b. Click **Add**.

      A new row appears at the bottom of the list of source columns.

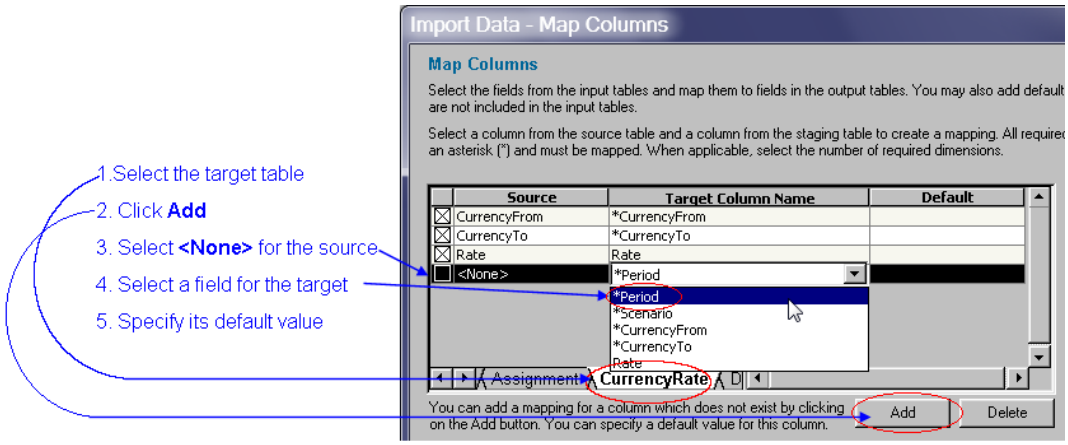
   c. Make sure that **<None>** is selected for the **Source** column.

      <None> signifies that the column does not appear in the source table.

d. Select the column to be created in the target table from the **Target Column Name** drop-down list.

e. Type a **Default** value.

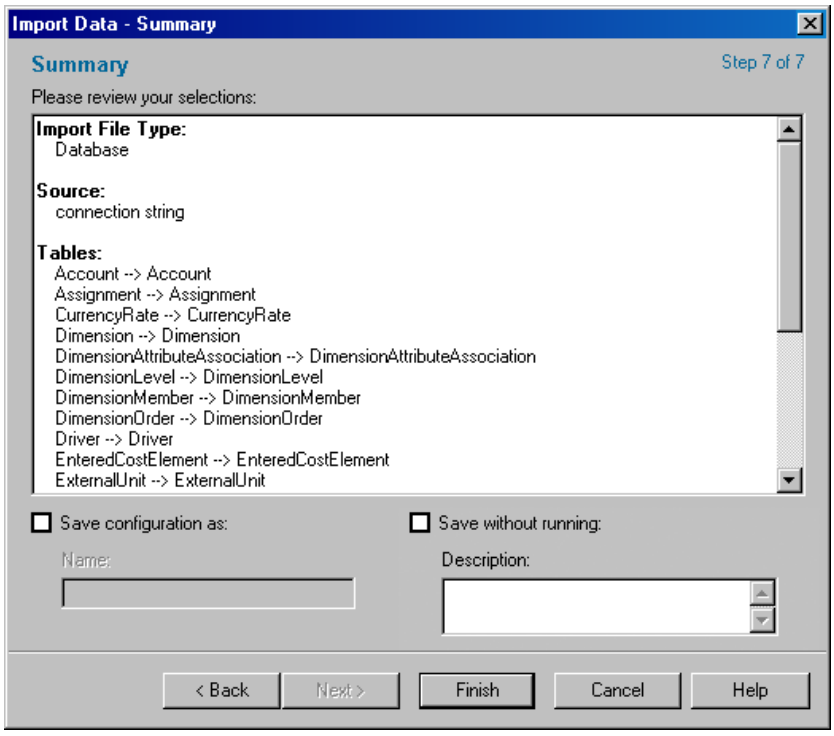
The following picture summarizes the process for adding to the target table a column that does not exist in the source table.



5. After adding all the desired missing columns to the target tables, click **Next**.

Step 7 – Review Your Choices

1. Review the import summary.



2. If you need to change any information, click **Back** until you reach the step that you need to change in the wizard.

All of the information that you have specified is saved. Click **Next** to advance through the wizard.

3. To save the import configuration so that the import can be easily run again, do the following:
   a. Select the **Save configuration as** option.
   b. Type the **Name**.
   c. Type the **Description**.
4. Select **Save without running** to save the import configuration with performing the import.
5. Click **Finish**.

Chapter 75

Import a Model from an XML File

Using the Import Wizard

To build a model or to add data to a model in SAS Cost and Profitability Management, you can either add data interactively with the user interface or you can import data into a model.

TIP Because manually building a model is time-consuming and error-prone and leaves little time for analysis, importing data is the recommended method.

You can import data from the following sources:

Database
See "Using the Import Wizard to Import from a Database" on page 636.

XML file
The XML file must be an XML file that was exported from SAS Cost and Profitability Management. You should not import a manually created XML file, because it is challenging to dynamically create an XML file with the correct format for directly importing model data.

See "Importing from an XML File" on page 646.

To import a model from an XML file, select **File** ⇨ **Import** ⇨ **Model Data**, and then select **XML or ZIP file**.

Oros model
Please contact SAS technical support if you want to import an Oros model into SAS Cost and Profitability Management.

Surveys
See "Importing Survey Data" on page 674.

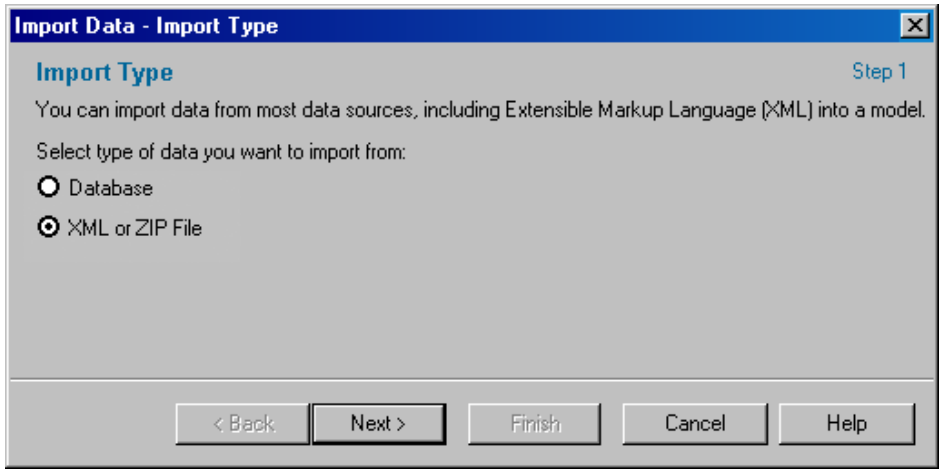
Importing from an XML File

When you import model data from an XML file, a new model is created. You can choose to give the model a new name or you can reuse the name of an existing model. You cannot import model data from an XML file to incrementally update an existing model or to combine several models into a single model. Whatever data is contained by an existing model is removed and replaced by the model data in the imported XML file.

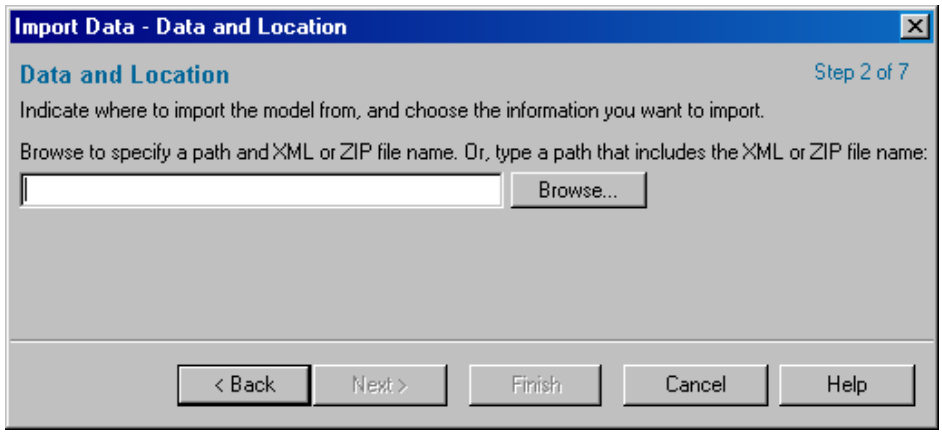
Note: You can perform this task without first opening a model.

1. Select **File** ⇨ **Import** ⇨ **Model Data**.

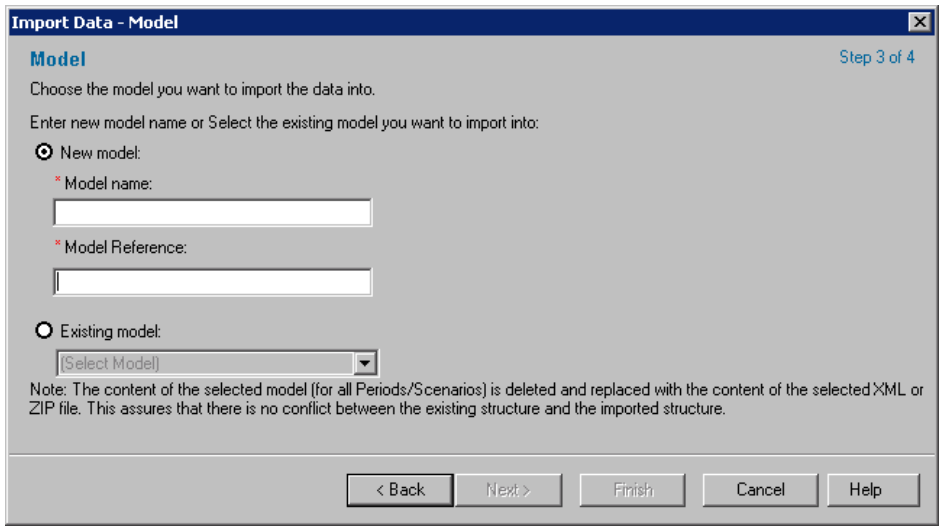
   The Import Data Wizard appears.



2. Select the **XML File** option.
3. Click **Next**.



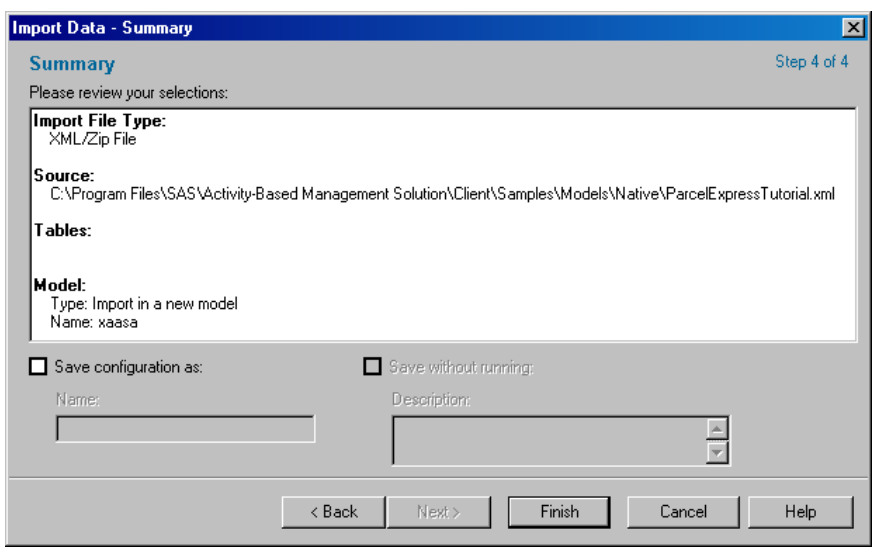
4. Type the absolute pathname to the XML file. Or, click Browse...
5. Click **Next**.



6. If you want to import the XML file into a new model, do the following:
   a. Select the **New model** option.
   b. Type the **Model name**.
   c. Type the **Model Reference**. The Model Reference is used in public views..
7. If you want to import the XML file into an existing model, do the following:
   a. Select the **Existing model** option.
   b. From the drop-down list, select a model.

      Note: If you attempt to replace a model that is currently open by importing a model, then it is possible that an error will occur when the system attempts to update the display of the model. To avoid any possible problem it is safer to close a model before attempting to replace it by importing a model.

   Note: Ensure that the existing model does not contain any published period/scenario associations. The published period/scenario association might cause an error when you try to replace the existing model by importing another.
8. Click **Next**.



9. Review the import summary.

10. If you need to change any information, click **Back** until you reach the step that you need to change in the wizard.

    All of the information that you have specified is saved. Click **Next** to advance through the wizard.

11. To save the import configuration so that the import can be easily run again, do the following:

    a. Select the **Save configuration as** option.

    b. Type the **Name**.

    c. Type the **Description**.

12. Select **Save without running** to save the import configuration with performing the import.

13. Click **Finish**.

Chapter 76

Export Model Data

Choosing the Export Format

You can export model data to a database or to an XML file. If you want to export only a portion of the model data, you must export to a database. If you export to an XML file, all model data is exported. If you want to export all of the model data, you can export to a database or to an XML file.

The following table lists some of the reasons why you might export model data. For each reason, the table shows whether you would export to a database or to an XML file:

Reason for exporting	Export destination
To export model data and to manipulate the data, and then to import the data back into the model or into another model	Database
To archive a model	Database XML file (requires less disk space)
To export some of the items in the model, but to exclude other items	Database

Note: When a table is exported to a database, SAS Cost and Profitability Management attempts to match the destination table's column names with the source table's column names as closely as possible, so that they are compatible with SAS. A name change is necessary when a database has certain naming limitations (such as a limited number of characters per field); those name changes are noted in the Field Name column of each table.

For information on exporting to a database, see "Choosing the Export Format" on page 649. Also, see Chapter 73, "Connecting to a Database," on page 617.

Note: If a network user from a different domain (than that of the SAS Cost and Profitability Management client machine) logs in to the client machine to export to XML, then that user must be granted WRITE access to the [ClientInstallpath]\bin folder for the operation to succeed.

When exporting a table to a database, observe the following restrictions on the table name:

SQL Server:

- Table name must begin with a letter.
- Table name should not start with special characters like $, @, #, %, |, !
- Special characters like %, -, |, ! are not allowed in the table name.
- Table name must be less than or equal to 128 characters.

Oracle:

- Table name must begin with a letter.
- Table name should not start with special characters like $, _, #, %,-,|, !
- Special character like @, %, - are not allowed in the table name.
- Table name must be less than or equal to 30 characters.

MySQL:

- Special characters -,%, @, |, ! are not allowed in the table name.
- Table name must be less than or equal to 64 characters.

Microsoft Access:

- Table name must be less than or equal to 64 characters.
- Special characters like $, %, #, !, - are not allowed in the table name.

Microsoft Excel:

- Table name must be less than or equal to 32 characters.
- Special characters like $, %, #, !, - are not allowed in the table name.

See Also

"Export a Model to an XML File" on page 666

Archiving a Model

Overview

When you archive a model, important model data is preserved so that the model can be restored to a saved state. Although not all model data is preserved, user-entered data and unique data are saved. Model data that is not saved is regenerated by SAS Cost and Profitability Management when the model is restored and calculated.

You might want to archive a model for the following reasons:

- to create a backup
- to save a version before making major changes
- to restore a model after upgrading a SAS Cost and Profitability Management server
- to transport a model between SAS Cost and Profitability Management servers

Determining the Tables to Export

When you export model data for business analysis, you must determine which tables to export with the appropriate calculated values. You can export the following SAS Cost and Profitability Management standard staging tables with calculated values.

Account table

Exporting the Account table with the calculated values enables you to perform additional analysis on any account, which is useful for static, calculated values analysis, but is not useful for cost-flow analysis. The types of fields in an exported Account table include:

- Definitional

  Keys: dimension signature and reference

  Model, module type, period, scenario, driver names, model name, unit of measure, periodic note
- Entered values: OutputQuantityUE, Revenue, SoldQty, TDQUE
- Calculated values

  Cost values: AllocatedCost, AssignedCost, AssignedIdleCost, AssignedNonReciprocalCost, AssignedReciprocalCost, DrivableCost, DrivenCost, DriverRate, EnteredCost, IdleCost, ReceivedAllocatedCost, ReceivedBocCost, ReceivedCost, ReceivedDrivenCost, ReceivedNonReciprocalCost, ReceivedReciprocalCost, UnassignedCost, UsedCost

  Driver data: AssignedIdleQuantity, DrivenQuantity, IdlePercentage, IdleQuantity, OutputQuantity, TDQ, TDQBasic, TDQCalculated, UnassignedQuantity, UsedQuantity

  Profitability analysis (uses both entered values and calculated values): Cost, Profit, Revenue, SoldQuantity, UnitCost, UnitProfit, UnitRevenue
- Attributes (specific to the model design)

  Dimensional attributes - used for grouping

  Numerical attributes - entered values

Calculated attributes

Assignment table

Exporting the Assignment table with the calculated values enables you to trace specific costs as they flow through a model. The Assignment table provides the cost flow and driver-quantity flows between each source account and destination account. It contains the content from the Single-stage Contributions OLAP cube fact table. The types of fields in an exported Assignment table include:

- Definitional

  Keys (for both the source account and destination account): dimension signature and reference

  Destination module type, driver name, model name, period, scenario, source module type

- Driver analysis: DriverQuantityBasic, DriverQuantityCalculated, DriverQuantityFixed, DriverQuantityVariable, DriverWeightFixed, DriverWeightVariable, IdleDriverQuantity, IdleDriverQuantityUE
- Cost flow: allocated cost, cost, idle cost, source cost
- Attributes (specific to the model design)

  Dimensional attributes - used for grouping

  Numerical attributes - entered values

  Calculated attributes

Multi-stage contributions cube fact table (with calculated values)

The types of fields in an exported Multi-stage Contributionscube include:

- Definitional

  Keys for each account: dimension signature and reference

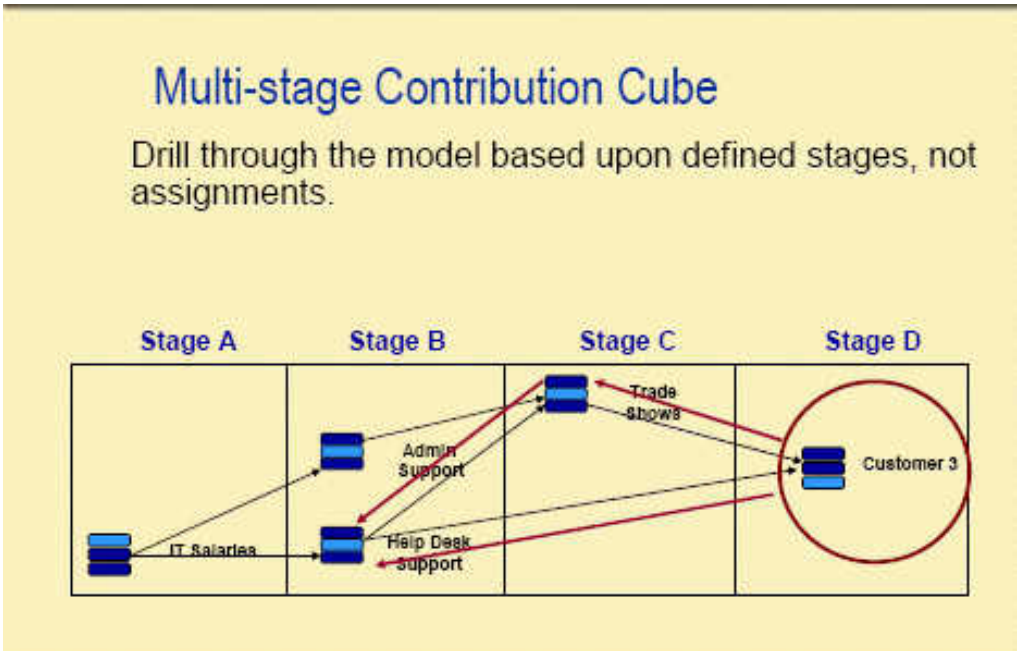
  Activity module type, Cost Object module type, model name, period, Resource module type, scenario

- Entered values: OutputQuantityUE, Revenue, SoldQty, TDQUE
- Calculated values: Cost, OutputQuantity
- Attributes specific to the model design

  Dimensional attributes - used for grouping

  Numerical attributes - entered values

  Calculated attributes



The cube fact table holds all of the calculations for cost flow through the model. The Multi-stage Contributions cube holds cost flow from each step through the model. Using the Multi-stage Contributions cube, an analysis can be created to trace a single, final cost object, through each contributing cost, to the activities back to the original resource.

In the previous example, all stages (A through D) are available for analysis in the Multi-stage Contributions cube. (About fact tables)

Resource Contributions cube

The types of fields in an exported Resource Contributions cube include:

- Definitional

  Keys for each beginning or final account (resources, cost objects): dimension signature and reference

  Destination module type, model name, period, scenario, source module type, source reference
- Entered values: DestinationSoldQuantity
- Calculated values: ContribCost, ContribPcnt, DestinationCost, DestinationOutputQuantity
- Attributes specific to the model design for each beginning or final account (resources, cost objects)

  Dimensional attributes used for grouping

  Numerical attributes entered values

  Calculated attributes

Working with Tables, Dimensions, Properties, and Attribute Values

For tables, dimensions, properties, and attribute values, you can export all items or specific items. You can change the name of each exported item from its default name. For maximum flexibility with tables, you can export the same table to multiple export

tables. For example, you can export the Account table to the tables named ResourceAccounts and ActivityAccounts.

TIP The Export wizard shows you which database fields are required for re-importing model data.

Exporting, Filtering, and Limiting Calculated Results

Following are the general steps that you perform when exporting model data:

- Export only certain period/scenario associations.

  You can export one or more period/scenario associations.

- Export only certain tables.

  You can select the specific SAS Cost and Profitability Management staging table you want to export. You can choose to change the names of the tables that you are exporting, which is useful when you are exporting multiple tables of the same type for distribution across a large audience (actual account and planning account).

- Choose the specific fields to be included in the dimension signature in all of the tables to be exported.

  The required fields for the dimension signature include a dimensional reference and the dimensional member reference. These fields are required if you want to re-import the exported data into SAS Cost and Profitability Management. However, for readability and integration with other systems (data warehouses), you might find it helpful to include the dimension name, the dimension member name, the dimension level, and the dimension level name.

- Export only certain fields in the tables.

  You can select specific fields within each table to be exported. The default (archive) selections of fields do not include any calculated values, so be sure to carefully select the calculated values you need to export for further analysis or to import into another system (data warehouse). For each field you export, you can change the names of the fields, which is particularly useful when creating a SAS Cost and Profitability Management system to import into another system (which might have predefined fields). You can change the numbers in a dimension signature to a more useful notation of the organizational structure.

- Export only certain dimensions in specific tables.

  You can eliminate unnecessary fields in the exported data. This is useful in the Multi-stage Contributions cube. The default behavior is to include all of the possible dimensions in each stage of the export, but in most SAS Cost and Profitability Management models, only a limited number of dimensions actually apply to a given stage in the model. By eliminating unnecessary fields, you can significantly decrease the size of the exported data.

- Export only certain members in a dimension in specific tables.

  You can define a specific point in the dimensional hierarchy to include in the exported data. This filter method is useful for creating specific exports for an organizational structure (specific departments, specific product lines, or specific customer types).

- Export multiple tables of a specific type -- add table and field -- filtering for content.

  When performing business analysis, you might want to export a single model into multiple tables, which is useful when providing specific results tables to specific

departments. The ability to export a single SAS Cost and Profitability Management staging table into multiple database tables might be useful when splitting the actual costs and budget costs. To export a single table into multiple tables, you need to add a table and select multiple versions of the same staging table type, and map the versions to different destination table names. Then, apply a filter to limit the results going to each destination table.

Filtering Data

Use the following methods to filter the data that you want to export:

- select parts of a dimension
- specify comparisons for the values of attributes
- specify comparison operators for the values of fields in a table

If you filter by a table field, you do not need to export the field.

The following table lists the comparison operators:

Operator	Description	Field type
LIKE	Wildcard Use the percentage symbol (%) to specify any amount of text, including spaces. For example, Name LIKE fiscal% will match fiscal, fiscally, and fiscal year 2004.	Text
	Use an underscore (_) to specify a single character. For example, Name LIKE account_ will match account1, account2, and accountX.	Text
	If you omit both % and _, the comparison is the same as when you use the operator =. For example, Name LIKE fiscal is equivalent to Name = fiscal.	Text
=	Equal	Text or numeric
<>	Not equal	Text or numeric
<	Less than	Numeric
>	Greater than	Numeric
<=	Less than or equal to	Numeric
>=	Greater than or equal to	Numeric

Using the Export Wizard

Overview

When you use the Export wizard to export model data, you can:

Export to a Database

To export a model to an database, select **File** ⇨ **Export** ⇨ **Model Data**, and then select **Database**. See "Export a Model to a Database" on page 656.

When you export to a database, you can select individual database tables and properties to export:

- "Determining the Tables to Export" on page 651
- "Exporting, Filtering, and Limiting Calculated Results" on page 654
- "Filtering Data" on page 655

Export to an XML or ZIP File

To export a model to an XML or ZIP file, select **File** ⇨ **Export** ⇨ **Model Data**, and then select **XML or ZIP file**. See "Export a Model to an XML File" on page 666.

Note: When you export to an XML file, you must export all of the model data. If the XML file already exists, it is overwritten.

Export Survey Model Data

To export survey model data, select **File** ⇨ **Export** ⇨ **Survey Model Data**. See "Exporting Survey Data" on page 669.

For more information on surveys, see Chapter 4, "Introduction to Surveys," in *SAS Cost and Profitability Management: Data Administration Guide*.

Note: You can export model data without first having to open the model.

See Also

- "Choosing the Export Format" on page 649
- "Archiving a Model" on page 651

Export a Model to a Database

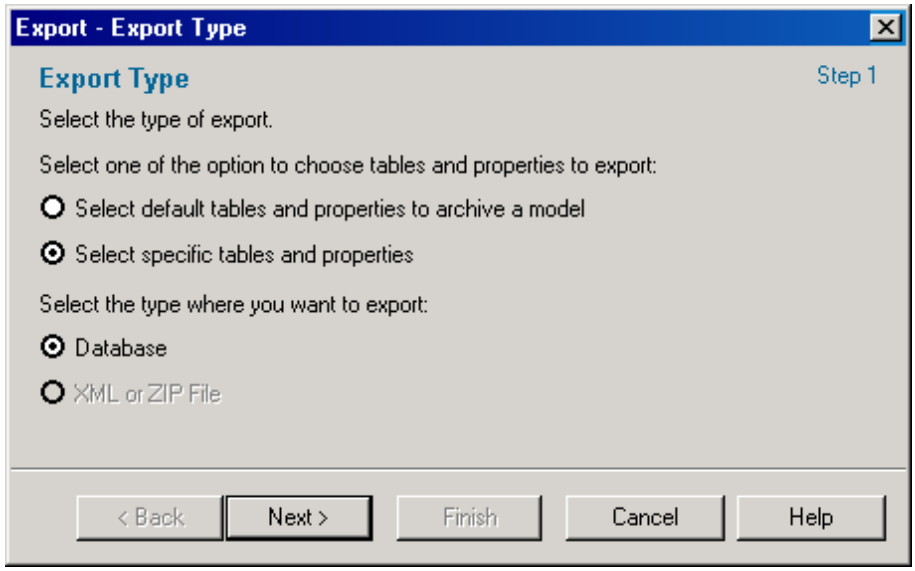
Before attempting to export data to a database, see Chapter 73, "Connecting to a Database," on page 617 which provides information about preparing to connect to the most common databases.

Note: You can perform this task without first opening a model.

1. Verify that the model is ready.
2. Select **File** ⇨ **Export Model Data**.

   The Export Wizard appears.

Step 1 of the Wizard

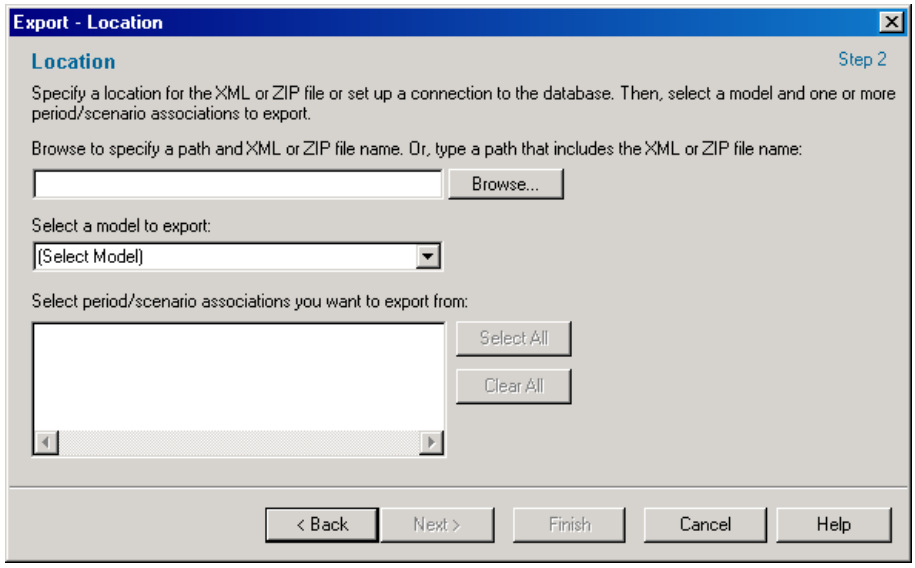


1. Select the **Select specific tables and properties** option.

   Notice that the only export type available is **Database**.

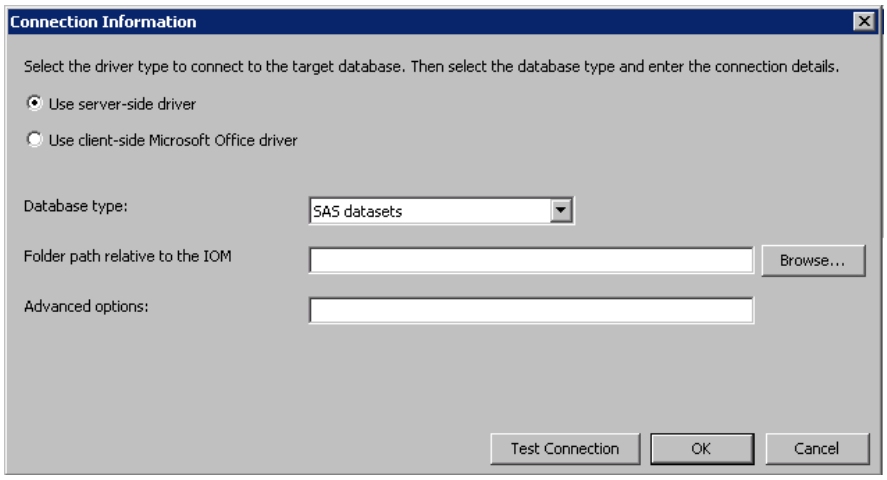
2. Click **Next**.

Step 2 of the Wizard



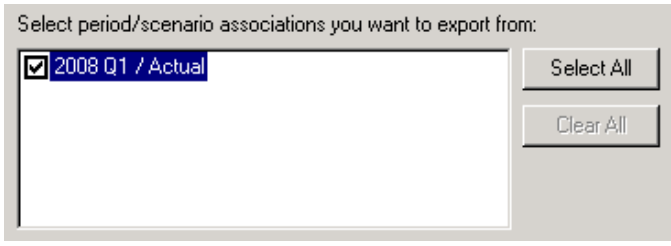
1. Click Browse...

   The Connection Information window opens.



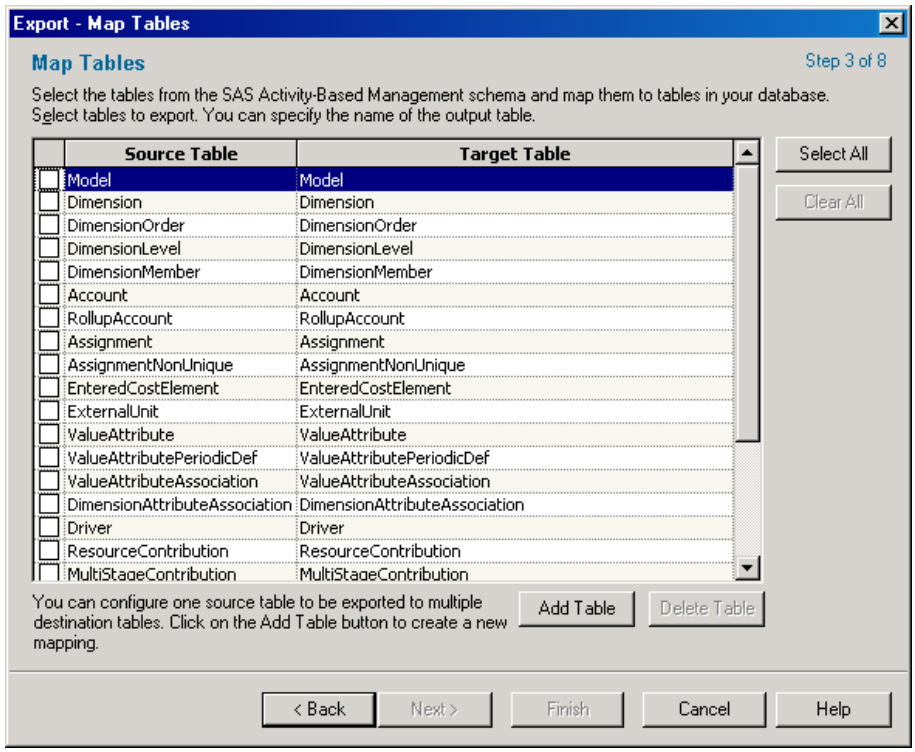
For information on using this window, see Chapter 73, "Connecting to a Database," on page 617.

2. From the **Select a model to export** drop-down list, select a model.
3. From the **Select period/scenario associations you want to export from** list, select the check box next to one or more period/scenario associations.



4. Click **Next**.

Step 3 of the Wizard



1. To select a table to export, select the check box to the left of the table in the **Source Table** column.

   You can select as many tables as needed.

2. To change the name of an exported table, click in the **Target Table** column to the right of a source table and type a new name.

3. To map a source table to more than one target table, do the following:

   a. Click **Add Table**.

      A new row is added with default information.

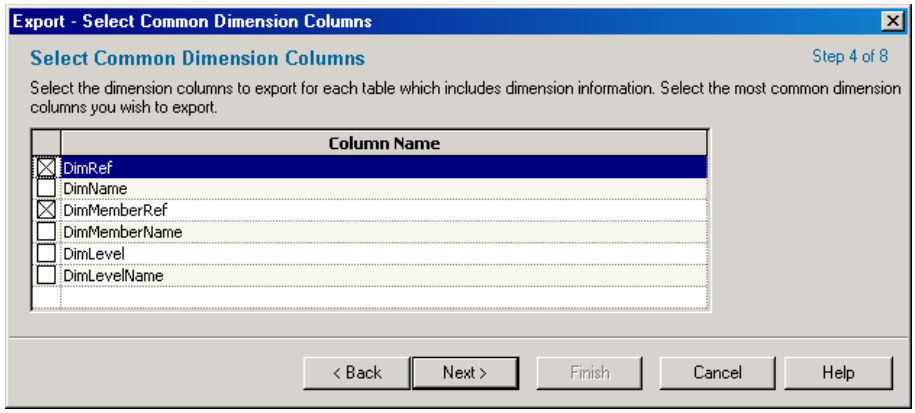
   b. Click in the **Source Table** column, and select a SAS Cost and Profitability Management table from the drop-down list.

4. Repeat steps 1 and 2 for every table that you want to export.

   Next, you will select dimensions to export for each table that contains dimension information.

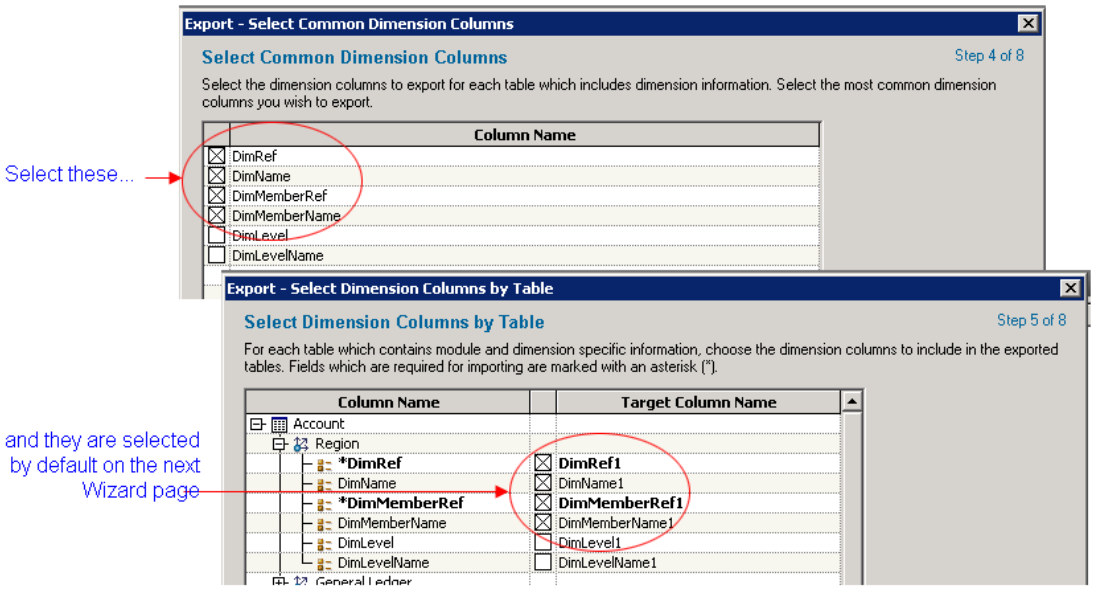
5. Click **Next**.

Step 4 of the Wizard



1. To select dimensions within tables to export, select the check box to the left of the dimension in the **Column Name** column.

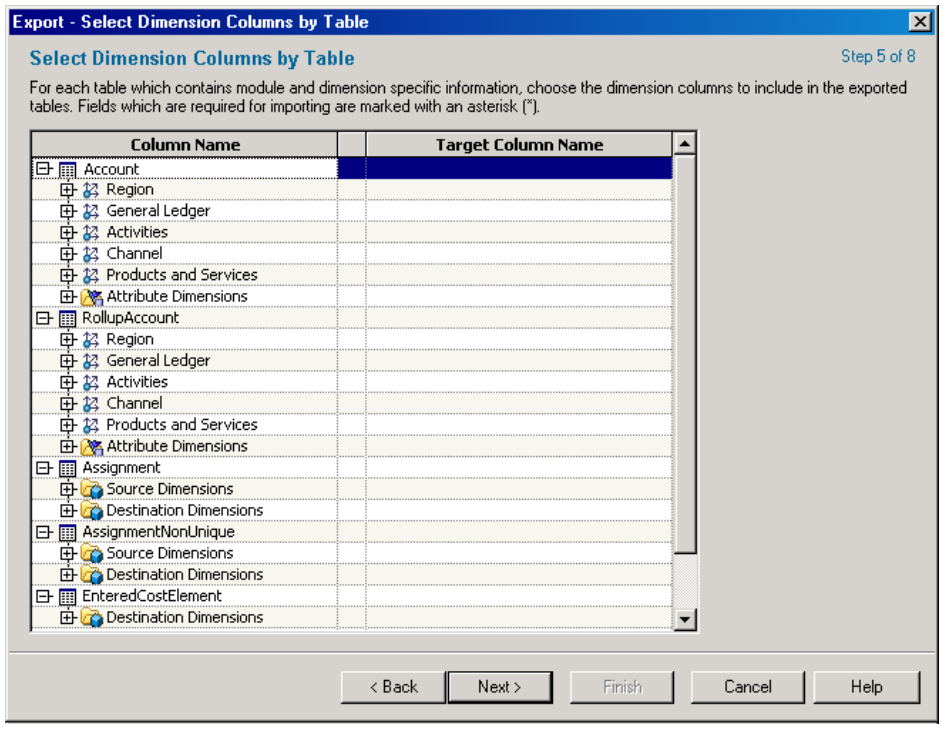
   Columns that you select are selected by default on the next Export Wizard screen. You can, however, change your selection on the next screen. That is, you can deselect a field that you had selected, or select a field that you had deselected.



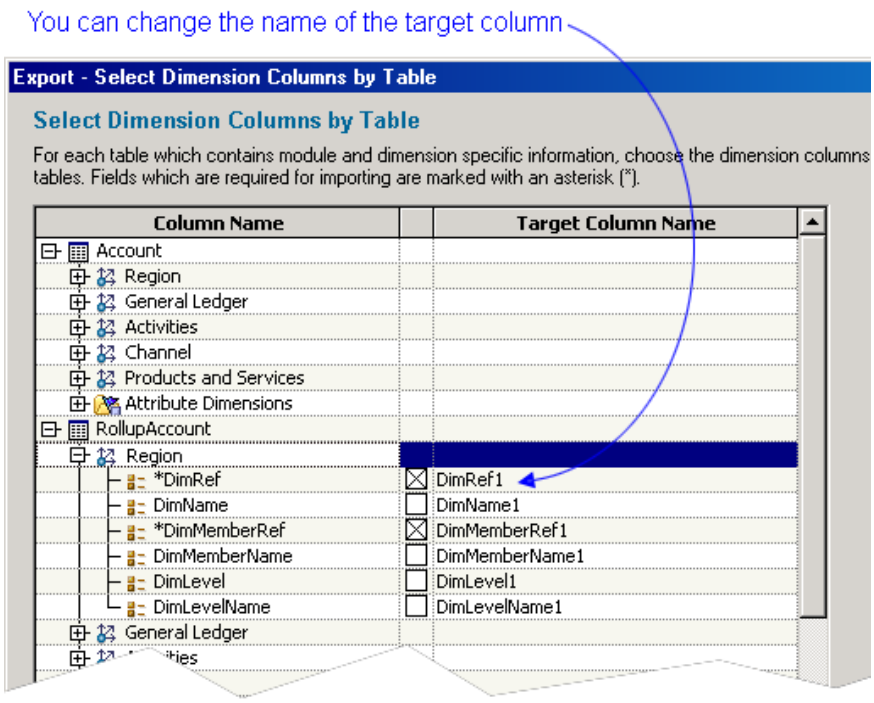
   Next, you will select dimensions and module information to export for each table that contains dimension information or module information.

2. Click **Next**.

Step 5 of the Wizard



1. Expand each table to review which columns are automatically selected to export.



2. To select a column to export, select the check box to the left of the **Target Column Name**.

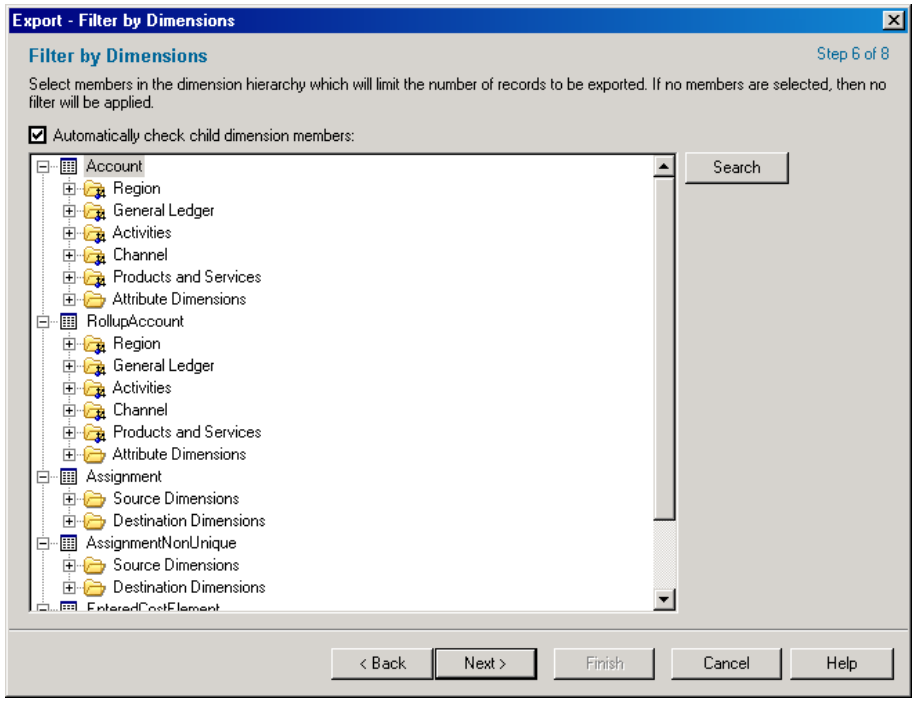
   Notice that columns marked with an asterisk (*) must be exported.

3. To change the name of an exported column, click in the **Target Column Name** column to the right of a column name, and type a new name.

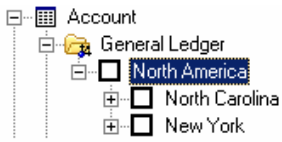
Next, you will select dimension members to export. By default, all data will be exported. By selecting dimension members, you can limit the amount of data that is exported.

4. Click **Next**.

Step 6 of the Wizard

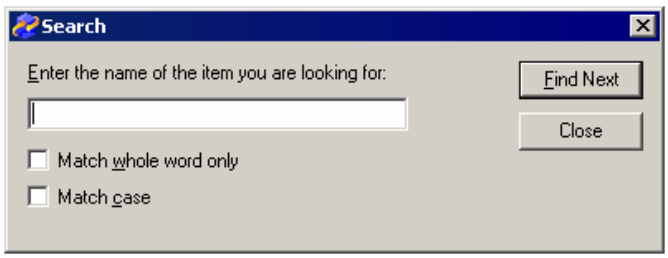


1. Expand each table to view the dimension members.



2. If you want to select all dimension members when a dimension is selected, select the **Automatically check child dimension members** option.
3. To select a dimension member, select the check box to the left of the dimension member.
4. To search for a dimension member, do the following:
   a. Click **Search**.

      The Search dialog box appears.



b. In the **Enter the name of the item you are looking for** box, type the name of the item.

   You can search for a table, a dimension, or a dimension member.

c. Select the **Match whole word only** option and the **Match case** option.

d. Click **Find Next**.

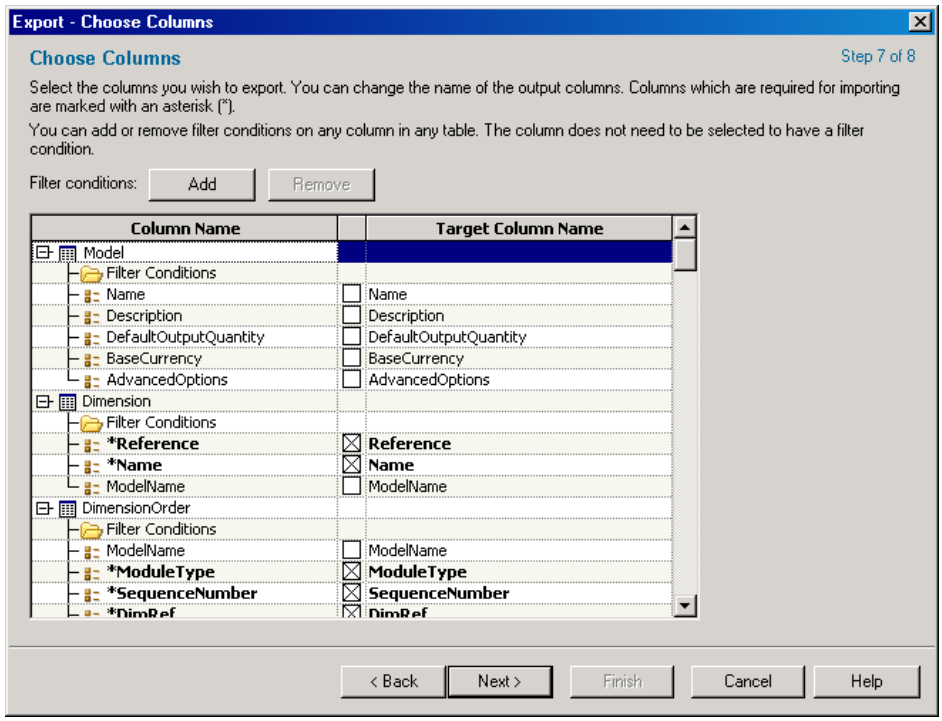
The next occurrence of the item is found.

5. Repeat steps 1 through 3 for each dimension member that you want to export.

   Next, you will select columns to export. You can limit which data to export by creating filters.

6. Click **Next**.

Step 7 of the Wizard



1. To select a column, select the check box to the left of the target column name.

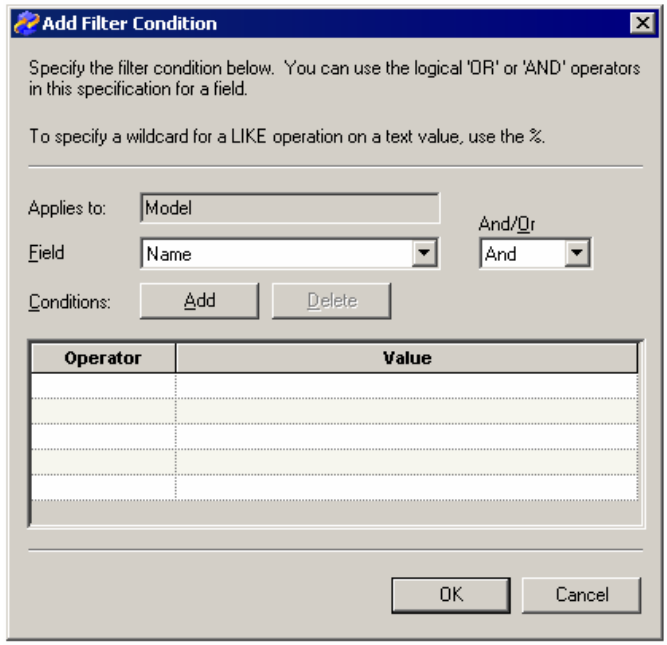
   Notice that columns marked with an asterisk (*) must be exported.

2. To change the name of an exported column, click in the **Target Column Name**column to the right of a column name, and type a new name.

3. To add a filter condition to a table, do the following:

   a. Click **Add**.

      The Add Filter Condition dialog box appears.



b. Select a database **Field** on which to base the filter condition.

c. From the **And/Or** drop-down list, select a logical operator.

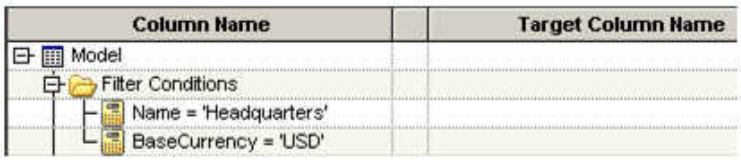
d. Click **Add**.

The field appears in the list.

e. Select an **Operator** from the drop-down list.

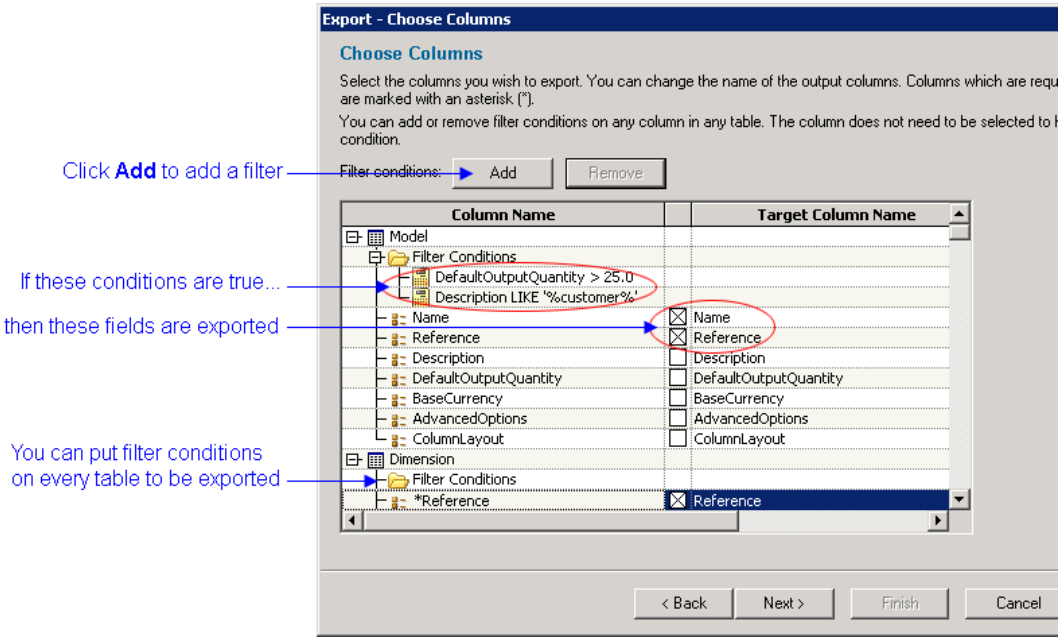
f. Type a **Value** for the operator.

You can create as many filter conditions as needed, but each filter condition can specify only one field. For example, if you need to limit the export to a model named Headquarters with a base currency of United States Dollars, you would have to open the Add Filter Condition dialog box twice. The first time, you would need to create the filter condition for the model name Headquarters. The second time, you would need to create the filter condition for the base currency United States Dollars. The completed filter conditions would look like the following:



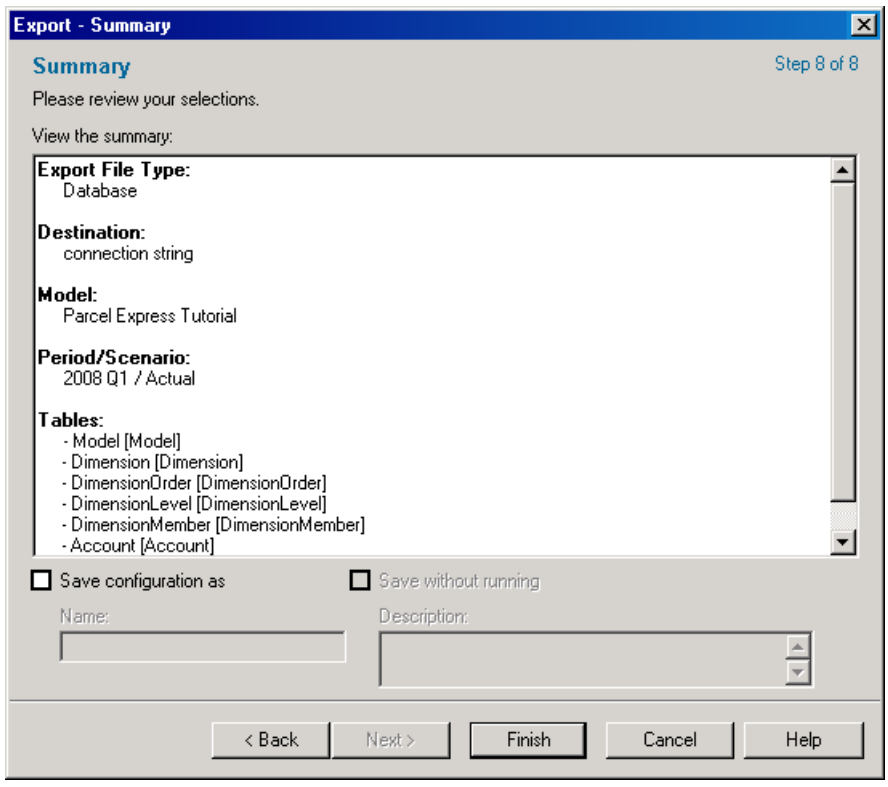
Selected columns are exported only if they pass the filter. In other words, for a column to be exported, it must both:

- be selected
- pass whatever filters exist for the table



4. Click **Next**.

Step 8 of the Wizard



1. Review the export summary.
2. If you need to change any information, click **Back** until you reach the step that you need to change in the wizard.

   All of the information that you have specified is saved. Click **Next** to advance through the wizard.

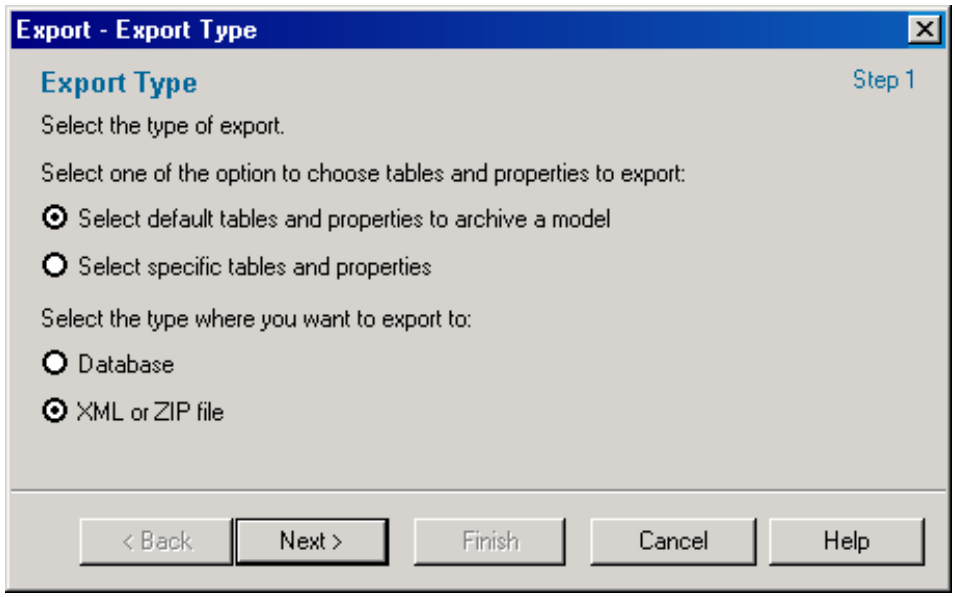
3. To save the export configuration so that the export can be easily run again, do the following:
   a. Select the **Save configuration as** option.
   b. Type the **Name**.
   c. Type the **Description**.
4. Select **Save without running** to save the export configuration without performing the export.
5. Click **Finish**.

Export a Model to an XML File

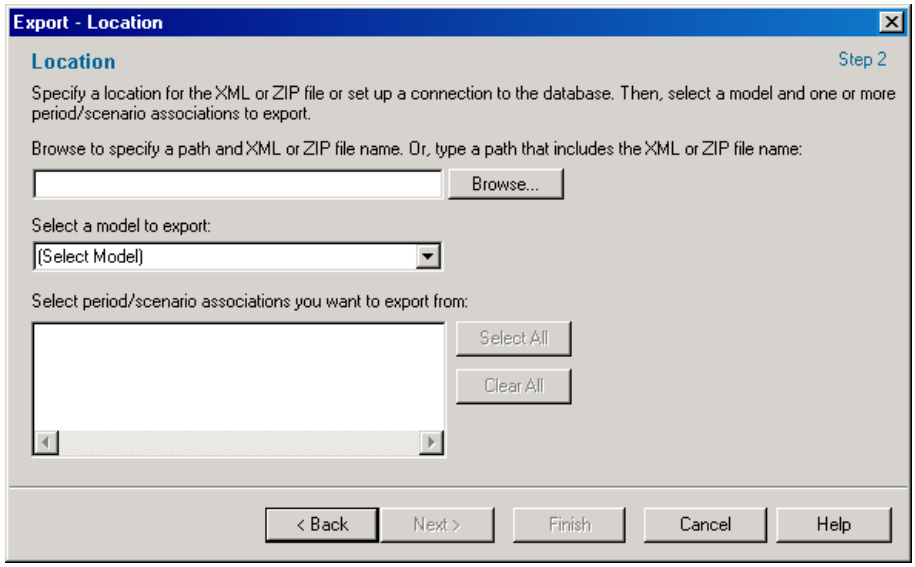
Note: You can perform this task without first opening a model.

1. Verify that the model is ready.
2. Select **File** ⇨ **Export Model Data**.

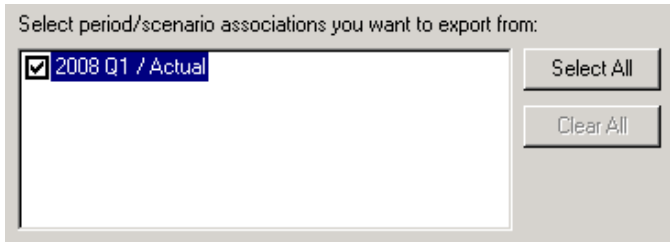
   The Export Wizard appears.



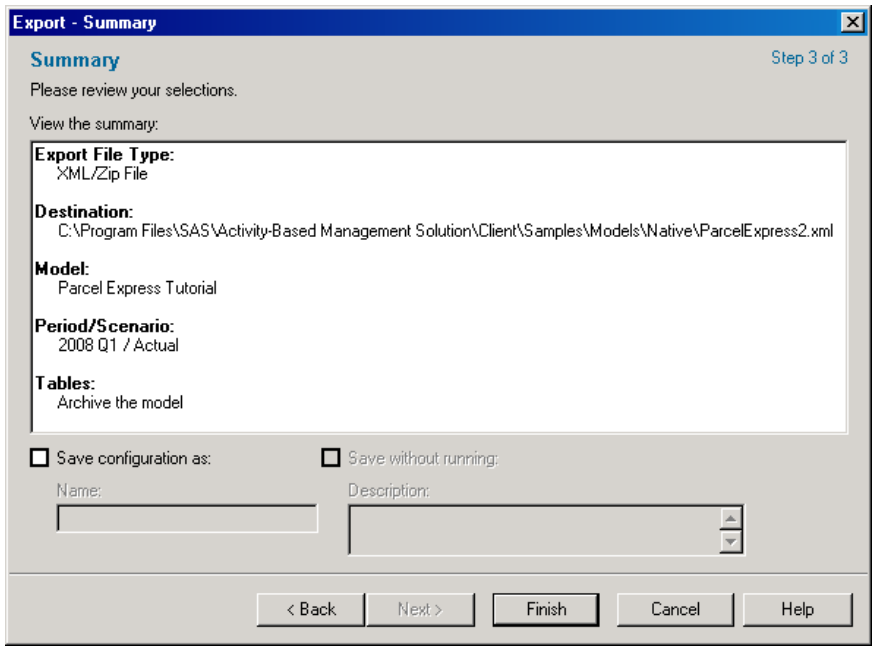
3. Select the **Select default tables and properties to archive a model** option.
4. Select the **XML File** option.
5. Click **Next**.



6. Type the absolute pathname to the XML file. Or, click Browse... .
7. From the **Select a model to export** drop-down list, select a model.
8. From the **Select period/scenario associations you want to export from** list, select the check box next to one or more period/scenario associations.



9. Click **Next**.



10. Review the export summary.

11. If you need to change any information, click **Back** until you reach the step that you need to change in the wizard.

    All of the information that you have specified is saved. Click **Next** to advance through the wizard.

12. To save the export configuration so that the export can be easily run again, do the following:

    a. Select the **Save configuration as** option.

    b. Type the **Name**.

    c. Type the **Description**.

13. Select **Save without running** to save the export configuration without performing the export.

14. Click **Finish**.

Export Survey Model Data

To do a survey, you do not have to export everything out of your model. For complete information about surveys, see Chapter 4, "Introduction to Surveys," in *SAS Cost and Profitability Management: Data Administration Guide*.

See Also

"Exporting Survey Data" on page 669

Chapter 77

Survey Data

Exporting Survey Data

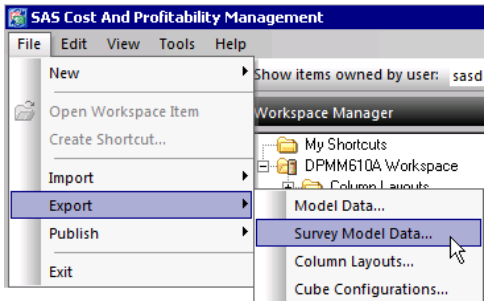
To create a survey, you do not have to export everything out of your model. Here are the steps to ensure you export only the required fields.

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

To export survey data, do the following:

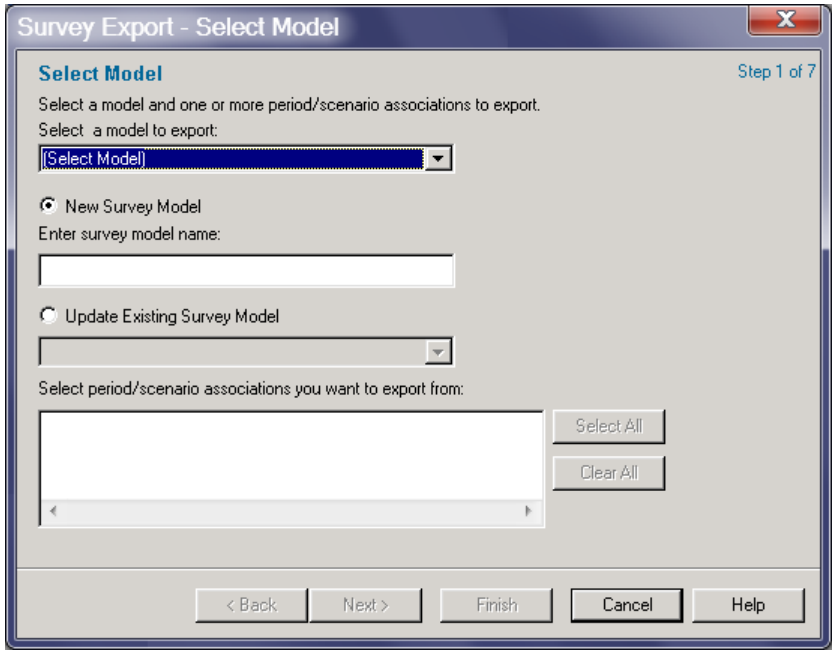
Note: The Surveys application should not be open when you export survey data because the Surveys application caches model data when it opens and will not be aware of newly exported survey data.

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.



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.

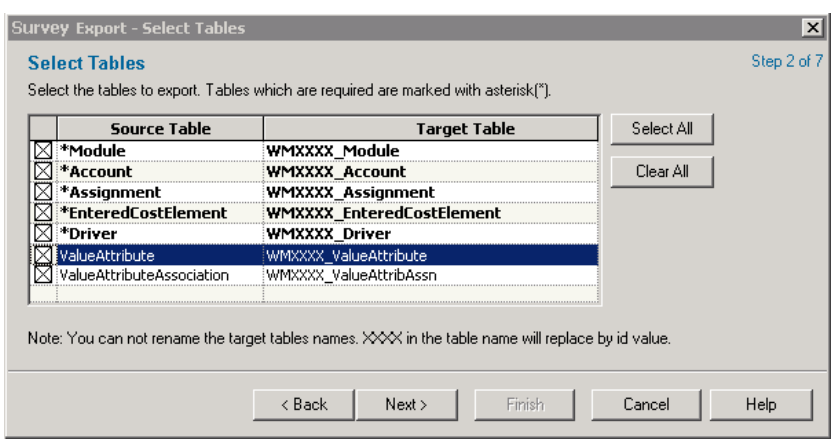
  This option is for updating the following two tables related to attributes: ValueAttribute table and ValueAttributeAssociation table. The two tables must have been exported previously.

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.



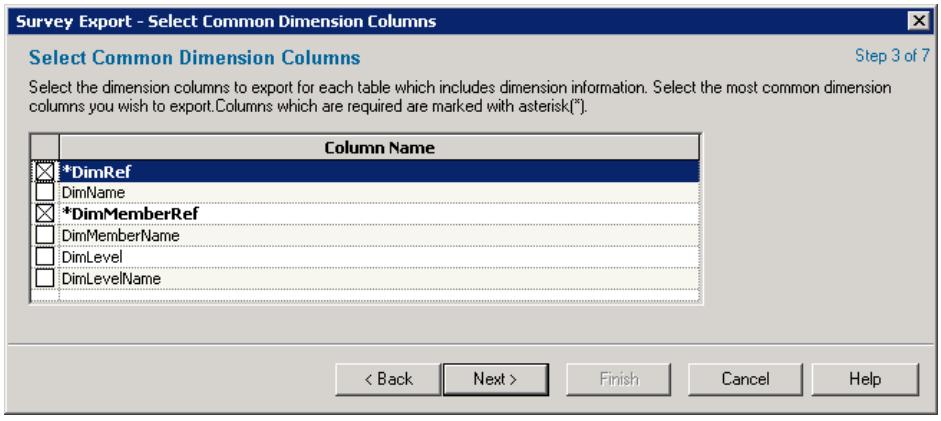
If you plan to survey Numeric Attributes, then you also must check the following two tables:

ValueAttribute table
See "ValueAttribute table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

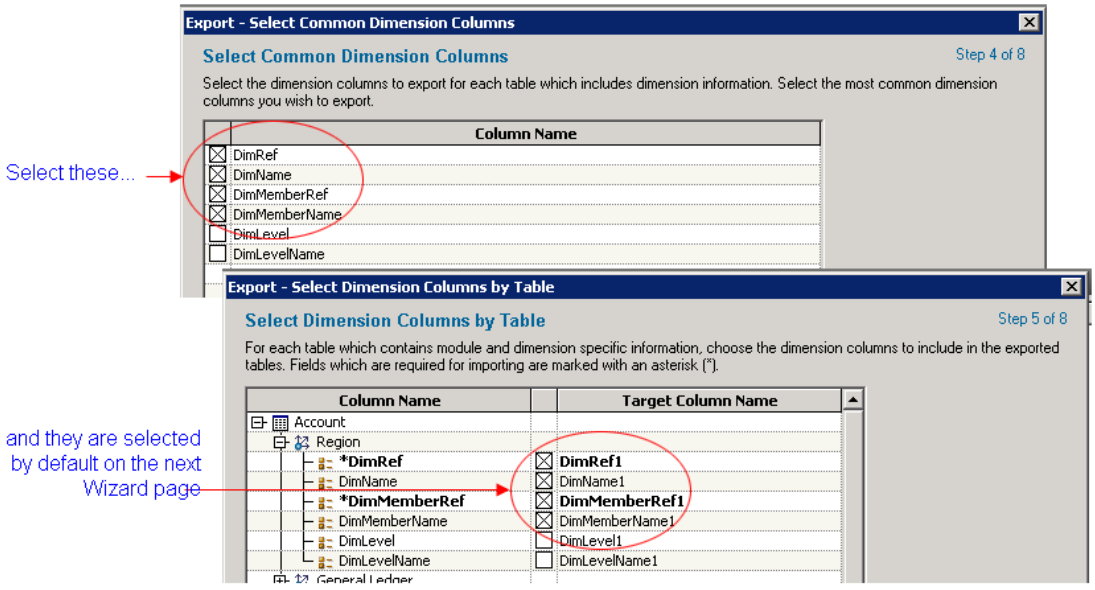
ValueAttributeAssociation table
See "ValueAttributeAssociation table" in Chapter 14 of *SAS Cost and Profitability Management: Data Administration Guide*.

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

   Note: You cannot unselect required columns.

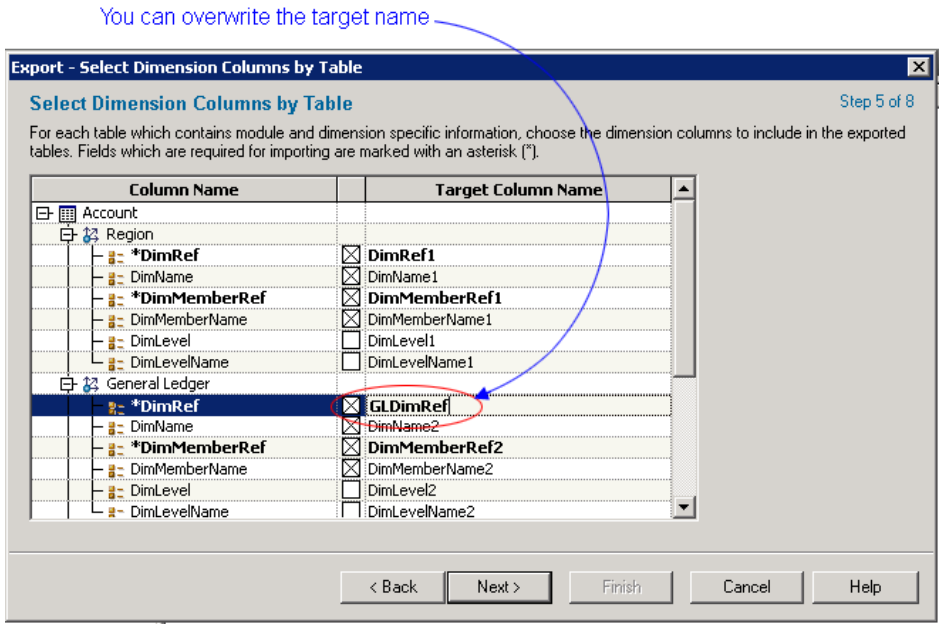


Columns that you select are selected by default on the next Export Wizard page. You can, however, change your selection on the next page. That is, you can deselect a field that you had selected, or select a field that you had deselected.

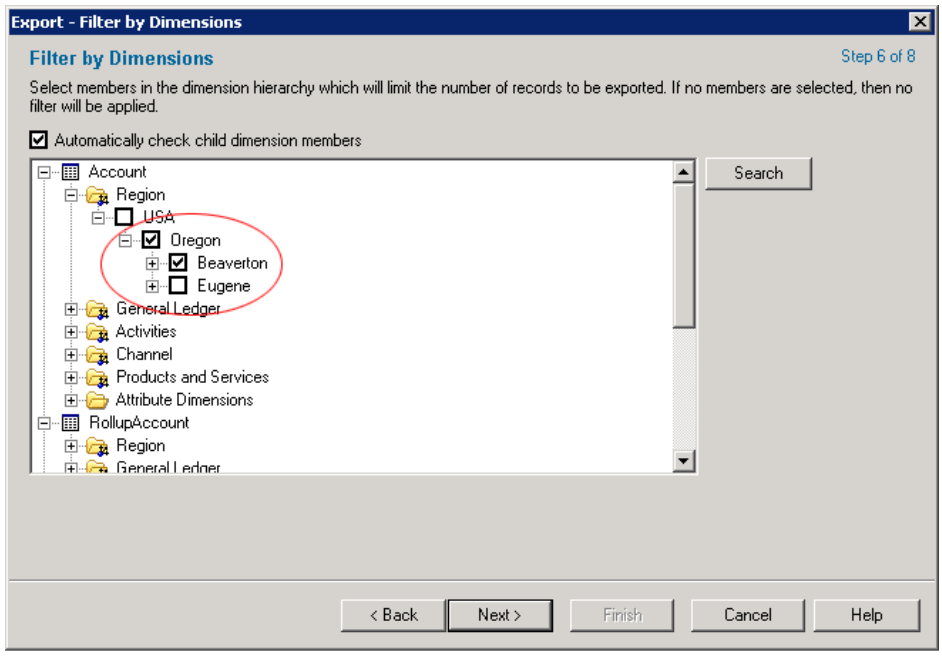


5. Select dimension columns to export for each table.

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



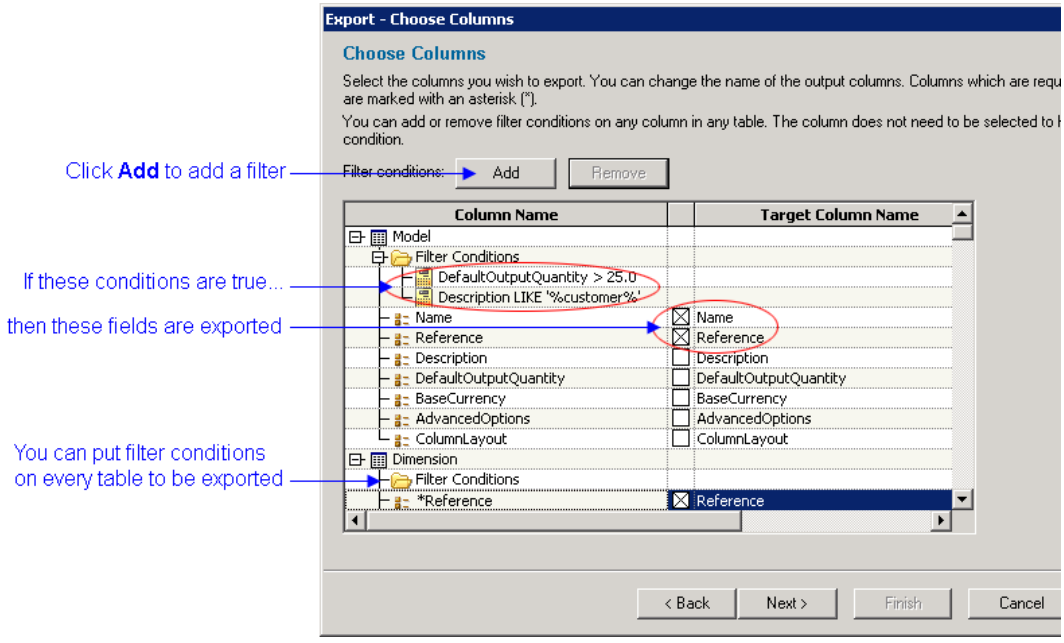
6. Filter by dimensions (that is, select those dimension members that will be exported for each table).



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)
```

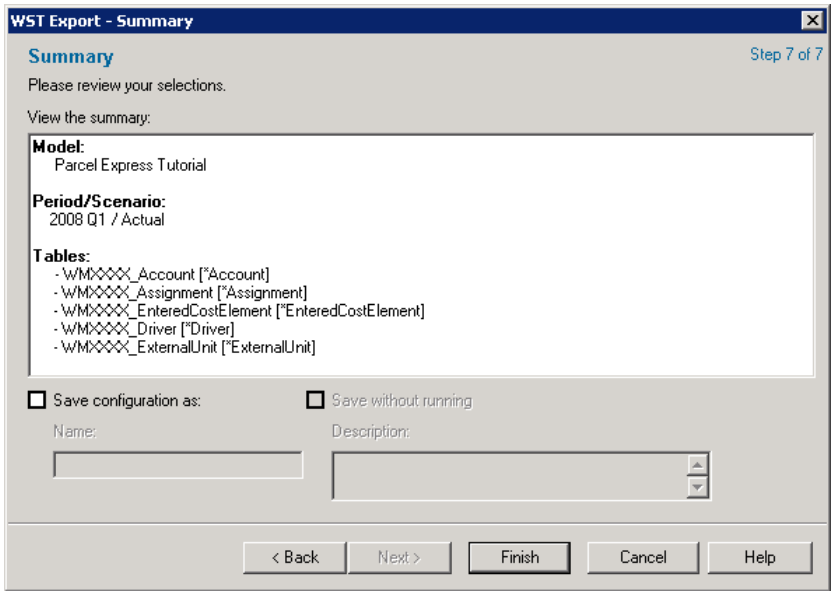
Assignment table

```
Source Accounts.DriverName
DriverQuantityFixed
```

EnteredCostElement table

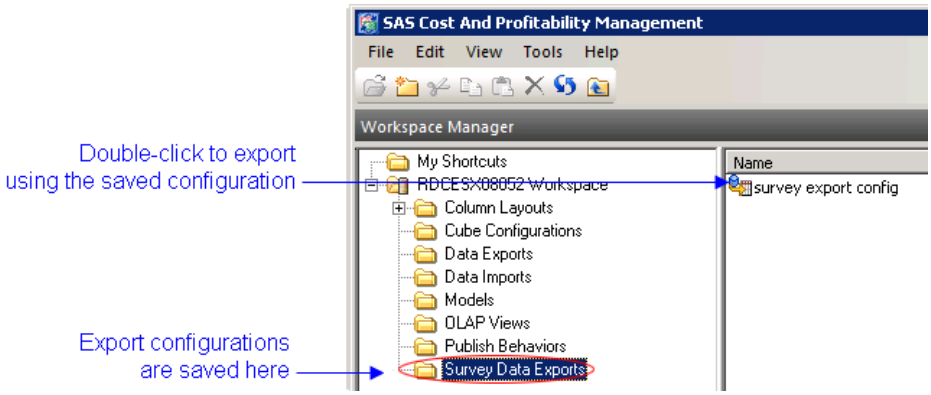
```
EnteredCost
EnteredUnitCost
```

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.



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.

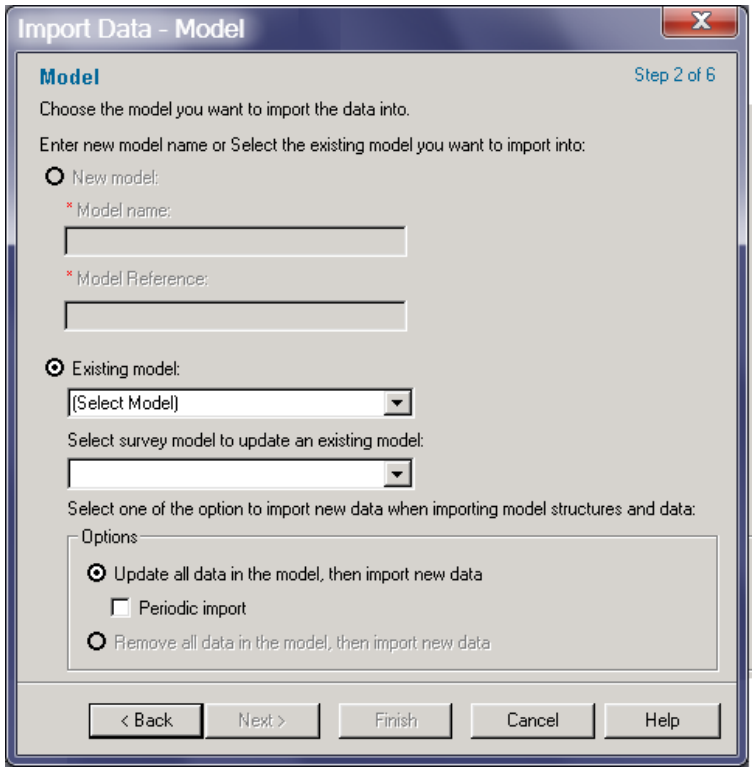
   b. Select the survey data to import.

      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 669.

   c. Select whether you want to do periodic import.

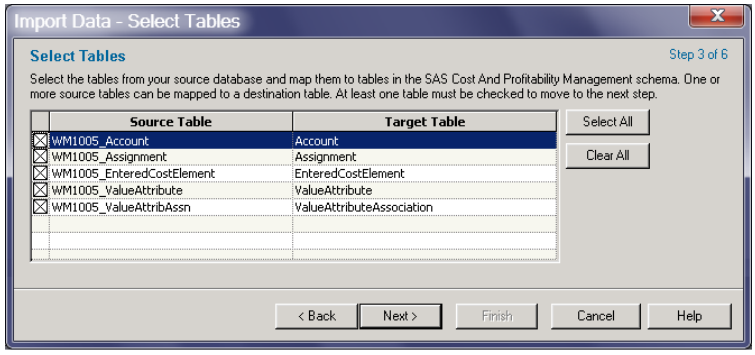
      Periodic import allows you to import only the periods that have changed in a model. For more information, see "Incremental Cube Generation" on page 569.

   d. Click **Next**.



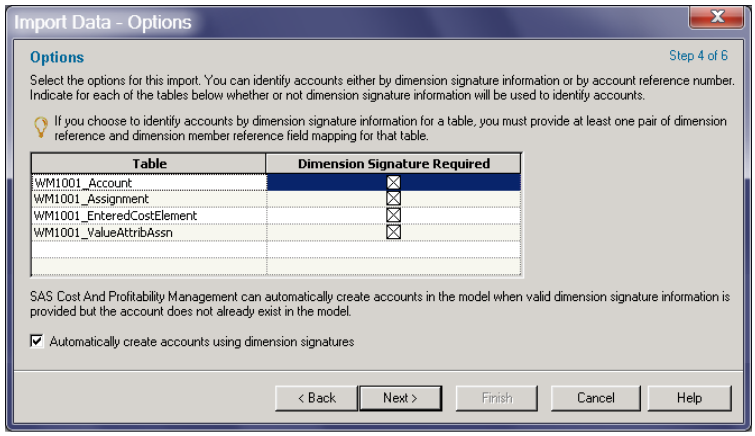
4. On the Import Data – Select Tables window, select tables from the survey data being imported and map them to tables in the model being updated.

   The tables being imported correlate to the tables that were previously exported. See Step 3 on page 670.



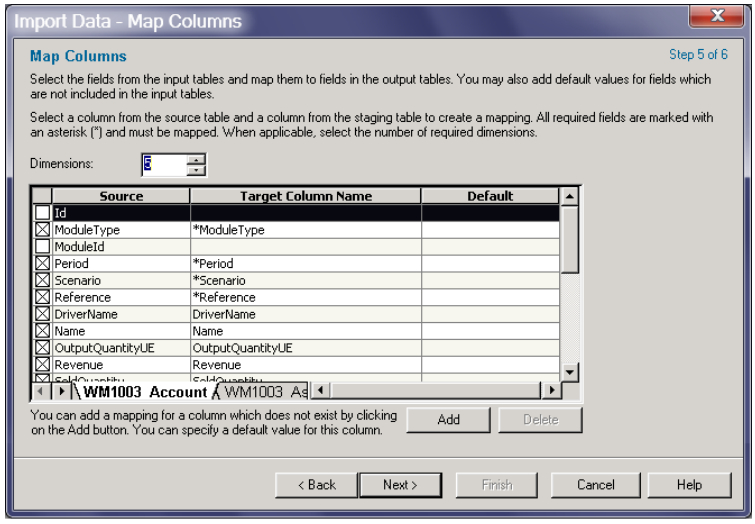
5. On the Import Data – Options window, for each table being imported, select whether you want to identify accounts by their dimension signature.

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

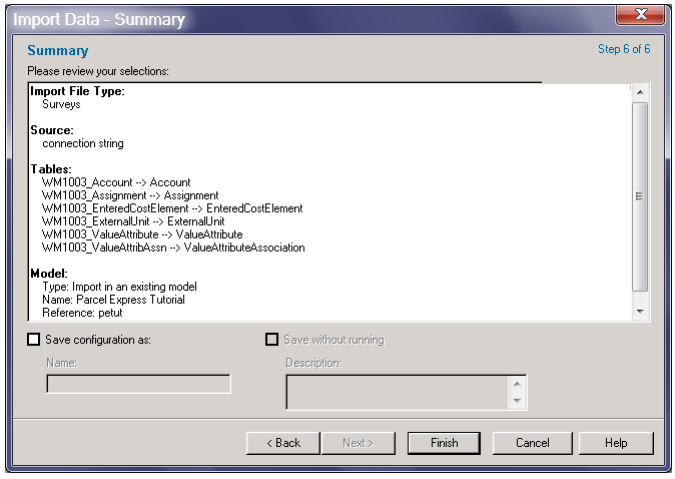


6. On the Import Data – Map Columns window, select the columns to be imported..

   The columns being imported correlate to the columns that were previously exported. See Step 4 on page 671. And see Step 5 on page 671.



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



Chapter 78

Cube Configurations

Import Cube Configurations

1. Go to the Workspace Manager.
2. Select **File** ⇨ **Import** ⇨ **Cube Configurations**. The Import Cube Configuration dialog box opens.
3. Select the file to import and import options:

   Rename cube configuration being imported
   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.

   Replace the existing cube configuration
   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.

   Do not import duplicate cube configurations
   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.
4. Click **Finish**.

Export Cube Configurations

1. Go to the Workspace Manager.
2. Select **File** ⇨ **Export** ⇨ **Cube Configurations**. The Export Cube Configuration dialog box opens.

3. Select the cube configurations to export and export options:

   Automatically check children
   Select this check box to recursively select cube configurations that are inside the selected folder.

   Note: This option applies only to subsequent folder selections. It does not select subfolders of folders that are already selected.

   Include folders
   Select this check box to store folder information in the exported file. As a result, folders can be recreated when you import the cube configurations.

   Note: This option does not create folders in the export directory. It stores the folder information in the export file so that the folders can be recreated later during import.

   Export to this path and file name
   Select the path and name of the export file.

   Note: Only one export file is created, even if you select multiple folders and cube configurations to export.

4. Click **Finish**.

Chapter 79

Column Layouts

Import a Column Layout

Note: You can perform this task without first opening a model.

1. Select **File** ⇨ **Import** ⇨ **Column Layouts**.

   The Import Column Layouts Wizard appears.

2. Follow the directions in the wizard.

Export a Column Layout

You can perform this task without first opening a model.

1. Select **File** ⇨ **Export** ⇨ **Column Layouts**.

   The Export Column Layouts Wizard appears.

2. Follow the directions in the wizard.

Chapter 80

Export Module Views to Excel

Export Module Views to Excel

Summary

If you want a printable form of the hierarchical structure of a module, and one that you can easily modify without affecting the underlying model, you can export the module to Excel. For example, using system-supplied column layouts, you can view modules and export the resulting module views to Excel so as to obtain printed reports on different aspects of your model.

You can either select particular rows of a module view to export, or you can export the entire module.

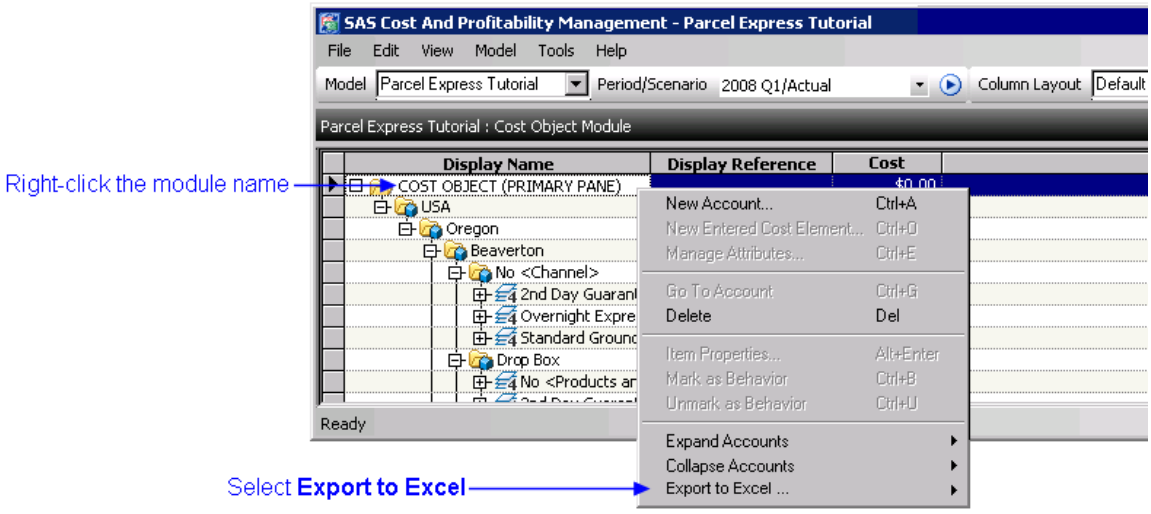
Note: Exported spreadsheets cannot be re-imported into SAS Cost and Profitability Management.

Export the Entire Module

To export the expanded rows of an entire module:

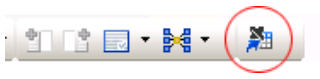
1. Select the primary pane of the module view.
2. Without selecting any rows, right-click the module name.

   Note: If more than one row is selected, then only the selected rows are exported.
3. Select **Export to Excel**.



The Export to Excel dialog box appears.

Note: You can also click the Export to Excel icon in the toolbar.

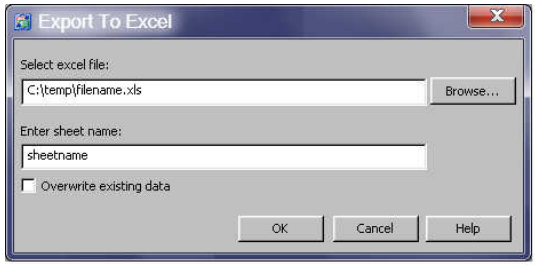


4. Specify the name of the Excel file and the name of the sheet to be created and whether you want to overwrite existing data.
   - If the Excel file does not exist, it is created with a sheet named as you specify.

     Note: You must specify a complete path and file name.

     Note: Do not name the sheet Sheet1 unless you are replacing Sheet1 of an already existing Excel file.

     Note: The sheet name cannot contain any special characters except underscores.
   - If the Excel file already exists but does not contain a sheet with the name that you specify, then a new sheet is added to the existing file.



5. Click **OK**.

Only Rows that are Expanded are Exported

Only rows that are expanded when you perform the export are exported. So, what you see in the module view is what you get in the spreadsheet.

It is not necessary, however, that either all the rows or all the columns are visible when you perform the export.

- Some rows that are expanded may not be visible if there are more rows than fit on the screen, such that vertical scrolling is necessary to see them.
- Some columns in the column layout may not be visible if there are more columns than fit on the screen, such that horizontal scrolling is necessary to see them.

Exporting works as though you had an infinitely large screen that is able to display, without scrolling, all the rows in the hierarchy that are expanded and all the columns in the column layout.

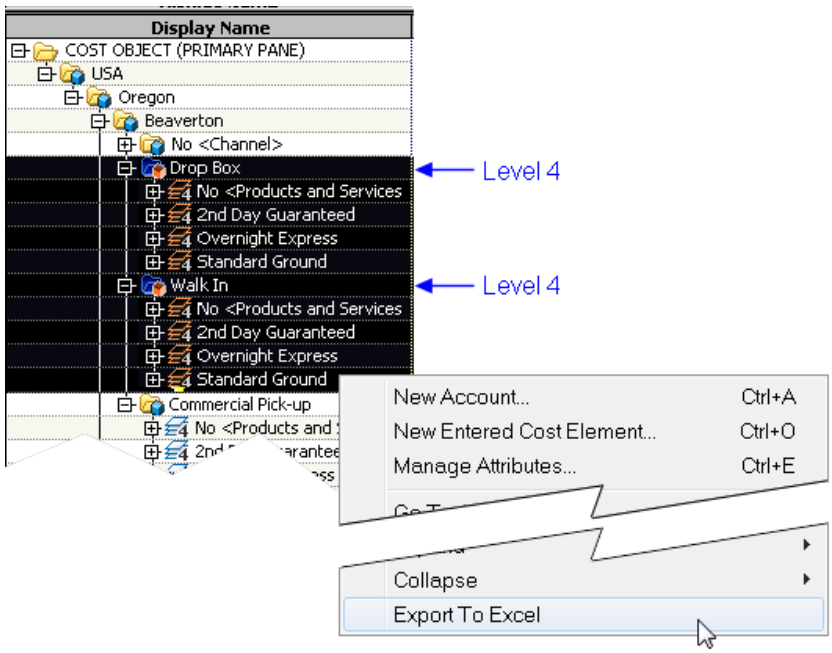
Note: Columns that you have added to the current column layout are exported even if you haven't yet saved the column layout with the additional columns.

Note: If there are no accounts in the module, then the export is not performed.

Select Particular Rows to Export

If you select a subset of the rows in the primary pane of a module view and select **Export to Excel**, then only the selected rows are exported.

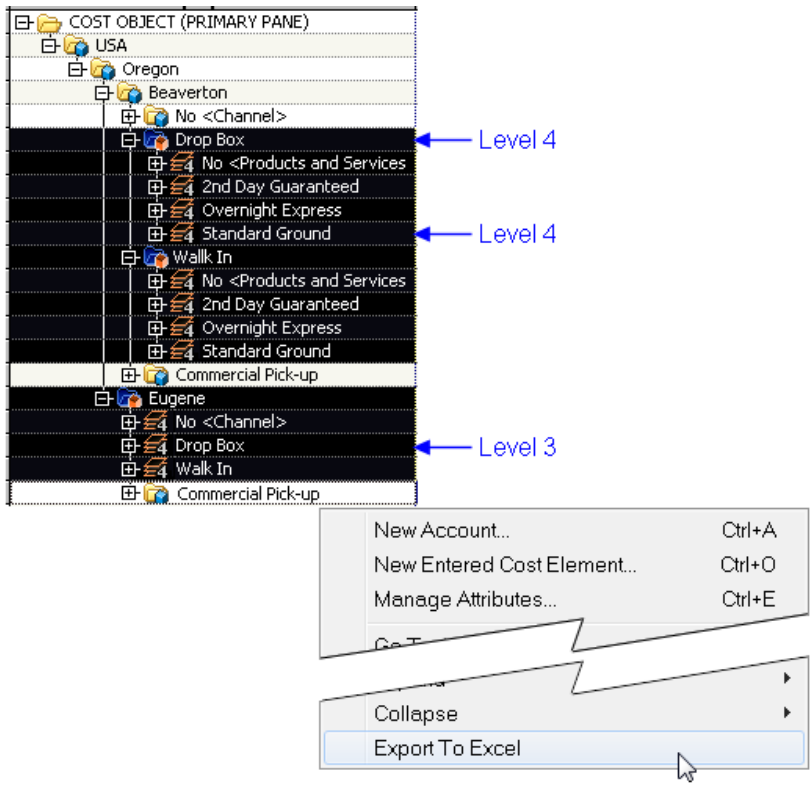
If the rows selected are contiguous and the first row selected is at the highest level of the selected rows, then the rows in the spreadsheet will exhibit the same hierarchy. For example, in the following picture, where all the rows for Drop Box and for WalkIn have been selected, their hierarchy will be reflected in the Excel spreadsheet.



The following picture shows the resulting spreadsheet.

Display Name	(Type)	Display Reference	Cost ($)	(Item View Order)
Drop Box	Rollup Account	Drop Box	0.00	6
No <Products and Services>	Account	No <Prod_Serv>	0.00	7
2nd Day Guaranteed	Account	2nd Day Guaranteed	0.00	8
Overnight Express	Account	Overnight Express	0.00	9
Standard Ground	Account	Standard Ground	0.00	10
Walk In	Rollup Account	Walk In	0.00	11
No <Products and Services>	Account	No <Prod_Serv>	0.00	12
2nd Day Guaranteed	Account	2nd Day Guaranteed	0.00	13
Overnight Express	Account	Overnight Express	0.00	14
Standard Ground	Account	Standard Ground	0.00	15

However, if you select non-contiguous rows that are at different levels, then it may not be possible to preserve the hierarchy in the spreadsheet. For example, in the following picture Eugene will be at the same level in the Excel spreadsheet as Drop Box and WalkIn. Because the rows selected are non-contiguous and at different levels, it is not practical for the export facility to reconstruct the larger hierarchy, of which the non-contiguous rows are a part.



The following picture shows the resulting spreadsheet. Notice that an empty row separates the discontinuous hierarchies.

An empty row separates the discontinuous hierarchies

11	Display Name	(Type)	Display Reference	Cost ($)	(Item View Order)
12	Drop Box	Rollup Account	Drop Box	0.00	6
13	No <Products and Services>	Account	No <Prod_Serv>	0.00	7
14	2nd Day Guaranteed	Account	2nd Day Guaranteed	0.00	8
15	Overnight Express	Account	Overnight Express	0.00	9
16	Standard Ground	Account	Standard Ground	0.00	10
17	Walk In	Rollup Account	Walk In	0.00	11
18	No <Products and Services>	Account	No <Prod_Serv>	0.00	12
19	2nd Day Guaranteed	Account	2nd Day Guaranteed	0.00	13
20	Overnight Express	Account	Overnight Express	0.00	14
21	Standard Ground	Account	Standard Ground	0.00	15
22					
23	Eugene	Rollup Account	Eugene	0.00	17
24	No <Channel>	Rollup Account	No <Chnnl>	0.00	18
25	Drop Box	Rollup Account	Drop Box	0.00	19
26	Walk In	Rollup Account	Walk In	0.00	20
27	Commercial Pick-up	Rollup Account	Commercial Pick-up	0.00	21

Other Features

Panes are Frozen for Better Scrolling

When you export a module view, the Excel panes are frozen so that the column header does not scroll vertically. The column header is always visible. And, the account names are frozen so that they do not scroll horizontally. They are always visible when you scroll horizontally to see all the account properties. You can unfreeze the panes in Excel if you want scrolling to perform differently.

Only this area scrolls

A B C D	E F G H I J
Model Name:	RushM (M1019)
Period/Scenario:	2000/Actual
Module Name:	ACTIVITY (PRIMARY PANE)
Column Layout Name:	Default
Model Base Currency:	US Dollar (USD ($))

Display Name	(Type)	Cost ($)	DrvName	DQF	DQV	DrvRate ($)	DV
ACTIVITY (PRIMARY PANE)		510,000.00					
Administration	Rollup Account	162,600.00					
Sales and Marketing	Rollup Account	106,500.00					
Preliminary RFQ	Account	16,221.83	No of Quotes	200.00		9,655.85	
Sales Call	Account	23,276.15	No of Days Visits	400.00		404.10	
Take Order	Account	16,721.27	No of Orders	1,000.00		1,797.99	
Process Returns	Account	7,046.06	No of Returns	300.00		391.45	
Support Web Site	Account	7,890.88	Per Product	600.00		986.36	

Additional Columns

When you export a module view to Microsoft Excel, SAS Cost and Profitability Management includes two additional columns in the spreadsheet: **(Type)** and **(Item View Order)**.

(Type)

Tells what sort of data the spreadsheet row is for:

Display Name	Cost	Type
RESOURCE (PRIMARY PANE)	$510,000.00	
Administration	$162,600.00	Rollup Account
Sales Marketing	$106,500.00	Rollup Account
Ads Shows	$15,000.00	Account
Deprec Faciltities	$9,000.00	Account
Salary Trade Shows	$31,200.00	Account
Salary Direct Sales	$46,500.00	Account
Salary Sale Manager	$4,800.00	Account
Engineering	$32,400.00	Rollup Account
Depre Facilities	$7,200.00	Account
Salary Engineer	$19,200.00	Account

Note: The **(Type)** column contains the same data as the **Type** column in a column layout. If the module view being exported already contains a **Type** column, then SAS Cost and Profitability Management does not add an additional **(Type)** column to the exported spreadsheet.

(Item View Order)

Use this column to restore a sorted spreadsheet to its original display order (see "Unsorting" on page 687).

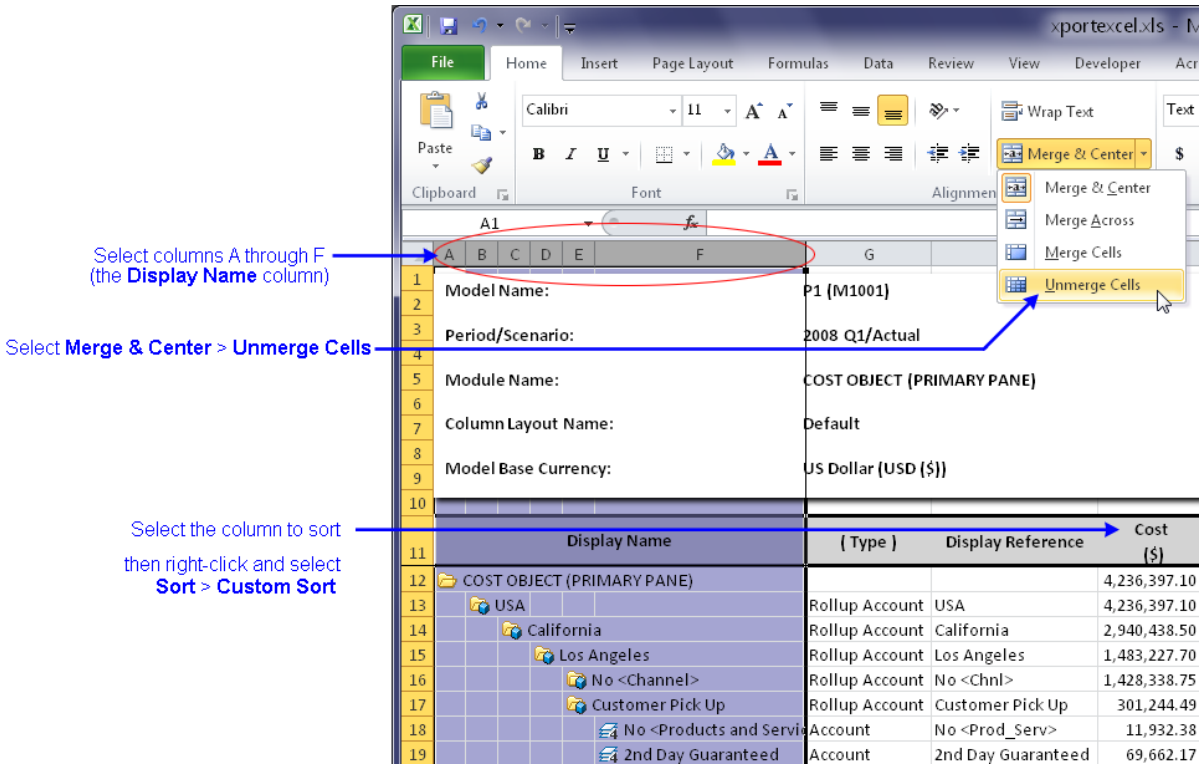
Display Name	Cost ($)	Type	Item View Order
RESOURCE (PRIMARY PANE)	510,000.00		1
Administration	162,600.00	Rollup Account	2
Sales Marketing	106,500.00	Rollup Account	3
Ads Shows	15,000.00	Account	4
Deprec Faciltities	9,000.00	Account	5
Salary Trade Shows	31,200.00	Account	6
Salary Direct Sales	46,500.00	Account	7
Salary Sale Manager	4,800.00	Account	8

Sorting

In order to sort columns in the exported Excel spreadsheet, you must first ungroup the **Display Name** column.

To ungroup the **Display Name** column:

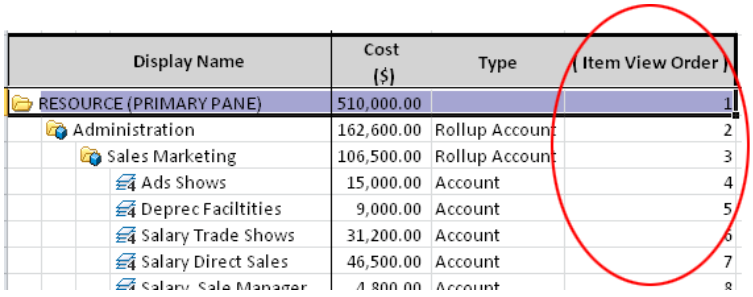
1. Select columns A through F (the **Display Name** column).
2. On the **Home** tab, select **Merge & Center** ⇨ **Unmerge Cells**.



3. Select the column that you want to sort and perform the sort.

Unsorting

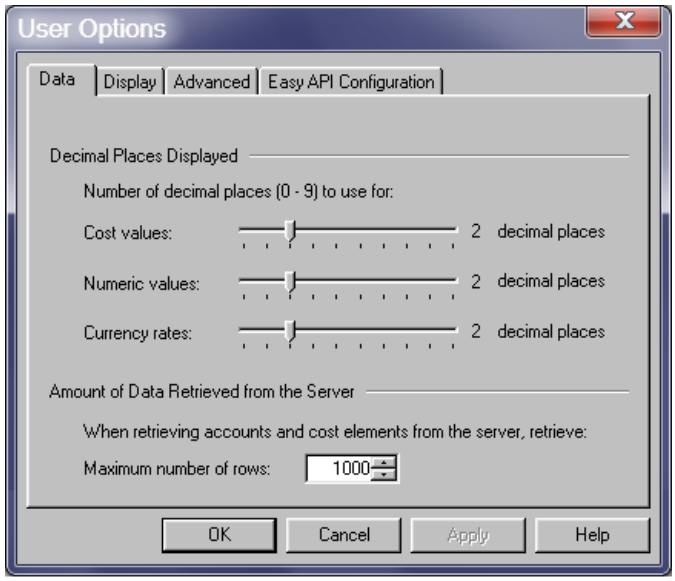
To restore a sorted spreadsheet to its original display order, you can use the **(Item View Order)** column that SAS Cost and Profitability Management adds to the Excel Spreadsheet for this purpose. The column contains the number of each row in the spreadsheet's original display order.



Display Name	Cost ($)	Type	(Item View Order)
RESOURCE (PRIMARY PANE)	510,000.00		1
Administration	162,600.00	Rollup Account	2
Sales Marketing	106,500.00	Rollup Account	3
Ads Shows	15,000.00	Account	4
Deprec Faciltities	9,000.00	Account	5
Salary Trade Shows	31,200.00	Account	6
Salary Direct Sales	46,500.00	Account	7
Salary Sale Manager	4,800.00	Account	8

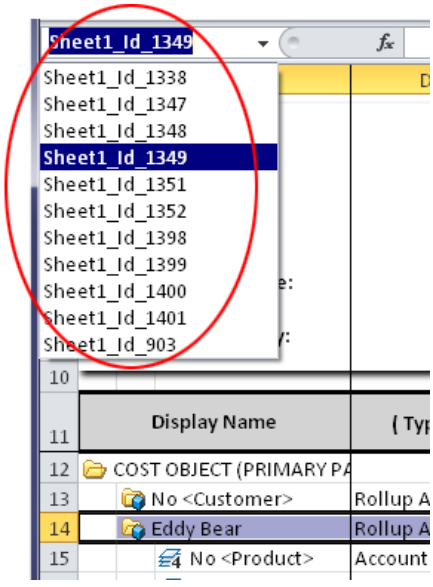
Decimal Places

The number of decimal places that are displayed in an exported Excel file are the number of decimal places that you specified in SAS Cost and Profitability Management by selecting **Tools** ⇨ **User Options**.



Row names

Every row in an exported Excel spreadsheet is named using the account ID. The format of the name is <*sheetname*>_ID_<*account ID*>. The name appears in the drop-down list located before the Excel formula bar, as shown in the following picture. If you select a name in the drop-down list, then the corresponding row is highlighted.



Checkboxes

Checkboxes in the module view are also displayed as checkboxes in the exported spreadsheet. However, the checkboxes in the exported spreadsheet only reflect what is in the module view. They cannot be edited.

Checkboxes cannot be edited

Display Name	(Type)	Display Reference	Cost ($)	Fixed
Engineering	Rollup Account	Dept 4000	32,400.00	
Depre Facilities	Account	4000 Deprec Facilities	7,200.00	☑
Salary Engineer	Account	4000 Salary Engineer	19,200.00	
Tools Supplies	Account	4000 Tools Supplies	6,000.00	
Accounting	Rollup Account	Dept 5000	23,700.00	
Deprec Facilities	Account	5000 Deprec Facilities	4,500.00	☑
Salary Benefits	Account	5000 Salary Benefits	16,200.00	

Restrictions

Currency formatting

Because Microsoft Excel does not support all the currency formats that SAS Cost and Profitability Management supports, currency fields are formatted as numeric columns in an exported spreadsheet. The currency symbol is displayed in the column header of the exported spreadsheet instead of in each spreadsheet cell. After exporting a module view, you can use Microsoft Excel to format currency cells as you like.

The following picture shows a module view in which costs are displayed both in dollars (the base currency) and in Euros (converted from dollars).

Rate table:

	EUR	USD
per EUR	1.00	1.28
per USD	0.78	1.00

Example: €1.00 = $1.28

Display Name	Display Reference	Cost in Euros	Cost in Dollars
ACTIVITY (PRIMARY PANE)		€3,299,422.61	$4,217,592.50
USA	USA	€3,299,422.61	$4,217,592.50
California	California	€2,290,647.55	$2,928,093.50
Los Angeles	Los Angeles	€1,155,154.74	$1,476,613.50
Customer Service	Customer Service	€75,854.55	$96,963.50
Parcel Delivery	Parcel Delivery	€1,079,300.20	$1,379,650.00
Parcel Handling	Parcel Handling	€499,046.38	$637,922.00
Oakland	Oakland	€1,135,492.80	$1,451,480.00
Oregon	Oregon	€1,008,775.07	$1,289,499.00

The following picture shows the resulting exported spreadsheet. Notice that the currency symbol is displayed in the spreadsheet column header, and the formula used for conversion is displayed in the formula area.

Notice the formula

H15 =PRODUCT(1476613.5, 0.7823)

Currency symbol in column header

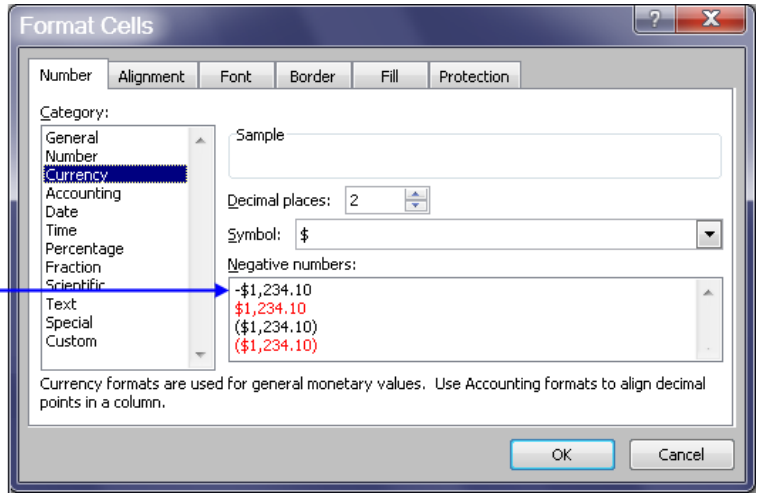
Model Name:	P1 (M1001)
Period/Scenario:	2008 Q1/Actual
Module Name:	ACTIVITY (PRIMARY PANE)
Column Layout Name:	Default
Model Base Currency:	US Dollar (USD ($))

Display Name	(Type)	Display Reference	Cost in Euros (€)	Cost in Dollars ($)	(Item View Order)
Los Angeles	Rollup Account	Los Angeles	1,155,154.74	1,476,613.50	4
Customer Service	Rollup Account	Customer Service	75,854.55	96,963.50	5
Parcel Delivery	Rollup Account	Parcel Delivery	1,079,300.20	1,379,650.00	6
Parcel Handling	Rollup Account	Parcel Handling	499,046.38	637,922.00	7
Oakland	Rollup Account	Oakland	1,135,492.80	1,451,480.00	8
Oregon	Rollup Account	Oregon	1,008,775.07	1,289,499.00	9

Negative numbers

Regardless of how negative numbers are formatted in a module view, they are preceded by a minus sign (-) and displayed in black when exported to Microsoft Excel. This is because the negative number formatting in the SAS Cost and Profitability Management module view might not be available in Microsoft Excel depending upon Excel's national language settings. Once the Excel file has been created, you can format negative numbers in it any way that you like.

Export uses this format for negative numbers



Chapter 81

OLAP Views

Import OLAP Views

To import OLAP views:

1. Go to the Workspace Manager.
2. Select **File** ⇨ **Import OLAP Views**. Step 1 of the Import OLAP Views dialog box opens.
3. Select the XML file (containing the views) to be imported.
4. Specify the model to associate with the views being imported.

   Note: Although a model was specified when the OLAP views were exported, you can associate the views with a different model on import.

   Rename OLAP view being imported
   If an OLAP view already exists with the same name as an OLAP view being imported, the one being imported is renamed. This applies to every OLAP view being imported if multiple views are imported.

   Replace the existing OLAP view
   If an OLAP view already exists with the same name as an OLAP view being imported, then the existing one is replaced with the imported one.

   Do not import duplicate OLAP views
   If an OLAP view already exists with the same name as an OLAP view being imported, then don't import the duplicate view. Leave the existing one in place.
5. Click **Next**. Step 2 of the Import OLAP Views dialog box opens.
6. For each OLAP view being imported, select a cube from the **Cube** drop-down list to associate with that view.

   The **Cube** drop-down list shows all the cubes that have been previously generated (on the import-to machine) for the model selected in Step 1.

If you are importing multiple views, you can choose to bypass importing individual ones.

7. Click **Finish**. The select views are imported.

See Also

"Export OLAP Views" on page 692

Export OLAP Views

To export OLAP views to an XML file:

1. Go to the Workspace Manager.
2. Select **File** ⇨ **Export OLAP Views**. The Export OLAP Views dialog box opens.
3. Select one or more OLAP views to be exported. You can export any view to which you have access.

   Automatically check children
   Selecting this causes cube configurations inside a folder to be recursively selected when you select the folder.

   Note: This only applies to **subsequent** folder selections. It does not select the children of folders that are **already** selected.

   Include folders
   Selecting this causes folder information to be stored in the export file so that the folders can be recreated when you subsequently import the cube configurations.

   Note: This option does not create folders in the export directory. It only stores the folder information in the export file so that the folders can be recreated later during import.

   Export to this path and XML file name
   Select the path and name of the export file.

   Note: Only one export file is created even if you select multiple folders and cube configurations to be exported.

4. Click **Finish**.

See Also

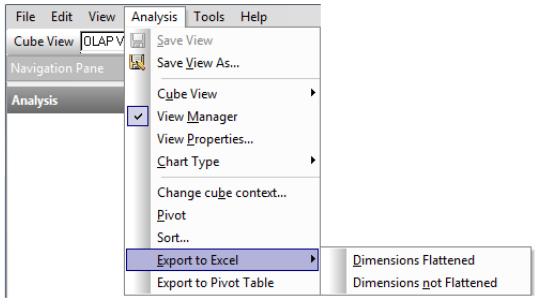
- "Import OLAP Views" on page 691
- "Export to Excel" on page 692

Export to Excel

To export an OLAP View to Excel, do the following:

1. Go to the **Analysis** workspace.

2. Open an OLAP view.
3. Select **Analysis** ⇨ **Export to Excel**



4. Choose either of the following:

   Dimensions Flattened
   The Excel table has more columns and fewer rows. The table has a single row of column headers, with one level per column.

   Dimensions not Flattened
   The Excel table has fewer columns and more rows. Each dimension has its own row.

The following picture shows an example of an Excel with flattened dimensions and without flattened dimensions.

Dimensions Flattened

(more columns)

	A	B	C	D
1	Level1	Level1	Level2	01JAN2008:00:00:00 - Cost
2				
3	All	All		.
4	All	None		.
5	All	USA	USA (direct)	.
6	All	USA	California	.
7	All	USA	Oregon	.
8	All	USA		.
9	None	All		.
10	None	None		47,114.10
11	None	USA	USA (direct)	.
12	None	USA	California	.
13	None	USA	Oregon	.
14	None	USA		.
15	Equipment Expenses	All		.
16	Equipment Expenses	None		.
17	Equipment Expenses	USA	USA (direct)	.
18	Equipment Expenses	USA	California	311,500.00
19	Equipment Expenses	USA	Oregon	194,300.00
20	Equipment Expenses	USA		505,800.00
21	Operating Expenses	All		.
22	Operating Expenses	None		.
23	Operating Expenses	USA	USA (direct)	.
24	Operating Expenses	USA	California	1,798,000.00
25	Operating Expenses	USA	Oregon	956,800.00
26	Operating Expenses	USA		2,754,800.00
27	Wages	All		.
28	Wages	None		.
29	Wages	USA	USA (direct)	.
30	Wages	USA	California	3,561,360.59
31	Wages	USA	Oregon	1,302,199.19
32	Wages	USA		4,863,559.77

Dimensions not Flattened

(fewer columns)

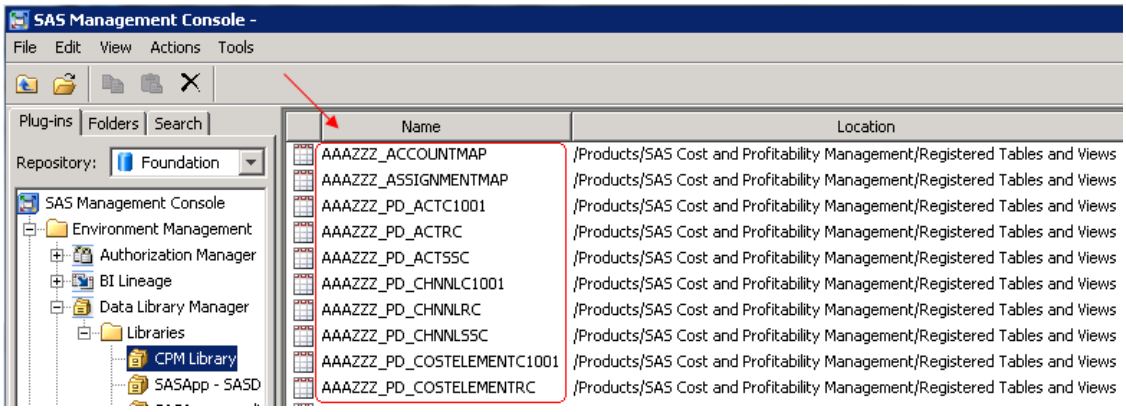
	A	B	C
1		All_Period	01JAN2008:00:00:00
2		Measures	Cost
3	Resource_General_Ledger	Resource_Region	
4	All	All	
5	All	None	
6	All	USA (direct)	
7	All	California	
8	All	Oregon	
9	All	USA	
10	None	All	
11	None	None	47,114.10
12	None	USA (direct)	
13	None	California	
14	None	Oregon	
15	None	USA	
16	Equipment Expenses	All	
17	Equipment Expenses	None	
18	Equipment Expenses	USA (direct)	
19	Equipment Expenses	California	311,500.00
20	Equipment Expenses	Oregon	194,300.00
21	Equipment Expenses	USA	505,800.00
22	Operating Expenses	All	
23	Operating Expenses	None	
24	Operating Expenses	USA (direct)	
25	Operating Expenses	California	1,798,000.00
26	Operating Expenses	Oregon	956,800.00
27	Operating Expenses	USA	2,754,800.00
28	Wages	All	
29	Wages	None	
30	Wages	USA (direct)	
31	Wages	California	3,561,360.59
32	Wages	Oregon	1,302,199.19
33	Wages	USA	4,863,559.77

Chapter 82

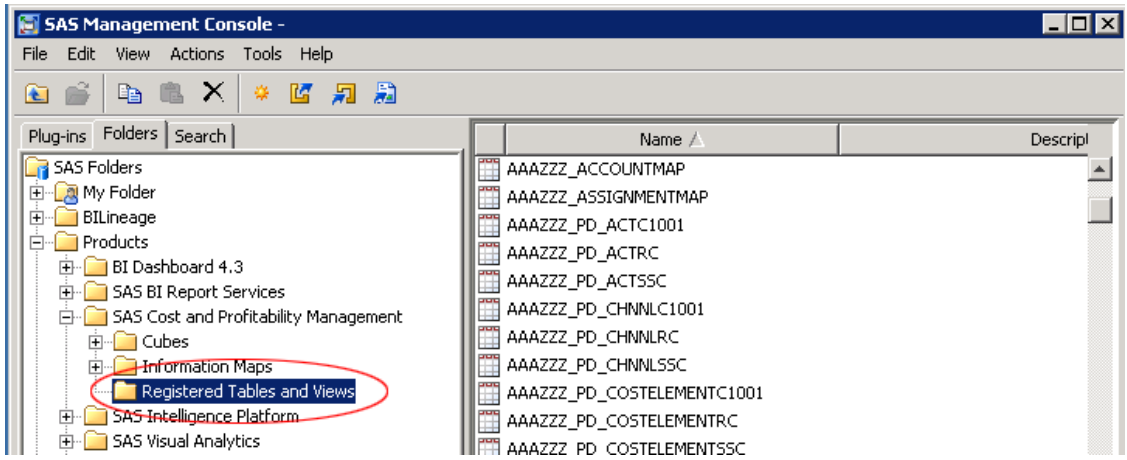
Export Registered Tables

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 696.

Model-independent views
These views are exported and registered automatically. See "Model-Independent Views" on page 698.

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 698.

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 701.

Table Name	Description
*ModelRef*_ACCOUNTMAP	A join of *ModelRef*_PV_ACCOUNT with other registered tables. Includes details for multiple periods, dimensional signature, and all numeric properties. See "modelRef_ACCOUNTMAP " in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_ASSIGNMENTMAP	A join of *ModelRef*_PV_ASSIGNMENT with other registered tables. Includes details for source and destination dimensional signatures, and all entered and calculated properties. See "modelRef_ASSIGNMENTMAP" in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

Table Name	Description
*ModelRef*_PV_ACCOUNT	Details for multiple periods, dimensional signature, and all numeric properties. See “modelRef_PV_ACCOUNT ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ASSIGNMENT	Details for source and destination dimensional signatures, and all entered and calculated properties. See “modelRef_PV_ASSIGNMENT ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ASSIGNMENTEXT	Details for source and destination dimensional signatures, and all entered and calculated properties. See “modelRef_PV_ASSIGNMENTEXT ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ATTRIBUTE	Details defining the attribute, attribute types, formulas for calculations, and default values. See “modelRef_PV_ATTRIBUTE” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ATTRIBUTEVALUE	Attribute attachment to the model accounts and the numeric value for the attribute. See “modelRef_PV_ATTRIBUTEVALUE ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_DIMENSION	Details defining the model's dimensions. See “modelRef_PV_DIMENSION ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_DRIVER	Details defining the drivers, driver types, and formulas for rules assignment and for calculations. See “modelRef_PV_DRIVER ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PV_ENTEREDCE	Details of all entered cost elements, with their corresponding account attachment and values. See “modelRef_PV_ENTEREDCE ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

Model-Independent Views

The following model-independent views are created automatically.

View Name	Description
PUBLICMODEL	Contains a list of models. See “PUBLICMODEL ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICMODELSTATUS	For each model, lists its period scenario associations and whether each one is calculated or not.. See “PUBLICMODELSTATUS” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICPERIOD	Contains a list of periods with their start and end dates. See “PUBLICPERIOD ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
PUBLICSCENARIO	Contains a list of scenarios. See “PUBLICSCENARIO ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

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
*ModelRef*_PD_COSTELEMENT<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> 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 “modelRef_PD_COSTELEMENT<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

View Name	Description
*ModelRef*_PD_*dimShortRef*<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_REGIONRC ABC_PD_REGIONSSC ABC_PD_REGIONC1001	Single dimension-member details: level by level, noting ID, Reference, and Name. See “modelRef_PD_dimShortRef<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_DRIVER<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_DRIVERRC ABC_PD_DRIVERSSC ABC_PD_DRIVERC1001	Driver ID and corresponding Driver Name. See “modelRef_PD_DRIVER<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_MODULE<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_MODULERC ABC_PD_MODULESSC ABC_PD_MODULEC1001	Details defining the types of modules. See “modelRef_PD_MODULE<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_PERIOD<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_PERIODRC ABC_PD_PERIODSSC ABC_PD_PERIODC1001	Details defining the periodic hierarchy. See “modelRef_PD_PERIOD<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.

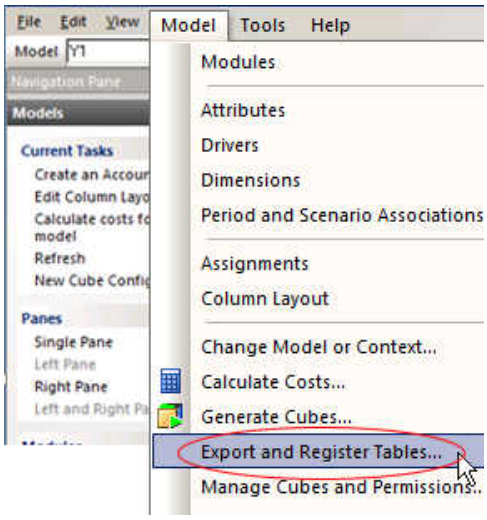
View Name	Description
*ModelRef*_PD_SCENARIO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_SCENARIORC ABC_PD_SCENARIOSSC ABC_PD_SCENARIOC1001	Details defining the types of available scenarios. See “modelRef_PD_SCENARIO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PD_YESNO<**suffix**> where <**suffix**> is RC, SSC, or C<**cube configuration ID**> for a multi-stage contributions cube. For example: ABC_PD_YESNORC ABC_PD_YESNOSSC ABC_PD_YESNOC1001	Dimensional definition for Boolean values: Text strings. See “modelRef_PD_YESNO<suffix> ” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide*.
*ModelRef*_PF_<**suffix**> where <**suffix**> is RC, SSC, or MSC_C<cubeID> for a multi-stage contributions cube. For example: ABC_PF_RC ABC_PF_SSC ABC_PF_MSC_C1001 If you have checked **Also load tables into a library for the SAS LASR Analytic Server**, 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: *ModelRef*_PF_<**suffix**> where <**suffix**> is RCSTAR, SSCSTAR, or MSC_C<cubeID>STAR for a multi-stage contributions cube. For example: ABC_PF_RCSTAR ABC_PF_SSCSTAR ABC_PF_MSC_C1001STAR	Fact Table: Stages and member IDs for each step through contribution. Source table for cube generation. See “modelRef_PF_<suffix>” in Chapter 16 of *SAS Cost and Profitability Management: Data Administration Guide* .

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:

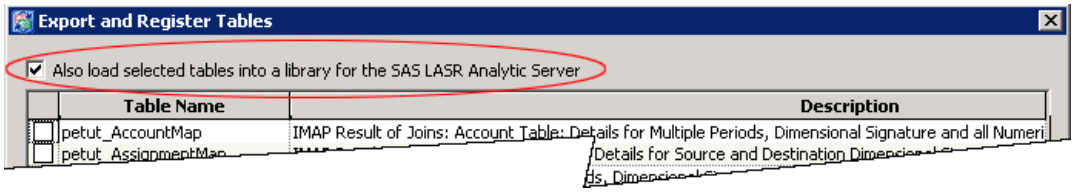
1. Select **Model** ⇨ **Export and Register Tables**.



The Export and Register Tables dialog box opens.

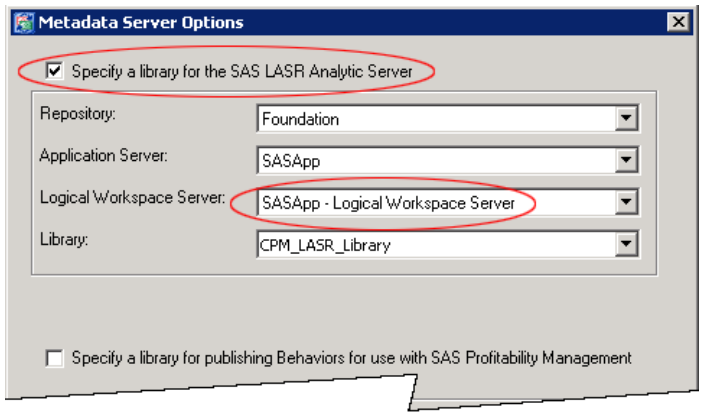
2. Select whether to: **Also load selected tables into a library for the SAS LASR Analytic Server**.

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



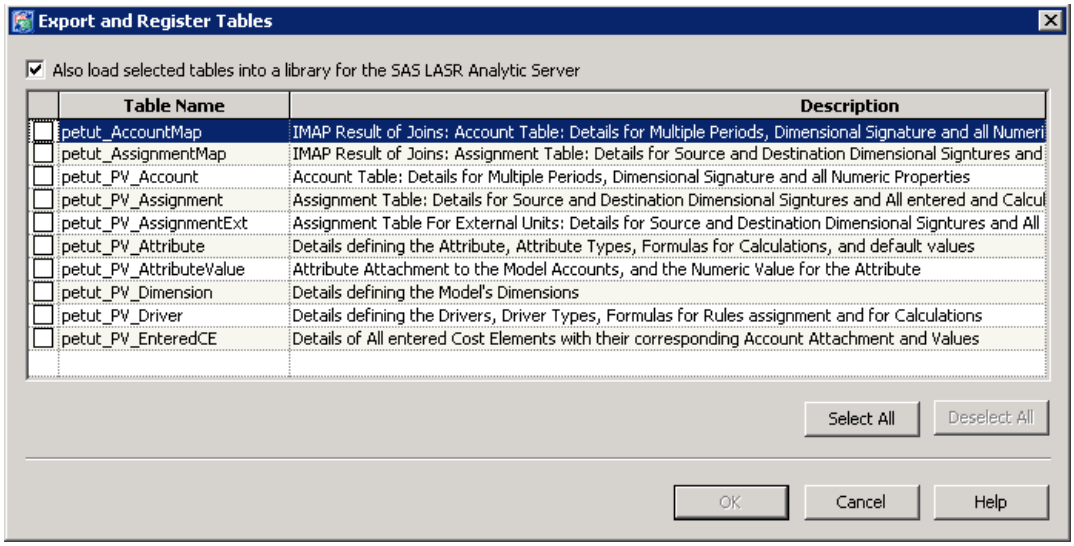
Note:

- 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" on page 504.
- 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" in Chapter 23 of *SAS Cost and Profitability Management: Data Administration Guide*.

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 698.
- 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 698. 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" on page 508.

Chapter 83

Import and Export with Easy API

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
- Export model data as a report

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 704

   The XML file describes the operation to be performed.

2. "Save Easy API Commands in a Text File" on page 705

   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 708

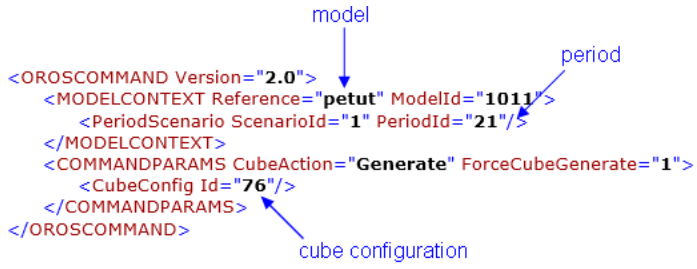
   Easy API uses the text file to run its commands.

See also

For more information, see Chapter 21, "Easy API," in *SAS Cost and Profitability Management: Data Administration Guide*.

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.



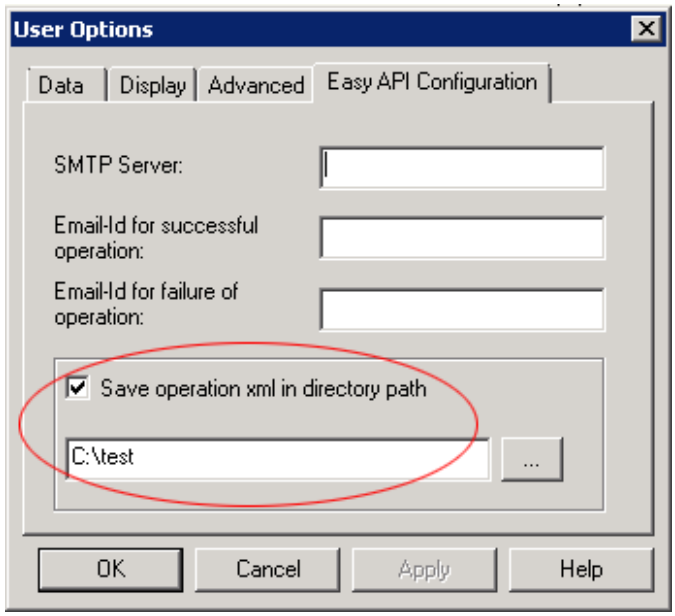

```
<OROSCOMMAND Version="2.0">
    <MODELCONTEXT Reference="petut" ModelId="1011">
        <PeriodScenario ScenarioId="1" PeriodId="21"/>
    </MODELCONTEXT>
    <COMMANDPARAMS CubeAction="Generate" ForceCubeGenerate="1">
        <CubeConfig Id="76"/>
    </COMMANDPARAMS>
</OROSCOMMAND>
```

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 20, “XML Passed to the API,” in *SAS Cost and Profitability Management: Data Administration Guide*.

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
acquiremodellock "read | write" "*modelRef*"	Acquires a Read or Write lock on the model. *Note:* If you do not issue acquiremodellock 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.
releasemodellock "*modelRef*"	Releases whatever lock (Read or Write) that is on the model. *Note:* If you do not issue releasemodellock, 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.
terminatewritelock “*modelRef*”	Terminates a Write lock that is acquired by a modeler on the model.
calculate "<*folder_path*>\your.xml"	Calculates and/or generates a cube. *Note:* 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"	Copies model data and pastes it as a new model.
copyperiod "<*folder_path*>\your.xml"	Copies model data from one period/scenario to another
export "<*folder_path*>\your.xml"	Exports model data.
export cube "<*folder_path*>\your.xml"	Exports cube configurations. The cube configuration name and cube configuration reference that appear in the XML file to export the cube configuration must adhere to the following guidelines: • Do not use the following characters in the cube configuration name: ' (single quotation mark), " (quotation mark), / (slash), \ (backslash), or | (pipe). • Do not use the following characters in the cube configuration characters: (space), . (period), , (comma), ; (semicolon), ' (single quotation mark), ` (grave accent), : (colon), ? (question mark), * (asterisk), & (ampersand), % (percent sign), $ (dollar sign), ! (exclamation point), - (en dash), + (plus sign), = (equal sign), ((opening parenthesis),) (closing parenthesis), [] (brackets), {} (braces), / (slash), \ (backslash), | (pipe), @ (at sign), # (number or pound sign), ^ (caret), " (quotation mark), or <> (angel brackets).
export and register tables "<*folder_path*>\your.xml"	Exports database tables that are registered in SAS Management Console for use with other SAS or non SAS programs. See Chapter 82, "Export Registered Tables," on page 695.
import "<*folder_path*>\your.xml"	Imports model data.
import cube "<*folder_path*>\your.xml"	Imports cube configurations. The cube configuration name and cube configuration reference that appear in the XML file to import the cube configuration must adhere to the following guidelines: • Do not use the following characters in the cube configuration name: ' (single quotation mark), " (quotation mark), / (slash), \ (backslash), or | (pipe). • Do not use the following characters in the cube configuration characters: (space), . (period), , (comma), ; (semicolon), ' (single quotation mark), ` (grave accent), : (colon), ? (question mark), * (asterisk), & (ampersand), % (percent sign), $ (dollar sign), ! (exclamation point), - (en dash), + (plus sign), = (equal sign), ((opening parenthesis),) (closing parenthesis), [] (brackets), {} (braces), / (slash), \ (backslash), | (pipe), @ (at sign), # (number or pound sign), ^ (caret), " (quotation mark), or <> (angel brackets).

Command and sample argument	What it does
export report "<*folder_path*>\your.xml"	Exports model data as a report. The XML file contains the required information (such as the name of the model, the dimensions, the type of report template, the periods and scenarios, and the database) to export the model data as a report. Suppose that you are exporting the model data as a report using the EasyAPI and a table with the same name already exists in the database. Then you must have exported the model data by selecting the **Replace the table if a table with the same name already exists in the database** check box in the Report Data wizard using the user interface of SAS Cost and Profitability Management at least once.
run "<*folder_path*>\your.xml"	Executes 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 ' (single quotation mark) or // (double slash) in front of the command line to comment out (skip) a particular Easy API command.

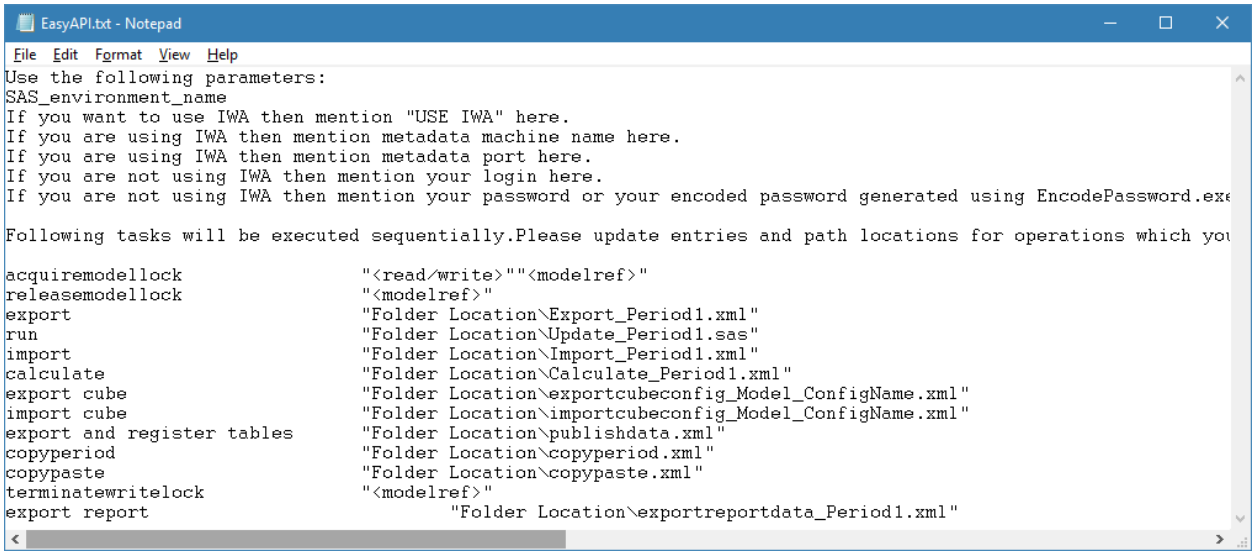
Note: Refer to the PeriodDefinition and ScenarioDefinition database tables to obtain the values of the PeriodId and ScenarioId tags to used in the XML file.

To issue an Easy API command, put it in a text file named EasyAPI.txt residing in the following directory:

```
<installation directory>SASCostandProfitabilityManagementClient\8.3\EasyAPI\
For example: C:\Program Files\SASHome\x86\SASCostandProfitabilityManagementClient\8.3\EasyAPI\
```

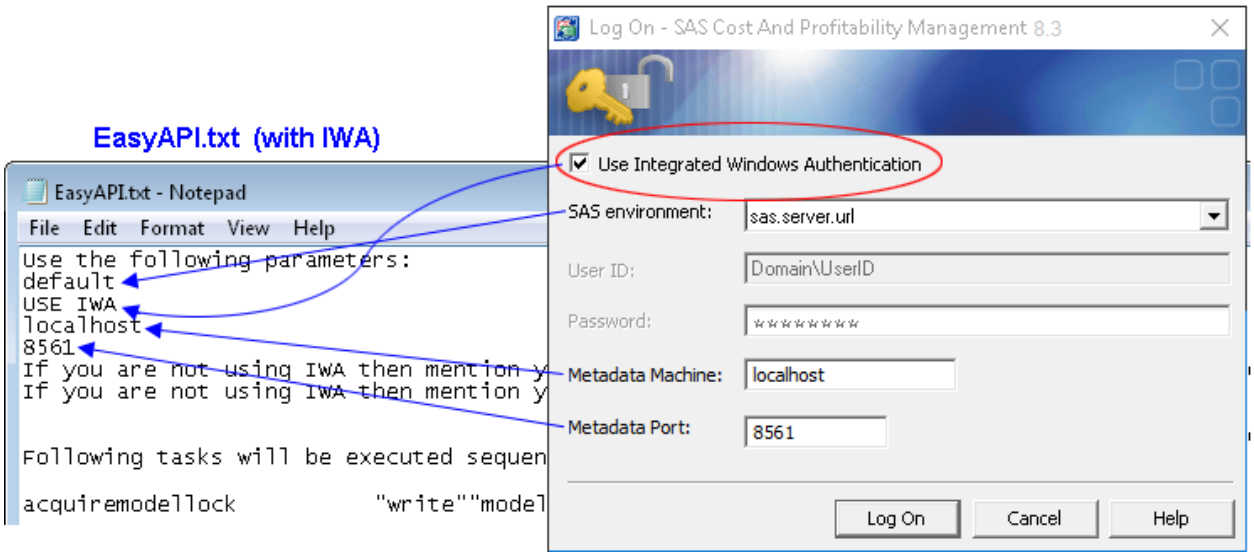
To open the EasyAPI.txt, select **Start** ⇨ **All Programs** ⇨ **SAS** ⇨ **SAS Cost and Profitability Management 8.3** ⇨ **EasyAPI** ⇨ **2. Specify parameters**.

The following picture shows EasyAPI.txt as it is appears on installation of SAS Cost and Profitability Management.

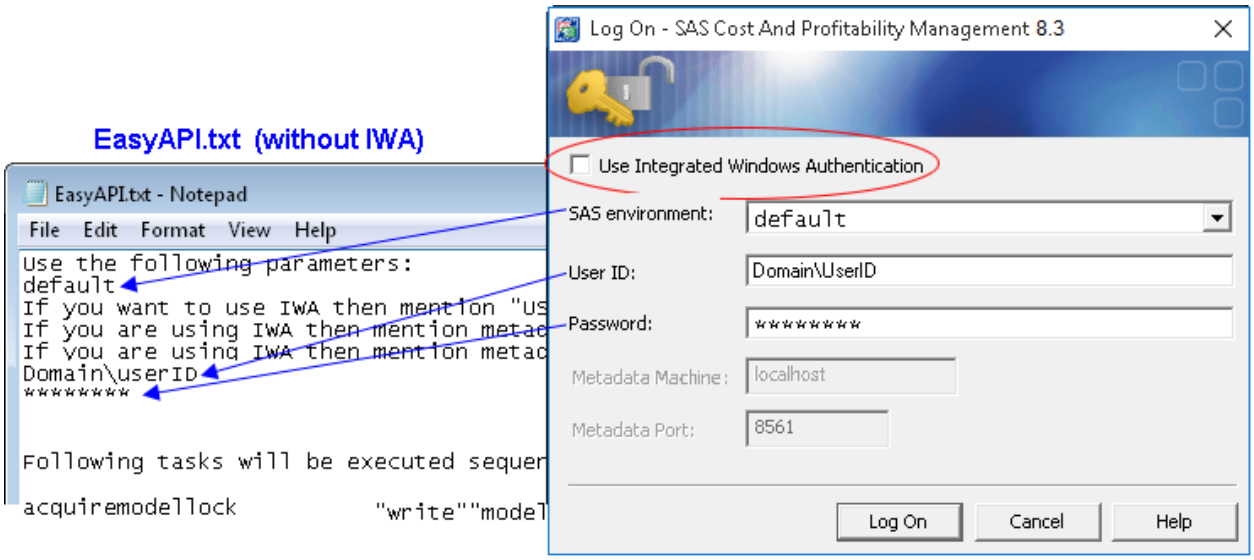


Make sure to include your login credentials in the txt file.

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

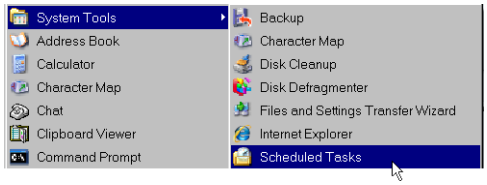
To invoke Easy API, select **Start** ⇨ **All Programs** ⇨ **SAS** ⇨ **SAS Cost and Profitability Management 8.3** ⇨ **EasyAPI** ⇨ **3. Run**.

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\8.3\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\8.3\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.

For More Information

See Chapter 21, “Easy API,” in *SAS Cost and Profitability Management: Data Administration Guide*.

Part 23

Reporting Model Data

Chapter 84

About Reports

Reports

You can export the model data for various periods and scenarios to generate the following types of reports:

- Destination Furthest
- Driver - Cost and Rate
- Multi-Level Contributions
- Resource Contribution
- Unassigned Costs
- Unit Cost
- Dimensional Attribute Cost

SAS Cost and Profitability Management contains a template for each type of report. SAS Cost and Profitability Management exports the model data in a table in the database. You can create a report to view, share, and analyze the data.

Report Templates

A report template specifies the layout of a report and the fields of data in a report (but not the report's data itself). When you create a report, you choose a report template.

SAS Cost and Profitability Management comes with several predefined report templates that provide set formats and permit great flexibility in the amount and type of data to include in a report. For information on the following templates, see “Working with Reports” in the *SAS Cost and Profitability Management: Data Administration Guide*, available from the Help menu or see a potentially more recent version at http://support.sas.com/documentation/onlinedoc/abm/.

Destination Furthest
Dimensional Attribute Cost
Multilevel Contributions
Driver - Cost and Rate
Resource Contributions
Unassigned Costs
Unit Cost

Chapter 85

Windows for Reports

Reports Folder

About the Reports Folder

In the Workspace Manager, the report configurations are available in the **Report Data Exports** folder. You can open a report configuration to export the model data. You can create a report configuration to save the new settings to export the model data.

Note: The availability of these features depends on your permissions.

How to Access the Reports Folder

After you log on to SAS Cost and Profitability Management, click **Workspace Manager**. The Workspace Manager displays a list of all folders, including the **Report Data Exports** folder.

Export Report Data to SAS Data Sets

To export report data to SAS data sets:

1. On the **Database Location** step of the Report Data wizard, select **Use alternate database** and then choose **Browse**.

The Connection Information dialog box appears.

For more information, see Step 7 on page 722.

2. In the **Database type** list, select **SAS datasets**.
3. Enter information in the following fields.

Table 85.1 *Description of Fields to Export Report Data to SAS Data Sets*

Field	Description
Folder path relative to the IOM server	The folder containing SAS data sets for import or export must be accessible by the SAS Integrated Object Model (IOM) server. • If the folder is on the same machine as the SAS IOM server, then you can specify the folder using a path such as **`c:\datasetexport`**. • If the folder is on a different machine than the SAS IOM server, then you must specify the machine name in specifying the path. For example, if the folder, **`datasetexport`**, is on machine **`xyz`**, then you might specify the path as **`\\xyz\datasetexport`**.
Advanced options	The advanced options that you might want to consider to connect to a SAS data set.

4. (Optional) Click **Test Connection** to test the database connection.
5. Click **OK**.

Export Report Data to the Microsoft SQL Server

To export report data to the Microsoft SQL server:

1. On the **Database Location** step of the Report Data wizard, select **Use alternate database** and then choose **Browse**.

   The Connection Information dialog box appears.

   For more information, see Step 7 on page 722.

2. In the **Database type** list, select **Microsoft SQL server**.
3. Enter information in the following fields.

Table 85.2 *Description of Fields to Export Report Data to the Microsoft SQL Server*

Field	Description
Host name	The machine name where the target database is installed.
Port number	The TCP port where the target database instance is running. If the database is on a Named Instance, then go to the SQL server configuration manager to check the TCP port on which it is running. Named Instances run on a dynamic TCP port.

Field	Description
Database name	A name for the database.
Username	The user name to connect to the database.
Password	The password to connect to the database.
Advanced options	Separate multiple options with a semicolon. For example: `driver_trace=all;driver_tracefile='c:\temp\trace_lower.log';traceflags=7;` *Note:* Use logging options only in case of failures where you want to provide internal driver logs to technical support. Otherwise, logging operations slow your operation.

4. (Optional) Click **Test Connection** to test the database connection.
5. Click **OK**.

Export Report Data to Oracle

To export report data to Oracle:

1. On the **Database Location** step of the Report Data wizard, select **Use alternate database** and then choose **Browse**.

   The Connection Information dialog box appears.

   For more information, see Step 7 on page 722.
2. In the **Database type** list, select **Oracle**.
3. Enter information in the following fields.

   Table 85.3 *Description of Fields to Export Report Data to Oracle*

Field	Description
Host name	The machine name where the target database is installed.
Port number	The TCP port where the target database instance is running.

Field	Description

Service name	You can find the values of **Host name** and **Service name** in the file **`tnsnames.ora`**.
	`ABMDSN2 =`
	`  (DESCRIPTION =`
	`  (ADDRESS_LIST =`
	`  (ADDRESS = (PROTOCOL = TCP)(HOST = abmblade1)(PORT = 1521))`
	`  )`
	`  (CONNECT_DATA =`
	`  (SERVICE_NAME = abmdsn)`
	`  )`
	`  )`

Username	The user name to connect to the database.
Password	The password to connect to the database.
Advanced options	The advanced options that you might want to consider while connecting to an Oracle database. You might want to open the **`sqlnet.ora`** file and verify that it has EZCONNECT specified under NAMES.DIRECTORY_PATH=(TNSNAMES, EZCONNECT). This file is at the same location as **`tnsnames.ora`**. For example: `C:\Oracle\product\11.1.0\client_1\network\admin\sqlnet.ora`

4. (Optional) Click **Test Connection** to test the database connection.
5. Click **OK**.

Export Report Data to PostgreSQL

To export report data to PostgreSQL:

1. On the **Database Location** step of the Report Data wizard, select **Use alternate database** and then choose **Browse**.

   The Connection Information dialog box appears.

   For more information, see Step 7 on page 722.
2. In the **Database type** list, select **PostgreSQL**.
3. Enter information in the following fields.

   Table 85.4 *Description of Fields to Export Report Data to PostgreSQL*

Field	Description
Host name	The host name.
Port number	The port number.

Field	Description
Database name	A name for the database.
Username	The user name to connect to the database.
Password	The password to connect to the database.
Advanced options	The advanced options that you might want to consider while connecting to a PostgreSQL database.

4. (Optional) Click **Test Connection** to test the database connection.
5. Click **OK**.

Export Report Data to Another Database

To export report data to another database:

1. On the **Database Location** step of the Report Data wizard, select **Use alternate database** and then choose **Browse**.

   The Connection Information dialog box appears.

   Step 7 on page 722For more information, see .
2. In the **Database type** list, select **Other**.
3. Enter information in the following fields.

   Table 85.5 *Description of Fields to Export Report Data to Another Database*

Field	Description
Host name	The host name.
ODBC DSN	Select from the drop-down list. The list includes all the System Open Database Connectivity (ODBC) data source name (DSN) that users have created on the SAS IOM server. *Note:* • It is recommended that you create a System ODBC DSN on the SAS IOM server machine and use it in the connection details. • A DSN for Postgres should be created with the Unicode version of the Postgres driver (PostgreSQL Unicode(x64)).
Username	The user name to connect to the database.
Password	The password to connect to the database.

Field	Description
Advanced options	By default, the catalog name is *Staging*. You can specify any arbitrary name for CATALOG= because it is a logical name that is used internally by the system. Most databases require a catalog name to be specified in the connection URL but not all. For example, the SQL server does not require a catalog name. If you attempt to connect to a database that does not require a catalog name, then you must remove it from the **Advanced options** field or the connection will fail. Click **Edit** to modify the connection URL.
Connection URL	When you click **Test Connection**, the system takes the information that you enter and constructs a connection URL that it displays in the **Connection URL** field. Click **Edit** to modify the connection URL. Following is a sample connection URL for a connection based on ODBC DSN: `DRIVER=ODBC;ODBC_DSN=DSNMySQL;UID=root;PWD=Pass99;CATALOG=Staging;` If you want to use an ODBC driver to connect to the target database, then the system creates a connection URL (which you can modify) for you. If you want to use a driver other than ODBC, then click **Edit** to enter the URL yourself.

4. (Optional) Click **Test Connection** to test the database connection.
5. Click **OK**.

Chapter 86

How To

Create a Report to Export Model Data

Creating a report exports data from model data to a database. Modelers (with Read or Read/Write access to a model) and viewers (with Read access to a model) can create a report to export the data for the model. You must also need permission to the **`SAS Folders\Products\SAS Cost and Profitability Management \Registered Tables and Views`** directory in SAS Management Console to register the report tables.

To create resource contributions and destination furthest reports, you must have already generated the RC fact table for the model.

Perform the following steps to create a report:

1. Select **File** ⇨ **Export** ⇨ **Report Data**.

   The Report Data Wizard appears.

2. On the **Select Type** step, select a report template and click **Next**.

   For information about what data each report template contains, see "Working with Reports" in the *SAS Cost and Profitability Management: Data Administration Guide*, available from the Help menu or from http://support.sas.com/documentation/onlinedoc/abm/.

3. On the **Select Model** step, select a model to export its data and click **Next**.

   Note: You must have required Read or Read/Write permission to the model.

4. On the **Select Period/Scenario** step, select the periods and scenarios for which you want to export the report data, and click **Next**.

5. On the **Select Module** step, select a module for which you want to export the data, and click **Next**.

   To include data from all the modules, select the **Use all modules** option.

6. On the **Select Dimensional Attributes** step, select the dimensional attributes, and click **Next**.

   Note: The check boxes of this step are available when you selected the Dimensional Attribute Cost report.

7. On the **Database Location** step, use the following options to export the report data to a database:

   Table 86.1 *Description of Fields of the Database Location Step*

Field	Description
Select Database Location	
Use default database	Select this option to export the report data to the default database that SAS Cost and Profitability Management uses.
Use alternate database	Select this option to export the report data to a database other than the default database.
Browse to set up an alternate database connection	Click **Browse** to set up an alternate database connection to export the report data. After you click this button, the Connection Information dialog box appears. This button is available only when you select the **Use alternate database** option.
Destination table name	Enter a name for the database table. The table name must include alphanumeric characters. The only special character allowed is the underscore. The table name must not be more than 22 characters. SAS Cost and Profitability Management displays the letter R followed by the model ID and underscore before the field. This displayed information is prefixed to the table name that you enter in this field. In other words, SAS Cost and Profitability Management creates a table with a name in the following format: **R<model_ID>_<user–specified name>**. For example, suppose that the model ID is 1234 and the table name that you enter is FY2018. Then, the **R1234_FY2018** table is created.
Load table into a library for SAS LASR Analytic Server	Click this check box to load the report data into a library for the SAS LASR Analytic Server. This check box is available when the SAS LASR Analytic Server is configured and you are exporting the report data to the default database.
Replace the table if a table with the same name already exists in the database	Select this check box to replace the table with the same name that is specified in the **Destination table name** field if it already exists in the database.

8. Click **Next**.
9. Review the summary.

10. (Optional) If you need to change any information, click **Back** until you reach the page that you need to change in the wizard.
11. (Optional) To save the report configuration so that the report can be easily run again, do the following:
    a. Select the **Save configuration as** option.
    b. Enter the **Name**.
    c. Enter the **Description**.
12. (Optional) To save the report configuration so that the report can be easily run without exporting the report data immediately, do the following:
    a. Select the **Save without running** option.
    b. Enter the new **Name**.
    c. Enter the **Description**.
13. Click **Finish**.

Use a Report Configuration to Export Report Data

You can use an already saved report configuration to easily export the report data. The configuration contains the following information:

- report template
- model
- periods and scenario associations
- module and dimension attributes
- information about the database connection

Note: Suppose that you created a report and its configuration definitions in SAS Activity-Based Management 7.1 and SAS Activity-Based Management 7.2 and you migrated to SAS Cost and Profitability Management 8.3. In such a scenario, the report configurations that were created in SAS Activity-Based Management 7.1 and SAS Activity-Based Management 7.2 are not migrated to SAS Cost and Profitability Management 8.3. You must create the report configurations again in SAS Cost and Profitability Management 8.3.

To use the report configuration to export the report data:

1. In the **Workspace Manager**, open the **Report Data Exports** folder.
2. Right-click a report configuration and click **Open**. The Report Data wizard appears and contains all the report-related information already selected in it.
3. (Optional) Change the report configuration information to meet any business requirements.

   You might want to change the name of the report table to avoid replacing the existing report table.
4. Click **Finish**.

Delete a Report Configuration

1. In the **Workspace Manager**, open the **Report Data Exports** folder.
2. Select a report configuration.
3. Click **Delete**.

   Note: Depending on your permissions, you may not have the ability to delete a report configuration.

Part 24

Publishing

Chapter 87

Publish Period/Scenario Associations for a Model

Publish a Period/Scenario Association for a Model

Overview

Publishing a period/scenario association:

- makes it available for viewing by anyone with View Model capability
- is necessary for the period/scenario association to be used for what-if analysis. See "Create a What-If Analysis" in Chapter 2 of *SAS Cost and Profitability Management: What-If Analysis*.

If a period/scenario association is published, then you can not:

- modify any periodic element belonging to that period/scenario association.

If a period/scenario association is published, then you can:

- modify periodic elements belonging to other period/scenario associations in the model
- modify any structural element belonging to the model.

See "Periodic and Structural Elements" on page 79.

How to Publish

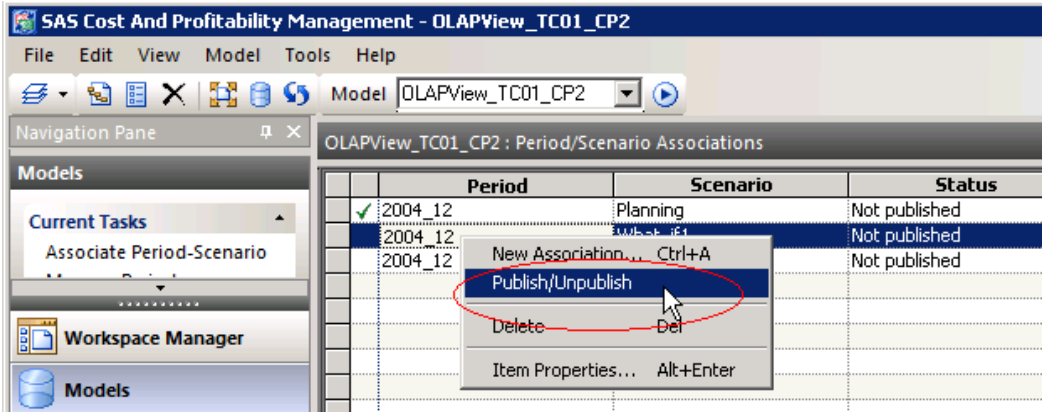
To publish a period/scenario association:

1. Select **Model** ⇨ **Period and Scenario Associations**.

   The Period and scenario associations page appears.

2. Select a period/scenario association.
3. Select **Edit** ⇨ **Publish/Unpublish**.

   Or, right-click a period/scenario association, and select **Publish/Unpublish**.



See Also

"Period and Scenario Associations View" on page 163

Chapter 88

Publish Behaviors to SAS Profitability Management

Overview

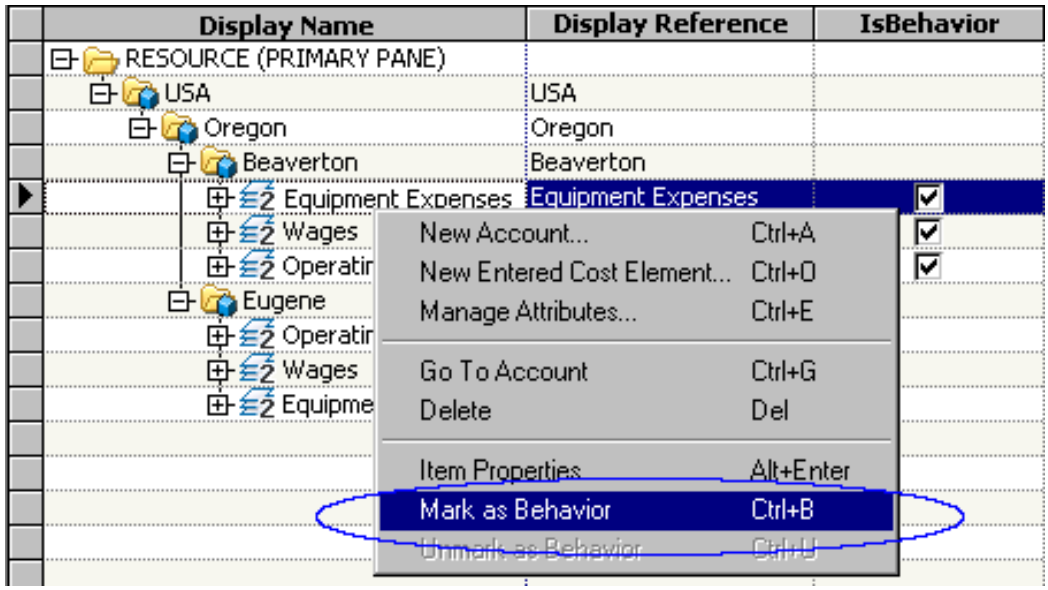
For complete information on working with SAS Profitability Management, see , see Chapter 25, “Working with SAS Profitability Management,” in *SAS Cost and Profitability Management: Data Administration Guide*.

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.



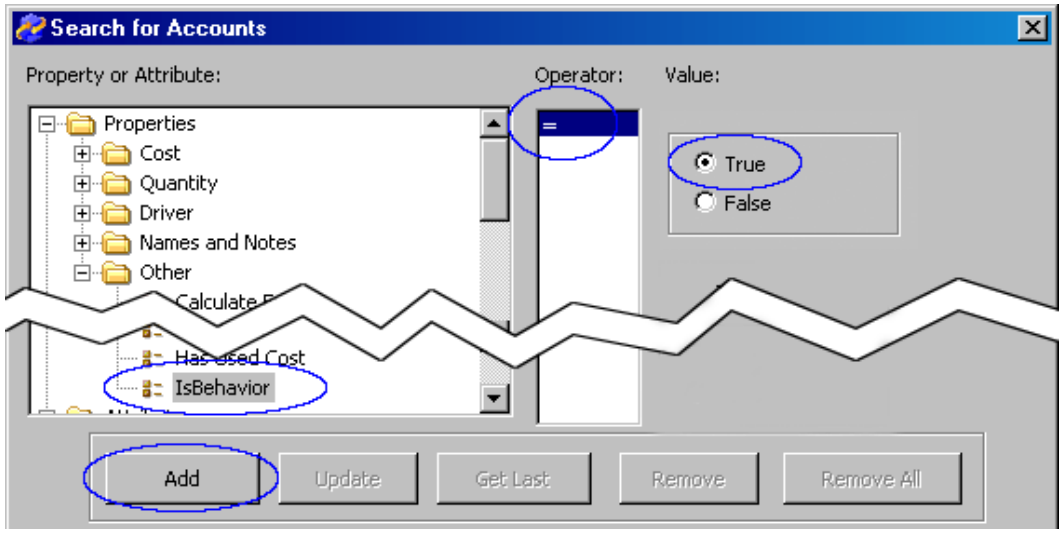
Mark Multiple Accounts

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.

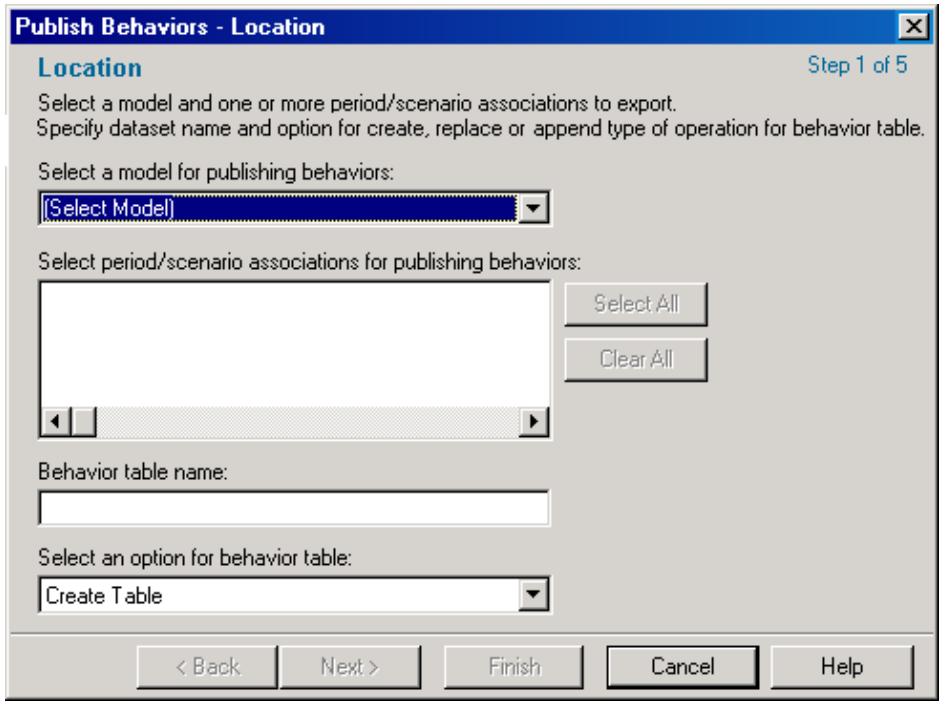


Publish Behaviors to SAS Profitability Management

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.



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.

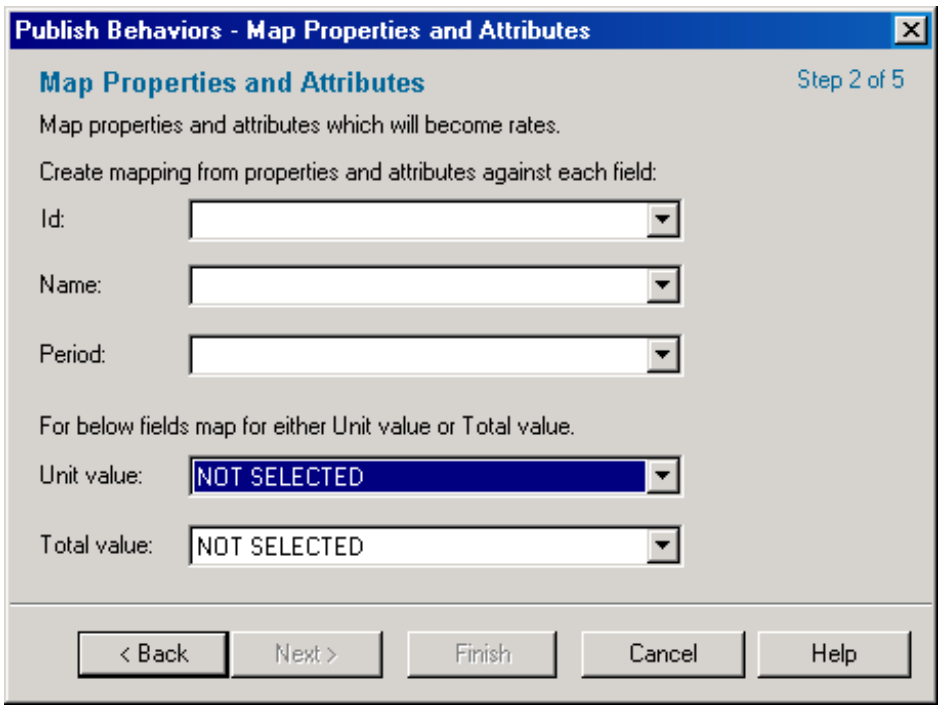
Behavior table name
Specify a name for the behavior table.

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.



Id
is the identifying reference for the behavior

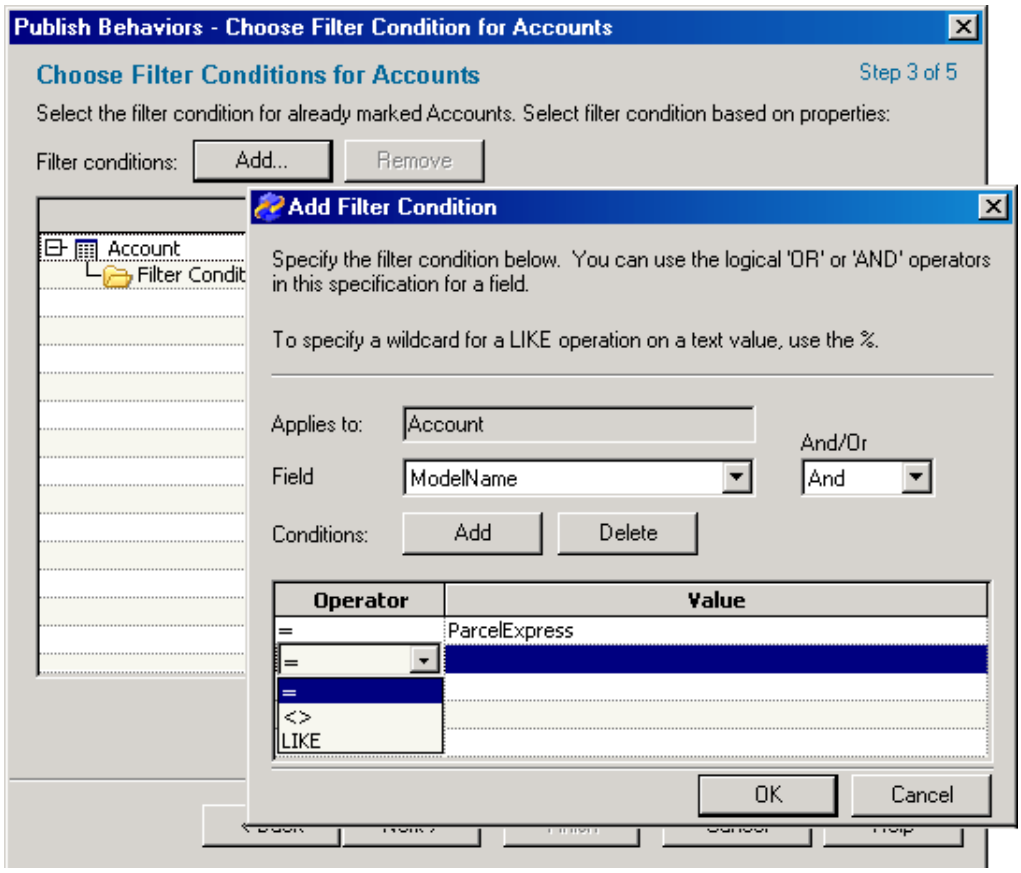
Name
is the name of the behavior

Period
defines the period for the costs

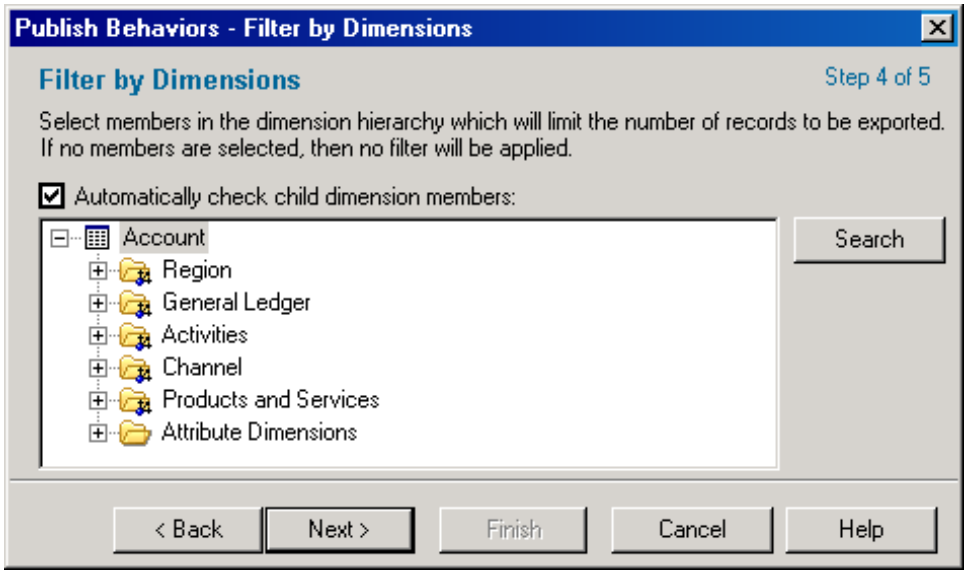
Unit value
is the unit cost for each transaction with this source. If you select a Unit value, then you may not select a Total value.

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 729
- Chapter 25, "Working with SAS Profitability Management," in *SAS Cost and Profitability Management: Data Administration Guide*

Part 25

Customizing the Interface

Chapter 89

User Options

Customizing the Interface: User Options

Overview

You can customize the SAS Cost and Profitability Management user interface to fit your needs. User interface settings change the interface for you only, not for all users. They do not change any data in a model. These settings are different from model properties.

Language Shown in the Interface

If multiple languages are installed on your computer, the language that is used is determined by your location setting in Windows (select **Settings** ⇨ **Control Panel** ⇨ **Regional and Language Options**).

Note: Error messages are generated by SAS Cost and Profitability Management and by the SAS Cost and Profitability Management server. Therefore, you might see error messages both in the language that you selected and in the language that is installed on the SAS Cost and Profitability Management server.

Default Display Precision for New Columns

Overview

You can specify the default number of decimal places that are displayed for costs, driver quantities, and rates. When you add a column on the Model view, the display precision that you set in your user options is used by default.

The display precision option does not affect current columns. It does not change the precision of the underlying data. And, it does not affect cubes.

These settings are saved on each computer. Therefore, any changes that you make to the default display precision affect only columns that you add on the computer on which you changed the default display precision. The changes affect all models.

You can format an individual column to change its display precision.

Default Precision for Cost Values

For the following properties, the default display precision for cost values determines the default display precision for new columns that have the Currency type:

```
Allocated Cost
Assigned Cost
Assigned Idle Cost
Assigned Non-reciprocal Cost
Assigned Reciprocal Cost
Cost
Drivable Cost
Driven Cost
Driver Allocated Cost
Driver Cost
Driver Driven Cost
Driver Idle Cost
Driver Percentage
Driver Used Cost
Entered Cost
Idle Cost
Idle Percentage
Profit
Received Allocated Cost
Received Assignment Cost
Received Cost
Received Driven Cost
Received Idle Cost
Received Non-reciprocal Cost
Received Reciprocal Cost
Received Used Cost
Revenue
Unassigned Cost
Used Cost
User Entered Cost Allocation
```

Default Precision for Numeric Values

For the following properties, the default display precision for numeric values determines the default display precision for new columns that have the Number type:

```
Assigned Idle Quantity
Driven Quantity
Driver Driven Quantity
Driver Quantity Basic
Driver Quantity Calculated
Driver Quantity Fixed
Driver Quantity Variable
Driver Sequence Number
Driver Weight Fixed
Driver Weight Variable
Idle Driver Quantity
Idle Driver Quantity UE
Idle Quantity
Output Quantity
Output Quantity UE
Sold Quantity
Total Driver Quantity
Total Driver Quantity Basic
Total Driver Quantity Calculated
Total Driver Quantity UE
Unassigned Quantity
Used Quantity
```

Default Precision for Currency Rates

The default display precision for currency rates determines the default display precision for columns in the Manage Exchange Rates dialog box.

```
Driver Rate
Unit Cost
Unit Profit
Unit Revenue
```

Appearance of Grids

You can select the font size of text that is contained in grids (tables). This setting is different from the Windows Control Panel setting that specifies the font size for all fonts that are displayed on your monitor.

You can select the colors for alternating rows in grids.

Display of Certain Dialog Boxes and Wizards

By default, when you create an entered cost element or create a dimension member, you specify information about the item in a dialog box. The dialog box enables you to specify all the information about the item when the item is created. However, you can bypass the dialog box to quickly create numerous items. In this case, system-generated default information is used, but you can change the default information later.

Cause Background Colors to Be Printed

1. Select **Help** ⇨ **Contents, Index, and Search** > .

   The SAS Cost and Profitability Management Help appears.

2. Select **Options ⇨ Internet Options.**

   The Internet Options dialog box appears.

3. Click the **Advanced** tab.
4. In the **Settings** list, under the **Printing options**, select the **Print background colors and images** option.

Hide or Show the Status Bar

Select **View ⇨ Status Bar**.

User Options Dialog Box

About the User Options Dialog Box

In the User Options dialog box, you can specify options that affect how you use SAS Cost and Profitability Management.

Note: You can perform the following tasks without first opening a model.

How to Access the User Options Dialog Box

Select **Tools ⇨ User Options**.

Data Tab

Select the Default Display Precision for New Columns

1. Click the **Data** tab.
2. In the Decimal Places Displayed section, select the number of decimal places for Cost values, Numeric values, and Currency rates.

Display Tab

Select the Grid Colors and Font Size

1. Click the **Display** tab.
2. In the Grid Colors section, select values for First row color, Second row color, and Text color from the menus.
3. In the Grid Font Size section, select an option.

Select Where to Display the Currency Symbol

1. Click the **Display** tab.

2. In the Currency section, select or clear the options.

Advanced Tab

Specify Whether Certain Dialog Boxes and Wizards Will Be Displayed

1. Click the **Advanced** tab.
2. In the Settings section, under Dialog Boxes and Wizards, select the dialog boxes and wizards.

Easy API Configuration Tab

Specify an SMTP server for sending emails

If you specify an SMTP server, then Easy API sends an email to report the results of each operation. You can specify one email ID for receiving reports of a successful operation and a different email ID for receiving reports of a failed operation.

- Click the **Easy API Configuration** tab.
- Specify an **SMTP server**.

Specify an email ID to receive good news

You can specify one email ID for receiving reports of a successful Easy API operation.

- Click the **Easy API Configuration** tab.
- Specify an **Email Id for successful operation**.

Specify an email ID to receive bad news

You can specify a different email ID for receiving reports of a failed operation.

- Click the **Easy API Configuration** tab.
- Specify an **Email Id for failure of operation**.

Turn on or off the saving of XML

When SAS Cost and Profitability Management performs an operation, it uses XML to pass the parameters for the operation. You can ask SAS Cost and Profitability Management to save this XML when it performs an operation so that you can use the same XML to perform the same operation using Easy API.

To begin saving XML:

- Click the **Easy API Configuration** tab.
- Select **Save operation xml in directory path.**
- Specify the directory path where the XML is to be saved.

To stop saving XML:

- Click the **Easy API Configuration** tab.
- Unselect **Save operation xml in directory path.**

Part 26

Properties Reference

Chapter 90

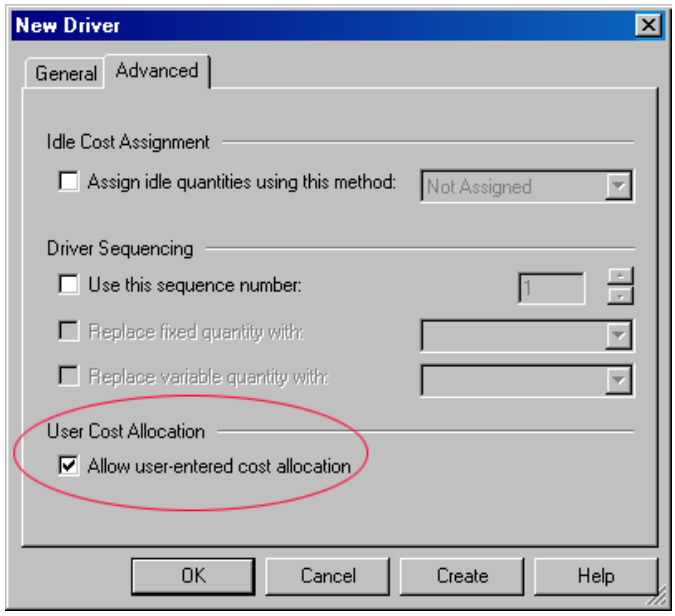
Alphabetic List of Properties

Allocated Cost (AllocCost)

Allocated Cost is the sum of DrvAllocCost on assignments going out of the account. DrvAllocCost is user-entered.

The cost that is flowing out of an account to other accounts based on the user allocation method; a direct placement of money to a specified destination account, and the remainder of the cost could be assigned using driver quantities.

The value of this property is generated by the system. You cannot change this value. It represents the sum of all user-entered allocated costs. You enter a user-entered allocated cost by setting the value of Driver Allocated Cost (DrvAllocCost). You enable a user allocation method by checking the **Allow user-entered cost allocation** check box on the **Advanced** tab of the New Driver window. Unless you check this, Driver Allocated Cost (DrvAllocCost) is protected for the driver.




```
Drivable Cost = Cost - Allocated Cost
Driver Rate = Drivable Cost / TDQ
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8–byte floating point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`AllocCost=∑ DrvAllocCost` See Chapter 60, "Properties in Calculation," on page 527.

See Also

"Create a Driver" on page 423

Assigned Cost (AsgnCost)

The cost that is flowing out of a cost element or account to other cost elements or accounts; the cost flowing out of an account to other accounts based on the driver method.

Assigned cost is calculated according to the following formula:

```
Assigned Cost = Driven Cost + Allocated Cost
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`Assigned Cost = Allocated Cost + Driven Cost`

See Also

- Assigned Reciprocal Cost on page 751
- Received Cost on page 785

Assigned Idle Cost (AsgnIdlCost)

The sum of costs on outgoing assignments that is specifically because of the source account's idle flow behavior.

Assigned Idle Cost is calculated according to the following formula:

```
Asgn IdlCost = (sum of DrvIdleCost of destination side
accounts)
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`AsgnIdlCost = AsgnIdlQty x DrvRate`
	Also:
	`AsgnIdlCost = sum of DrvIdlCost on outgoing assignments`

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412

Assigned Idle Quantity (AsgnIdlQty)

The amount of an account's idle quantity that causes cost on outgoing assignments because of the account's idle flow behavior.

AsgnIdlQty for a driver is the sum of IdlDrvQty for the individual assignments.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`AsgnIdlQty = ∑ IdlDrvQty` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412

Assigned Non Reciprocal Cost (AsgnNRecipCost)

The sum of costs on an account's outgoing assignments where the destination accounts are not part of the same reciprocal system as the source account.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency

Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	N/A

Assigned Reciprocal Cost (AsgnRecipCost)

The assigned cost that is flowing from a reciprocal account to other reciprocal accounts. For a non-reciprocal account, this value is NULL.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	N/A

See Also

- "Reciprocal Costing" on page 414
- Cost on page 752
- Received Reciprocal Cost on page 787

Calculate Error (CalcError)

The error status for a calculated driver or calculated attribute value.

This property rolls up in the module view and displays the highest priority error at the module level. This error information can be useful when debugging calculation errors.

System generated or user entered?	Generated by system
Default format in a column layout:	Text
Data type:	8–byte floating-point number

Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	N/A

Cost (Cost)

The calculated cost of a cost element or account.

Cost is calculated according to the formula:

```
Cost = EnteredCost + Received Cost
```

System generated or user entered?	System-generated
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	`Cost=EnteredCost + ReceivedCost` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- Chapter 59, "Calculating Costs," on page 525
- Assigned Reciprocal Cost on page 751
- Received Cost on page 785

Dimension Level Name (DimLevelName)

The name of the level for the dimension that completes an intersection.

System generated or user entered?	User modifiable But this property is read-only in a grid. You cannot change this value in a grid.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters
Type of property in an assignments pane:	N/A

Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- Dimension Level Number (DimLevelNumber) on page 753

Dimension Level Number (DimLevelNum)

The depth in the dimension hierarchy of the dimension that completes an intersection.

System generated or user entered?	Generated by system
Default format in a column layout:	Text
Data type:	4-byte integer value
Type of property in an assignments pane:	N/A
Formula	N/A

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- “Dimension Level Number (DimLevelNum)” on page 753
- Dimension Level Name (DimLevelName) on page 752

Dimension Member Name (DimMemName)

The name of the dimension member that defines an account (dimensional intersection).

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	The maximum length is 256 alphanumeric Unicode characters
Type of property in an assignments pane:	N/A

Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- Chapter 24, "What Are Dimensions?," on page 195
- Dimension Member Reference (DimeMemberReference) on page 754

Dimension Member Reference (DimMemRef)

The reference of the dimension member that defines an account (dimensional intersection).

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- Chapter 24, "What Are Dimensions?," on page 195
- Dimension Member Name (DimMemberName) on page 753

Dimension Name (DimName)

The name of the dimension that completes an intersection.

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.

Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- Dimension Reference (DimReference) on page 755

Dimension Reference (DimRef)

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- Chapter 24, “What Are Dimensions?,” on page 195
- Dimension Name (DimName) on page 754

Display Name (Display Name)

The name for the displayed item.

System generated or user entered?	Generated by system. However, you can change Name.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters
Type of property in an assignments pane:	N/A

Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Display Reference on page 756

Display Reference (Display Reference)

The reference for the displayed item.

System generated or user entered?	Generated by system. However, you can change Reference.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data , this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Display Name on page 755

Drivable Cost (DrvblCost)

The remainder of an account's cost that can be assigned by its driver after subtracting user-allocated costs.

You cannot change this value. It is generated by the system according to the following formula:

```
DrivableCost = Cost - AllocatedCost
```

Also:

```
DrivableCost = UsedCost + IdleCost
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency

Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`DrvblCost = Cost - Allocated Cost`
	Also::
	`DrvblCost = IdleCost + UsedCost`
	See Chapter 60, "Properties in Calculation," on page 527.

See Also

"Step 9–DrvblCost for Resource Account" on page 550

Driven Cost (DrvnCost)

The amount of cost that is driven by the account's driver, including both Used Cost and Assigned Idle Cost.

Driven Cost is calculated according to the following formula:

```
Driven Cost = Used Cost + Assigned Idle Cost
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`Driven Cost = DrvnQty x DrvRate`
	Also:
	`Driven Cost = Used Cost + AsgnIdlCost`

See Also

- "Used Cost (UsedCost)" on page 799
- "Assigned Idle Cost (AsgnIdlCost)" on page 749

Driven Quantity (DrvnQty)

The sum of the driver quantity on an account's outgoing assignments, including both Used Quantity and Assigned Idle Quantity.

The value of this property is generated by the system. You cannot change this value. It is defined by the following formula:

```
DrivenQuantity = AssignedIdleQty + UsedQuantity
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`Driven Quantity = UsedQty + AsgnIdlQty` See Chapter 60, "Properties in Calculation," on page 527.

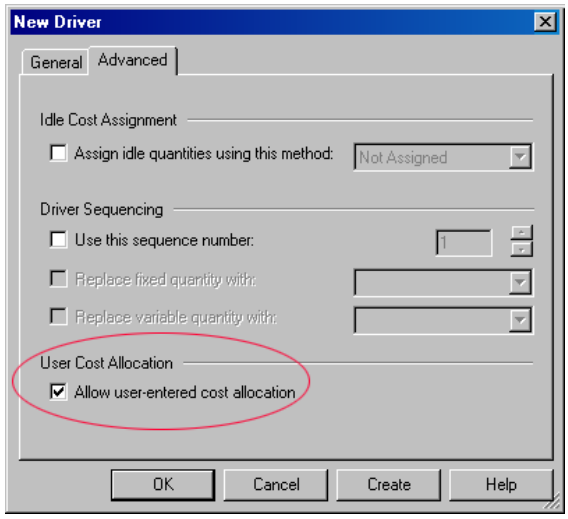
See Also

- "Assigned Idle Quantity (AsgnIdlQty)" on page 750
- "Driver Quantity Calculated (DrvQtyCalc)" on page 764

Driver Allocated Cost (DrvAllocCost)

The user-entered Allocated Cost value on an assignment path.

You enable the setting of Driver Allocated Cost by checking the **Allow user-entered cost allocation** check box on the **Advanced** tab of the New Driver window. Unless you check this, Driver Allocated Cost is protected for the driver.



System generated or user entered?	User-entered
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	User-entered value

When you add this property to a column in a grid, the default format is Currency.

This is an.

See Also

"Create a Driver" on page 423

Driver Cost (DrvCost)

The cost that is flowing out of or flowing into an account through an assignment .

Driver Cost is calculated according to the following formula:

```
Driver Cost = (Driver Driven Cost + Driver Allocated Cost)
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number

Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`Driver Cost = DrvAllocCost + DrvDrvnCost` That is, the outgoing cost for an assignment is the sum of the cost that is allocated plus the sum that is driven by the driver. See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "Example 6: Weighted Driver with Fixed and Variable Driver Quantities" on page 391
- "Step 12–DrvCost for Assignments to Activity Accounts" on page 553
- "Step 16–DrvCost for Assignments to Cost Object Accounts" on page 557
- Cost on page 752

Driver Driven Cost (DrvDrvnCost)

the cost going to each destination account. It is calculated by multiplying DrvQtyCalc (the number of units) times DrvRate (the cost per unit)

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.

Formula	`Driver Driven Cost = DrvQtyCalc x Driver Rate`
	So:
	`Driver Driven Cost = DrvQtyCalc  x (DrvblCost/TDQ)`
	Also:
	`DrvDrvnCost = DrvIdlCost + DrvUsedCost`
	See Chapter 60, "Properties in Calculation," on page 527.

See Also

- Cost on page 752
- "Step 11–DrvDrvnCost for Assignments to Activity Accounts" on page 552
- "Step 15–DrvDrvnCost for Assignments to Cost Object Accounts" on page 556

Driver Driven Quantity (DrvDrvnQty)

The assigned quantity that is flowing on an assignment path, including both Used Quantity and Assigned Idle Quantity.

Driver Driven Quantity is calculated according to the following formula:

```
DrvDrvnQty = DrvQtyCalc + IdleDrvQty
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`DrvDrvnQty = DrvQtyCalc + IdlDrvQty` See Chapter 60, "Properties in Calculation," on page 527.

When you add this property to a column in a grid, the default format is Number.

The value of this property is generated by the system. You cannot change this value.

This is an .

Driver Formula (DrvFormula)

The formula that is attached to a source account's driver.

System generated or user entered?	User modifiable. But This property is read-only in a grid. You cannot change this value in a grid—only in the Driver properties dialog.
Default format in a column layout:	Text
Data type:	Large text value
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

When you import or export model data, this value must translate into the TEXT or NTEXT data types in Microsoft SQL Server.

See Also

- Chapter 44, "Calculated Driver," on page 393
- Formula on page 771

Driver Idle Cost (DrvIdlCost)

A source account's Idle Cost. The assigned idle cost that is flowing on an assignment path.

Driver Idle Cost is calculated according to the following formula:

```
Driver Idle Cost = Idle DriverQty * Driver Rate
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`DrvIdlCost = IdlDrvQty x DrvRate` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- Driver Cost on page 759
- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412

Driver Name (DrvName)

`Driver Name (DrvName)` The name of the driver that is associated with an account.

System generated or user entered?	User-entered.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.

Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

Driver Percentage (DrvPcnt)

The percentage of the total driver quantity for an assignment.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	DrvPcnt = (DrvQtyCalc/TDQ) x 100

Driver Quantity Basic (DrvQtyBasic)

The sum of the fixed and variable quantities that flows from one account to another.

Driver Quantity Basic is calculated according to the following formula:

```
DrvQtyBasic = DQF + DQV* Dest.TDQ
```

The formula for DrvQtyBasic is a limiting case of the general formula for DrvQtyCalc:

```
DrvQtyCalc = (DQF x DWF)+ (DQV x DWV x Dest.TDQ)
```

where DWF and DWV both equal 1.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`DrvQtyBasic = DQF + DQV* Dest.TDQ`

See Also

- "What Is a Driver?" on page 355
- Driver Quantity Calculated on page 764
- Driver Quantity Fixed on page 764
- Driver Quantity Variable on page 765

Driver Quantity Calculated (DrvQtyCalc)

Represents the number of units going to each destination account.

The value of this property is generated by the system. You cannot change this value. It is calculated according to the following formula:

```
DrvQtyCalc = (DQF x DWF) + (DQV x DWV x Dest.TDQ)
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`DrvQtyCalc = (DQF x DWF)+(DQV x DWV x Dest.TDQ)` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "DrvQtyCalc" on page 356
- "What Is a Driver?" on page 355
- Chapter 61, "Detailed Example of Calculation," on page 539
- Driver Quantity Basic on page 763
- Driver Quantity Fixed on page 764
- Driver Quantity Variable on page 765

Driver Quantity Fixed (DQF)

The user-entered fixed quantity that flows from one account to another.

System generated or user entered?	You can change this value only on assigned cost elements with a driver that allows fixed driver quantities.

Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	User-entered value

See Also

- "Fixed Driver Quantities" on page 400
- "Variable Driver Quantities" on page 401
- Driver Quantity Basic on page 763
- Driver Quantity Calculated on page 764
- Driver Quantity Variable on page 765

Driver Quantity Variable (DQV)

The user-entered variable quantity that flows from one account to another.

System generated or user entered?	You can change this value only on assigned cost elements with a driver that allows variable driver quantities.
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	User-entered value

See Also

- "Fixed Driver Quantities" on page 400
- "Variable Driver Quantities" on page 401
- Driver Quantity Basic on page 763
- Driver Quantity Calculated on page 764
- Driver Quantity Fixed on page 764

Driver Rate (DrvRate)

Represents the cost per unit for every destination account It is calculated according to the following formula:

```
DriverRate = IF ExternalUnit THEN UnitCostEntered ELSE DrivableCost / TDQ
```

And:

```
DrivableCost = Cost - AllocatedCost
```

Note: Driver Rate can be different from Unit Cost if any of the following is true:

- TDQUE ne 0 (which overrides TDQ in the formula DrvRate = DrvblCost/TDQ)
- AllocCost ne 0 (which affects DrvblCost in the formula DrvRate = DrvblCost/TDQ)
- OutQtyUE ne 0 (which overrides OutQty in the formula UnitCost = Cost/OutQty)

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`Driver Rate = DrvblCost / TDQ` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "DrvRate" on page 356
- "Step 10–DrvRate for Resource Account" on page 551
- "Step 14–DrvRate for Outgoing Drivers from Activity Accounts" on page 555

Driver Sequence Number (DrvSeq)

The sequence number that is associated with a source account's driver.

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Number

Data type:	4-byte integer value
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

See Also

"Driver Sequencing" on page 407

Driver Type (DrvType)

The type of driver: Evenly Assigned, Percentage, or Standard. The default driver is Standard.

System generated or user entered?	User modifiable. But this property is read-only in a column-layout.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Chapter 40, "Introducing Drivers," on page 355

Driver Used Cost (DrvUsedCost)

The sum of costs on an account's outgoing assignments that is caused by the account's driver. This amount does not include Allocated Cost or Idle Cost.

Driver Used Cost is calculated according to the following formula:

```
Driver Used Cost = DriverQtyCalc * Driver Rate
```

or, the following formula if there is no SoldQuantity

```
DriverUsedCost = DrivableCost * DQCalc/TDQCalc
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.

Formula	`DrvUsedCost = DrvDrvnCost - DrvIdlCost`
	Also:
	`DrvUsedCost = DrvQtyCalc x DrvRate`
	See Chapter 60, "Properties in Calculation," on page 527.

Driver Weight Fixed (DWF)

The numeric factor that modifies a fixed driver quantity. The default weight is 1.

System generated or user entered?	You can change this value only on assigned cost elements with a driver that allows fixed driver weights.
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	N/A

See Also

- "Fixed Driver Quantities" on page 400
- "Variable Driver Quantities" on page 401
- Driver Weight Variable on page 768

Driver Weight Variable (DWV)

The numeric factor that modifies a variable driver quantity. The default weight is 1.

System generated or user entered?	You can change this value only on assigned cost elements with a driver that allows variable driver weights.
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	N/A

See Also

- "Fixed Driver Quantities" on page 400
- "Variable Driver Quantities" on page 401
- Driver Weight Fixed on page 768

Entered Cost (EntCost)

The user-entered cost for a cost element or the calculated total of all account entered cost elements for an .

System generated or user entered?	User entered
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	User-entered value

See Also

- "Entered Cost Elements" on page 260
- "Entered Unit Cost (EntUnitCost)" on page 770
- Cost on page 752

Entered Unit Cost (EntUnitCost)

The unit cost, as entered by a user, for a cost element. Or, the sum of the entered unit cost of all the cost elements belonging to an account.

System generated or user entered?	User entered
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	User-entered value

See Also

- "Entered Cost Elements" on page 260
- "Entered Cost (EntCost)" on page 769
- Cost on page 752

Fixed Driver Quantity Override

For a sequenced driver, the name of the property that replaces the Driver Quantity Fixed value after the previous sequence pass.

System generated or user entered?	User entered, but not in a column layout.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- "Driver Sequencing" on page 407

- Variable Driver Quantity Override on page 801

Formula

The formula for a calculated driver or calculated attribute.

System generated or user entered?	User entered, but not in column layout
Default format in a column layout:	Text
Data type:	This is a large text-based value.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into the TEXT or NTEXT data types in Microsoft SQL Server.

See Also

- Chapter 51, "Formulas," on page 457
- Chapter 44, "Calculated Driver," on page 393
- "Calculated Attributes" on page 274
- Driver Formula on page 761

For What-If (ForWhatIf)

Indicates whether an item has been marked for What-If Analysis.

System generated or user entered?	User-entered. See Chapter 71, "Marking Items for What If Analysis," on page 605.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

Has Assignments (HasAsgn)

Indicates whether an account receives costs from another account.

System generated or user entered?	System generated
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	N/A

Has Attributes (HasAttr)

Indicates whether an item has at least one attribute.

System generated or user entered?	System generated
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

Has Entered Cost (HasEntCost)

Indicates whether an account has at least one user-entered cost element.

System generated or user entered?	System generated
Default format in a column layout:	Checkbox
Data type:	Boolean

Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

See Also

Chapter 30, "What Are Cost Elements?," on page 259

Has Idle Cost (HasIdlCost)

Indicates whether an item has a non-zero idle cost—that is, costs that don't flow to other accounts.

System generated or user entered?	System generated
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	N/A

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412

Has Notes (HasNotes)

Indicates whether an account has a Periodic Note or Model Note.

System generated or user entered?	System generated
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Account See "Account Properties" on page 329.

Formula	N/A

See Also

- "Model Note (ModNote)" on page 781
- "Periodic Note (PerNote)" on page 783

Has Used Cost (HasUsedCost)

Indicates whether an item has a non-zero used cost—that is, it flows costs to other accounts.

System generated or user entered?	Calculated by system
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	N/A

Idle Cost (IdlCost)

The idle cost is the remaining cost after all direct assignments have been made; the calculated cost that does not flow out of an account. It is the account unit cost x idle quantity (where idle quantity = user-entered total driver quantity – output quantity). Or, more succinctly:

```
Idle Cost = IdlQty x Driver Rate
```

Also:

```
Idle Cost = AssignedIdleCost x UnassignedCost
```

Also:

```
IdleCost = IdleQty x Driver Rate
```

This remaining cost can be assigned using a specific idle cost flow method (driver, percentage, user-entered, or evenly).

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency

Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`Idle Cost = IdlQty x DrvRate`
	Also:
	`Idle Cost = AsgnIdlCost + UnAsgnCost`

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412
- Cost on page 752
- Idle Quantity on page 778
- Idle Percentage on page 777

Idle Driver Quantity (IdlDrvQty)

The quantity that causes cost on an assignment path that is specifically because of the source account's idle flow behavior.

IdlDrvQty for an assignment is determined by its idle flow method (user-entered, user proportion, driver quantity, evenly assigned). For information, see "Idle Quantities" on page 410.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	N/A

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412
- Idle Driver Quantity UE on page 776
- Idle Flow Method on page 776

Idle Driver Quantity UE (IdlQtyUE)

The optional, user-entered idle driver quantity value for assigned cost elements and internal cost elements.

This value is used to drive idle costs that are based on source account's idle flow behavior.

System generated or user entered?	User entered
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	User-entered value

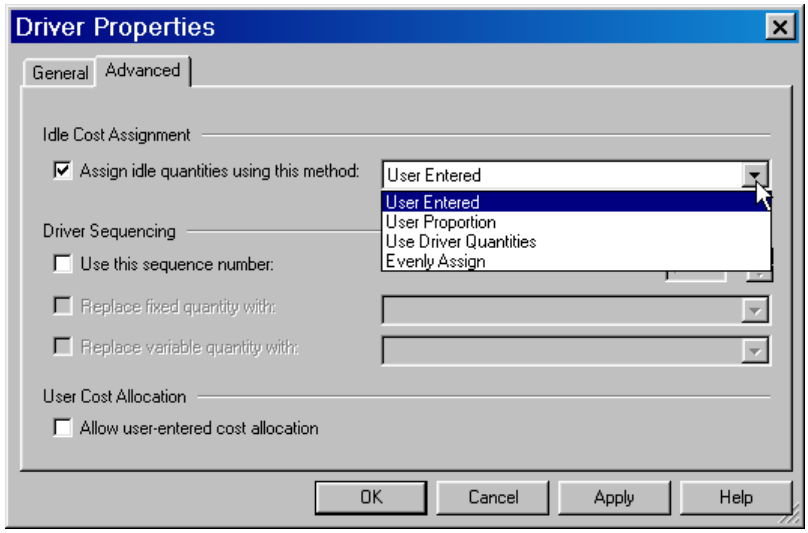
See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412
- Idle Driver Quantity on page 775
- Idle Flow Method on page 776

Idle Flow Method (IdleFlow)

The method by which idle cost through a driver is calculated for a source account's outgoing assignments.

Interactively within the interface, the values can be: User Entered, User Proportion, User Driver Quantities, and Evenly Assigned. The default is User Entered. The Idle Flow Method is specified in a drop-down box on the Advanced tab of the Driver Properties window.



When you import model data, the values can be: Dont Assign, Evenly Assigned, User Driver Quantities, User Entered, User Proportion. The default is Dont Assign.

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412

Idle Percentage (IdlPcnt)

The calculated percentage of an account's cost that does not flow out of the account. The idle percentage is the remaining cost after all direct assignments have been made. It is 100 - (user-entered total driver quantity -' calculated total driver quantity)/user-entered total driver quantity.

Idle Percentage is calculated according to the following formula:

```
IdlePercentage = IdleQty/TDQ
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Percentage
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Assignment See "Assignment Properties" on page 325.
Formula	`IdlPcnt = 100 * IdlQty / TDQ` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "Idle Quantities" on page 410
- "Example of Idle Flow" on page 412
- Idle Cost on page 774
- Idle Quantity on page 778

Idle Quantity (IdlQty)

The calculated amount of an account's quantity that does not flow out of the account. The idle quantity is the remaining cost after all direct assignments have been made. It is (user-entered total driver quantity – output quantity).

```
IdlQty = TDQUE - OutQty
```

```
If TDQUE > OutputQty then Positive Idle
```

```
If TDQUE < OutputQty/OutputQtyUE then Negative Idle
```

Because TDQ equals UsedQty unless overridden by TDQUE, the only way that IdlQty can be non-zero is if TDQUE is greater than UsedQty.

Note: If TDQUE is less than UsedQty, then the following error message appears: "Overdriven source account. The value entered for TDQUE is less than the Driven Quantity."

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`IdlQty = TDQ - UsedQty` See Chapter 60, “Properties in Calculation,” on page 527.

See Also

- “Idle Quantities” on page 410
- “Example of Idle Flow” on page 412
- Idle Cost on page 774
- Idle Percentage on page 777

Intersection Name (IntsctnName)

The concatenation of all the names of the dimensions that intersect to create an item. Each name is separated by a special character.

System generated or user entered?	System generated
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Intersection Reference on page 779

Intersection Reference (IntsctnRef)

The concatenation of all the references of the dimension members that intersect to create an item. Each reference is separated by a special character.

System generated or user entered?	System generated
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.

Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Intersection Name on page 779

IsBehavior (IsBehavior)

Indicates whether an account has been marked as a behavior for use in SAS Profitability Management..

System generated or user entered?	User-entered. See "Mark Accounts as Behaviors" on page 729.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

IsReciprocal (IsRecip)

Indicates whether an assignment or account is part of a reciprocal system. You can use this property in account search.

The property applies both to accounts and to assignments. For assignments, the property tests true when the source and destination accounts are in the same reciprocal system.

System generated or user entered?	System-generated
Default format in a column layout:	Checkbox
Data type:	Boolean

Type of property in an assignments pane:	Account See "Account Properties" on page 329. Assignment See "Assignment Properties" on page 325.
Formula	N/A

See Also

- "Reciprocal Id (RecipId)" on page 788
- "Reciprocal Costing" on page 414

Model Note (ModNote)

Text describing an account. The text applies to every period/scenario in which the account occurs—unlike a Periodic Note which applies only to an account in one period/scenario.

System generated or user entered?	User-entered
Default format in a column layout:	Text
Data type:	32K alphanumeric Unicode characters.
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

See Also

"Periodic Note (PerNote)" on page 783

Module Order (ModOrdr)

The relative position of the module in the ordered set of modules that determines the direction of cost flow. Valid values are in the range -1000 to +1000, exclusive. The order of modules is defined during model creation and cannot be changed afterwards.

System generated or user entered?	User-defined during model creation
Default format in a column layout:	Number

Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

See Also

"Create a Model" on page 104

Name (Name)

The user-entered or default name of an item.

System generated or user entered?	User-entered
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Reference on page 789

Output Quantity (OutQty)

The calculated amount that is produced for an account for a period/scenario association, or the user-entered amount (Output Quantity UE). The user-entered amount overrides the calculated amount.

You cannot change this value. It is generated by the system according to the following formula:

```
OutputQty = IF OutputQtyUE is not null THEN OutputQtyUE ELSE UsedQty
```

Output Quantity is used to determine Unit Cost according to the following formula:

```
UnitCost = Cost/OutQty
```

Also: **`UnitRevenue = Revenue / OutQty`**

Also: `UnitProfit = Profit / OutQty`

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	`Output Quantity = Used Quantity` See Chapter 60, "Properties in Calculation," on page 527.

Output Quantity UE (OutQtyUE)

The user-entered output quantity.

System generated or user entered?	User entered
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	User-entered value

See Also

Output Quantity on page 782

Periodic Note (PerNote)

Descriptive text attached to an account for a particular period/scenario—unlike a Model Note whose text applies to an account in every period/scenario in which it occurs..

System generated or user entered?	User entered
Default format in a column layout:	Text
Data type:	The maximum length is 32,000 alphanumeric Unicode characters.

Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	N/A

See Also

- "Model Note (ModNote)" on page 781
- Has Notes on page 773

Profit (Profit)

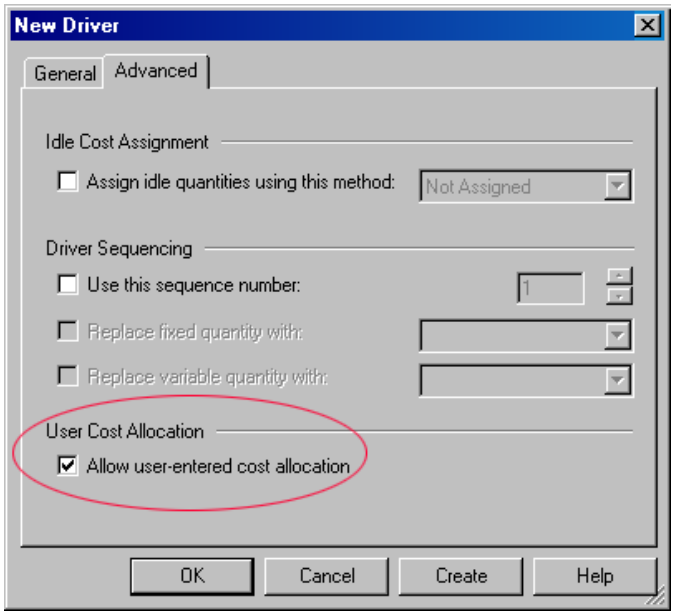
The calculated difference between revenue and cost.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	Total Revenue – Total Cost

Received Allocated Cost (RcvAllocCost)

The calculated cost of all allocated cost elements within an account. The sum of all allocated costs that flows into an account.

Note: You enable allocated cost elements by checking the **Allow user-entered cost allocation** check box on the **Advanced** tab of the New Driver window.



System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	`RcvAllocCost = ∑ DrvAllocCost` RcvAllocCost is the sum of costs allocated on individual assignments, DrvAllocCost, to the destination account.

Received Cost (RcvCost)

The calculated cost that is received by an account from all assignments.

Received Cost is calculated according to the following formula:

`Received Cost = Received Driven Cost + Received Allocated Cost`

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number

Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	`Received Cost = RcvDrvnCost + RcvAllocCost` Also: `Received Cost = RcvDrvnCost + RcvAllocCost`

Received Driven Cost (RcvDrvnCost)

The sum of costs of assignments to an account that is specifically caused by driver quantities and by excluding allocated cost amounts.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.

Formula	`RcvDrvnCost = ∑ DrvDrvnCost`
	RcvDrvnCost is the sum of costs driven by individual assignments, DrvDrvnCost to the destination account.
	Also:
	`RcvDrvnCost = RcvIdlCost + RcvUsedCost`

Received Idle Cost (RcvIdlCost)

The cost amounts on assignments to an account that are caused by the source accounts' idle flow behaviors.

This value appears only on a cost element, not on an account.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.

Formula	`RcvIdlCost = ∑ DrvIdlCost` RcvIdlCost is the sum of idle costs from individual assignments, DrvIdlCost , to the destination account

Received Non Reciprocal Cost (RcvNRecipCost)

The calculated difference between the cost that is received by an account from all assignments and the cost that is received from all reciprocal assignments.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	RcvNRecipCost = RcvCost – RcvRecipCost

See Also

Received Reciprocal Cost on page 787

Received Reciprocal Cost (RcvRecipCost)

The calculated cost that an account receives from other accounts in the same reciprocal system.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	RcvRecipCost = RcvCost – RcvNRecipCost

See Also

Received Non Reciprocal Cost on page 787

Received Used Cost (RcvUcost)

The cost amounts on assignments to an account that are caused by non-idle driver quantities (Driver Quantity Calculated).

This value appears only on a cost element, not on an account.

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	`RcvUsedCost = ∑ DrvUCost` RcvUsedCost is the sum of used costs from individual assignments, DrvUCost, to the destination account.

Reciprocal Id (RecipId)

A system-assigned ID to identify a set of reciprocal assigments. The same ID is assigned to both accounts and assignments in a reciprocal set.

You can use this property in account search, but not in driver formulas or rule formulas because it is not stable across different period/scenarios.

System generated or user entered?	System-generated
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329. Assignment See "Assignment Properties" on page 325.
Formula	N/A

See Also

- "IsReciprocal (IsRecip)" on page 780
- "Reciprocal Costing" on page 414

Reference (Reference)

The user-entered or default label that uniquely identifies an item.

System generated or user entered?	User-entered
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters
Type of property in an assignments pane:	N/A
Formula	N/A

When you mport or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Name on page 782

Revenue (Revenue)

The user-entered revenue.

System generated or user entered?	User entered
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	User-entered value

Sold Cost (SoldCost)

Sold Cost is calculated according to the following formula:

```
Sold Cost = SoldQty x DrvRate
```

System generated or user entered?	System-generated
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	`SoldCost = SoldQty x DrvRate` See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "Sold Quantity (SoldQty)" on page 790
- "Driver Rate (DrvRate)" on page 766

Sold Quantity (SoldQty)

The user-entered number of units sold.

System generated or user entered?	User entered
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Incoming See "Incoming Properties" on page 328.
Formula	User-entered value

See Also

- "Example 2: Entered Unit Cost with Variable Driver Quantities and SoldQty" on page 374
- "What Is a Driver?" on page 355

Total Driver Quantity (TDQ)

The calculated output quantity or the user-entered total driver quantity (Total Driver Quantity UE (TDQUE)). The user-entered amount overrides the calculated amount.

You cannot change this value. It is generated by the system according to the following formula:

```
TDQ = IF TDQUE is not null THEN TDQUE ELSE UsedQty
```

TDQ = Used Quantity (6) unless overridden by TDQUE

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`TDQ = Used Quantity` (unless overridden by TDQUE) See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "What Is a Driver?" on page 355
- Total Driver Quantity Basic (TDQBasic) on page 791
- Total Driver Quantity Calculated (TDQCalculated) on page 792

Total Driver Quantity Basic (TDQBasic)

The calculated quantity of all outgoing Basic driver quantities (Driver Quantity Basic) for an account.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	???

See Also

- "What Is a Driver?" on page 355
- Total Driver Quantity (TDQ) on page 791
- Total Driver Quantity Calculated (TDQCalculated) on page 792
- Total Driver Quantity UE (TDQUE) on page 792

Total Driver Quantity Calculated (TDQCalc)

The calculated quantity of all outgoing Calculated driver quantities (Driver Quantity Calculated) for an account.

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`TDQCalc = ∑DrvQtyCalc` TDQCalc for a driver is the sum of DrvQtyCalc for the individual outgoing assignments of that driver. See Chapter 60, "Properties in Calculation," on page 527.

See Also

- "What Is a Driver?" on page 355
- Total Driver Quantity (TDQ on page 791
- "Total Driver Quantity UE (TDQUE)" on page 792
- Total Driver Quantity Basic (TDQBasic on page 791
- Total Driver Quantit UE (TDQUE) on page 792

Total Driver Quantity UE (TDQUE)

The user-entered total driver quantity.

System generated or user entered?	You cannot change this value for accounts with a driver type of Evenly Assigned, or Percentage.
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	User-entered value

See Also

- "What Is a Driver?" on page 355
- Total Driver Quantity (TDQ on page 791
- "Total Driver Quantity UE (TDQUE)" on page 792
- Total Driver Quantity Basic (TDQBasic on page 791
- Total Driver Quantity Calculated (TDQCalculated) on page 792

Type (Type)

The type of item.

System generated or user entered?	User entered, but not in a column layout
Default format in a column layout:	Text
Data type:	4-byte integer value
Type of property in an assignments pane:	N/A
Formula	N/A

Unassigned Cost (UnAsgnCost)

The calculated cost that does not flow out of an account.

Unassigned Cost is calculated according to the following formula:

```
Unassigned Cost = Cost - Assigned Cost
```

System generated or user entered?	Calculated by system

Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.

Formula	`UnAsgnCost = DrvblCost + AllocCost - AsgnCost`
	Also:
	`UnAsgnCost = Cost - AsgnCost`
	Also:
	`UnAsgnCost = IdlCost - AsgnIdlCost`

See Also

- Assigned Cost on page 749
- Unassigned Quantity on page 794

Unassigned Quantity (UnAsgnQty)

The calculated number of driver quantity units that do not cause cost on an account's outgoing assignments.

Unassigned Quantity is calculated according to the following formula:

```
UnassignedQty = (OutputQty - UsedQty + UnassignedIdleQty)
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`UnAsgnQty = IdlQty - AsgnIdlQty`

See Also

Unassigned Cost on page 793

Unique Driver Quantities (UniqDvrQty)

Indicates whether a driver quantity is not shared by assignments from other accounts.

System generated or user entered?	User entered, but not in a column layout.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

Unit Cost (UnitCost)

The calculated cost of one unit. It is calculated according to the following formula:

```
UnitCost = IF ExternalUnit THEN UnitCostEntered ELSE DrivableCost/OutputQty
```

And:

```
OutputQty = IF OutputQtyUE is not null THEN OutputQtyUE ELSE UsedQty
```

So:

```
UnitCost = IF ExternalUnit THEN UnitCostEntered ELSE DrivableCost/UsedQty
```

Remember:

```
DrivableCost = Cost - AllocatedCost
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`UnitCost = Cost / OutQty`

Unit Of Measure (UoM)

There are two instances of Unit of Measure:

- The user-entered name for the type quantity (such as DQF) going out from an account. For example, you can set UoM to “tires” to specify that what goes out of an account is number of tires.
- The user-entered name for the type of measure of a numeric attribute.

This property is periodic. You can enter a different Unit of Measure in different periods.

System generated or user entered?	User entered
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A
Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

Unit Profit (UnitProfit)

The calculated profit for a sold unit.

You cannot change this value. It is calculated according to the following formula:

```
UnitProfit = Profit/OutputQty
```

And:

```
OutputQty = IF OutputQtyUE is not null THEN OutputQtyUE ELSE UsedQty
```

So:

```
UnitProfit = IF OutputQtyUE is not null THEN Profit/OutputQtyUE ELSE Profit/UsedQty
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See “Account Properties” on page 329.

Formula	UnitProfit = IF OutQtyUE is not null THEN Profit / OutQtyUE ELSE Profit/UsedQty

See Also

- Unit Cost on page 795
- Unit Revenue on page 797

Unit Revenue (UnitRevenue)

The calculated revenue for a sold unit.

You cannot change this value. It is calculated according to the following formula:

```
UnitRevenue = Revenue/OutputQty
```

And:

```
OutputQty = IF OutputQtyUE is not null THEN OutputQtyUE ELSE UsedQty
```

So:

```
UnitRevenue = IF OutputQtyUE is not null THEN Revenue/OutputQtyUE ELSE Revenue/UsedQty
```

Table 90.1 *General information concerning the property*

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Account See "Account Properties" on page 329.
Formula	UnitRevenue = IF OutQtyUE is not null THEN Revenue/OutQtyUE ELSE Revenue/UsedQty

See Also

- Unit Cost on page 795
- Unit Profit on page 796

Use Fixed Quantities (UseFixQty)

Indicates that Driver Quantity Fixed is editable on an assignment whose source account uses the fixed quantity driver.

Table 90.2 General information concerning the property

System generated or user entered?	User entered, but not in a column layout.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

See Also

- Use Variable Quantities on page 798
- Use Weighted Quantities on page 799

Use Variable Quantities (UseVarQty)

Indicates that Driver Quantity Variable is editable on an assignment whose source account uses the variable quantity driver.

Table 90.3 General information concerning the property

System generated or user entered?	User entered, but not in a column layout.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

See Also

- Use Fixed Quantities on page 797
- Use Weighted Quantities on page 799

Use Weighted Quantities (UseWeightedQty)

Indicates whether Driver Weight Fixed and/or Driver Weight Variable values are editable on an assignment whose source account uses the weighted quantity driver.

System generated or user entered?	User entered, but not in a column layout.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	Driver See "Driver Properties" on page 329.
Formula	N/A

See Also

- Use Fixed Quantities on page 797
- Use Variable Quantities on page 798

Used Cost (UsedCost)

The calculated used cost for outgoing driver quantities and sold quantities for an account.

Used Cost is calculated according to the following formula:

```
Used Cost = UsedQty*Driver Rate
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Currency
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`Used Cost = UsedQty x DrvRate`

See Also

Used Quantity on page 800

Used Quantity (UsedQty)

The calculated amount for outgoing driver quantities and sold quantities for an account.

Used Quantity is calculated according to the following formula:

```
UsedQty = (TDQCalc + SoldQty)
```

System generated or user entered?	Calculated by system
Default format in a column layout:	Number
Data type:	8-byte floating-point number
Type of property in an assignments pane:	Outgoing See "Outgoing Properties" on page 326.
Formula	`UsedQty = TDQCalc + Sold Quantity` See Chapter 60, "Properties in Calculation," on page 527.

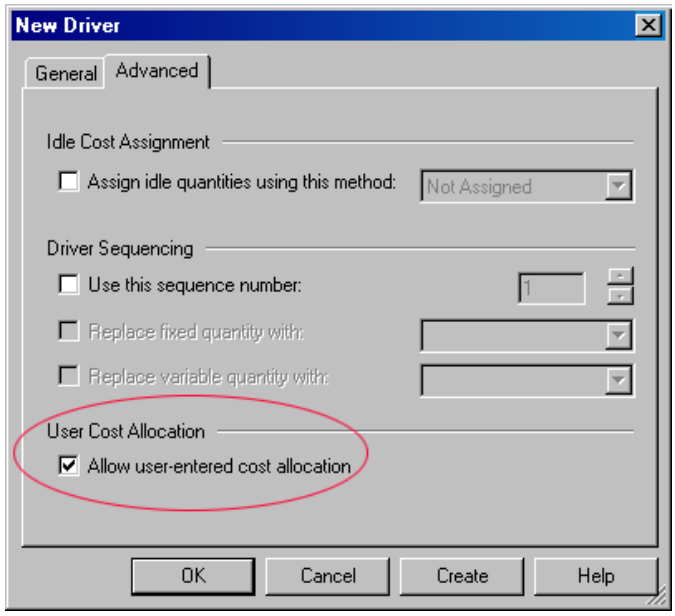
See Also

Used Cost on page 799

User-Entered Cost Allocation

indicates whether user-entered allocated cost values (Driver Allocated Cost) are editable on an assignment whose source account uses a user-entered cost allocation driver.

You enable this property by checking the **Allow user-entered cost allocation** check box on the **Advanced** tab of the New Driver window.



This property is displayed on the drivers page as **UECostAllocation**:

System generated or user entered?	User-modifiable, but not in a column layout.
Default format in a column layout:	Checkbox
Data type:	Boolean
Type of property in an assignments pane:	N/A
Formula	N/A

Variable Driver Quantity Override

For a sequenced driver, the name of the property that replaces the Driver Quantity Variable value after the previous sequence pass.

System generated or user entered?	User modifiable, but not in a column layout.
Default format in a column layout:	Text
Data type:	The maximum length is 64 alphanumeric Unicode characters.
Type of property in an assignments pane:	N/A

Formula	N/A

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

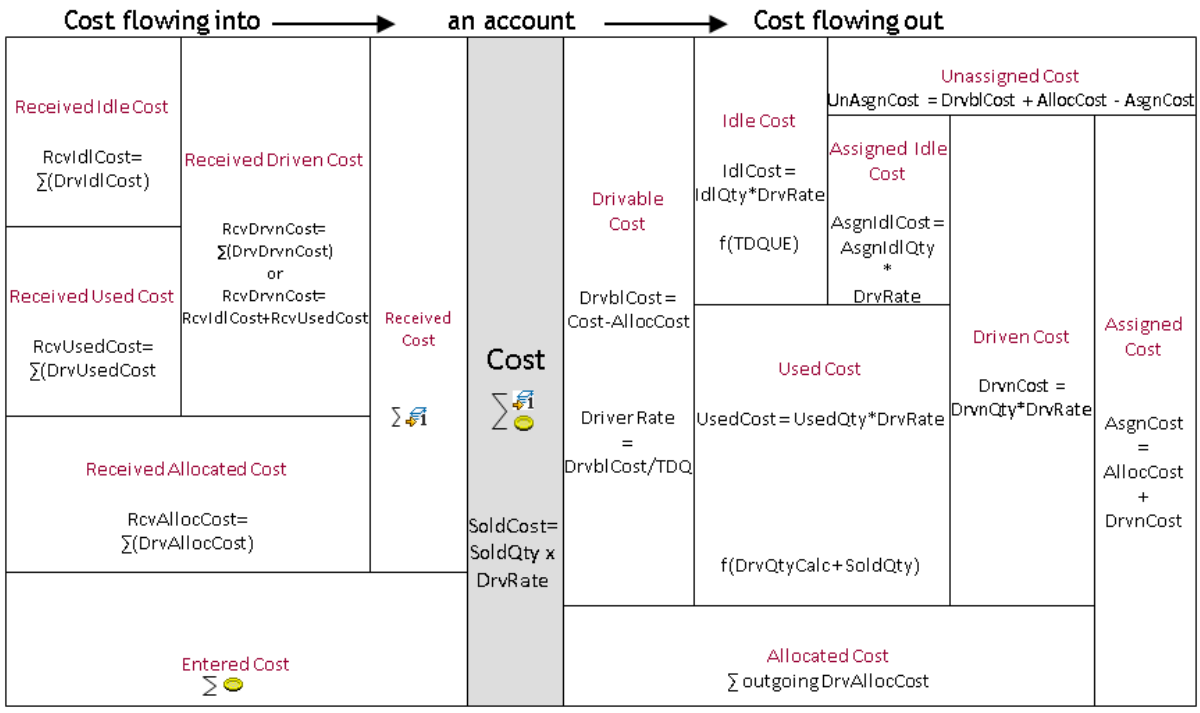
- "Driver Sequencing" on page 407
- Fixed Driver Quantity Override on page 770

Chapter 91

Property Relationship Diagrams

Cost Flowing Into and Out of an Account (Non-Reciprocal)

Incoming and Outgoing Cost



Legend:

Entered Cost Element

See "Entered Cost Elements" on page 260.

Received Cost Elements

1 2 3 See "Received Cost Elements" on page 262.

See Also

- "How to Read these Diagrams" on page 810
- Chapter 60, "Properties in Calculation," on page 527

Cost Properties on the Source Side

```
Drivable Cost = (Cost - Allocated Cost)

Driver Rate = Drivable Cost/TDQ

Drivable Cost = (Used Cost + Idle Cost)

Idle Cost = (Assigned Idle Cost + Unassigned Cost)
```

```
Driven Cost = (Used Cost + Assigned Idle Cost)

Assigned Cost = (Driven Cost + Allocated Cost)

Unassigned Cost = Cost - Assigned Cost
```

Cost and Quantities Math on the Source Side

```
Driver Rate = Cost/TDQ (or TDQUE)

Used Cost = UsedQty*Driver Rate

IdleCost = IdleQty*Driver Rate

IdlePercentage = IdleQty/TDQ
```

Costs on the Destination Side

```
Received Driven Cost + Received Allocated Cost = Received Cost

Received Cost + Entered Cost = Cost
```

Costs on an Assignment

```
Driver Driven Cost = (Driver Used Cost + Driver Idle Cost)

Driver Cost = (Driver Driven Cost + Driver Allocated Cost)
```

Cost and Quantity Math on an Assignment

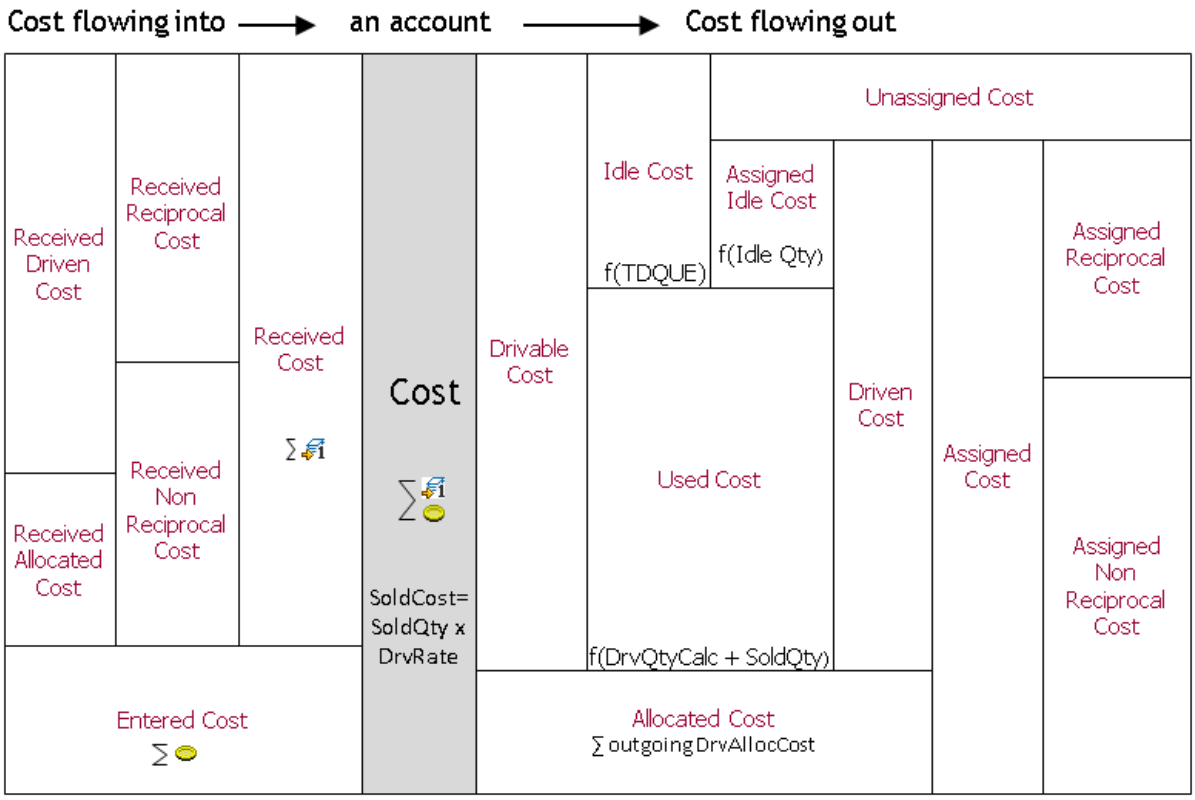
```
Driver Used Cost = DriverQtyCalc * Driver Rate

Driver Idle Cost = Idle DriverQty * Driver Rate

Driver Driven Cost = (Driver Used Cost + Driver Idle Cost) or
(DrvDrvnQty * Driver Rate)

Driver Cost = (Driver Driven Cost + Driver Allocated Cost)
```

Some Cost and Quantity Math in a Nutshell

```
TDQ = IF TDQUE is not null THEN TDQUE ELSE UsedQty

OutputQty = IF OutputQtyUE is not null THEN OutputQtyUE ELSE
UsedQty

DrivableCost = Cost - AllocatedCost

DriverRate = DrivableCost/TDQ

UnitCost = DrivableCost/OutputQty

UnitCost = IF OutputQtyUE is not null THEN DrivableCost/
OutputQtyUE ELSE DrivableCost/UsedQty

UnitRevenue = Revenue/OutputQty
```

```
UnitRevenue = IF OutputQtyUE is not null THEN Revenue/
OutputQtyUE ELSE Revenue/UsedQty

UnitProfit = Profit/OutputQty

UnitProfit = IF OutputQtyUE is not null THEN Profit/
OutputQtyUE ELSE Profit/UsedQty
```

Cost Flowing Into and Out of an Account (Reciprocal)



Legend:

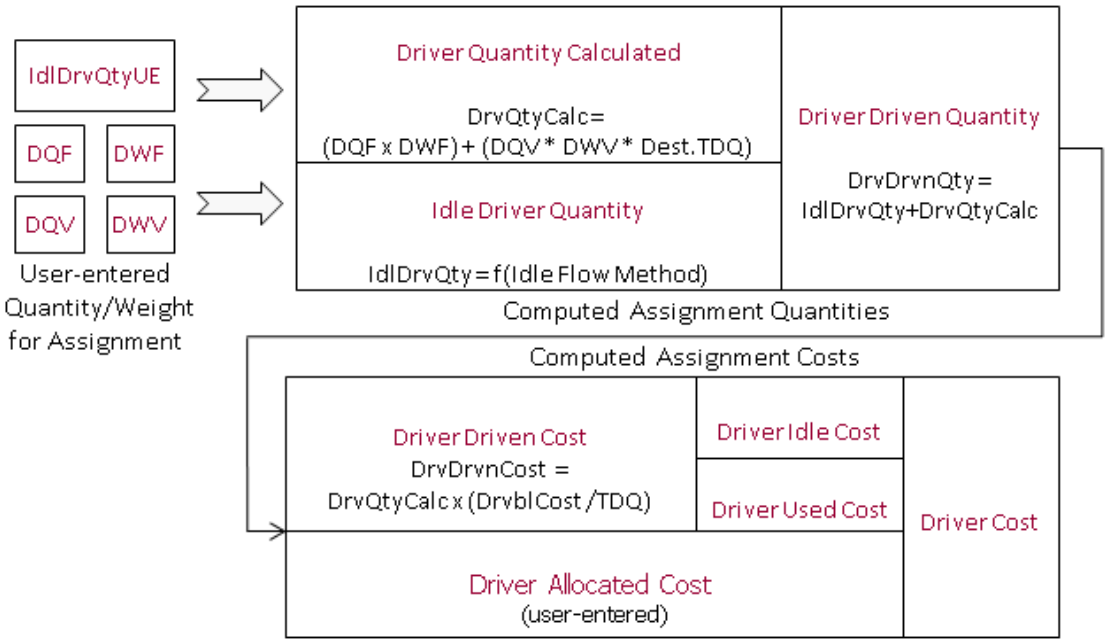
Entered Cost Element

See "Entered Cost Elements" on page 260.

Received Cost Elements

See "Received Cost Elements" on page 262.

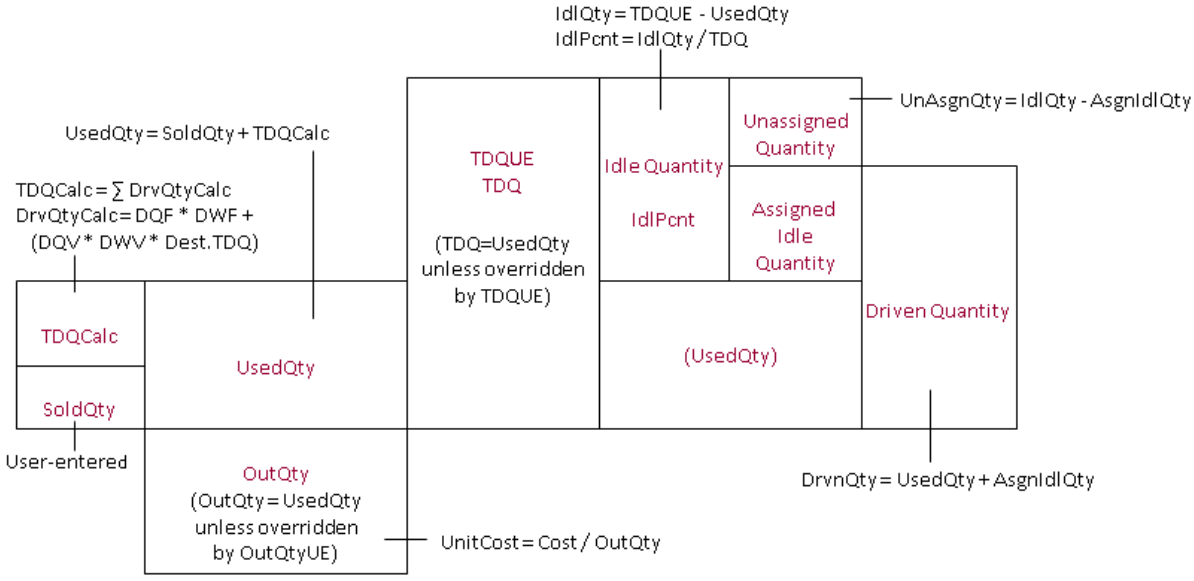
Assignment Properties



See "How to Read these Diagrams" on page 810.

Account Driver Properties

Account Driver Quantities



See "How to Read these Diagrams" on page 810.

Quantities on an Assignment

```
DrvQtyCalc = (DQF * DWF) + (DQV * DWV * Dest.TDQ)
DQBasic = DQF + DQV* Dest.TDQ
Idle Driver Quantity = Idle Driver Quantity UE
```

Cost and Quantity Math on an Assignment

```
Driver Used Cost = DriverQtyCalc * Driver Rate
Driver Idle Cost = Idle DriverQty * Driver Rate
Driver Driven Cost = (Driver Used Cost + Driver Idle Cost) or
(DrvDrvnQty * Driver Rate)
Driver Cost = (Driver Driven Cost + Driver Allocated Cost)
```

Quantities on the Outgoing Side

```
UsedQty = (TDQCalc + SoldQty)
DrivenQty = ∑ Assigned Idle Quantity + ∑ Used Qty
IdleQty = (TDQUE - OutputQty) (TDQUE and OutputQtyUE override
respective quantities)
```

If `TDQUE > OutputQty` then `Positive Idle`

If `TDQUE < OutputQty/OutputQtyUE` then `Negative Idle`

Cost and Quantity Math on the Source Side

`Driver Rate = DrivableCost/TDQ` (or `DrivableCost/TDQUE`)

```
DrivableCost = Cost - AllocatedCost
Used Cost = UsedQty*Driver Rate
IdleCost = IdleQty*Driver Rate
IdlePercentage = IdleQty/TDQ
```

Idle Cost Quantity Math on an Assignment

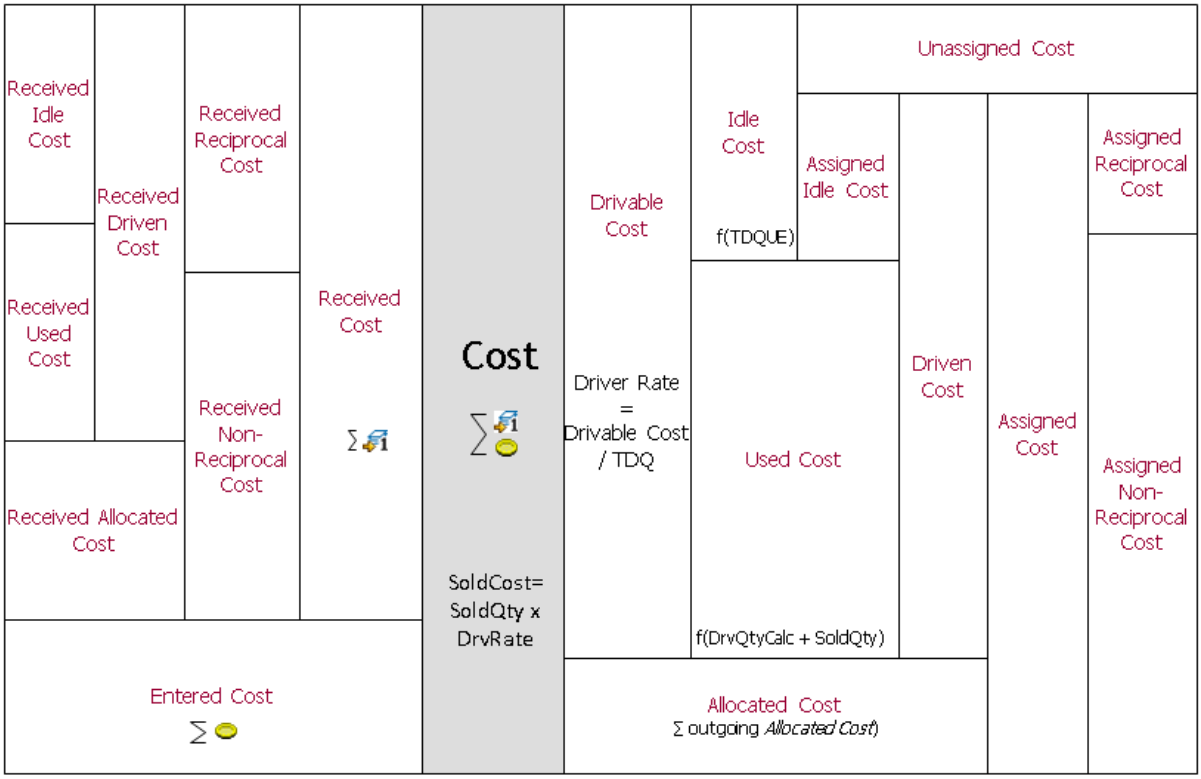
```
IdleDrvQty = IdleQtyUE
DrvDrvnQty = DrvQtyCalc + IdleDrvQty
DrvUsedCost = DrvQtyCalc * Driver Rate
DrvIdleCost = IdleDrvQty * Driver Rate
DrvDrvnCost = DrvDrvnQty * Driver Rate
DrvDrvnCost = (DrvUsedCost + DrvIdleCost)
```

```
DrvCost = (DrvDrvnCost + DrvAllocCost)
```

Idle Cost Quantity Math on the Source Side

```
DrvRate = DrvbleCost /TDQ

IdleQty = TDQUE-UsedQty

IdlCost = IdleQty * DriverRate

Asgn IdlCost = (sum of DrvIdleCost of destination side
accounts)

UnassignedCost = (IdleCost - AsgnIdlCost) or (Cost - Assigned
Cost)

AssignedCost = (DrivenCost + AllocCost)
```

Combined Account Cost Properties



See "How to Read these Diagrams" on page 810.

Legend:

Entered Cost Element


See "Entered Cost Elements" on page 260.

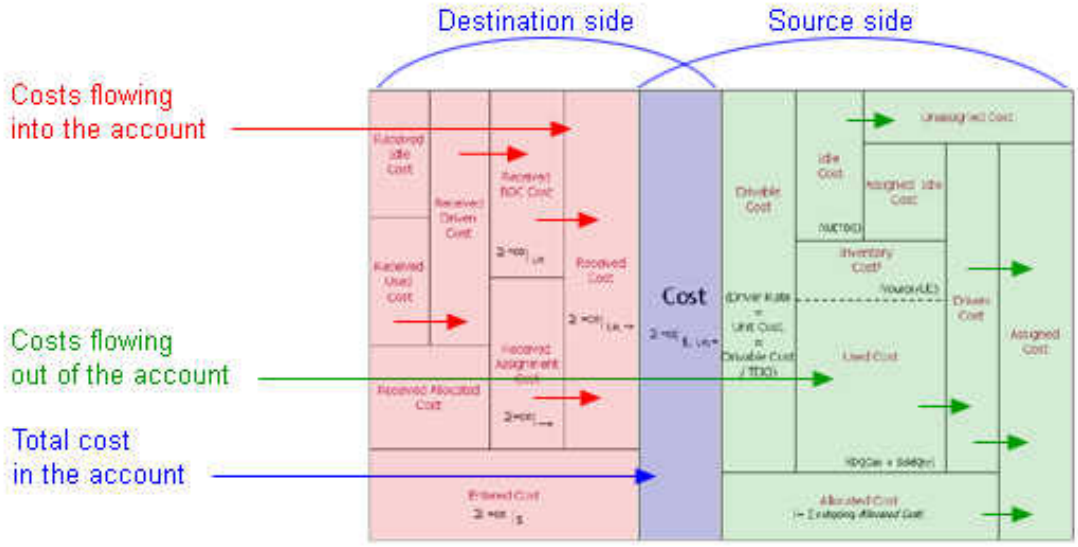
Received Cost Elements


See "Received Cost Elements" on page 262.

How to Read these Diagrams

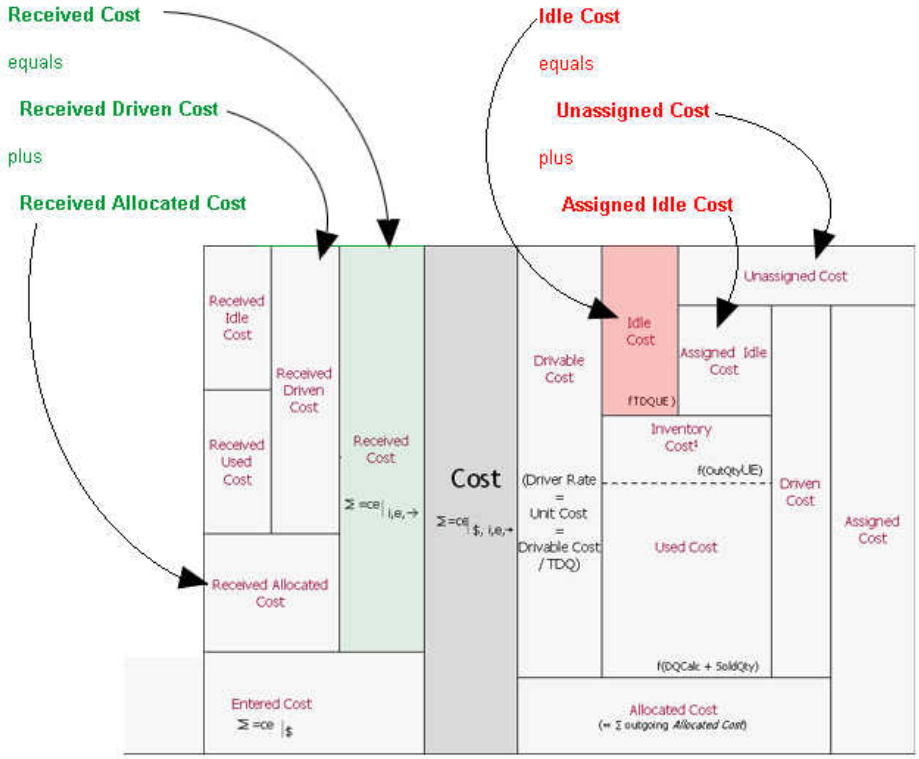
A Single Account

The diagrams show costs flowing into and out of a single account. Read the chart from left to right. The diagram shows costs coming in from the left, passing through from left to right, and going out on the right.



X and Y Axes

The vertical Y-axis represents a fixed quantity of cost along the entire horizontal X-axis. So, the quantity between any two parallel horizontal lines is constant along the horizontal X-axis. The following picture should make this clear:


Received Cost
equals
Received Driven Cost
plus
Received Allocated Cost
Idle Cost
equals
Unassigned Cost
plus
Assigned Idle Cost
Received Idle Cost
Received Driven Cost
Received Used Cost
Received Cost
Received Allocated Cost
Entered Cost
Cost
Drivable Cost
Idle Cost
Assigned Idle Cost
Unassigned Cost
Inventory Cost
(Driver Rate = Unit Cost = Drivable Cost / TDQ)
Used Cost
Driven Cost
Assigned Cost
Allocated Cost

Index

D

E

M

N

O

P

Q

R

S

T

U

V

W