THE POWER TO KNOW.

SAS® Activity-Based Management 7.1

User's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® Activity-Based Management 7.1: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Activity-Based Management 7.1: User's Guide

Copyright © 2010, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-59994-904-8

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

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

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

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

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

1st electronic book, July 2010
1st printing, July 2010

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

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

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

Contents

Part 1

Getting Started

Chapter 1

Overview

Activity-Based Management

Activity-based management is a system that enables an organization to manage activities and processes. Once these aspects are managed, organization performance and value can improve. By applying direct and indirect business costs to activities, SAS Activity-Based Management enables managers to get a true understanding of the costs and profits that are associated with a product, customer, service, or business process. SAS Activity-Based Management supports ongoing profitability analysis, cost-management initiatives, shared services management, planning and budgeting efforts, and capacity optimization.

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

Activity-Based Costing

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 report 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 Activity-Based 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.

SAS Activity-Based Management

Benefits

With SAS Activity-Based 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.

SAS Activity-Based Management guides you through the steps for Web-enabled data analysis and reporting from a model. With access over the Internet to interactive views of a model, to interactive analysis capabilities, and to customizable reports, decision-makers can complete the following tasks:

- learn the true costs of producing and delivering products or services
- identify areas in which profits could be increased
- examine processes and act to improve them

Licenses

Overview

SAS Activity-Based Management is sold with either an unlimited site license or with any number of the following licenses:

- Analyst Power User
- Business User

In very simple terms, someone with an Analyst Power User license can create models and generate cubes, whereas someone with a Business User license can view cubes. However, to completely understand the licenses, it is necessary to understand how user permissions are granted in SAS Activity-Based Management 7.1. SAS Activity-Based Management now uses the same permission scheme that is used by all of SAS Business Intelligence applications. To understand this scheme, it is necessary to understand the following entities:

- Capabilities
- Roles
- Groups
- Users

Capabilities

Capabilities are the operations of SAS Activity-Based Management that a user can do. There as nine capabilities:

- Create Models
- View Models
- Create Cubes
- View Cubes
- Create Reports
- View Reports
- View Contributions
- Audit
- Administer

Capabilities are not granted to users directly. To understand how users acquire capabilities, it is first necessary to understand roles and groups.

Roles

A role is assigned one or more capabilities. A role can also have any number of groups as members. A group inherits the capabilities of each role of which it is a member. There can be any number of roles at a SAS installation, and each role can have any number of capabilities.

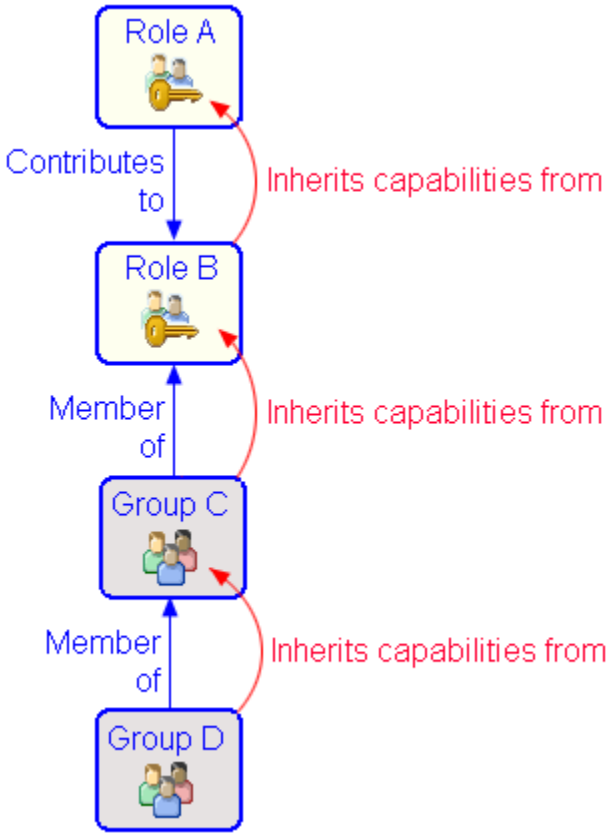
Note: A role can also contribute its capabilities to another role, in which case a group that is a member of this latter role inherits the capabilities of both roles—the role of which the group is a member and the role that contributes to the other role.

Groups

A group can have any number of users as members. A user who is a member of a group inherits the capabilities of the group. There can be any number of groups at a SAS installation, and each group can have any number of users.

Note: A group can also be a member of another group, in which case a user that is a member of this latter group inherits the capabilities of both groups—the group of which the user is a member and the group that this group is a member of.

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



On installation of SAS Activity-Based Management, two groups are created automatically:

- Activity-Based Management Users
- Activity-Based Management Cube Creators

These two groups are important for the following reasons:

1. Any user who wants to use SAS Activity-Based Management 7.1 must be a member, directly or indirectly, of the Activity-Based Management Users group.

   Note: A user is indirectly a member of that group if the user is a member of a group that is a member of the Activity-Based Management Users group (or of a group that is a member of a group that is a member of the Activity-Based Management Users group—and so forth.)

2. Any user who wants to create SAS OLAP cubes must be a member, directly or indirectly, of the Activity-Based Management Cube Creators group. (Membership in this group is not necessary for creating Microsoft Analysis Services cubes.)

   Note: The Activity-Based Management Cube Creators group is a member of the Activity-Based Management Users group—so members of the Activity-Based Management Cube Creators group can use SAS Activity-Based Management.

A user with the capability to create an entity (model, report, cube) who creates such an entity becomes its owner. The owner of an entity can give Read or Read/Write access to that entity to members of a group. Users who receive group access to an entity thereby acquire access to other SAS Activity-Based Management features related to the entity. For a complete list of the features of SAS Activity-Based Management that the capabilities and group access provide access to, see the topic “Table of Capabilities and Model Access” in the online help.

For information on creating additional roles, groups, and users at your installation, see the section "Creating SAS Activity-Based Management Users" in the SAS Activity-Based Management 7.1: Installation, Migration, and Configuration Guide. Also see the instructions in the online help.

Now you are able to fully understand the licenses.

Business User

A Business User license gives its holder license to acquire any of the nine capabilities except for Create Models. Its holder can be a member of any group with any or all of the other eight capabilities.

Analyst Power User

By contrast, the Analyst Power User license enables its holder to be a member of any group with any or all of the nine capabilities—including, in particular, the Create Models capability which gives a user the ability to create a model and to generate cubes from it.

Migration from a Previous Release

If a customer migrates from a previous release of SAS Activity-Based Management to SAS Activity-Based Management 7.1, then the following three groups are created automatically during migration:

- Activity-Based Management Administrators
- Activity-Based Management Modelers
- Activity-Based Management Viewers

Each group has the capability that is shown in the following table.

Group	Capability
Activity-Based Management Administrators	Administer
Activity-Based Management Modelers	Create Models
Activity-Based Management Viewers	View Models

Note: Each of these groups is a member of the SAS Activity-Based Management Users group—so any member of any of these three groups can use SAS Activity-Based Management. The Activity-Based Management Modelers group is also a member of the Activity-Based Management Cube Creators group. Therefore, members of the Modelers group can create SAS OLAP cubes.

During migration, each business user in a previous release of SAS Activity-Based Management becomes a member of the Activity-Based Management Viewers group. Each such user must have a Business User license for SAS Activity-Based Management 7.1.

Each Modeler in a previous release of SAS Activity-Based Management becomes a member of the Activity-Based Management Modelers group. Each such user must have an Analyst Power User license for SAS Activity-Based Management 7.1.

Chapter 2

Working in SAS Activity-Based Management

Connecting to a SAS Environment

Overview

When you log on to SAS Activity-Based Management, the file **`environments.ini`** (whose default installation location is C:\Program Files\SAS \SASActivityBasedManagementClient\7.1) provides the URL of the file **`sas-environments.xml`**. This latter file associates each entry in the SAS environment drop-down list of the Log On dialog with the URL of a SAS WIP Server (Web Instrstructure Platform). The WIP Server authenticates your logon information and provides an interface to a SAS Activity-Based Management Metadata Server. Because you, as a client user, log on to a WIP Server instead of directly to a SAS Activity-Based Management Metadata Server, the Metadata Server can be changed without affecting your client log-on procedure.

Current Credentials

Your SAS Activity-Based Management user name and password might not be the same as your network user name and password. If they are the same, then you can use your network logon information, also known as your current credentials, to connect to the SAS environment.

If your current credentials are not applicable to the server, then you must supply your domain and user name.

Model Availability

Sometimes, you might not be able to perform certain tasks in a model. For example, if you are a Designer and you attempt to edit a part of the model that another Designer is simultaneously editing, you will not be able to edit. However, you will always be able to view model data.

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. For example, if you create a report, the costs in the report should reflect a specific point in time. You do not want someone to change the costs while the report is being generated. Likewise, when SAS Activity-Based 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 Activity-Based Management prevents changes from being made during the following tasks:

- validating a model
- creating a report
- calculating costs
- generating cubes
- exporting data

Saving Data

SAS Activity-Based Management is an application that 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 actively save data, data is saved automatically.

  Automatic saving enables all users to see the most current data immediately. (The availability of data is controlled by 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 Activity-Based Management writes directly to the model.

  Writing directly to the model 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 current model before you make changes, you can export the model before you change data in the model. Then, if necessary, you can revert to the previous version of the model by importing the model.

Organization of Features

Modes

SAS Activity-Based Management groups features by the following categories:

- Workspace Manager

  The Workspace Manager provides a treeview of all SAS Activity-Based Management elements— from models to cube configurations to reports — and gives you access to them all. You can start frequently used tasks. For more information, see Chapter 3, "Workspaces and Workspace Manager," on page 17.

- Model mode

  In Model mode, you perform the following tasks:

  - build a model and edit a model on the Resource module page, the Activity module page, and the Cost Object module page
  - manage external units on the External Units module page
  - manage attributes on the Attributes page
  - manage drivers on the Drivers page
  - manage dimension members on the Dimension page

- OLAP mode

  In OLAP mode, you use cubes on the OLAP page to analyze data. For more information, see Chapter 13, "Analyzing Model Data with OLAP," on page 89.

- Reports mode

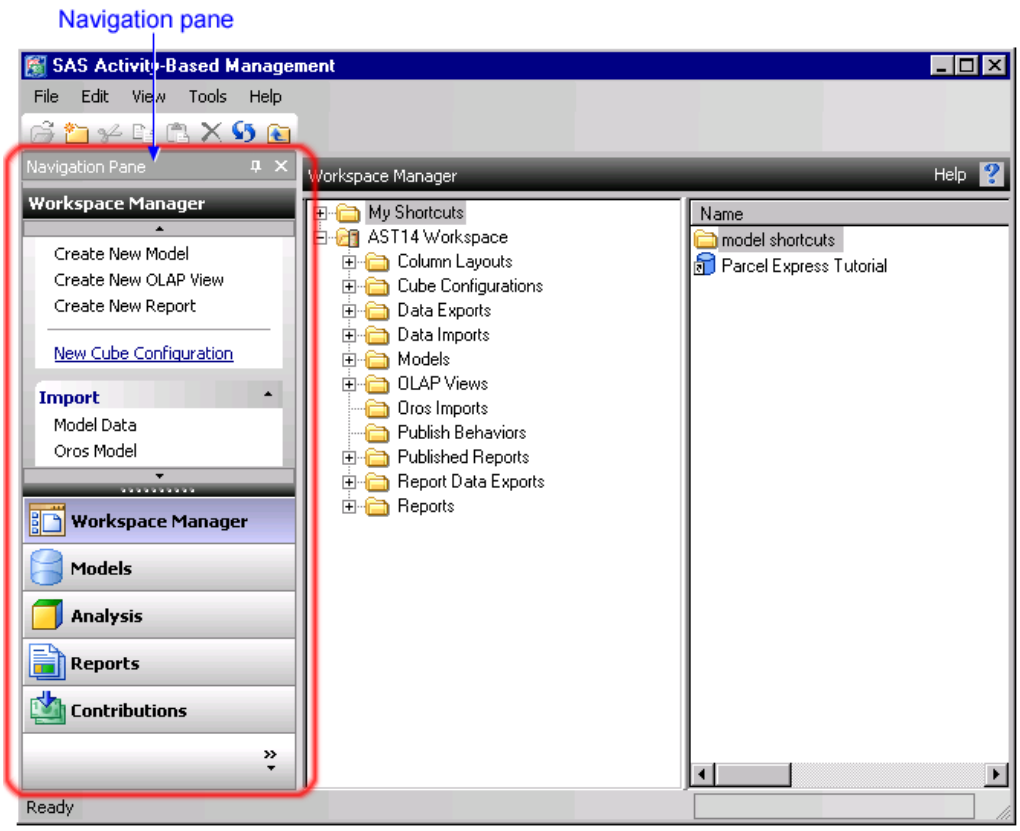
  In Reports mode, you open or create a report configuration. For more information, see Chapter 14, "Reporting Model Data," on page 95.

- Contributions mode

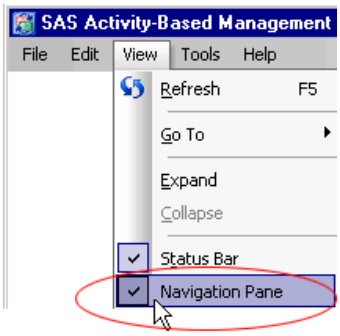
  In Contributions mode, you perform ad-hoc queries on a "cube on the fly" to view cost flows throughout a model. For more information, see Chapter 29, "Working with Querying Contributions," on page 195.

Changing Modes with the Navigation Pane

To change modes, you use the Navigation Pane. The Navigation Pane allows you to navigate among the workspaces that constitute SAS Activity-Based Management.



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

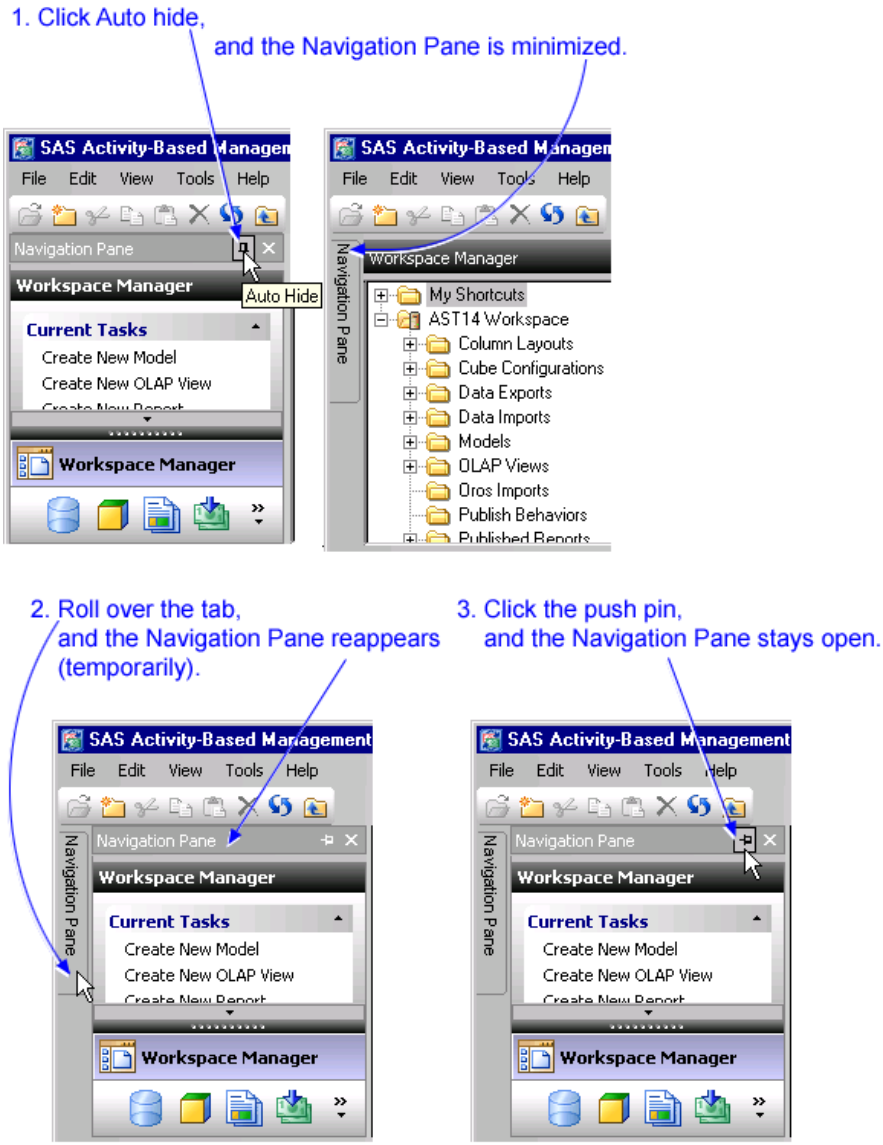


Minimizing the Navigation Pane

1. Click the Auto hide icon 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 (the push-pin) to make the Navigation Pane stay open (or select **View** ⇨ **Navigation Pane**).



Partitioning 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.

When you change modes, a new menu appears for that mode between the **View** menu and the **Tools** menu.

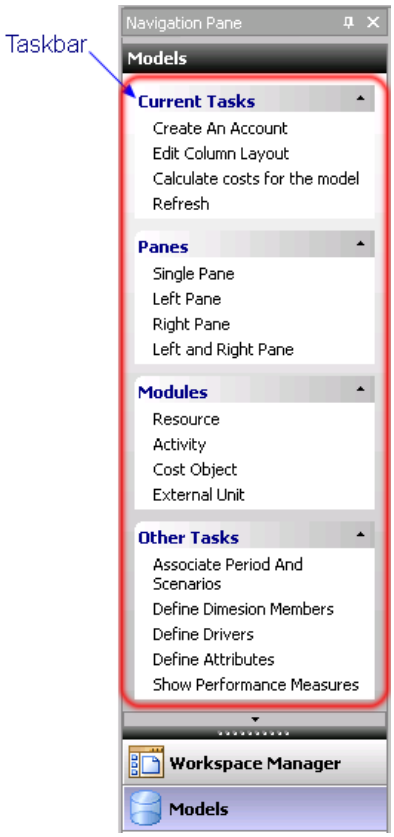
For example, the **Reports** menu appears when you are in Reports mode.

Note: Each mode can display a different model or a different period and scenario association. When you first change to another mode, you can select a model and a period and scenario association.

Task Bars

Each workspace has a taskbar that shows you the most frequently used operations that are related to the workspace.

For example, the taskbar for the Models workspace contains these operations:



Status Bar

The status bar, which is at the bottom of the window, displays informational messages, such as whether the current period and scenario association is published. The status bar also displays the tasks that are currently being performed. From the status bar, you can manage server tasks.

Refreshing Information

As Designers work in SAS Activity-Based Management, some information is updated automatically. However, this information might not be updated immediately. Whenever the refresh icon turns red (), new information is available and you can refresh the information.

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

Viewers can see changes to global items, such as exchange rates and period/scenario associations, only after they restart SAS Activity-Based Management.

Canceling a Server Task

You can cancel a SAS Activity-Based Management server task that you initiate. However, after you cancel a task, the model might be in an unusable state.

You might need to cancel a server task if you realize that the task requires too much time or when you start an operation by mistake.

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

The Work Flow

Overview

By storing model data in a structure that is optimized for analytical purposes, SAS Activity-Based Management provides fast and intuitive analysis of broad trends and relationships. SAS Activity-Based Management is designed specifically for analyzing data. The following design features enable SAS Activity-Based 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.

The following is an overview of the general work flow.

Enter Business Data

1. Plan periods and scenarios. Set up periods and scenarios for each SAS Activity-Based Management server.

   On each server, all 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. Build a model interactively or import model data.
3. Calculate costs and generate cubes.

Analyze Business Data

- Use modules, cubes, and reports to analyze business data.

Make Business Analyses Available to Others

- Set up and save column layouts, OLAP views, published reports, and report configurations.
- Export cubes and reports.

Chapter 3

Workspaces and Workspace Manager

Workspaces and Workspace Items

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

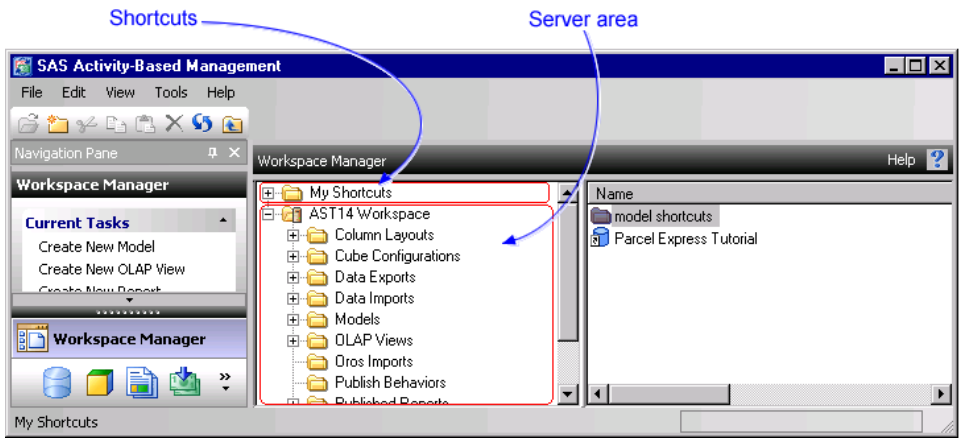
Workspace Manager

Work Panes

Workspace Manager is your personalized view of all your workspaces. Workspace Manager enables you to select and organize items to fit your needs.

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

Workspace Manager displays information in two panes, as shown here:



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 all of the items on a SAS Activity-Based 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 system administrator for SAS Activity-Based Management, you see every item, regardless of who owns the item, and you can interact with every item.

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: You cannot change the model that is associated with an item. To avoid the error message about a missing model, delete the workspace item, and 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. You can use a shortcut to open and use an item.

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

My Shortcuts holds only folders and shortcuts. Each user of SAS Activity-Based Management has a different My Shortcuts, 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.

Naming Guidelines for Workspace Items

Your organization can develop its own naming guidelines. However, these guidelines must conform to the SAS Activity-Based Management naming conventions. For more information, see the Help.

Ownership and Permissions for Server Area Items

Overview

When you create a server area item, SAS Activity-Based Management assigns ownership to you. You can modify the item, rename it, or move it. You can assign permissions to other users, based on the users groups.

To each group, 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.

To assign permissions to a group, you must be a member of that group. This restriction prevents sensitive company data from unintentional publication. If you need to assign permissions to a group of which you are not a member, ask the SAS Activity-Based Management system administrator to temporarily assign you to the group.

Part 2

Modeling Concepts

Chapter 4

Models

Introduction

The basic container for activity-based management information in SAS Activity-Based Management is the model. A meaningful activity-based management model reflects the organization that it is modeling and uses terms that are familiar to the people who work at the organization. Therefore, a good understanding of your organization's resources, activities, and cost objects is required.

A model contains two types of information: the structure of the model, and the data for specific amounts of time (periods) and for specific circumstances (scenarios).

For more information, see Chapter 5, “Periods and Scenarios,” on page 29.

Structural Information

The structural elements of a model should be named after elements that are present in the organization's environment.

For example, an organization's general ledger account names, such as Wages and Depreciation, can be used to name and reference the resource accounts in the activity-based management model; the hierarchy of processes in an organization can be applied to the activity accounts in the model. For more information, see Chapter 7, “Modules,” on page 43.

Structural information includes the following:

- accounts to hold costs and information that pertain to resources, activities, and products.

  For more information, see Chapter 8, “Accounts and Cost Elements,” on page 51.

- assignment paths that assign resource expenses to activities, and activity costs to products.

  For more information, see Chapter 10, “Assignments,” on page 75.

- drivers for measuring the frequency or intensity of demands that are placed on resources by activities, and on activities by products.

  For more information, see Chapter 9, “Drivers,” on page 57.

- lists of components that products consist of.

Structural information can change for different periods of time. For example, if your organization changes products or activities seasonally, you can reflect this seasonality in the model.

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 reports or 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 details of analysis that users perform after the model is built depend on the structure that you define.

- How well do you know your organization?

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

- 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. Include enough detail to generate any required reports.

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.

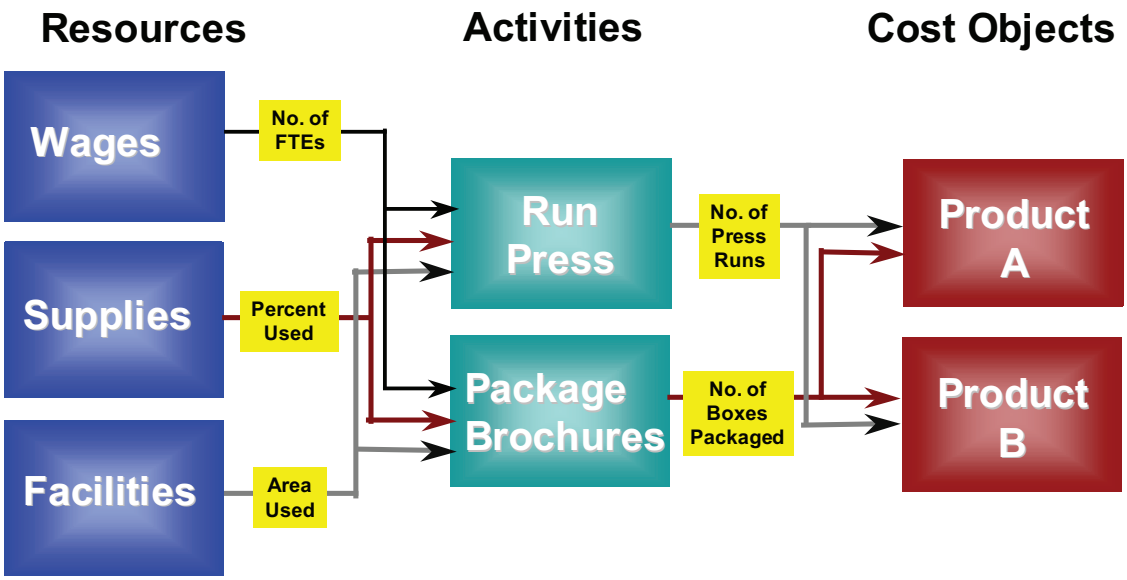
Communicate often with those people who will use the information provided by the model. Are you including the information that meets their needs? Is the amount of detail enough? Do these people need to analyze the data in ways you have not accommodated in the model?

The best method to design a model is to create a paper model before attempting to build the model in SAS Activity-Based 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 and reports
- gathering a panel of experts or focus groups
- observing people and work flow

Guidelines for Sharing Models, Configurations, and Data

Overview

Before you build a model and populate it with data, your organization should establish guidelines for when data is saved, the names used to save items, and how data is managed.

After you build a model and analyze it, you will have data that helps you make business decisions (or that helps others in your organization make business decisions). You will want to share your results (model data, analysis results) so that coworkers can use the data that you have collected. However, because a significant amount of 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.

For example, Workspace Manager lists items that are available for all SAS Activity-Based Management users who are using the same server. Without a naming guideline, items within Workspace Manager might be difficult to locate. For more information, see Chapter 3, “Workspaces and Workspace Manager,” on page 17.

Saving Items

Establish guidelines for when to save and when not to save column layouts, OLAP views, published reports, and report configurations.

Naming Items

To manage the lists of items that are on a SAS Activity-Based Management server, establish naming conventions. Decide whether a name should include the Designer's initials. 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 of 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

Naming Report Configurations

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

- the description that was entered when the report configuration was saved
- the date on which the report configuration was saved

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 pre-defined OLAP cube
- the network logon information of the person who saved the OLAP view

Sharing Data with Others

For people who do not use SAS Activity-Based Management, you can export OLAP cubes to a Microsoft Excel spreadsheet, and you can export reports to popular application formats.

Calculating Costs

Overview

As you begin entering data from the paper model into the model in SAS Activity-Based Management, you should notice how the costs are conforming to expectations. At any time during the development of the 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 model's resource account references match the general ledger's account numbers, it is more efficient to add costs to the model by importing them from the general ledger. For more information, see Chapter 17, “Importing Data and Exporting Data,” on page 107.

When you calculate costs, you can choose a specific period/scenario association, or you can choose all associations. For more information, see Chapter 5, “Periods and Scenarios,” on page 29.

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
- adding, deleting, or changing the cost of an external unit cost element in a bill of costs
- changing the cost of an account that is used as an internal unit cost element in a bill of costs

- changing an account's driver, driver quantities, attribute quantities, or output quantities
- adding or deleting a roll-up account, an account, or a cost element
- creating new assignments or deleting existing ones

In all of these actions, costs are updated if you calculate costs. However, it might not be necessary to calculate costs for a minor update.

Errors

You should choose to report all errors at least once while you are calculating costs and generating OLAP cubes. Fix any significant errors that are reported. If you determine that the remaining errors are insignificant, you can then choose to not report all errors. By limiting the number of errors that are reported, you might improve performance when costs are calculated and when OLAP cubes are generated.

Chapter 5

Periods and Scenarios

Introduction

A period can represent any unit of time: 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.

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

Setting Up Periods and Scenarios

Overview

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

Note: Because periods and scenarios are independent of a particular model, you can create them without a model being open. Further, you can define periods and scenarios before you create the first model.

Some scenarios can be agreed upon and can be set up before people start to use SAS Activity-Based Management. Other scenarios can be set up 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.

On the OLAP page, SAS Activity-Based Management automatically rolls up values for each period and scenario that are in the hierarchy above the hierarchy level that contains data. This rolled-up data is not available for reports. For more information, see Chapter 13, "Analyzing Model Data with OLAP," on page 89.

Period Start Dates and End Dates

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

A period must meet these criteria:

- 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.

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 in the next higher hierarchy level.

Organizing Periods and Scenarios

You can create a hierarchy of periods, such as FY2005 > Q1 > January, and a hierarchy of scenarios, such as Budget > Aggressive. By default, each hierarchy level is given a name such as Period L1 or Scenario L1. However, these names are not descriptive during OLAP analysis. So, you can rename period levels and scenario levels if necessary.

Deleting Periods or Scenarios

If a period or scenario is used in a period/scenario association, you cannot delete the period or scenario. You must first delete the period/scenario association in each model, and then you can delete the period or scenario.

Guidelines for Creating Periods and Scenarios

Before you create periods and scenarios, consider the following:

- You can copy data from one period/scenario association to another period/scenario association. Copying data from one period/scenario association to another period/

scenario association enables you to propagate association data to similar period/scenario associations.

- Model structure is period-specific.

  Each period/scenario association can have a unique account and assignment structure. This structure could reflect seasonal products, regional variations, or planning scenarios.

Period/Scenario Associations

Overview

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.

If you are a Viewer, then the **Period/Scenario** drop-down lists in SAS Activity-Based Management show only period/scenario associations that have been published. For more information about a published period/scenario association, see the section "Period/Scenario Associations" on page 31.

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 be changed. In fact, the entire model structure might disappear if the new period/scenario association contains no model structure data.

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 period/scenario associations.

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 Viewers to see because the Designer is in the process of entering the data.

Once the data for a period/scenario association is entered and calculated, the association is Published. This status enables Viewers to see and analyze the data in that period/scenario association.

Typically, a Designer publishes a period/scenario association at the end of a reporting period, such as at the end of a fiscal quarter.

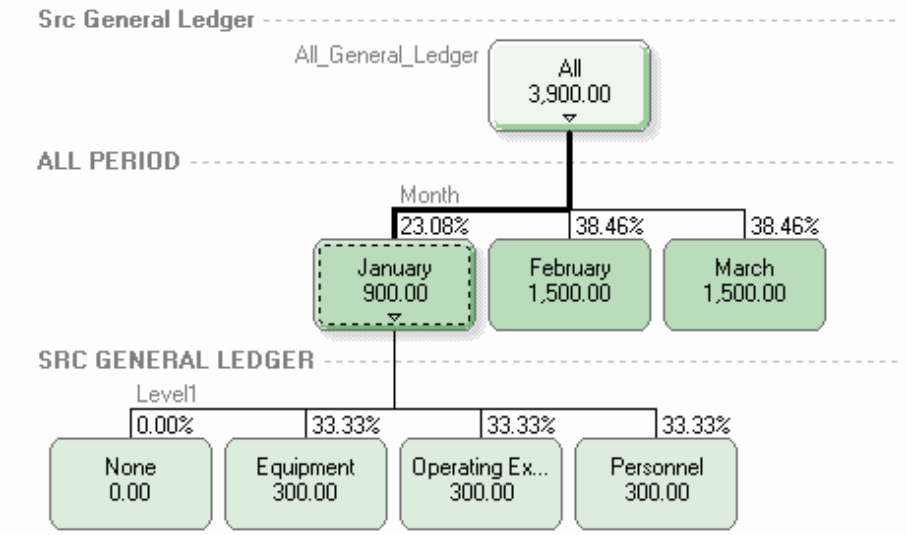
Note: As long as a period/scenario association has a status of Published, the data within the association cannot be edited.

Periods and Scenarios in OLAP Analysis

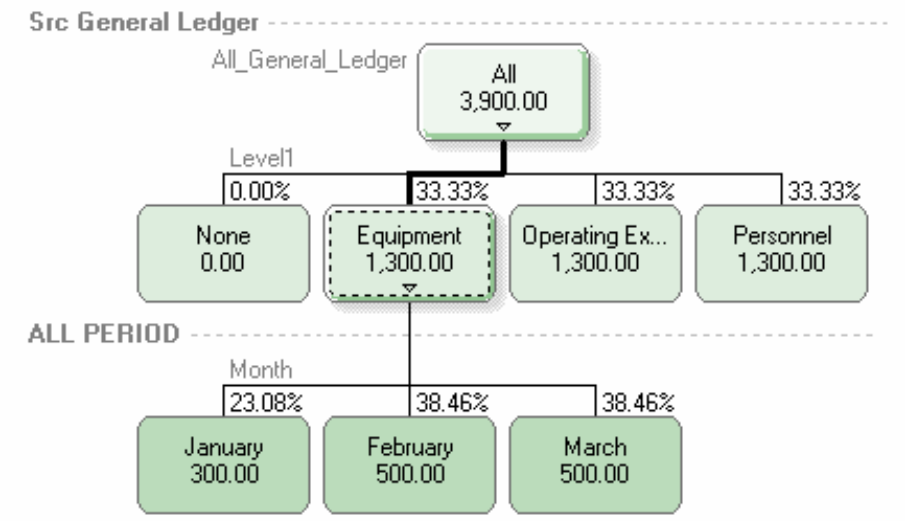
Periods and scenarios are dimensions and can be used by business users for OLAP analysis. Periods and scenarios can aggregate data or separate data. (For more information about dimensions and how to organize them for OLAP analysis, see Chapter 6, “Dimensions,” on page 35.)

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 page. 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 6

Dimensions

Introduction

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.

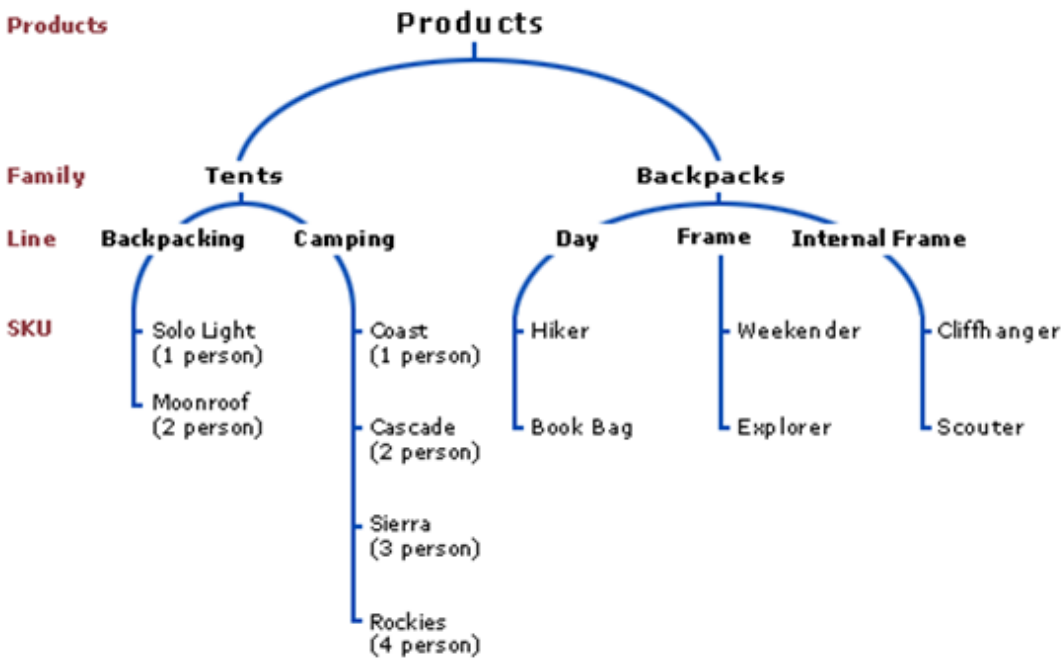
To further illustrate, suppose that a manager says, "I need to see the data items x, y, and z grouped by a, by b, and by c." Data items x, y, and z represent measures, and a, b, and c represent dimensions. For example, perhaps the manager needs to see revenue, cost, and profit grouped by region, by customer, and by product. For more information, see Chapter 13, “Analyzing Model Data with OLAP,” on page 89.

Dimension Levels

To present data in a manageable 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.

For example, in the following 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).



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 to reveal more detail. For example, a Viewer might first ask to see product costs for the past three fiscal years. The Viewer might notice that the costs for 2001 are higher than the costs for the other years. The Viewer 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.

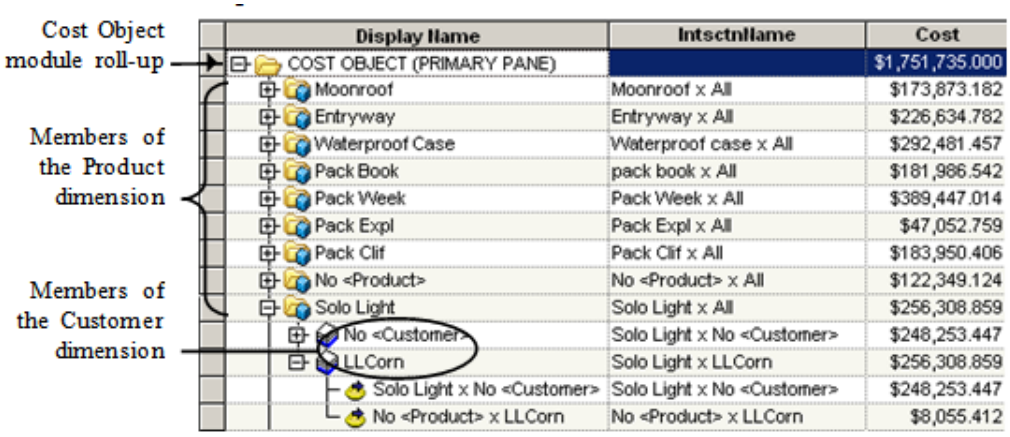
Dimension Members

Dimension members are the unique elements in the dimension levels. In the previous example, Backpacking, Frame, and Hiker are examples of dimension members.

All and No <dimension name> Displayed in Model Mode

To help understand what you see on a Model page in SAS Activity-Based Management, you can add a column to display the Intersection Name property. The intersection name lists the dimension members in the dimension order from left to right.

In the following illustration, the Cost Object module roll-up cost represents all product costs and all customer costs. 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** roll-up 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** roll-up account represents all the product costs of Solo Light, which includes the costs for all customers and all other costs.



Display Name	IntsctnName	Cost
COST OBJECT (PRIMARY PANE)		$1,751,735.000
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
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

As you drill down into the **COSTOBJECT** module roll-up, 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.

For information about how All and No are represented on the OLAP page, see Chapter 13, “Analyzing Model Data with OLAP,” on page 89.

Types of Dimensions

There are two types of dimensions in a model: structural and attribute.

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 dimensions that uniquely identifies a model is the dimension signature. For more information, see Chapter 7, “Modules,” on page 43.

As a Designer, 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 (the business users) 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 business 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 automatically created whenever dimension attributes are created. You cannot explicitly create an attribute dimension. For information, see Chapter 11, "Attributes," on page 81.

The SAS Activity-Based Management OLAP tool makes no distinction between attribute dimensions and structural dimensions. For more information, see Chapter 13, "Analyzing Model Data with OLAP," on page 89.

Guidelines for Creating Dimensions

Before you create dimensions, consider the following:

- 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 business users pull apart (deconstruct) information in a model. Consider the reporting needs of the business users.

  Identify a dimension by the fact that business 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 business users.

  Attribute dimensions are for business users needs for reporting. Do not put these needs into the model's 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:

  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

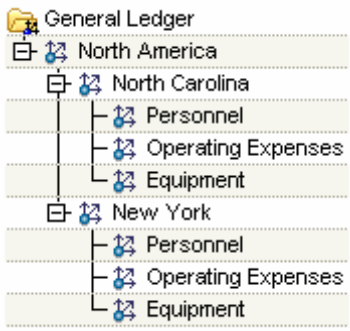
creating two dimensions: one dimension for the office location and one dimension for the general ledger expenses.

Also, a dimension member can be identified by whether it gives or receives costs (it is part of an assignment). For more information, see Chapter 10, "Assignments," on page 75.

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.

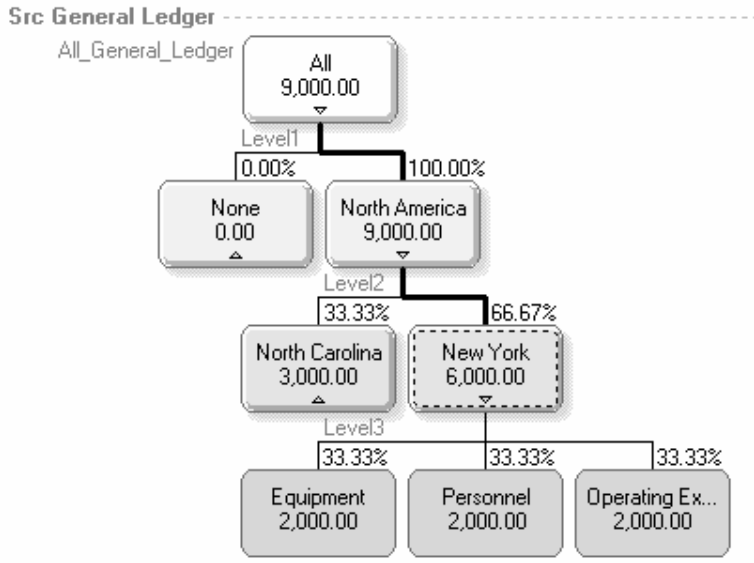
From your perspective as model builder, you can create a single, structural dimension on the Dimensions page as shown in the following figure:



In the Resource module, the structure looks like the following figure (with example values entered):

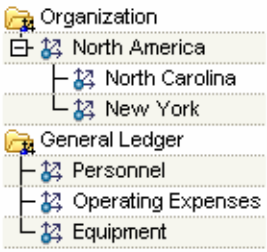
Display Name	Cost
RESOURCE (PRIMARY PANE)	$9,000.00
North America	$9,000.00
North Carolina	$3,000.00
Personnel	$1,000.00
Personnel_1017-1	$1,000.00
Operating Expenses	$1,000.00
Operating Expenses_	$1,000.00
Equipment	$1,000.00
Equipment_1019-1	$1,000.00
New York	$6,000.00
Personnel	$2,000.00
Personnel_1026-1	$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. Your business users can analyze the cost data in the OLAP tool, as shown in the following figure of the Cube Explorer View:

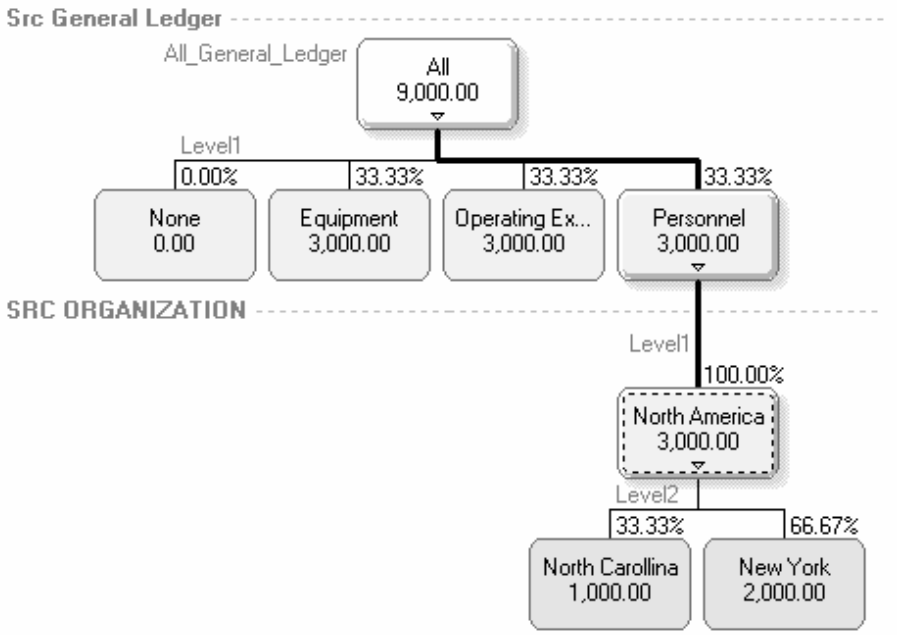


Business users can examine the costs associated with New York or North Carolina (which is not expanded in the figure). However, they cannot combine the costs of individual accounts for both North Carolina and New York. For example, business users cannot examine the total cost of personnel for both North Carolina and New York.

Now, suppose that you want to allow business 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 page:



In the Resources module, the structure looks identical to the previous example. However, in addition to the OLAP analysis shown in the previous example, business 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 7

Modules

Introduction

A module contains a specific type of cost information in a model, such as information about resources or information about activities. Cost information is contained in accounts and costs elements. For information on accounts and cost elements, see Chapter 8, “Accounts and Cost Elements,” on page 51.

The organization of each module corresponds to the organization of structural dimensions. The structural dimensions of each module are addressed later in this chapter.

A model can contain the following modules:

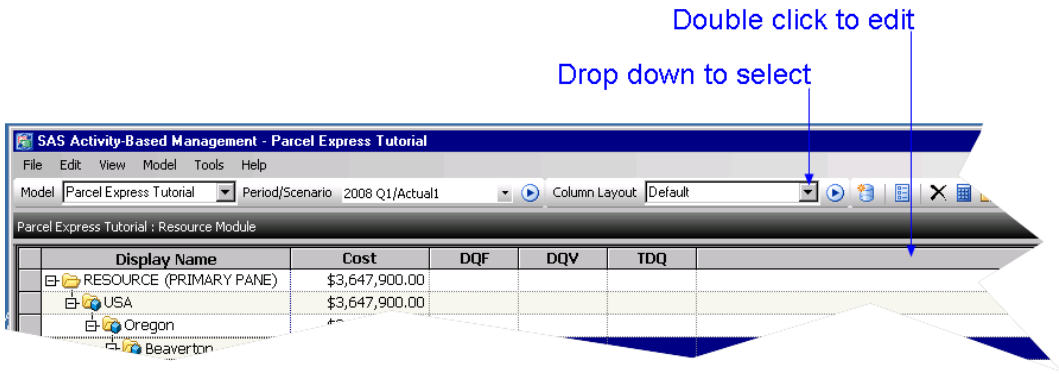
- Resource module
- Activity module
- Cost Object module
- External Units module

If a model is not a complete activity-based costing implementation, the model is not required to use all of the available modules. For example, suppose that your organization performs work for customers on a customized basis, and you need a model to estimate job costs. You can enter accounts and their costs in the Resource module, Activity module, and the External Units module (if needed), and you can calculate the model costs. When

you later identify the needs of a specific customer, you can create a cost object account that receives costs from the existing accounts. When you calculate the model costs, you can determine the costs of the customized job.

Columns

SAS Activity-Based Management displays information in each module in user-defined columns in a grid, as shown in the following illustration (select a column layout from the drop-down list):



You can add or remove columns from the grid and specify how information in the columns appears. (Double-click the column header to edit the column layout.) Once you are satisfied with the appearance, you can save the display configuration (called a saved column layout). Later, you can display the saved column layout and the columns are displayed as when you saved the layout. For more information, see Chapter 15, “Column Layouts,” on page 99.

Note: Adding or removing columns from the grid does not change the table. Column layouts provide a view of tables they do not affect the data in the tables. Select **Model** ⇨ **Column Layout** ⇨ **Edit Columns** to add or remove columns. Or, double-click the column header to open the Column Layout dialog box.

Note: The **Display** column, which is the left-most column in the column layout, is always required, so you cannot remove it, change it, or reorder it.

References

A reference is a unique identifier for an item in a module, such as an account 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. For information about reference conventions, see the Help.

The 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 Product Group Product SKU (if needed)	Shoes Summer Sandals Leather Weave Tan576830

Customers Dimension

The Customers dimension typically contains information about customer categories and individual customers. For example:

Dimension Level Name	Example
Customer Group Customer Category Specific Customer Location (if needed)	Retail Discount Big Mart Phoenix

Channels Dimension

The Channels dimension typically contains information about distribution channel categories and individual channels. For example:

Dimension Level Name	Example
Channel Group Channel Category Channel Channel Details (if needed)	Traditional Wholesale Catalog Targeted Mail

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 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 Process Activity	Receive Packages Collect by Region 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 Function Department	North Carolina Manufacturing Inspection

The Resource Module

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 Activity-Based 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 Activity-Based Management model is the next higher level in the example (10x Salaries and Wages). For example:

Dimension Level Name	Example
GL Account Group GL Account	Wages, Salaries, and Benefits 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. For more information, see Chapter 8, "Accounts and Cost Elements," on page 51.

Organization Dimension

The Organization dimension that was created in the Activity module can be reused here.

The External Units Module

The External Units 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 Activity-Based Management model, but needs to be accounted for in the model. For information about bills of costs, see Chapter 10, "Assignments," on page 75.

Like the other modules, the External Units 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.

Guidelines for Creating the 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. (For more information about validating a model, see Chapter 8, “Accounts and Cost Elements,” on page 51.)

  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.

Chapter 8

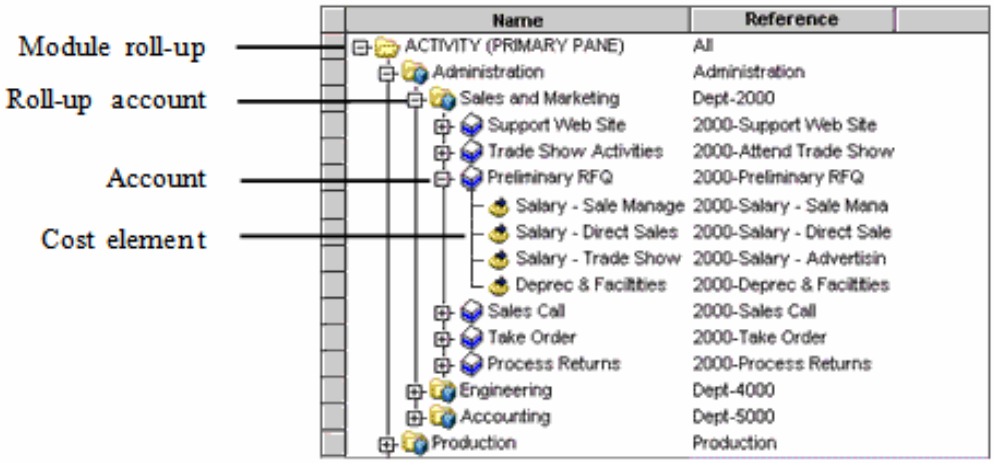
Accounts and Cost Elements

Accounts

Overview

An *account* is an intersection of two dimensions that is similar to a line item in a chart of accounts. Each account contains cost elements.

A *roll-up account* is a cluster of accounts or a cluster of other roll-up accounts that are related by function, department, location, or group. The cost of a roll-up account is the sum of costs for all the accounts and roll-up accounts in the immediate subordinate level.



Each module contains a **module roll-up**. A module roll-up is the highest level in the module. A module roll-up represents all the accounts and roll-up accounts in the module.

Creating and Deleting Accounts

An account corresponds to a dimension member. 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.

Guidelines for Creating Accounts

- 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 reporting 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
Direct Mail Advertising
In-Store Advertising Circulars
Radio Advertising
TV Advertising
Yellow Pages Advertising
	Advertising
	Print
	  Magazine Inserts
	  Direct Mail
	  In-Store Circulars
	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. For more information, see Chapter 9, “Drivers,” on page 57.

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.

Cost Elements

Overview

The cost of an account is the sum of its cost elements. There are four types of cost elements:

- entered cost element
- assigned cost element
- internal unit cost element
- external unit cost element

Entered Cost Element

An entered cost element is the cost that you specify. In the interface, it is represented by this image:  .

Assigned Cost Element

An assigned cost element is the cost that is flowed from one account to another account. In the interface, it is represented by this image:  .

Internal Unit Cost Element

An internal unit cost element is a unit (for example, a part that is produced within an organization) whose cost is already represented in the model. This cost flows from one account to another account's bill of costs. In the interface, it is represented by this image:  .

External Unit Cost Element

An external unit cost element is a unit (for example, a part that is purchased from a supplier) whose cost is maintained outside of a SAS Activity-Based Management model, but needs to be accounted for in the model. In the interface, it is represented by this image:  .

In SAS Activity-Based Management, external unit cost elements are treated like accounts. When an external unit cost element's cost is flowed to an account, the flowed cost is listed as an external unit cost element. External unit cost elements always contribute costs, but they do not receive costs.

Validating a Model

As you create each module by creating accounts and cost elements, validate the model to ensure that it is structurally correct.

Note: The existence of any of the following conditions does not mean that there are problems in the model. Ensure that the messages produced by the validation are what you expect. For example, if you use negative driver quantities in a model, you should expect the validation to report the negative driver quantities.

You can choose to validate the following conditions of a model:

- Overdriven source accounts do not exist.

  Ensures that an account does not assign to another account(s) more than 100% of its cost.

- Unassigned accounts do not exist.

  Ensures that all accounts in the Resource module and in the Activity module are assigned. Accounts in the Cost Object module are typically at the end of an assignment path, so they are not assigned to other accounts.

- Accounts that have zero costs do not exist.

  Ensures that all accounts have costs that are not zero.

- Empty attributes do not exist.

  Ensures that all attributes that have been added to accounts have values. For numeric attributes, this ensures that the values are not zero. For text attributes, this ensures that the values are not blank.

- Negative driver quantities do not exist.

  Ensures that an account does not have negative driver quantities.

Chapter 9

Drivers

Introduction

A *driver* controls how costs flow from one account to another account by instructing how costs flow from an account. A driver attempts to accurately measure an account's consumption or to assign costs to an account. For example, a driver can be the number of full-time employees (FTEs), the percent of supplies used, or the number of boxes packaged.

Types of Drivers

Basic Driver

A basic driver (named Basic) includes only one factor that represents the relationship between accounts. For example, a basic driver can be the number of hours a machine is running or the number of full-time equivalents that are required for producing a product.

Bill of Costs Driver

A bill of costs driver (named Bill of Cost) represents an account that is a bill of costs.

Note: Every driver from an External Unit account is considered a BOC driver regardless of whether it is an explicit BOC driver or a Calculated driver. From any other account only an explicit BOC driver is considered a BOC driver. Apart from External Unit accounts, a Calculated driver from any other account is not considered a BOC driver.

For information about bills of costs, see Chapter 10, “Assignments,” on page 75.

Calculated Driver

Overview

A calculated driver (named Calculated) includes several factors that represent the relationship between accounts. The factors are determined by a formula that you define. A formula that includes numeric information about the destination accounts produces a calculated driver quantity from multiple variables. For information about formulas and functions, see Chapter 12, “Using Formulas,” on page 87.

A formula for a calculated driver is not periodic; the formula remains the same for all periods.

Example: Shipping a Product

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 OutputQuantity) is divided by the number of units that a pallet can hold (the user-defined numeric attribute UnitsPerPallet), as follows:

```
OutputQuantity/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 formula can be used for multiple products.

Example: 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 cost than a customer who is closer to the company.

Example: 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. 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.

Example: 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 formula could be altered to calculate the financial carrying cost of the inventory, not the storage cost, as follows:

```
(OutputQuantity/InventoryTurns)*Cost
```

Example: 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 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 formula could be used for all customers.

Rule-based Driver

You use rule-based drivers to generate cost assignments automatically during calculation. The feature works by allowing you to attach a formula to an existing driver. When you attach the driver with its formula to a source account, calculation evaluates for which potential destination accounts the formula tests true and creates an assignment from the source account to each such destination account.

For information on Rule-based Drivers, see the Online Help for the product.

Evenly Assigned Driver

An evenly assigned driver (named Evenly Assigned) assigns equal percentages of an account's costs to each account that receives costs. For example, if you use an evenly assigned driver to assign an account's cost to two accounts, then each account automatically receives 50 percent of the cost.

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, replace this driver with a driver that is more accurate.

Note: The types of driver quantities are preset for the evenly assigned driver; you cannot change them.

Percentage Driver

A percentage driver (named Percentage) assigns a specified percentage of an account's costs to each account that receives costs. The percentage driver tracks the total percentage and issues a warning if costs exceed 100 percent or if less than 100 percent of costs are used.

The percentage driver is often used as a first approximation of the costs that flow between accounts when actual driver quantities are not known. Unlike an evenly assigned driver that allocates the same percentage to each account that receives costs, a percentage driver enables you to assign a percentage to each account. When actual driver quantities are known, replace this driver with a driver that is more accurate. Or, if determining the actual driver quantities is not practical, ensure that the percentages are reasonably accurate.

Note: The types of driver quantities are preset for the percentage driver; you cannot change them.

Example: Time Spent on Tasks

Suppose that an organization decides that it is 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.

Sales Volume Driver

A Sales Volume driver generates a cost flow whose magnitude is determined by the quantity of an item sold. To use a Sales Volume driver you must do two things:

1. Attach a Sales Volume driver to an account. The account is the source of a cost that is associated with selling an item. Usually such accounts are in the Cost Object module.

   Note: For a Sales Volume driver to work, the dimensions of an account to which it is attached must be the same as the dimensions in the Profit Analysis module. The dimensions of the Cost Object module must be the same as the dimensions of the Profit Analysis module.

2. Assign a value to the Sold Quantity (SoldQty) property of one or more accounts (presumably a Cost Object account) that are the final destination of the item before sale.

During calculation, then, the system does two things:

1. It generates an assignment between each account to which a Sales Volume driver is attached and every appropriate destination account.

   Note: A destination account is appropriate if its Sold Quantity (SoldQty) is non-zero and if its intersection dimension members are the same as the intersection dimension

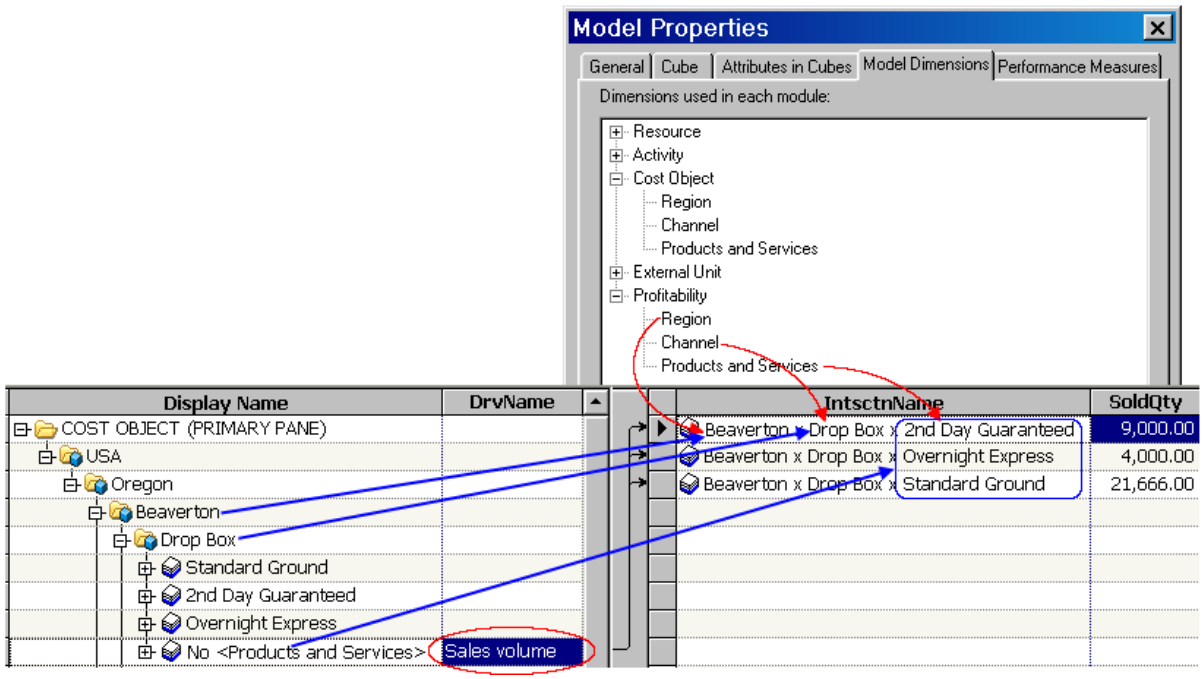
members of the source account (where "None" matches any intersection dimension member). The intersection dimension members for the destination account are determined by the dimensions of the Profit Analysis module.

The following picture shows an example in which the Sales Volume driver is attached to the following source account:

```
Beaverton x Drop Box x No <Products and Services>
```

During calculation, the system generates assignments to the following destination accounts that have a non-zero value for SoldQty:

```
Beaverton x Drop Box x 2nd Day Guaranteed
Beaverton x Drop Box x Overnight Express
Beaverton x Drop Box x Standard Ground
```



You can see that the "None" intersection (No <Products and Services>) in the source account matches the intersection in three destination accounts. "None" acts like a wildcard whether it appears in a source account or in a destination account.

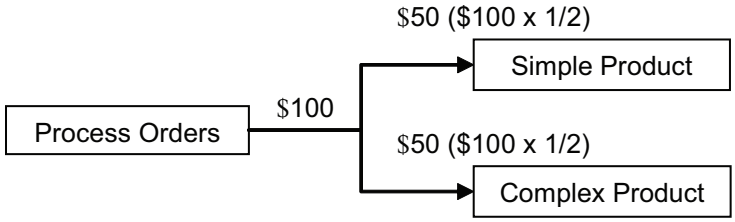
2. It calculates the Driver Quantity Calculated (DrvQtyCalc) for the driver. DrvQtyCalc is equal to the SoldQty on the destination account. The system then uses the value of DrvQtyCalc to calculate the magnitude of the cost flow.

Weighted Driver

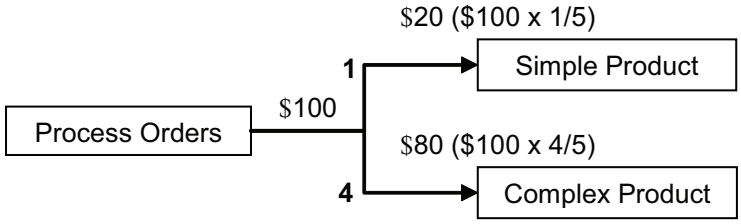
A weighted driver (named Weighted) contains two factors that represent the relationship between accounts. The first factor measures volume (or frequency), and the second factor measures intensity (or complexity). The two factors are multiplied to produce a calculated value that is used for distributing 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. For information about the types of driver quantities, see "Types of Driver Quantities" on page 63.

Example: Processing Customer Orders

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 that is needed for an average customer call.

Default Driver

The default driver is the driver that SAS Activity-Based Management associates with an assignment when you first create the assignment. You can change the driver later.

There is a default driver for each module.

Types of Driver Quantities

Overview

You can specify the types of driver quantities that a driver will accept as input, preventing inappropriate data from being entered into a model.

For example, if you create a driver that accepts only fixed quantities, SAS Activity-Based Management prevents variable quantities and weighted quantities as follows:

- In the interface, variable quantities and weighted quantities cannot be entered.
- When you import model data, variable quantities and weighted quantities are reported as warnings and are ignored.

Fixed Driver Quantities

For *fixed driver quantities*, the cost that flows to a destination account from a source account does not depend on the output quantity: the source accoun'ts unit cost is multiplied by the driver quantity (represented by the property Driver Quantity Fixed (abbreviated DQF)). For information about source accounts and destination accounts, see Chapter 10, "Assignments," on page 75. For information about properties, see the Help.

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 basic, fixed-quantity driver are shown in the following figure:

The driver quantity for **Salary** is 3.00: **2.00** for **Take Orders** and **1.00** for **Process Orders** (the column DQF in the previous figure).

The unit cost that flows from **Salary** is calculated by dividing the total cost of **Salary** by the driver quantity of **Salary**: $100.00/3.00 = $33.33.

So, the cost for each account is calculated as follows:

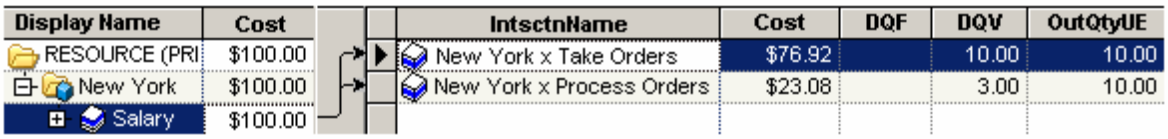
Account	Unit Cost x DQF	Cost
Take Orders	$33.33 x 2.00	$66.67
Process Orders	$33.33 x 1.00	$33.33

Note: All values are rounded.

Variable Driver Quantities

For *variable driver quantities*, the cost that flows to a destination account from a source account depends on the output quantity: the source account's unit cost is multiplied by the driver quantity (the property Driver Quantity Variable (abbreviated DQV)) and by the output quantity (either the property Output Quantity (abbreviated OutQty)) or the property Output Quantity UE (abbreviated OutQtyUE)).

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 basic, variable-quantity driver are shown in the following figure:



The driver quantity for Salary is 130.00, which is calculated as follows:

Account	DQV x OutQtyUE	Total
Take Orders	10.00 x 10.00	100.00
Process Orders	3.00 x 10.00	30.00
		130.00

The unit cost that flows from **Salary** is calculated by dividing the total cost of **Salary** by the driver quantity of **Salary**: $100.00/130.00 = $00.7692.

So, the cost for each account is calculated as follows:

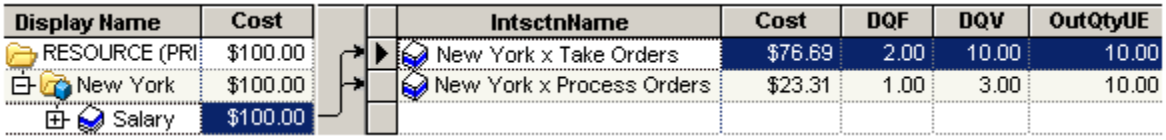
Account	Unit Cost x DQV x OutQtyUE	Cost
Take Orders	$00.7692 x 10.00 x 10.00	$76.92
Process Orders	$00.7692 x 3.00 x 10.00	$23.08

Note: All values are rounded.

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 basic, 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	OutQtyUE
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 driver quantity for Salary is 133.00, which is calculated as follows:

Account	(DQF) +	(DQV x OutQtyUE)	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 unit cost that flows from **Salary** is calculated by dividing the total cost of **Salary** by the driver quantity of **Salary**: $100.00/133.00 = $00.7519.

So, the cost for each account is calculated as follows:

Account	(Unit Cost x DQF) +	(Unit Cost x DQV x OutQtyUE)	Cost
Take Orders	$00.7519 x 2.00	$00.7519 x 10.00 x 10.00	$76.69
Process Orders	$00.7519 x 1.00	$00.7519 x 3.00 x 10.00	$23.31

Note: All values are rounded.

Weighted Driver Quantities

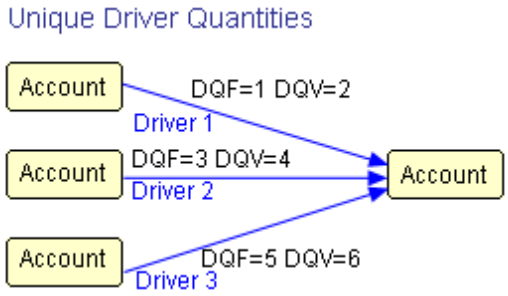
Weighted driver quantities are used with a weighted driver. For information about weighted drivers, see the Weighted Driver section.

Unique Driver Quantities

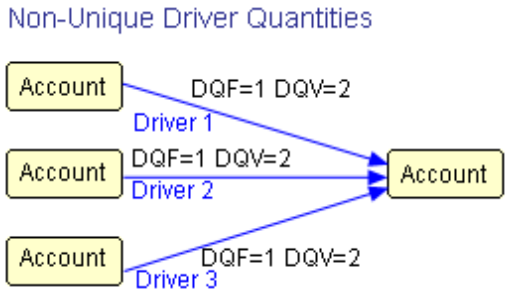
Overview

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.



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.



For example, 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

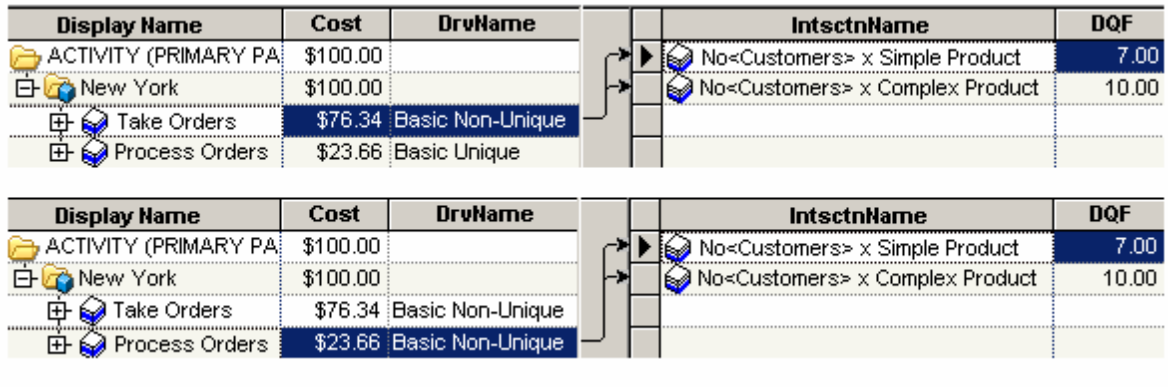
IntsctnName	DQF
No<Customers> x Simple Product	7.00
No<Customers> x Complex Product	10.00

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 **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 the activity **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.

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.

Idle Quantities

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 SAS Activity-Based Management (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 that are described in the following section. 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).

Therefore, the driver idle cost (the 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 x $100.00).

Driver Sequencing

Overview

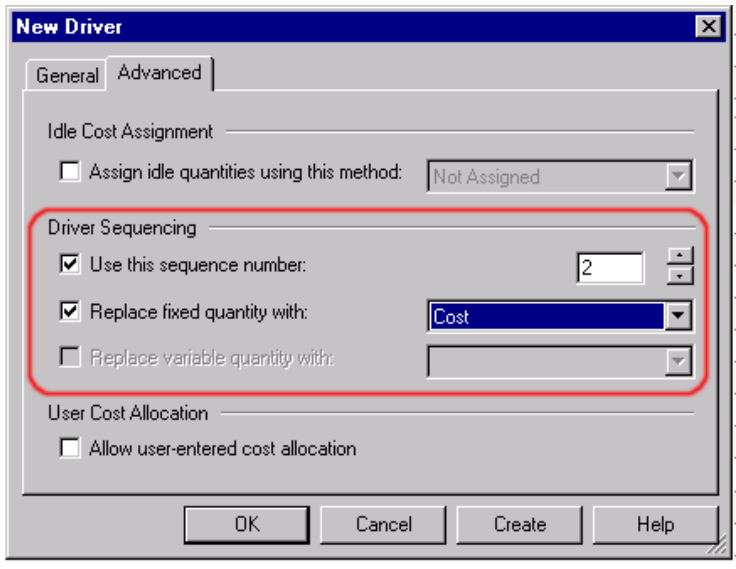
Driver sequencing allows you define multiple calculation passes through assignments such that the driver quantity (DQF or DQV) of a sequenced driver (in a subsequent pass) is based on a cost resulting from drivers that execute in a previous pass.

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.

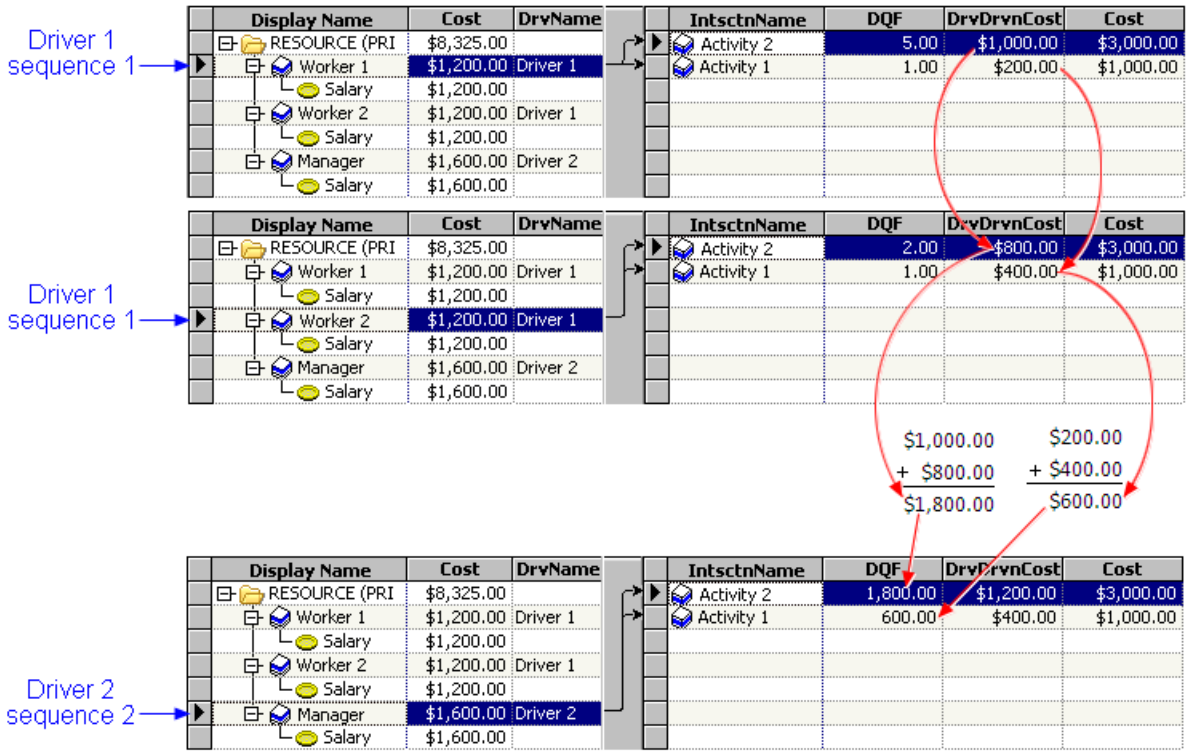


	DQF Activity 1	DQF Activity 2
Worker 1	1	5

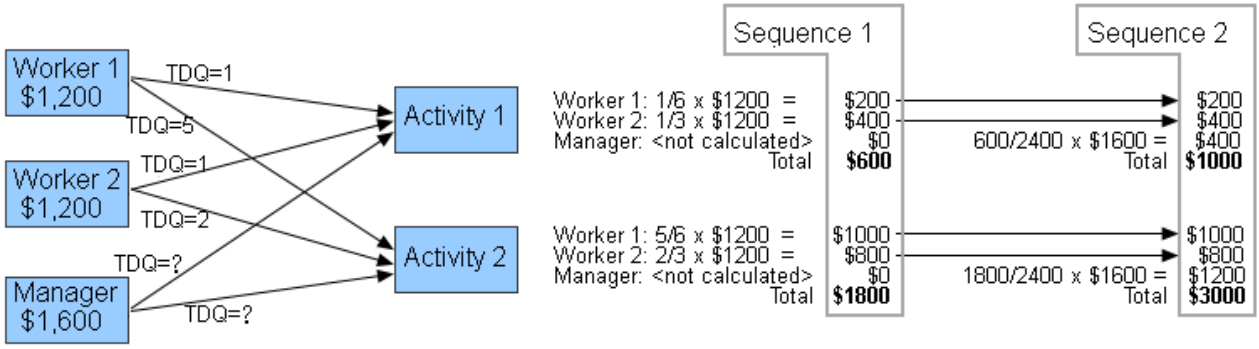
	DQF Activity 1	DQF Activity 2
Worker 2	1	2
Manager	to be determined	to be determined

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.

As the following illustration 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	$1,200	1	5	$200	$1,000
Worker 2	$1,200	1	2	$400	$800
Managaer	$1,600	200+400=600	1,000+800=1800	$400	$1,200

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 the 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

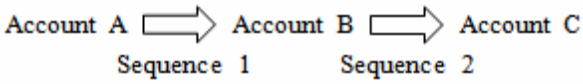
The following presents common misconceptions about driver sequencing:

- Every driver needs a sequence number.

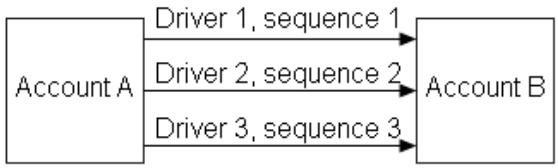
  Specify a sequence number for a driver only when the driver quantities on the destination accounts need to be replaced. This replacement is the only function of driver sequencing.

- Every driver in an assignment path needs a sequence number.

  Do not specify a driver sequence to indicate the calculation sequence between accounts in an assignment path. For example:

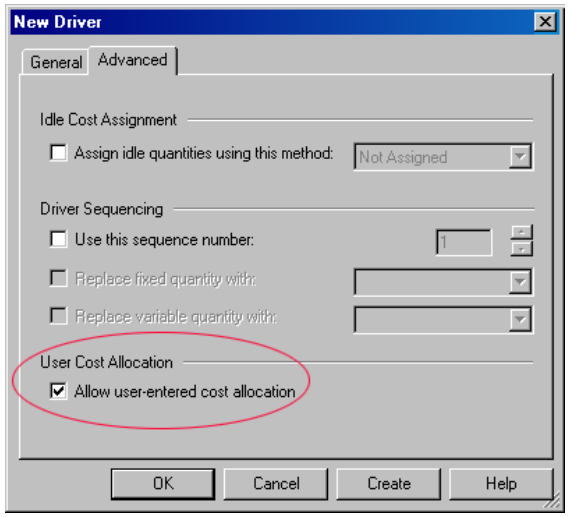


- A sequenced driver is not paired with any particular previous driver. The cost that replaces the DQF or DQV for a sequenced driver is the cost for an account at the time the sequenced driver runs, regardless of what drivers have contributed to that cost. For example in the following picture, the DQF 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.



For information about assignment paths, see Chapter 10, “Assignments,” on page 75.

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 (DrvAllocatedCost.

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.

Guidelines for Creating Drivers

Before you create drivers, consider the following:

- 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.

- 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.

- In general, use fixed driver quantities for basic drivers. Use variable driver quantities for bill of costs drivers.

Chapter 10

Assignments

Introduction

An *assignment* is a way to distribute costs between accounts. Costs that are assigned between accounts that are in the same module are called *inner-module assignments*. Costs that are assigned between accounts that are in different modules are called *cross-module assignments*.

When you assign a cost from one account to another account, the cost is said to flow from the first account to the second account. The first account is called a *source account* because it is the source of costs for the second account. The second account is called the *destination account* because it receives the first account's cost flows.

The manner in which the cost flows is controlled by a driver. A single source account can flow its costs to any number of destination accounts, but each source account can have only one driver. (For information about drivers, see Chapter 9, “Drivers,” on page 57.)

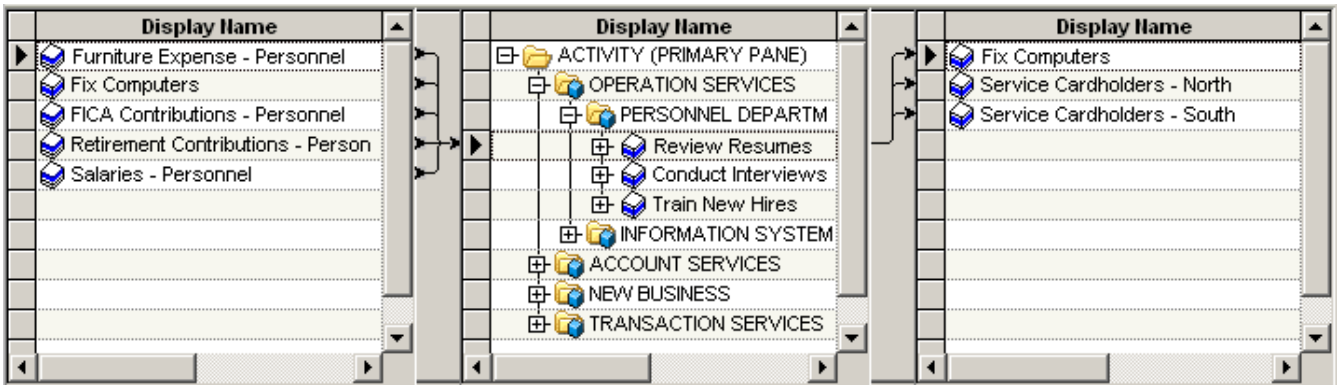
Assignment Paths

As you create assignments between accounts on the Resource module page, the Activity module page, and the Cost Object module page, you create a sequence of accounts through

which costs flow. This sequence is called the assignment path. For example, suppose that costs are assigned first 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.

Assignments Panes

You can divide the Resource module page, Activity module page, and Cost Object module page into up to three panes. The panes enable you to easily create assignments or to see the assignments between accounts.



For example:

- One pane

  The primary pane shows you one module at a time. The primary pane always displays the hierarchy of the module that you are viewing. The left and right panes display accounts only, not module hierarchy.

- Two panes

  The primary pane and the left or right pane show you assignments from one module to another module, as well as assignments that are in the same module.

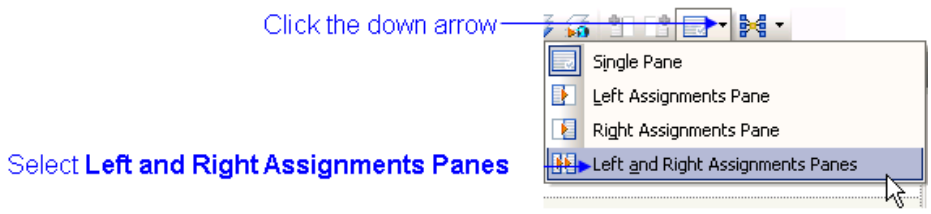
- Three panes

  The left pane, the primary pane, and the right pane show you assignments that flow the costs from resources, through activities, to cost objects.

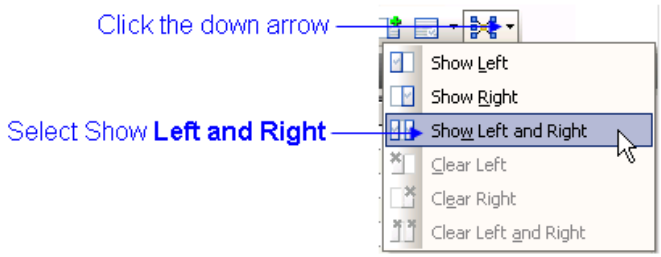
Costs flow from the left to the right. The space between the panes is called the splitter bar. Lines and arrows indicate the assignments between accounts.

To display assignments, such as in the picture above:

1. Select **Left and Right Assignments Panes** from the View Assignments Panes icon.



2. Select **Show Left and Right** from the Show Assignments icon.



If there are no assignments, you see nothing additional when you show assignments.

Bills of Costs

Overview

A bill of costs (BOC) provides an easy way to add material costs and internal unit costs directly to accounts.

Use a bill of costs for these circumstances:

- External Unit cost elements outside of the Resource module general ledger need to be included in a model.

  Not all costs that are assigned through a model come from the general ledger. For example, material costs (the cost of purchased components) can be additional product cost information, which often comes from a Material Requirements Planning (MRP) system.

- Internal Unit cost elements in a model that are associated with product families need to be tracked.

If you create an assignment from an external unit, the bill of costs driver is automatically specified. If you create an assignment from an internal unit, you must specify the bill of costs driver.

A bill of costs driver quantity can include variable quantities, fixed quantities, or both. For information about drivers quantities, see Chapter 9, “Drivers,” on page 57.

Discrete-Manufacturing Environments

In a discrete-manufacturing environment, a bill of materials is a familiar concept. Usually, each product has a bill of materials that lists all the product's components and their costs. In the simplest case, a bill of materials is a list of the product's components, the number of units of each component that the product includes, and the unit cost for each component. Because bills of materials exist in many environments, bills of costs correspond to them to some degree.

In discrete-manufacturing environments, products that are components of other products are called internal unit cost elements. Each internal unit cost element can have its own bill of costs.

Process-Manufacturing Environments

In a process-manufacturing environment, an activity might have a bill of costs. A bill of costs for an activity lists all materials that were added during that activity in the process.

Because material costs are added to the specific activities that consume them, you can trace the cost of finished products more accurately.

Types of Bill of Costs Quantities

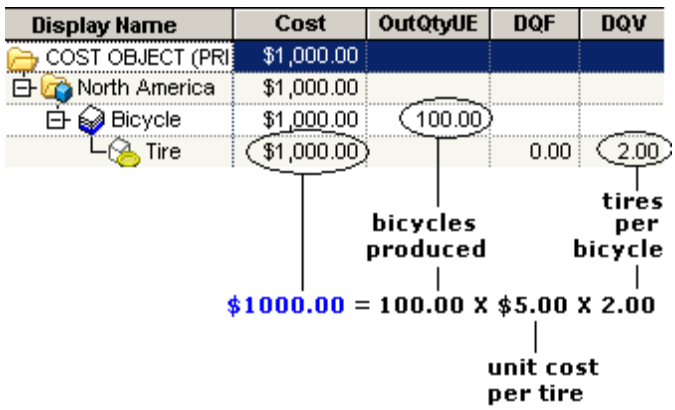
A bill of costs quantity can include variable quantities, fixed quantities, or both. For information about types of drivers quantities, see Chapter 9, "Drivers," on page 57.

Variable Quantities

For variable quantities, the cost that flows to an account from the bill of costs depends on the output quantity: the unit cost of the bill of costs is multiplied by the driver quantity (the property Driver Quantity Variable; the column **DQV**) and by the output quantity (either the property Output Quantity or the property Output Quantity UE, the column **OutQtyUE**).

For example, suppose that a company manufactures bicycles. The rubber tires are purchased from another company for $5 each. (This value was specified when the external unit was created.) Because each bicycle requires two tires, the destination unit cost of the bill of costs is $10. If the company produces 100 bicycles, the total cost is $1,000 (100.00 x $5.00 x 2.00).

(Note that all other cost elements have been removed to better illustrate the bill of costs.)

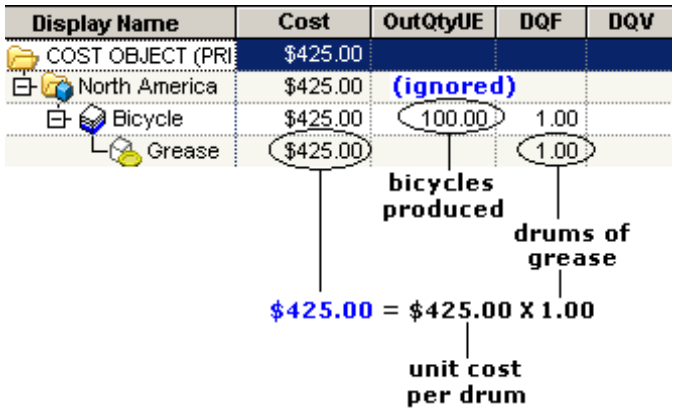


Fixed Quantities

For fixed quantities, the cost that flows to an account from the bill of costs does not depend on the output quantity: the unit cost of the bill of costs is multiplied by the driver quantity (the property Driver Quantity Fixed, the column **DQF**).

For example, suppose that the bicycle manufacturer buys a drum of grease at the beginning of each year. The unit cost of the bill of costs is the current price of the drum of grease, regardless of how many bicycles are produced.

(Note that all other cost elements have been removed to better illustrate the bill of costs.)



Variable Quantities and Fixed Quantities

You can use both variable quantities and fixed quantities in a single bill of costs.

(Note that all other cost elements have been removed to better illustrate the bill of costs.)

Display Name	Cost	OutQtyUE	DQF	DQV
COST OBJECT (PRI	$1,425.00			
North America	$1,425.00			
Bicycle	$1,425.00	100.00	1.00	
Tire	$1,000.00		0.00	2.00
Grease	$425.00		1.00	

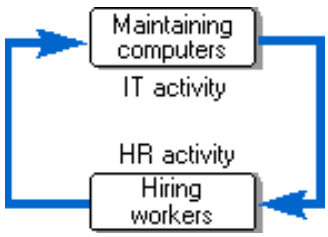
Reciprocal Costing

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.

You create reciprocal costs by creating an assignment from an account or from an internal bill of costs.

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.



Guidelines for Creating Reciprocal Costs

When you use an internal bill of costs to create a reciprocal cost, consider the following:

- You can attach more than one reciprocal internal bill of costs within the same module.
- You must attach an internal bill of costs to a third account so that costs flow out of the reciprocal cost assignment. Otherwise, the two accounts in the reciprocal cost assignment create a continuous loop.

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 follows:

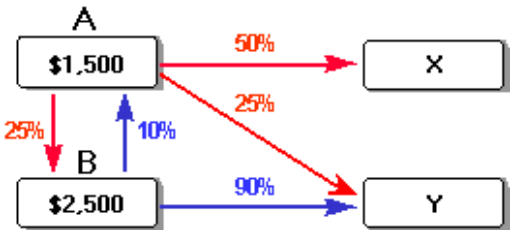


Table 10.1 *Reciprocal Costs Calculation*

The equations:	The results:
A = 1,500 + .10B	A = $1,794.87
B = 2,500 + .25A	B = $2,948.72
X = .5A	X = $897.44
Y = .25A + .9B	Y = $3,102.56

Guidelines for Creating 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 Activity-Based Management issues warnings when you calculate costs.

Chapter 11

Attributes

Introduction

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. An attribute conveys information about the account.

You create an attribute on the Attributes page. You add and delete (manage) the attributes that have been added to an account on the Resource module page, the Activity module page, and the Cost Object module page.

Tag Attributes

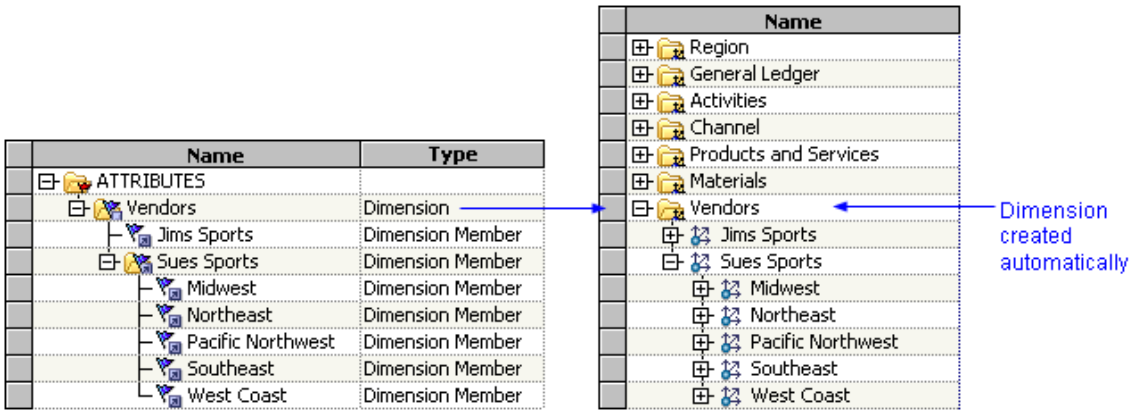
A tag attribute is either added to an account or it is not. A tag attribute does not store a value.

Note: In previous releases of SAS Activity-Based Management, tag attributes were referred to as Boolean attributes.

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.

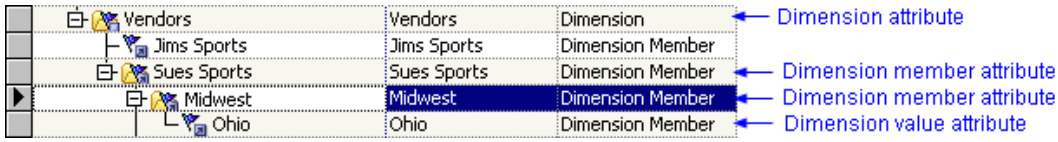
Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes

A dimension attribute corresponds to a dimension. When you create a dimension attribute on the Attributes page, a dimension is automatically created on the Dimensions page, as shown:

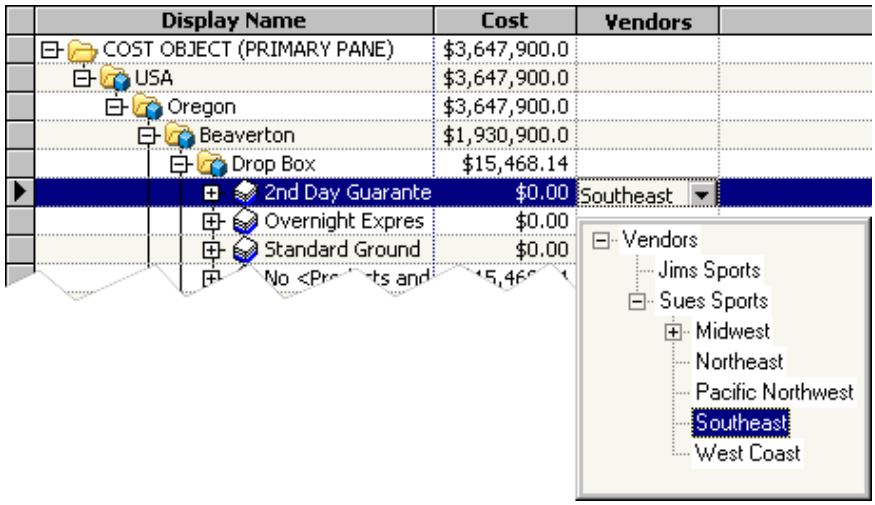


The leaf node of a dimension attribute is called a dimension value attribute. It is the dimension value attribute that applies to an account. Whereas a dimension member attribute contains a dimension value attribute, a dimension value attribute cannot contain other attributes.

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 illustration, 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 documented 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 model's financial measures or non-financial measures.
- A numeric attribute can track activity performance over time.
- A numeric attribute can be a performance measure for special reporting purposes.

The data that a numeric attribute contains can be used in calculated drivers and in calculated attributes.

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 a formula that you define. For more information, see Chapter 12, “Using Formulas,” on page 87.

Property values and attribute values are derived from the account 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 account to which the attribute is attached, because each account 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

Suppose that a company provides order-processing services to other companies. To determine the average cost of processing an order, the cost (the property Cost) 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
```

Text Attributes

A text attribute stores alphabetic and numeric characters. A text attribute provides information about the account 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.

Attribute Folders

An attribute folder organizes attributes. An attribute folder can contain any type of attribute and multiple types of attributes, because the folder itself has no type. However, a dimension attribute folder can contain only dimension member attributes or 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 Activity-Based Management automatically creates it as a dimension attribute folder.

Stages Attributes

Stages Attributes

Stages are used for analyzing the multi-stage contributions cube. To define stages, add specific dimension value 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 naming conventions. (For information about the multi-stage contributions cube, see Chapter 13, “Analyzing Model Data with OLAP,” on page 89.)

To see the stages in the multi-stage contributions cube, you must specify the properties for the cube. (For information about the multi-stage contributions cube, see Chapter 13, “Analyzing Model Data with OLAP,” on page 89.

Guidelines for Adding Stages to Accounts

Follow these guidelines when you add Stages to accounts:

- Do not add more than one stage (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 to accounts that you do not need for analysis.

  Every account that has a stage is included in the multi-stage contributions cube. The cube can be generated more quickly if accounts not needed for analysis are omitted.
- To avoid performance problems, do not exceed six to eight stages.

Chapter 12

Using Formulas

Introduction

You must define a formula to use with the following:

- calculated driver
- rule-based driver
- calculated attribute

A formula can contain the following elements:

- basic functions
- system-defined numeric properties
- user-defined numeric attributes

For information about calculated drivers, see Chapter 9, “Drivers,” on page 57. For information about calculated attributes, see Chapter 11, “Attributes,” on page 81. For information about rule-based drivers and about the functions that you can use in formulas, see the Help.

Defining Formulas

Overview

To define a formula, select numeric properties, numeric attributes, and operators. To include user-defined numeric attributes, you must create these attributes before you define a formula.

If you use a numeric 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 numeric 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 Accounts and Destination Accounts

A formula can use the numeric properties and numeric attributes of a source account or a destination account. In the SAS Activity-Based 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 or , followed by a property or attribute. For example, or .

Property Names in Formulas

The property names in formulas are the same as the property names that you see elsewhere in SAS Activity-Based 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.

Testing Formulas

Testing a formula validates the operators and ensures that the syntax is correct. However, numeric properties and numeric attributes are not validated. Formula 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 Activity-Based 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 Activity-Based 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 a numeric property name.

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 the name contains spaces.

```
OutputQuantity/"Units Per Pallet"
```

Chapter 13

Analyzing Model Data with OLAP

Introduction

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 Activity-Based 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.

Measures

Data that is most likely to be summarized are stored as values, either as currency or as numbers. *Measures* are the values that are aggregated and analyzed. A model can have values for costs, for sales quantities, for revenue, and so on. These values form the measures. For example, a model has information about product sales: product costs, sales quantities, and revenue. If you want to evaluate the success of a particular region, you can compare that region's total revenue to the total revenue of each other region.

Cubes

Overview

A cube is the basic unit of analysis: it represents a particular domain of inquiry in 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 Activity-Based Management cubes are standard OLAP cubes.

You use SAS Activity-Based Management to connect to the cubes on a SAS Activity-Based Management server and to interact with these cubes. For each model, a Designer can generate cubes. Viewers can then manipulate these cubes using 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 columns and rows.

A cube in which costs are indexed by two dimensions (such as customer and product) is a two-dimensional cube. A cube in which costs are indexed by three dimensions (such as region, customer, and product) is a three-dimensional cube.

It does not matter how many dimensions a cube has; the storage unit is 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 Activity-Based 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. In addition, you can force the model to be calculated.

Generating Cubes

After you calculate costs, you can generate cubes to analyze a model with the OLAP Analyzser. Generating a cube is a two-part process. First, you create a cube configuration to specify options for generating the cube. Then, you invoke the cube configuration to do the generation. The process is separated into two steps so that your generation options are

remembered instead of your having to re-specify them each time you generate. xxxChapter 31, “Working with Cube Generation,” on page 217.

Note: When you generate a cube, SAS Activity-Based 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. In addition, you can force the model to be calculated.

When SAS Activity-Based Management generates cubes, it can optionally perform calculations that pre-aggregate numeric data to give you faster performance when you work with the OLAP Analyzer. 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

  For information about stages, see Chapter 11, “Attributes,” on page 81.

- the number of dimensions

  When you import model data, select fewer than six or eight dimensions. For information about dimensions, see Chapter 6, “Dimensions,” on page 35.

- the length of assignment paths

  Assignment paths should contain fewer than 10 items.

- the number of accounts and the number of assignment paths

  For information about assignments and assignment paths, see Chapter 10, “Assignments,” on page 75.

After you calculate costs and generate cubes, check for errors.

Note: Whenever you generate cubes, cubes that were generated previously are no longer available for viewing.

Pre-Defined Cubes

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 subassembly 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. For information about stages attributes, see Chapter 11, “Attributes,” on page 81.

If you need to see driver quantities during OLAP analysis, use the single-stage contributions cube because it is the only cube that contains driver quantities.

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. For information about stages attributes, see Chapter 11, "Attributes," on page 81.

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 the resource contributions 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 in which 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.

Working with the OLAP Analyzer

Overview

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.

Note: Models are not displayed in the **Model** drop-down list of the OLAP Analyzer until you generate the model's cubes.

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.

OLAP Analyzer Performance

If the SAS Activity-Based 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.

OLAP Views

An OLAP view is a collection of information, such as a cube and a model, that controls how a cube is displayed by the OLAP Analyzer.

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

Saving an OLAP View

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 a Cube Explorer View are retained during a session, even when you return to the OLAP Analyzer after viewing other tabs. However, changes are lost when you close SAS Activity-Based Management or when you close a grid view, a chart view, or a Cube Explorer View.

If you want changes to be available later, save the OLAP view. However, the window positions and window states are not saved.

Chapter 14

Reporting Model Data

Reports

Overview

A report is created as an Adobe Acrobat PDF file.

The Report Wizard guides you through a series of steps for selecting data, running a report, and saving a report.

You can view, print, and export a report.

Model Size and Report Performance

Model size affects the performance of the reports. This is particularly true of the Multi-level Contributions report, which contains a large amount of detailed data. The time required to gather the data depends on the size of the model. Likewise, creating a report that includes data from multiple period/scenario associations requires more time than from a single period/scenario association.

When creating a report for a large model, consider creating a report for each module, instead of creating a single report for all modules. Likewise, create a report for a single period/scenario association, instead of creating a report for multiple period/scenario associations.

Report Header

Each report has a header that lists pertinent information for that report. All or some of the following information can be listed in a report's header:

Table 14.1 *Report Header*

Information	Description
Model Name	the model selected for the report
Module	one or more modules selected for the report; each module starts on a new page
Period	the period selected for the report
Scenario	the scenario selected for the report
View Perspective	the dimension selected for the report
Filtered	indicates that one or more attributes were used to select data for the report; attributes that are used to select data are listed on a report's last page

Setting up a Report

To create a report, the Report Wizard guides you through a series of steps for selecting data, running a report, and saving a report. The modules, period/scenario association, and dimensions that you select determine the data that is included in the report.

Note: To create a report you do not have to generate a cube. However, to create the following reports you must have already generated the Fact table for the model:

- Resource Contributions
- Destination Furthest
- Profit and Loss (Resource Contribution)

Saving Report Data

To save the report, you have the following options: exporting and publishing.

Exporting a Report

If you want to edit a report after running it, such as adding a company logo, you must export the report.

When exporting a report to SAS Enterprise Guide, you must add a task to the SAS Enterprise Guide add-ins before you can view the report. The programmatic identifier

(ProgID) is AbmEGPlugin.ReportData. For more information about SAS Enterprise Guide add-ins, see the SAS Enterprise Guide Help.

Publishing a Report

If you want other users to see the report with the data reflecting a specific point in time, you can publish the report. Publishing the report enables other users to see the report without rerunning the report. For a report that requires considerable time to run, publishing the report can save other users time.

Report Configurations

Once you open a report, you can save selections in the report to use later for a specific model or multiple models. This saved information is called a report configuration.

Saving a Report Configuration

You are not saving the report's data. Rather, you are saving the report configuration for later use.

All report configurations saved by Designers and Viewers on the same SAS Activity-Based Management server are listed in Workspace Manager. Therefore, your organization might want to establish guidelines for saving and naming report configurations. Although all users see the report configuration listed in Workspace Manager, each user's access to the report configurations is controlled by each report configuration's ownership and permissions. For information about ownership and permissions, see Chapter 3, "Workspaces and Workspace Manager," on page 17.

Selecting a Period/Scenario Association

Before opening a saved report configuration, select the period/scenario association for the current model. When you open a saved report configuration, the current model's period and scenario are used.

Report Templates

A report template is a file that specifies the layout of a report and the fields of data in a report (but not the data itself). When you create a report, you choose a report template.

Pre-Defined Report Templates

SAS Activity-Based Management has predefined report templates that provide formats and that permit great flexibility in the amount and type of information to include in a report. For a list of the predefined report templates and their descriptions, see the Help.

Chapter 15

Column Layouts

Introduction

Each column in a grid in SAS Activity-Based Management displays a property. A property refers to any model item that holds the values that were entered by a user or that were calculated by SAS Activity-Based Management. For information about properties, see the Help.

A column layout is a collection of displayed columns, column formats, and the column order. You can customize column layouts on the following pages to display various information, such as properties, attributes, periods, and scenarios:

- the Resource module page
- the Activity module page
- the Cost Object module page
- the External Units module page

Fore more information, see Chapter 7, “Modules,” on page 43.

When you save a customized column layout, you assign a name to the column layout so that you can retrieve it later. If a model is open, the column layout is applied to that model. If you select a column layout from Workspace Manager when a model is not open, the original model that was saved with the column layout is opened as well. The selected

column layout is applied to the corresponding model, and the Resource module page is opened.

Each named column layout has four different aspects, one for each module. For example, instead of creating different named column layouts for each module, such as My Favorite for Resource module or My Favorite for Activity module, you can create one column layout named My Favorite that changes based on which module is displayed.

All column layouts that have been saved by all users on the same server are listed in Workspace Manager. Therefore, your organization might want to set up guidelines for saving and naming column layouts.

Showing Multiple Period/Scenario Associations

By default, a new column displays values for a model's current period/scenario association. When you want to compare measures across period/scenario associations, create columns that have different period/scenario associations.

Display Name	Display Referenc	2003 Cost	2001 Cost	2000 Cost
RESOURCE (PRIMARY PANE)		$2,417,843.00	$2,352,343.00	$2,366,843.00
PERSONNEL	PER	$123,100.000	$57,600.000	$72,100.000
COMPENSATION	PER-COMP	$104,000.000	$38,500.000	$53,000.000
Salaries - Personnel	PER-SALARIES	$104,000.000	$38,500.000	$53,000.000
Temp Salaries	510-8110350	$0.000	$1,000.000	$15,000.000
Part Time Salaries	510-8110200	$4,000.000	$2,500.000	$5,000.000
Non-Exempt Salaries	510-8110100	$8,000.000	$10,000.000	$8,000.000
Exempt Salaries	510-8110050	$75,000.000	$25,000.000	$25,000.000

Default Column Layout

The default column layout is the column layout that is shown to all users by default, before any columns are added or removed and before any columns are formatted. The columns that are included are Display Name, Display Reference, and Cost.

After changing the default column layout, do not save the column layout. If you save changes to the default column layout, the default column layout is changed for all users. Instead, save your changes to the default column layout in a separate column layout.

Note: The column Display Name is the system-generated name for an item, not the name that you choose or that you can edit. To display the name that you can edit, add to the grid a column to display the property Name. For information about adding a column, see Chapter 28, “Working with Column Layouts,” on page 189.

Formatting Columns

Overview

When you add a column to the following pages, SAS Activity-Based Management assigns default formatting to the column:

- the Resource module page
- the Activity module page
- the Cost Object module page
- the External Units module page

The default column formatting is based on the following factors:

- the type of underlying data
- your Microsoft 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 column's general appearance, such as text color, background color, cell alignment, and font style. Additionally, you can specify the currency and the format type in the column. Changing the currency or the format type affects how the underlying data is displayed; 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 page display the model's base currency in the view title bars. Measures on the OLAP page are formatted as numbers.

Some reports can display any two currencies that were specified in exchange rates tables. In addition, you can export a report to another application, and then change the formatting. For information about exchange rates, see Chapter 16, “Currencies and Exchange Rates,” on page 105.

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.

Type

A column's format type controls the column's basic appearance.

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 the underlying data:

Underlying data	Available format types
Text	Text
Cost or rate	Currency (default) Number
Number or quantity	Number (default) Currency Percentage
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 Microsoft Windows Regional Options.

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, if necessary, set the precision for any column in which you want the precision to be different from the default precision.

Note: 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 your Microsoft 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
Digit grouping	Customize Regional Options	123 456 789

Displaying the Currency Symbol

You can add columns on the Resource module page, the Activity module page, and the Cost Object module page 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.

Saving a Column Layout

When you save a column layout, the following information is saved:

- the description
- the column headings
- the information that is displayed in each column, and the period/scenario association to which the information pertains
- the format for each column, and the order of the columns from left to right
- the model
- the column widths
- the open module

The following information is not saved:

- the number of assignments panes that are visible

Chapter 16

Currencies and Exchange Rates

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 model's costs in more than one currency simultaneously. However, you must first define the currencies and their exchange rates. SAS Activity-Based Management can display a model's costs in whatever currency you select, using the exchange rate that is associated with each currency. For more information, see Chapter 27, “Working with Currencies,” on page 185.

Base Currency

When you create a model, you can select the base currency that you want to use for calculating the model s costs. SAS Activity-Based Management saves the base currency with the model.

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

Exchange Rates

An exchange rate is a multiplier that is used to convert values from one currency to another currency. You enter exchange rates in an exchange rates table. Because exchange rates tables are shared by all the models on the same SAS Activity-Based 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 your user options.

For more information about working with exchange rates, see Chapter 27, “Working with Currencies,” on page 185.

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 Microsoft Windows to set the locale to French (France). In SAS Activity-Based Management, you can format a unit cost column for francs with a currency of France (F) 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. For more information about selecting currencies, see Chapter 27, “Working with Currencies,” on page 185.

Chapter 17

Importing Data and Exporting Data

Introduction

This chapter discusses importing and exporting data at a conceptual level. For task-based instructions and procedures related to importing and exporting data, see Chapter 33, "Working with Importing and Exporting Data," on page 257.

In general, there are two ways to build a model:

- interactively enter data into a model
- import data into a model

Because manually building a model is time-consuming, error-prone, and leaves little time for analysis, importing data is the recommended method.

Building a Model by Importing Data

Overview

To build a model or to add data to a model in SAS Activity-Based Management, you can interactively enter data in Model mode or you can import data. You can import data from the following sources:

- a database
- an XML file

When you import data, you can use the wizards in SAS Activity-Based Management, or you can use the SAS Services API to write a program that imports data from a database or from an XML file. (Writing a program is not addressed in this document.)

Importing Data: General Steps

Overview

The following steps describe how to import data:

1. 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. This step 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 and numerical attributes for performance measures (cost per unit). 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.

2. Check for errors and warnings.

For a detailed summary about which tables to group in each step and in what order to import tables, see the following table. For information about the tasks of importing data and exporting data, see Chapter 33, “Working with Importing and Exporting Data,” on page 257.

Import Process Summary

The following table summarizes the process for importing your data in steps:

Table 17.1 *Import Process Summary*

Table name	Requires dimension signature	Requires reference numbers	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
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.				

For a more detailed explanation of dimension signatures, reference numbers, and their requirements, see "Understanding Keys" on page 111 later in this chapter.

Updating a Model with New Structure and New Periodic Values

The following list enumerates the required tables for updating a model with new structure and new periodic values:

- Period

- Scenario
- Account
- DimensionLevel
- DimensionMember
- EnteredCostElement
- ExternalUnit
- Assignment
- Driver
- DimensionAttributeAssociation
- ValueAttribute
- ValueAttributeAssociation
- ValueAttributePeriodicDef

Updating a Model with New Periodic Values

The following list enumerates the required tables for updating a model with new periodic values:

- Account
- EnteredCostElement
- ExternalUnit
- Assignment
- ValueAttributeAssociation
- ValueAttributePeriodicDef

Preparing the Source Data

As you prepare the source data, consider the importance of designing a sound model. The design of a SAS Activity-Based Management model determines the data that you need to provide.

- What are the dimensions to use in defining the model?
- What sources will you have available for the expenses?
- What sources will you have available for transactional quantities to use as drivers?

The availability of this data in an existing electronic format affects your data collection and transformation requirements. For more detailed model design information, see Chapter 4, "Models," on page 23.

Importing Data from a Database

Overview

You can import information from the following data sources:

- SAS Datasets
- Microsoft SQL Server

- Oracle
- MySQL
- Microsoft Access
- Microsoft Excel workbook (spreadsheet ranges)
- Other databases using a JDBC driver.

The Data Schema

The database to be imported must match the data schema. For task-based information regarding the data schema, see the Help. When the database is imported, SAS Activity-Based Management attempts to convert all values in the database 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 Activity-Based Management database. SAS Activity-Based Management attempts to convert all numeric values.

Understanding Keys

The SAS Activity-Based 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 or its reference.

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	DimRef	DimMemRef	IntsctnName	IntsctnRef
RESOURCE (PRIMARY PANE)					
USA	USA	Region	USA	USA x All	USA x All
Oregon	Oregon	Region	Oregon	Oregon x All	Oregon x All
Beaverton	Beaverton	Region	Beaverton	Beaverton x All	Beaverton x All
Wages	B_WG	GL	Wages	Beaverton x Wages	Beaverton x Wages
Operating Expenses	B_OE	GL	Operating Expenses	Beaverton x Operating Expenses	Beaverton x Operating Expenses
Equipment Expenses	B_EE	GL	Equipment Expenses	Beaverton x Equipment Expenses	Beaverton x Equipment Expenses
Eugene	Eugene	Region	Eugene	Eugene x All	Eugene x All
Wages	E_WG	GL	Wages	Eugene x Wages	Eugene x Wages
Operating Expenses	E_OE	GL	Operating Expenses	Eugene x Operating Expenses	Eugene x Operating Expenses
Equipment Expenses	E_EE	GL	Equipment Expenses	Eugene x Equipment Expenses	Eugene x Equipment Expenses

The dimension signature for this account is based on the dimensions that are used and the dimension members that are used, so the dimension signature for this account requires multiple values as shown here:

Table 17.2 *Dimension signature*

Property	Examples
DimRef1	Region
DimMemberRef1	Beaverton, Eugene

Property	Examples
DimRef2	General Ledger Account (GL)
DimMemberRef2	Wages, Operating Expenses, Equipment Expenses

Reference key

In the previous example, the reference for the account is **B_WG**, and this single value uniquely identifies the account.

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 Activity-Based Management, you should be aware of the advantages and disadvantages of using one method 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 any new accounts that are found in the transaction tables and fails to import them.

The Database Tables in the Data Schema

The following tables define the SAS Activity-Based Management data schema. Each table corresponds to a specific structural or periodic aspect of a model, such as periods, scenarios, dimensions, and so on. Use the data schema to create a database that will hold the information you want to import into a model. This data schema is also used by SAS Activity-Based Management to export models to XML files.

When you build a new model by importing data, you must import the following tables:

- Dimension
- DimensionOrder
- DimensionLevel
- DimensionMember
- Driver
- Account

- EnteredCostElement
- Assignment

  Note: You must define at least one dimension for each of these tables or the import will fail or an empty model will be created.

The following table provides a brief description of each table. For information about the columns contained in each table and the fields that are required when importing data, optional when importing data, and calculated (available only when exporting data), see the Help.

Table Name	Description
Model	Specifies the model name, the description, and the default values for the model. This table is not required when importing data because you must specify the model name when importing data.
Period	Specifies the period names, the descriptions, and the start dates and the end dates. If the required periods already exist on the SAS Activity-Based Management server, then this table is not required when importing data.
PeriodLevel	Specifies the level names for period hierarchies. If the required period levels already exist on the SAS Activity-Based Management server, then this table is not required when importing data.
Scenario	Specifies the scenario names and the descriptions. If the required scenarios already exist on the SAS Activity-Based Management server, then this table is not required when importing data.
ScenarioLevel	Specifies the level names for scenario hierarchies. If the required scenario levels already exist on the SAS Activity-Based Management server, then this table is not required when importing data.
CurrencyRate	Specifies currency exchange rates for each period. If multiple currencies are not required in the model, then this table is not required when importing data.
Dimension	Specifies the dimensions in the model. This table must include the dimensions required for building the structure of the Resource module, the Activity module, the Cost Object module, and the External Units module. The Dimension table includes the dimension attributes, if dimension attributes are used in the model. Do not include numeric attributes, text attributes, and tag attributes. This table is required when importing data.

Table Name	Description
DimensionOrder	Specifies the order of the dimensions in the Resource module, the Activity module, the Cost Object module, and the External Units module. The records in the DimensionOrder table must be sorted according to module type and sequence number. This table is required when importing data.
DimensionLevel	Specifies level names in a dimension hierarchy. This table is required when importing data.
DimensionMember	Specifies the hierarchy for each dimension. This table is required when importing data.
Account	Specifies the dimension intersections in the Resource module, the Activity module, and the Cost Object module. The combination of the columns DimRef<n> and DimMemberRef<n> for each valid dimension is the dimension signature. Do not include dimension signatures for the External Units module in this table. This table is required when importing data.
ExternalUnit	Specifies the dimension intersections for the External Units module. The combination of the columns DimRef<n> and DimMemberRef<n> for each valid dimension is the dimension signature. If the model does not use external units, then this table is not required when importing data.
EnteredCostElement	Specifies the entered cost elements and the account that is associated with each entered cost element. The account that is associated with an entered cost element can be specified with the column AccountReference or with the columns DimRef<n> and DimMemberRef<n>. If the model does not use entered cost elements, then this table is not required when importing data.
Driver	Specifies the drivers. This table is required when importing data.
Assignment	Specifies the assignments, including assignments from the External Units module. The accounts in an assignment can be specified with the columns SourceReference and DestinationReference or the columns SourceDimRef<n> and SourceDimMemberRef<n>, and DestinationDimRef<n> and DestinationDimMemberRef<n>. This table is required when importing data.

Table Name	Description
AssignmentNonUnique	Specifies the driver quantities on destination accounts for all drivers where the driver quantity type is nonunique. The driver quantity on a destination account can be specified with the column DestinationReference or the columns DimRef<n> and DimMemberRef<n>. This table is not required when importing data because the unique and nonunique driver quantities can be specified in the Assignment table.
ValueAttribute	Specifies the attribute hierarchy for numeric attributes, text attributes, and tag attributes. This table is not required when importing data.
ValueAttributePeriodicDef	Specifies the default value and formula for the numeric attributes. This table is required when importing data that has value attributes.
ValueAttributeAssociation	Specifies the accounts that are associated with numeric attributes, text attributes, and tag attributes. The accounts can be specified with the column ItemReference or the columns DimRef<n> and DimMemberRef<n>. This table is not required when importing data.
DimensionAttributeAssociation	Specifies the accounts that are associated with dimension attributes. The accounts can be specified with the column ItemReference or the columns DimRef<n> and DimMemberRef<n>. This table is not required when importing data.

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:

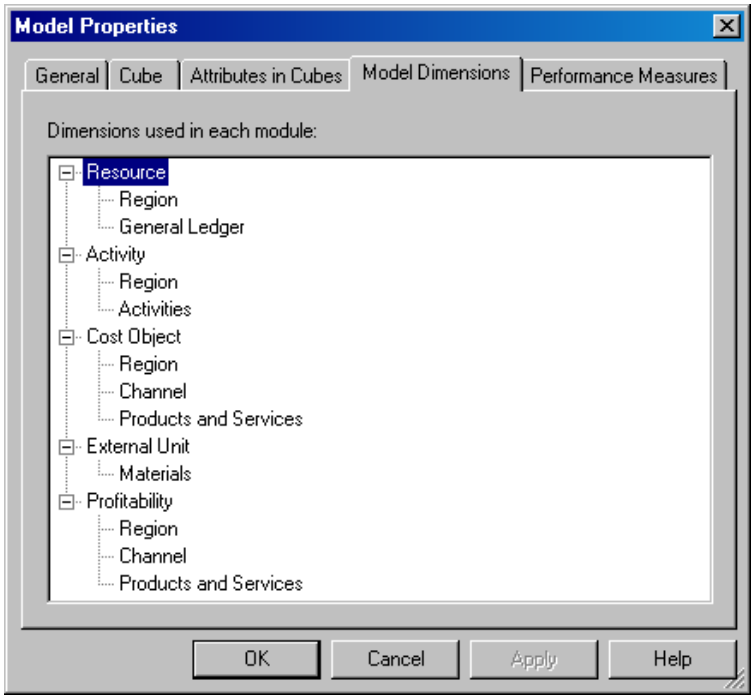
1. Import a sample model in XML format. The sample models are located in the following directory:

   C:\Program Files\<Client_Installation_Location>\Activity-Based Management Solution\Client\Samples\Models\Native

2. Create a target database and ensure that it exists to export your sample database.
3. Archive the model.

   This creates a set of database tables. For information about this task, see Chapter 33, “Working with Importing and Exporting Data,” on page 257.

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 previously shown, you can use the sample model directly. The External Units 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 model 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 Units module: two dimensions	Add 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 Activity-Based Management model.

Database Table Relationships, Dependencies, and Data Values

There are no database-enforced relationships between any of the SAS Activity-Based 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 Activity-Based 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 section. Here 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.

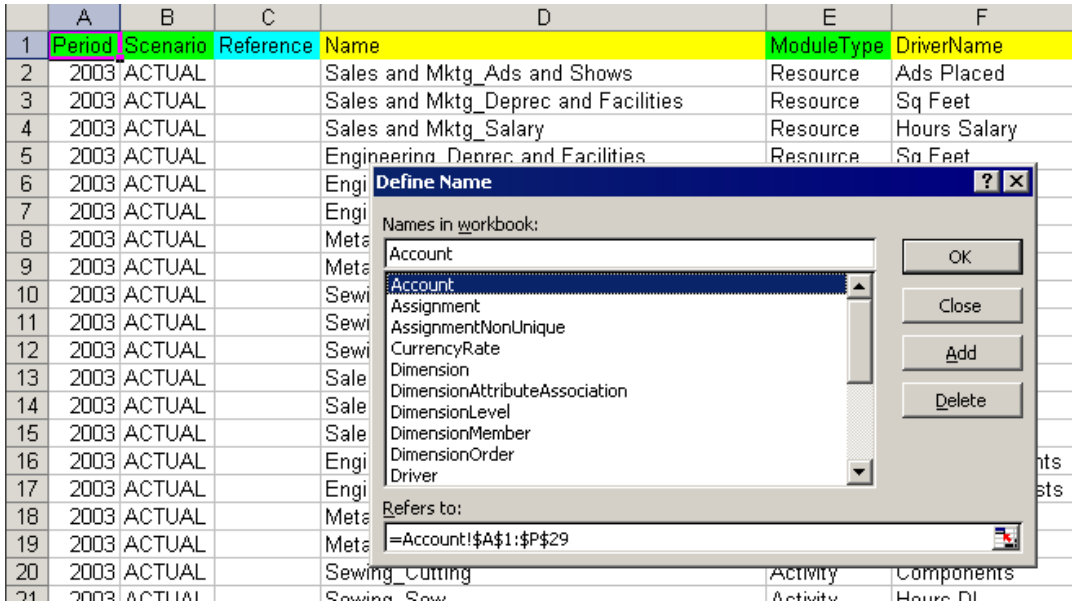
Data from Multiple Databases

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 | ABM Web Services Integration APIto write a program that uses an XML import configuration to specify each database in a separate StagingArea element. For information about the StagingArea element, see the Help. For information about the SAS | ABM Web Services Integration APIYour_Server_Name

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, 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:



Importing Data from an XML File

Overview

Before you import data from an XML file, ensure that the SAS Activity-Based Management server can access the XML file. To import the data, you can use the wizard, or you can write a program.

The XML file should be an XML file that was exported from SAS Activity-Based 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.

XML Import Configurations

Create one XML import configuration for new structures. Create a different XML import configuration to delete or update existing structures. Do not add, delete, and update structures in the same XML import configuration. For examples of XML import configurations, see the Help.

Period/Scenario Associations

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.

Exporting Model Data

Overview

You can export model data to a database (Microsoft SQL Server, Oracle, or Microsoft Access) 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 model data	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

Using the Export Wizard to Export Data

When you use the Export Wizard to export model data, you can choose to export to a database or to an XML file. If you export to a database, you can select individual database tables and properties to export. If you export to an XML file, you must export all of the model data.

Note: If the database or the XML file already exists, it is overwritten.

Archiving a Model

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 Activity-Based 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 Activity-Based Management server
- to transport a model between SAS Activity-Based Management servers

Exporting Data for Further Business Analysis

Exporting model data enables you to make better business decisions in the following areas:

Product costing

Determining a product's true cost is critical. Exporting results for product costing improves business by:

- focusing on leading, rather than lagging, indicators to corporate success
- indicating what products truly cost and focusing management efforts
- determining what products you should sell and your competitive advantage
- identifying how you can change your processes to improve costs

Results include:

- detailed product cost report
- product cost activity contribution report
- resource contribution analysis report
- activity and driver unit rates report for forecasting

Profitability analysis

The challenge is to determine the profitable products produced and the services delivered. Exporting results for profitability analysis improves business by:

- determining the most and least profitable product sold
- determining the most profitable customers and market segments served
- facilitating focused strategic marketing to emphasize profitable products
- steering corporate process improvement efforts to the greatest benefits

Results include:

- product profitability cliff analysis
- product profitability analysis
- customer and market segment profitability analysis
- activity effectiveness analysis
- resource consumption analysis

Planning and budgeting

The ability to predict future financial results and appropriately allocate resources is critical. Exporting results for planning and budgeting improves business by:

- identifying resource constraints or areas of excess capacity that restrict future growth
- increasing accuracy and accountability for the budget using activity-based analysis
- providing scenario analysis to evaluate strategic alternatives
- reducing the budgetary cycle time and improving responsiveness to market conditions

Results include:

- forecasted profitability report
- capacity utilization reports
- resource capacity constraints and cost reports
- activity effectiveness analysis
- budget variance reports for costs and demands

Service costing

Understanding the cost behavior of the services your organization delivers is critical. Effective management of services depends on understanding the activities required to deliver the services and the relative costs of those activities. Exporting results for service costing improves business by:

- increasing accountability between the customer demands and the costs incurred
- identifying the cause-and-effect nature of services and work performed
- determining the activity consumption for services delivered

- providing scenario analysis to evaluate service-level alternatives

Results include:

- services trend analysis report
- services activity consumption report
- services resource contribution report
- services profitability analysis by customer and service category

Shared services costing

Understanding the cost behavior of the shared services your organization delivers to internal customers is critical. Exporting results for shared service costing improves business by:

- increasing accountability between the customer demands and the costs incurred
- identifying the cause-and-effect nature of services and work performed
- determining the activity consumption for services delivered
- determining an appropriate benchmark for make-or-buy decisions for services
- providing scenario analysis to evaluate service-level alternatives

Results include:

- shared services trend analysis report
- shared services report for internal billing or cost allocations
- shared services activity consumption report
- shared services resource contribution report

Determining the Tables to Export for Business Analysis

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 Activity-Based 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, InventoryCost, ReceivedAllocatedCost, ReceivedAssignmentCost, ReceivedBocCost, ReceivedCost, ReceivedDrivenCost, ReceivedNonReciprocalCost, ReceivedReciprocalCost, UnassignedCost, UsedCost

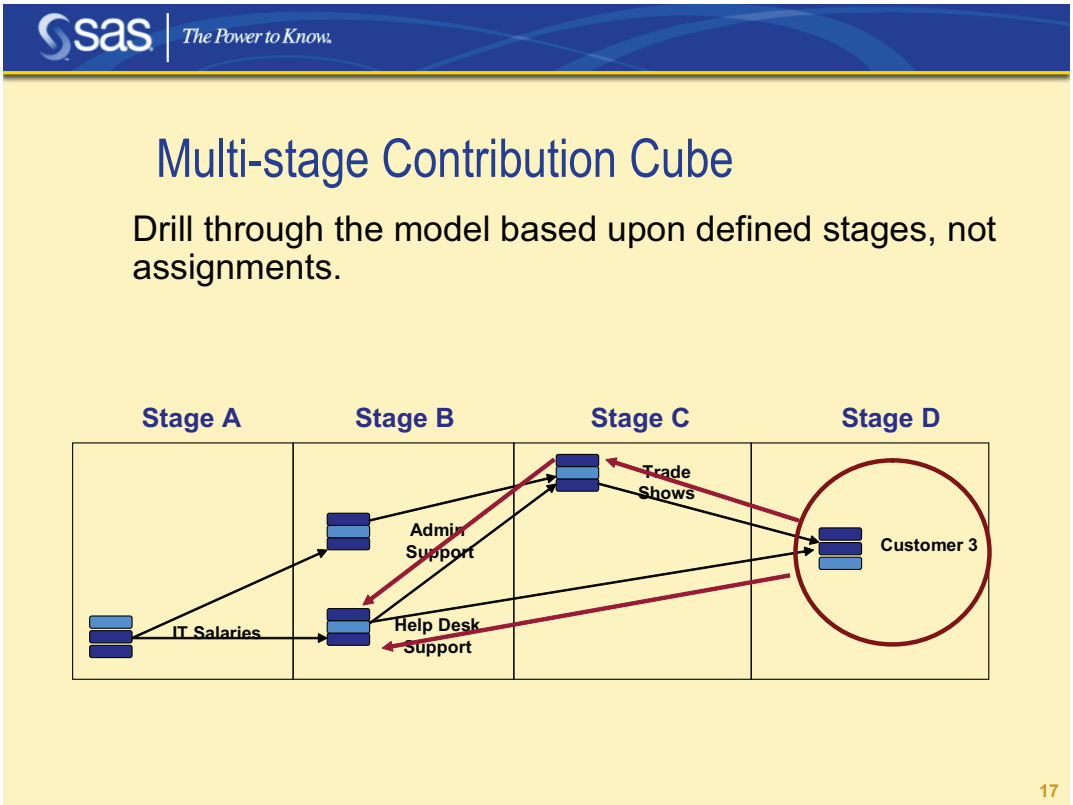
    - Driver data: AssignedIdleQuantity, DrivenQuantity, IdlePercentage, IdleQuantity, InventoryQuantity, OutputQuantity, TDQ, TDQBasic, TDQCalculated, UnassignedQuantity, UsedQuantity
    - Profitability analysis (uses both entered values and calculated values): Cost, Profit, Revenue, SoldQuantity, UnitCost, UnitProfit, UnitRevenue
- Attributes (specific to the model design)
    - Dimensional attributes used for grouping
    - Numerical attributes entered values
    - Calculated attributes performance measures

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 performance measures

Multi-stage contributions cube fact table (with calculated values)



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, then back to the original resource. In the previous example, all stages (A through D) are available for analysis in the multi-stage contributions cube.

For more information about fact tables, see Chapter 13, "Analyzing Model Data with OLAP," on page 89.

The resource contributions cube holds the cost flow from the end points of an assignment; the resource contributions cube enables you to compare the final destination with the original-entered cost elements. In the previous example, only Stage A and Stage D are available in the cube. The resource contributions cube is most useful in determining (for a given product) which departments contribute effort to produce the product.

Multi-stage cube

The types of fields in an exported multi-stage contributions cube include:

- Definitional
  - Keys for each account (resources, activities, cost objects, external units): 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 (resources, activities, cost objects, external units)

- Dimensional attributes used for grouping
- Numerical attributes entered values
- Calculated attributes performance measures

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
- Quantity
- 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 performance measures

Report table exports

The types of reports that you can generate include:

- Reports exported from SAS Activity-Based Management
-
- You can choose to export the standard staging tables or you can export the standard fields and contents, which are created in the standard reports that are installed with SAS Activity-Based Management.
- Reports on imported data content directly inside SAS Enterprise Guide
-
- Using the SAS Activity-Based Management Add-In for SAS Enterprise Guide, the SAS Activity-Based Management reports can be run directly from within SAS Enterprise Guide. This produces a SAS table that can be used in further analysis.

The exported report tables are consistent with the standard report templates that are formatted and shipped with SAS Activity-Based Management. For more information about the reports, see Chapter 14, “Reporting Model Data,” on page 95.

Report	Number of standard report templates	Multiple currencies supported
Destination furthest	1	1
Dimensional attribute cost	1	
Dimensional attribute unit cost	1	

Report	Number of standard report templates	Multiple currencies supported
Dimensional view	1	
Driver - cost and rate	1	
Idle capacity	1	
Module hierarchy	1	1
Multi-level contributions	1	
Multi-level contributions with cost elements	1	
Profit and loss depth horizontal	1	
Profit and loss depth vertical	1	
Profit and loss resource horizontal	1	
Profit and loss resource vertical	1	
Profit cliff	1	
Resource contributions	1	1
Resource contributions by attribute	1	
Resource contributions with cost elements	1	1
Single-stage assignments	1	1
Single-stage contributions	1	1
Unassigned costs	1	
Unit cost	1	1
Total number of export report tables	**21**	**7**

For more information on the standard report templates and their content, see the Help.

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. The Export Wizard shows you which fields are required for reimporting data.

Exporting, Filtering, and Limiting Calculated Results

Here 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 Activity-Based 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 dimensional reference and the dimensional member reference. These fields are required if you want to reimport the exported data into SAS Activity-Based 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 Activity-Based 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 Activity-Based 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 Activity-Based 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:

- selecting parts of a dimension
- specifying comparisons for the values of attributes
- specifying 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 2006". Use an underscore (_) to specify a single character. For example, Name LIKE account_ will match "account1", "account2", and "accountX". 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

Part 3

Building a Model

Chapter 18

Working in SAS Activity-Based Management

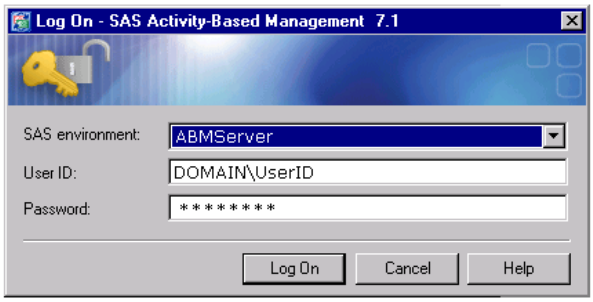
Introduction

This chapter contains task-based information about working in SAS Activity-Based Management. For more conceptual information, see Chapter 2, "Working in SAS Activity-Based Management," on page 9.

Connect to a Server

1. Start SAS Activity-Based Management.

   The Log On dialog box appears.



2. From the **SAS environment** drop-down list, select a SAS Metadata server.
3. Type your domain and User ID, for example HQ\JohnD.
4. Click **Log On**.

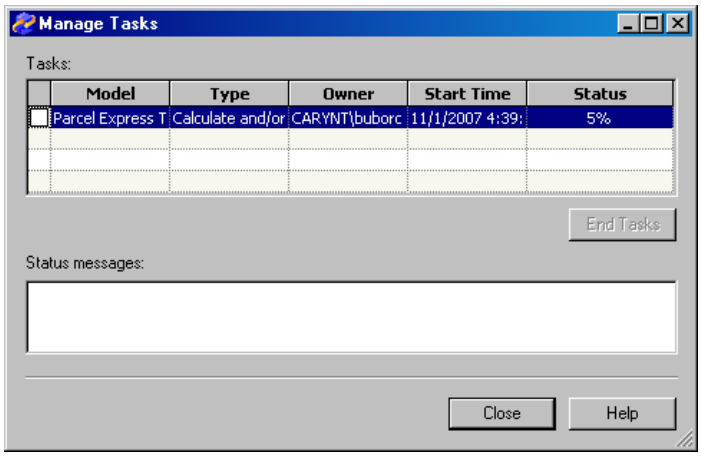
   You see the SAS Activity-Based Management Workspace Manager.

   Note: 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**).

Cancel a Server Task

1. **Select Tools ⇨ Manage Tasks**.

   The Manage Tasks dialog box appears.



2. 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.

3. Click **End Tasks**.

   There might be a delay before the task(s) is canceled.

Chapter 19

Working with Workspaces and Workspace Manager

Introduction

This chapter contains task-based information about workspaces and Workspace Manager. For more conceptual information, see Chapter 3, "Workspaces and Workspace Manager," on page 17.

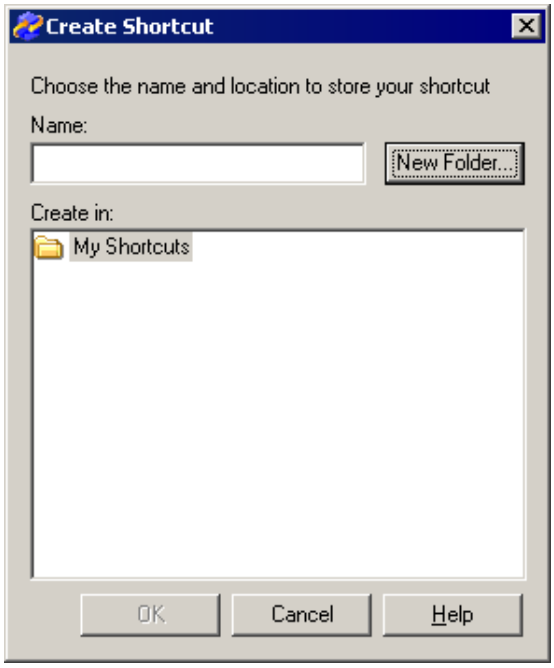
Create a Shortcut to a Workspace Item

1. Click the **Workspace Manager** tab in the Navigation Pane.

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

   The Create Shortcut dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. For information, see the Help.

5. In the **Create in** list, select the folder in which you want to place the shortcut.

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

   The New Folder dialog box appears.

7. Type the folder **Name**.

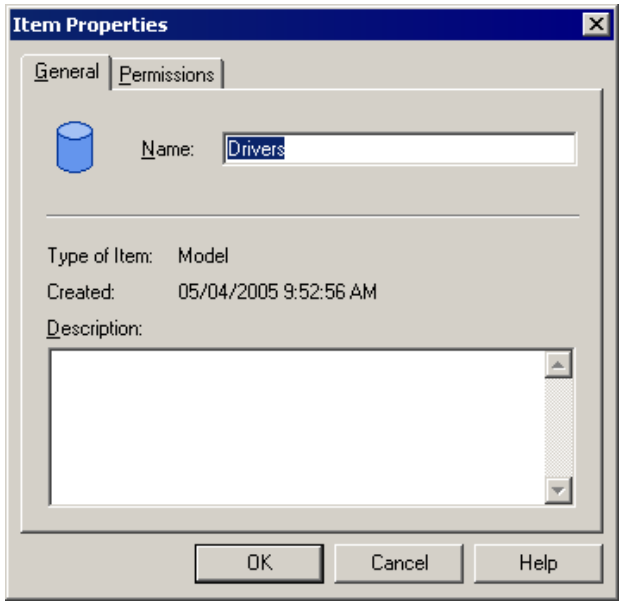
Tip: You can create a shortcut by dragging an object from the server area into **My Shortcuts**.

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.



3. Click the **General** tab.
4. Type the **Name**.

   The name must follow the naming conventions. For information, see the Help.

   Tip: When you rename an item, notify other users. Otherwise, they might look for a name that no longer exists.
5. Type the **Description**.
6. Click the **Permissions** tab.

   Note: Permissions are not applicable to shortcuts.
7. Select an **Owner** if you want to change the owner of the item.
8. In the **Privileges** list, select or clear the boxes that are next to each listed **Group**.

Create a Folder

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

   The New Folder dialog box appears.
3. Type the **Name**.

Move a Workspace Item

1. In Workspace Manager, select an item.
2. Drag the item to a new location.

   Note: You can drag the item only to a new location that is of the same type as the item.

Open a Workspace Item

1. In Workspace Manager, select an item.
2. Select **File** ⇨ **Open Workspace Item**.

Chapter 20

Working with Models

Introduction

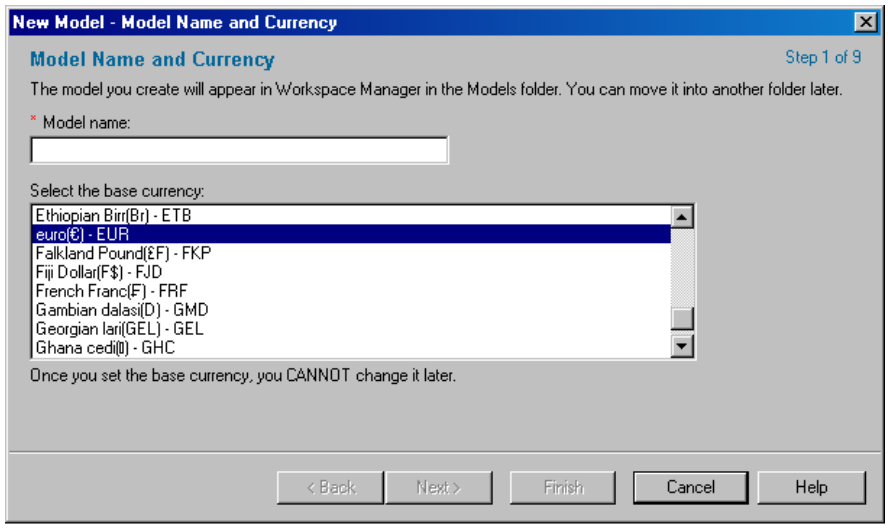
This chapter contains task-based information about working with models. For more conceptual information, see Chapter 4, “Models,” on page 23.

Create a Model

To be able to create a model, you must first understand the concepts of periods, scenarios, dimensions, and the modules. For information about periods and scenarios, see Chapter 5, “Periods and Scenarios,” on page 29. For information about dimensions, see Chapter 6, “Dimensions,” on page 35. For information about the modules, see Chapter 7, “Modules,” on page 43.

1. Select **File** ⇨ **New** ⇨ **Model**.

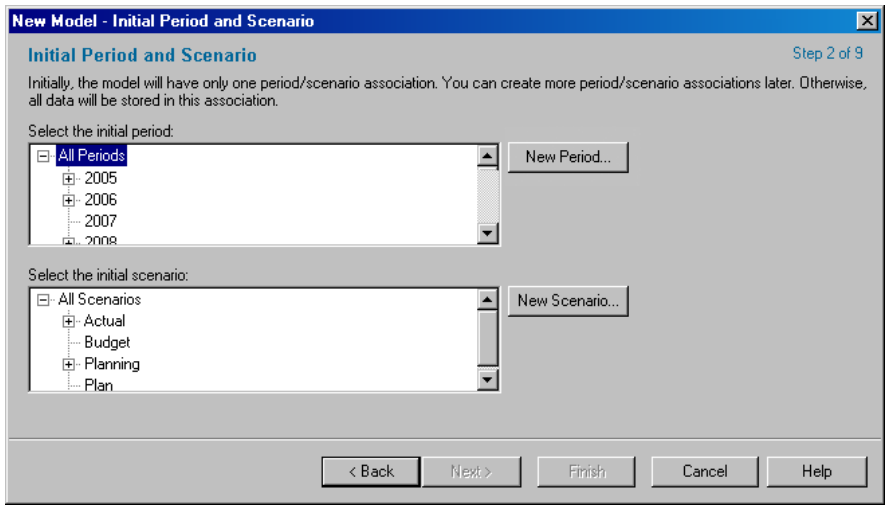
   The New Model Wizard appears.



2. In the **Name the model** box, type the name.
3. From the **Select the base currency** list, select a currency.

   Notice that once you select a base currency, you cannot change it later.

   Next, you will specify the initial period/scenario association. You can create additional periods, scenarios, and period/scenario associations later.
4. Click **Next**.

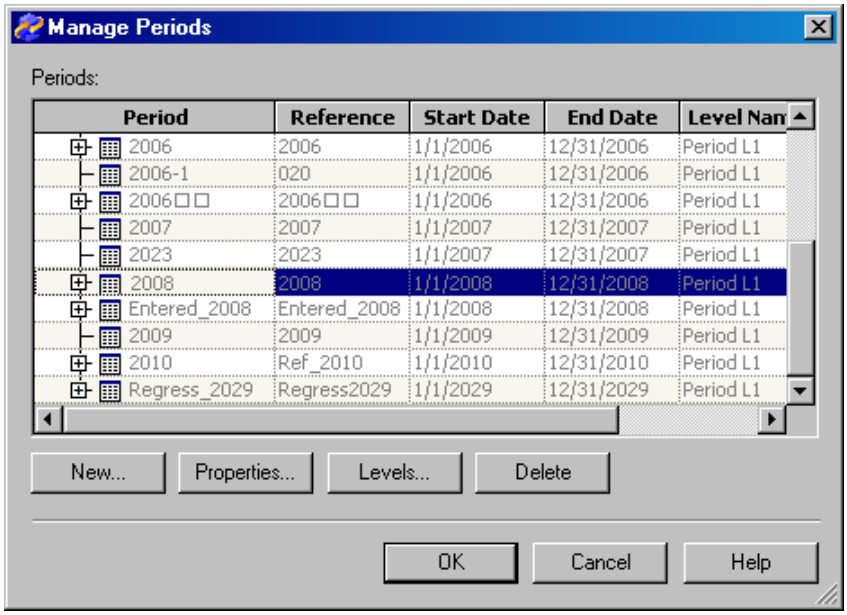


5. From the **Select the initial period** list, select a period.

   For information about periods, see Chapter 5, “Periods and Scenarios,” on page 29.
6. To create a new period, do the following:

   a. Click **New Period**.

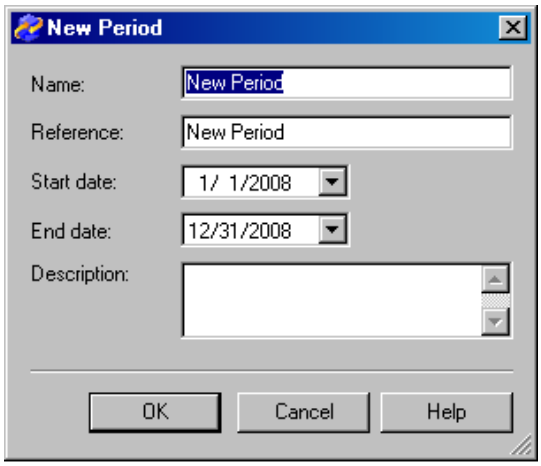
      The Manage Periods dialog box appears.



b. From the **Period** column, select a period under which to create the new period.

c. Click **New**.

The New Period dialog box appears.



d. Type the **Name**.

The name must follow the naming conventions. For information, see the Help.

e. 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. For information, see the Help.

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

g. Type the **Description**.

h. Click **OK**.

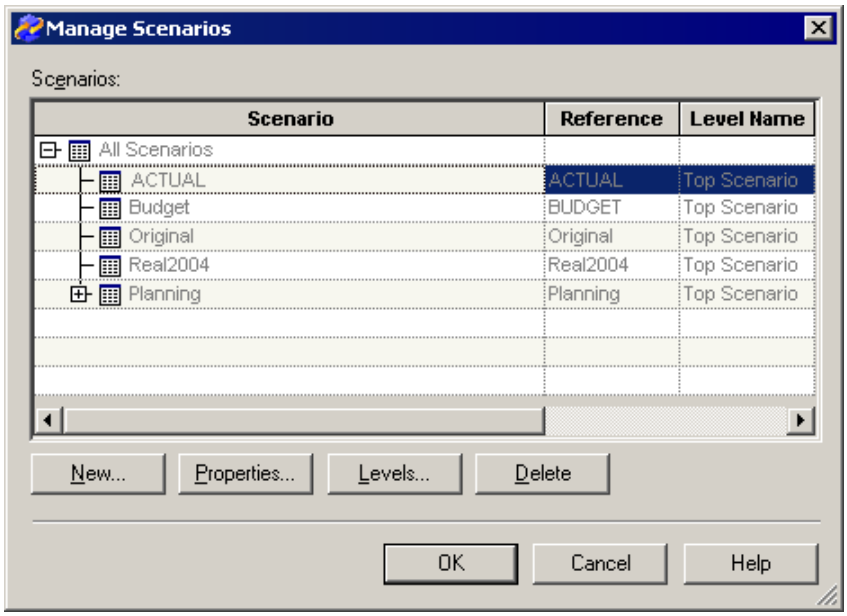
7. From the **Select the initial scenario** list, select a scenario.

For information about scenarios, see Chapter 5, “Periods and Scenarios,” on page 29.

8. To create a new scenario, do the following:

   a. Click **New Scenario**.

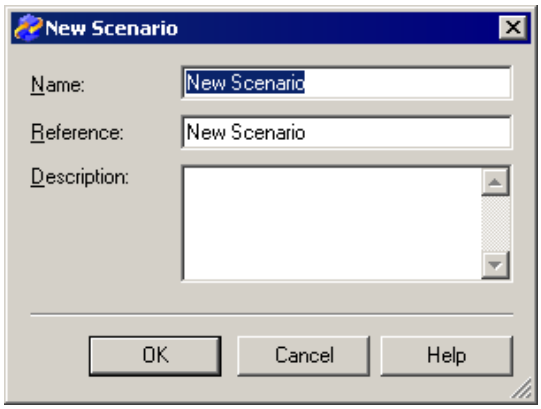
      The Manage Scenarios dialog box appears.



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

   c. Click **New**.

      The New Scenario dialog box appears.



   d. Type the **Name**.

      The name must follow the naming conventions. For information, see the Help.

   e. Type the **Reference**.

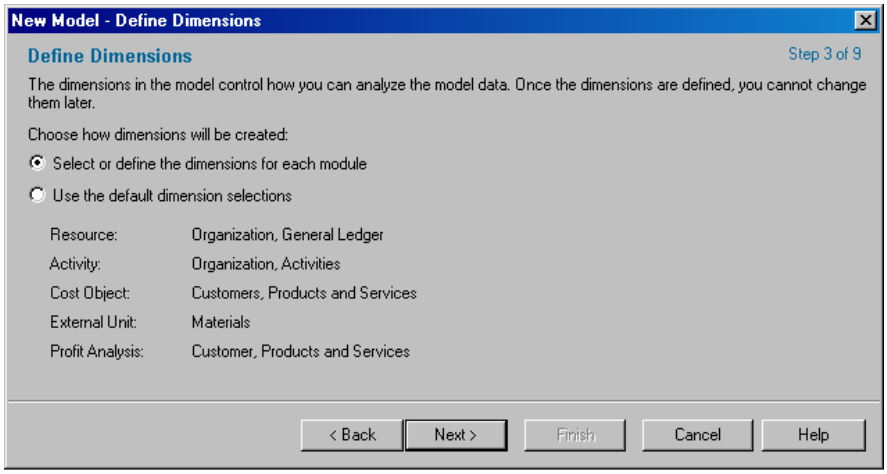
      A default reference is created from the scenario's name. If you change the reference, the new reference must follow the reference conventions. For information, see the Help.

   f. Type the **Description**.

   g. Click **OK**.

Next, you will specify the dimensions for each module: the Resource module, the Activity module, the Cost Object module, and the External Units module. In addition, you will specify the dimension for profit analysis.

9. Click **Next**.



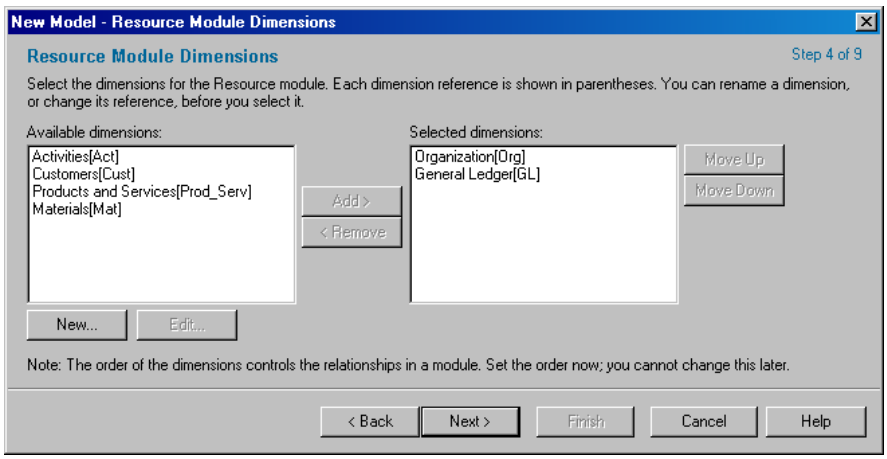
10. Select either the **Select or define the dimensions for each module** option or the **Use the default dimension selections** option.

    The default dimensions for each module are listed. These dimensions are routinely used by a variety of organizations, but they might not be appropriate for your organization or you might want additional dimensions.

    For information about dimensions, see Chapter 6, "Dimensions," on page 35.

    The remainder of this task assumes that you want to create additional dimensions for each of the modules. Each module (and profit analysis) will be presented in the wizard.

11. Click **Next**.



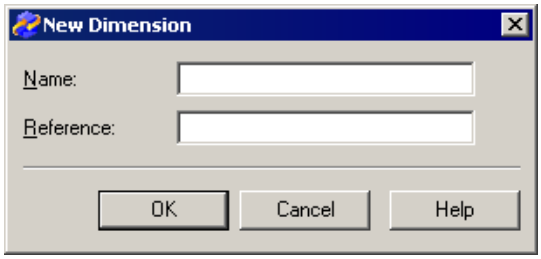
12. From the list of **Available dimensions** for the Resource module, select a dimension, and click  .

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

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

    a. Click **New**.

       The New Dimension dialog box appears.



b. Type the **Name**.

   The name must follow the naming conventions. For information, see the Help.

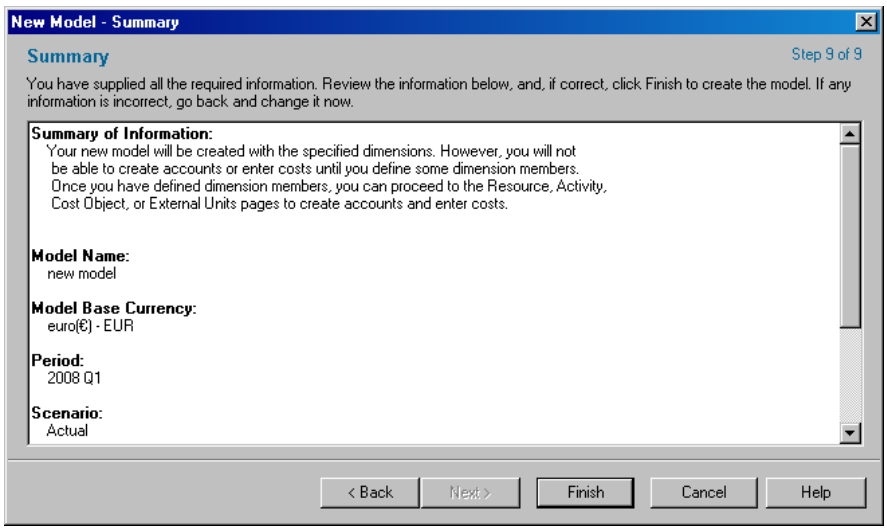
c. Type the **Reference**.

   A default reference is created from the dimension's name. If you change the reference, the new reference must follow the reference conventions. For information, see the Help.

The wizard presents each of the remaining modules and profit analysis so that you can specify the dimensions. Each of these pages is identical to the page previously described.

After you have defined the dimensions for profit analysis, you will review the summary of what will be created.

14. Click **Next**.



15. Review the summary information.

16. If you need to change any information, click **Back** until you reach the page that you need to change in the wizard.

    All the information that you have specified is saved. Click **Next** to advance through the wizard.

17. Click **Finish**.

    The model is created and the Dimensions page appears. Next, dimensions must be created. For information about the task of creating dimensions, see Chapter 22, “Working with Dimensions,” on page 155.

Start Model Mode

To perform any of the modeling tasks (except creating a model), you must be in Model mode.

Click the **Models** tab in the Navigation Pane.

Open a Model without a Model Already Open

If you start Model mode without a model already open, you must open a model from the Models workspace.

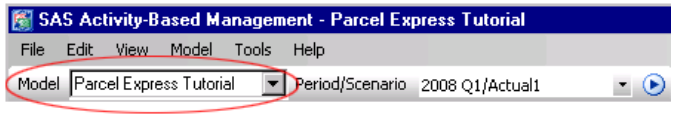
The list of folders and the list of models correspond to the Models branch of the server area in Workspace Manager. For more information, see Chapter 3, "Workspaces and Workspace Manager," on page 17.

1. From the list of **Folders**, select a folder.
2. From the list of **Models**, click a model.

Open a Model with a Model Already Open

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  .



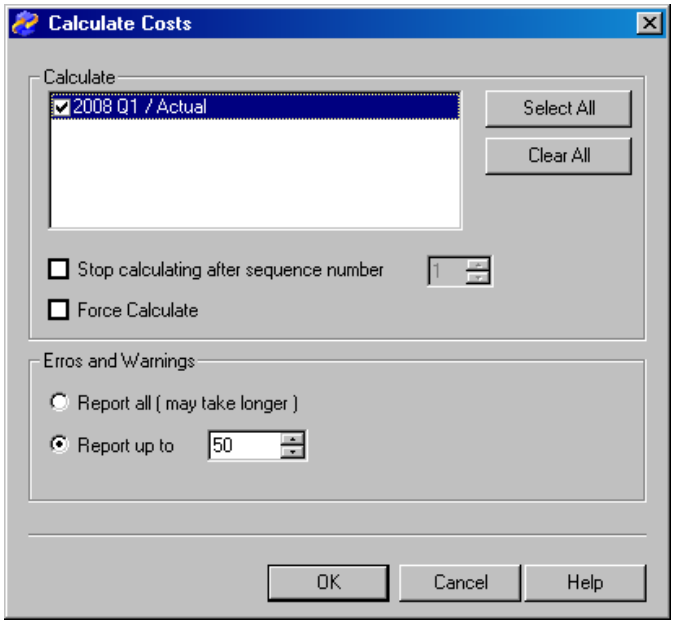
  If you do not see the model that you want to use, you probably do not have permission to view it. For information about permissions, see Chapter 3, "Workspaces and Workspace Manager," on page 17.

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

Calculate Costs

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

   The Calculate Costs dialog box opens.



2. Select one or more period/scenario associations.
3. Select or clear the **Stop calculating after sequence number** option. If applicable, specify the sequence number.

   For information about driver sequencing, see Chapter 9, “Drivers,” on page 57.
4. Select or clear the **Force calculate** option.

   You can select to perform calculations even if the flag is set that says that calculations are up to date. To speed processing, SAS Activity-Based Management skips calculating if it thinks that calculations are up to date. Use this option to force calculations anyway.
5. For **Errors and Warnings**, select an option.

   For information about errors, see Chapter 4, “Models,” on page 23.
6. Click **OK**.

   You see a message in the status bar, at the bottom of the window, indicating that the calculation is in progress.

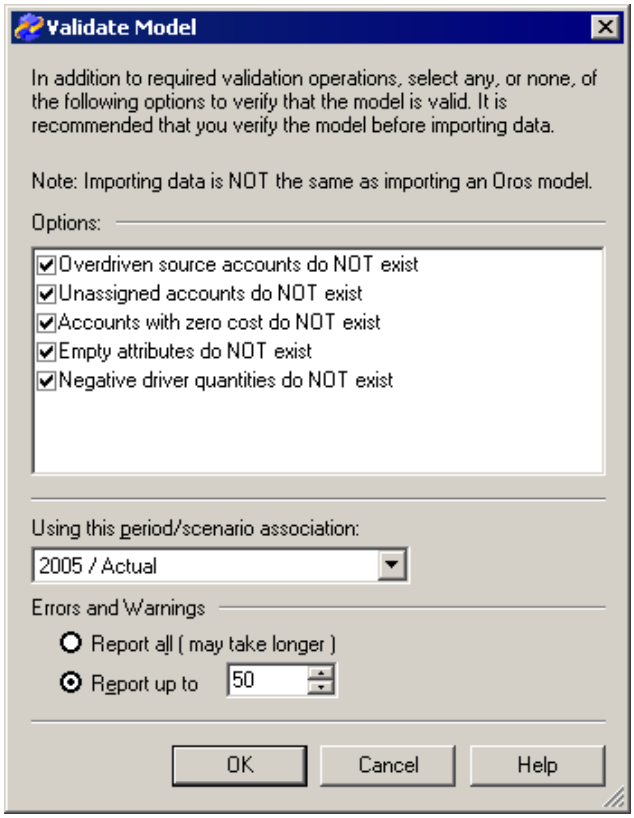
Validate a Model

For information about validating a model, see Chapter 8, “Accounts and Cost Elements,” on page 51.

1. Open a model in Model mode.

2. Select **Model** ⇨ **Validate**.

   The Validate Model dialog box appears.



3. From the list of **Options**, select or clear the options.
4. From the **Using this period/scenario association** drop-down list, select either a period/scenario association or **<All>**.
5. For **Errors and Warnings**, select an option.

   For information about errors, see Chapter 4, “Models,” on page 23.

Chapter 21

Working with Periods and Scenarios

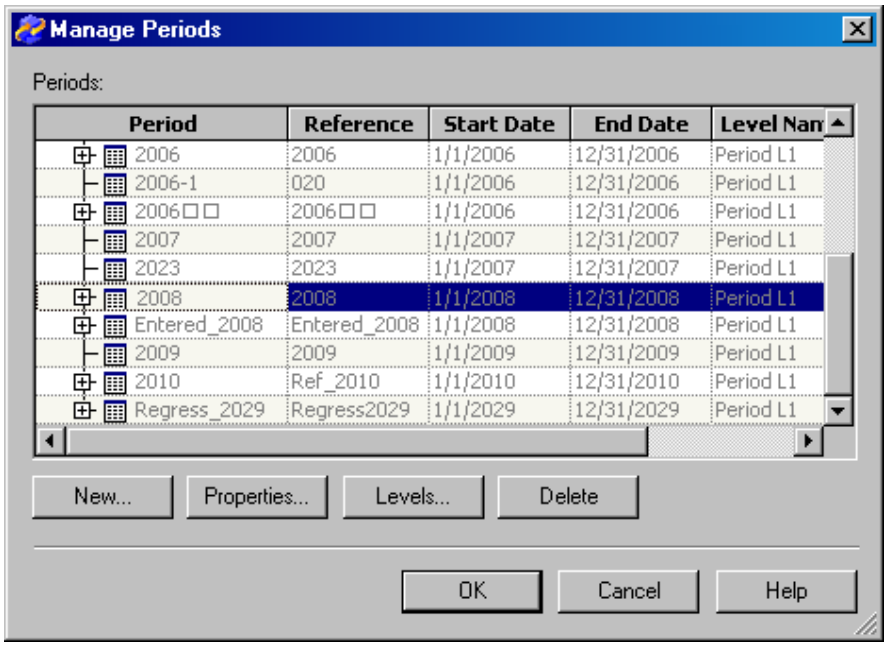
Introduction

This chapter contains task-based information about working with periods and scenarios. For more conceptual information, see Chapter 5, “Periods and Scenarios,” on page 29.

Create a Period

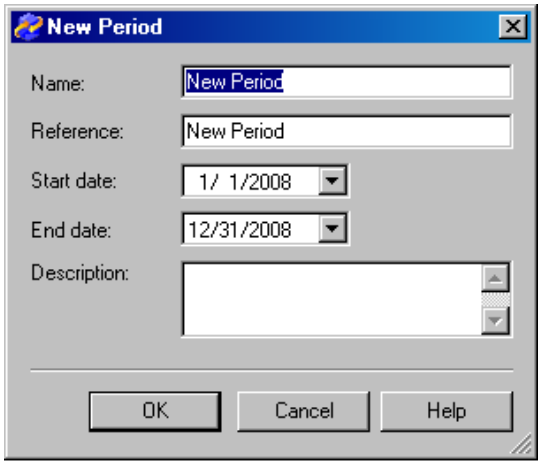
1. Select **Tools** ⇨ **Manage Periods**.

   The Manage Periods dialog box appears.



2. From the **Period** column, select a period under which to create the new period.
3. Click **New**.

   The New Period dialog box appears.



4. Type the **Name**.

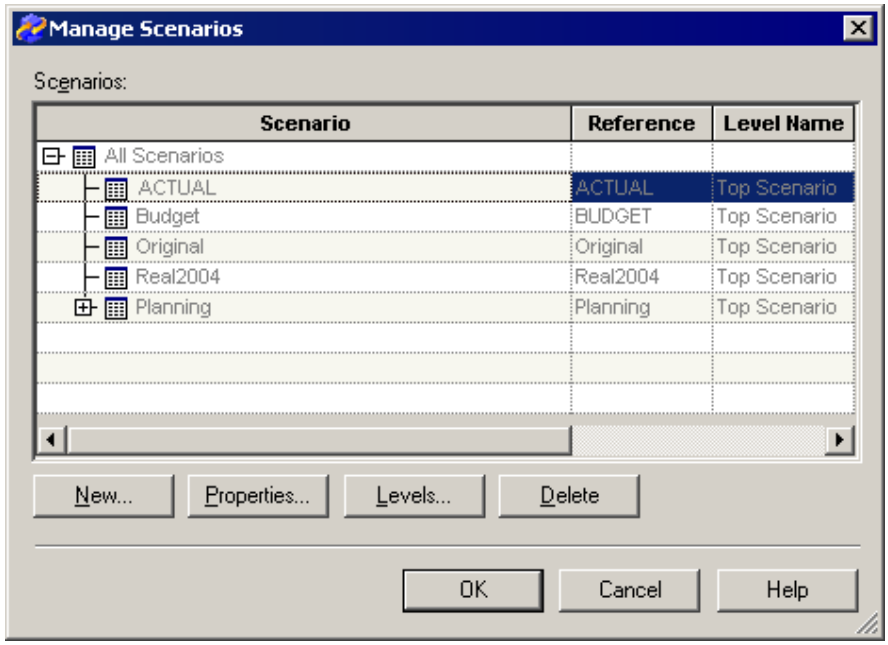
   The name must follow the naming conventions. For information, see the Help.
5. Type the **Reference**.

   A default reference is created from the period's name. If you change the reference, the new reference must follow the reference conventions. For information, see the Help.
6. Type the **Start date** and the **End date**, or select dates from the drop-down calendar.
7. Type the **Description**.

Create a Scenario

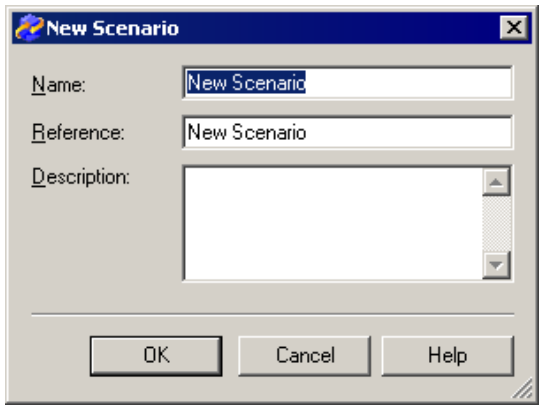
1. Select **Tools** ⇨ **Manage Scenarios**.

   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. For information, see the Help.

5. Type the **Reference**.

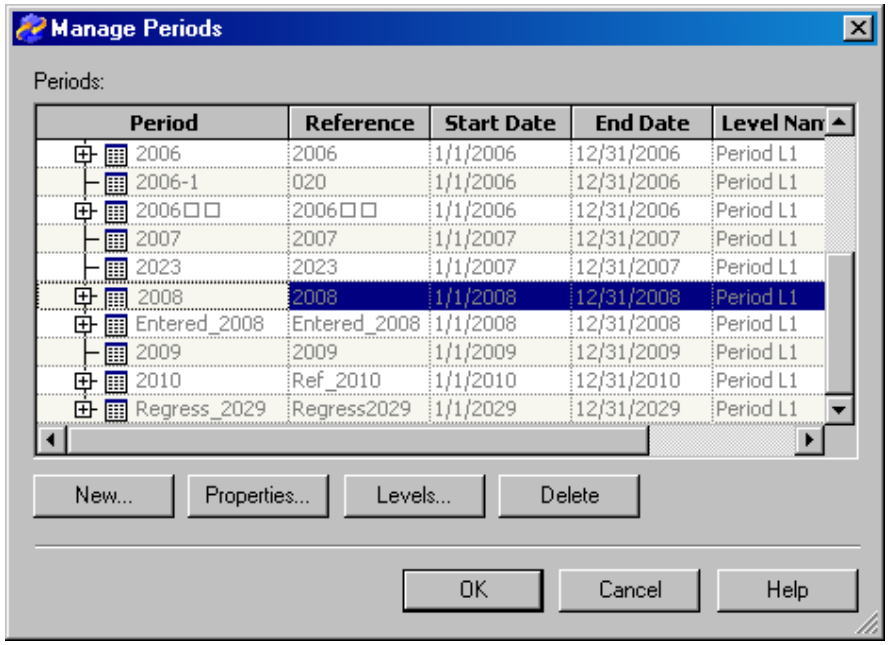
   A default reference is created from the scenario's name. If you change the reference, the new reference must follow the reference conventions. For information, see the Help.

6. Type the **Description**.

Manage Period Level Names

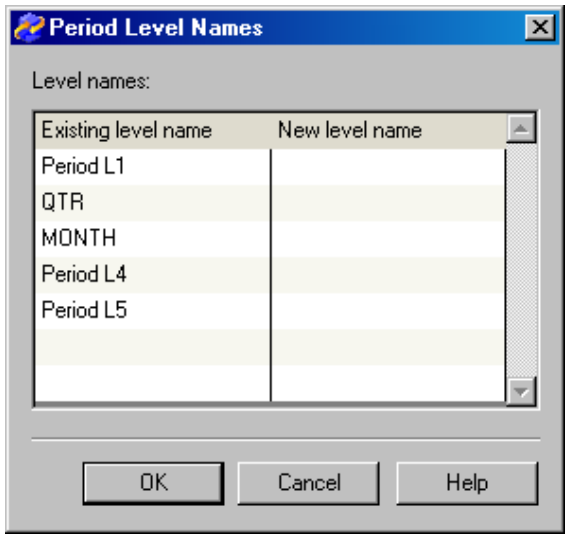
1. Select **Tools** ⇨ **Manage Periods**.

   The Manage Periods dialog box appears.



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

   The Period Level Names dialog box appears.



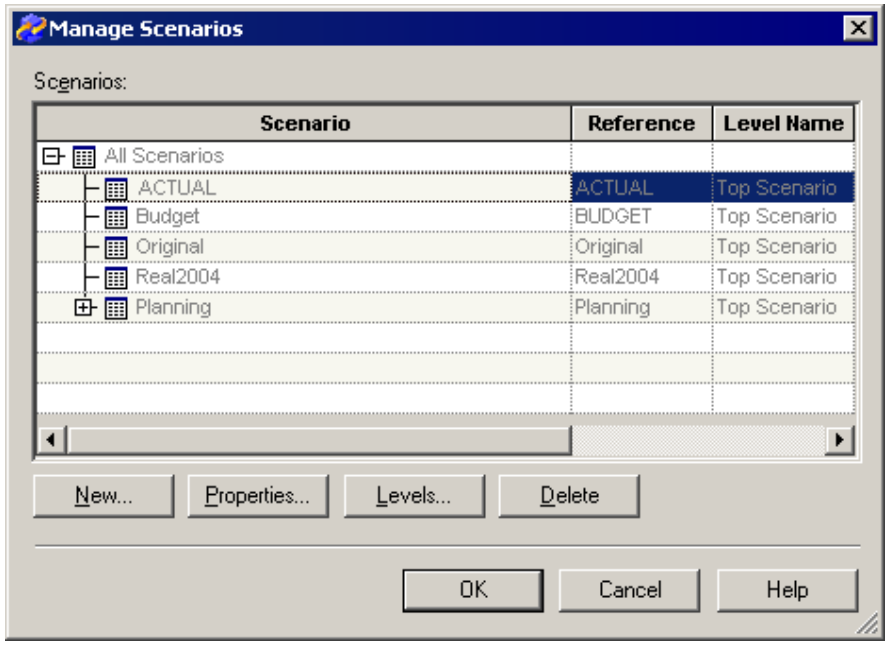
4. In the **New level name** column, double-click in a row and type a name.

   The name must follow the naming conventions. For information, see the Help.

Manage Scenario Level Names

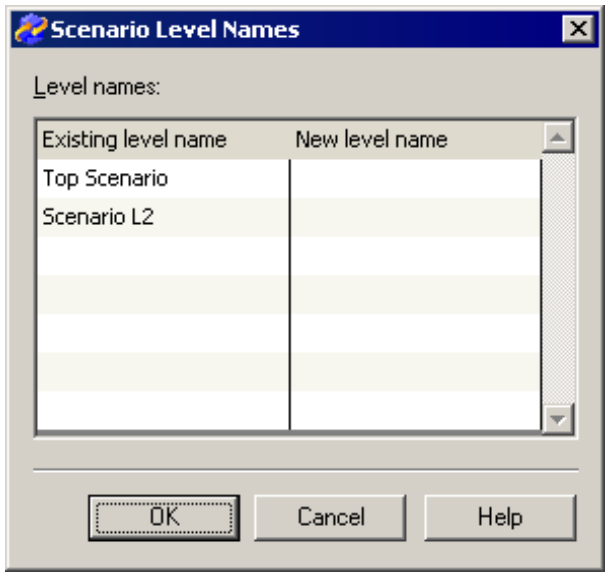
1. Select **Tools** ⇨ **Manage Scenarios**.

   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. For information, see the Help.

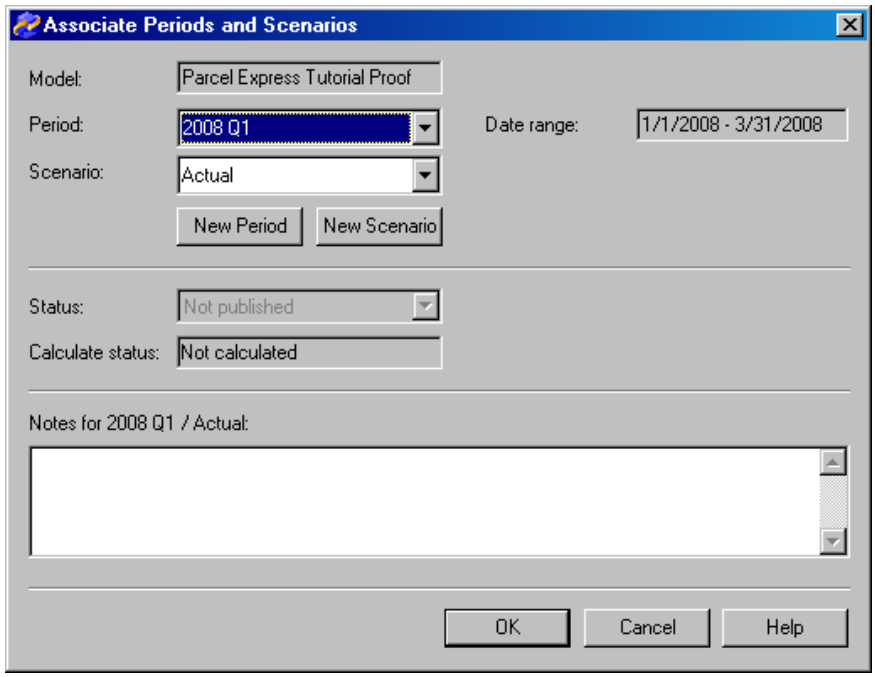
Create a Period/Scenario Association

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.

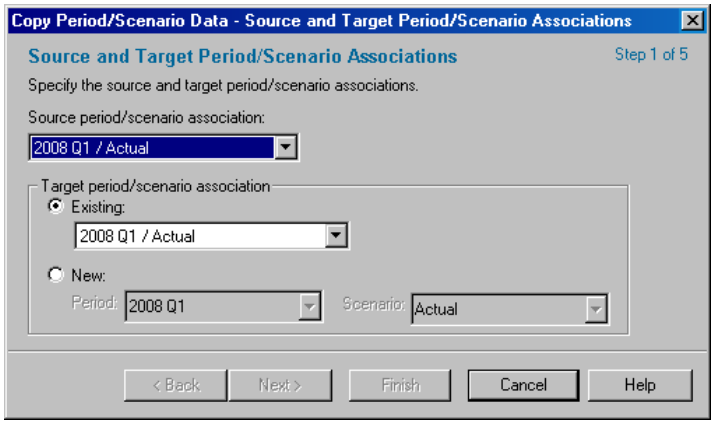
   b. Select the **Scenario**.

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

Copy Period/Scenario Data

1. Select **Model** ⇨ **Copy Period/Scenario Data**.

The Copy Period/Scenario Data dialog box appears.



2. Select a **Period/scenario**.
3. To copy data to an existing period/scenario association, do the following:
   a. Select the **Existing period/scenario association** option.
   b. From the drop-down list on the right, select a period/scenario association.
4. To copy data to a new period/scenario association, do the following:
   a. Select the **New period/scenario association** option.
   b. Select a **Period** and a **Scenario**.

Publish or Unpublish a Period/Scenario Association

1. Select **Model** ⇨ **Period and Scenario Associations Page**.

   The Period and scenario associations page appears.
2. Select a period/scenario association.
3. Select **Edit** ⇨ **Publish/Unpublish**.

Chapter 22

Working with Dimensions

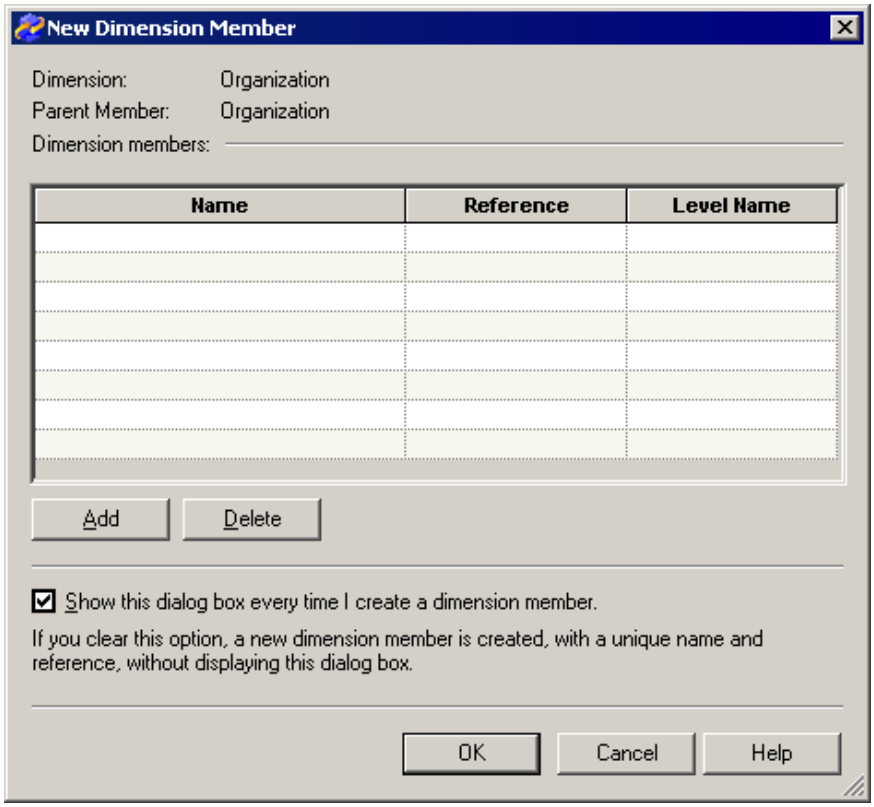
Introduction

This chapter contains task-based information about working with dimensions. For more conceptual information, see Chapter 6, “Dimensions,” on page 35.

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. For information, see the Help.

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. For information, see the Help.

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.

Delete a Dimension Member

For information about deleting accounts, see Chapter 8, “Accounts and Cost Elements,” on page 51.

1. On a module page, delete the account that corresponds to the dimension member.

2. On the Dimensions page, select the dimension member.

3. Select **Edit** ⇨ **Delete**.

Reorder Dimension Members

You can control the display order of accounts by reordering dimension members on the Dimensions page. The order that you establish is preserved when you export and import a model. You can reorder accounts using either the mouse or the keyboard.

To reorder dimension members using a mouse, do the following:

1. Go to the Dimensions page 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.

When you open an assignments pane, you can notice that its accounts are ordered according to the order that you establish for its dimension members. And, the new order is reflected in cubes

To reorder dimension members using the keyboard, do the following:

1. Go to the Dimensions page.
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.

Changing the Parent of a Dimension Member

You can use either 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.

To change the parent of a dimension member using the mouse, do the following:

1. Go to the Dimensions page and select the dimension member that you want to reparent.
2. Hover over the dimension member until 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 reparent the item. If you confirm, then the item is moved under the new parent.

   Note: 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.

Chapter 23

Working with Accounts and Cost Elements

Introduction

This chapter contains task-based information about working with accounts and cost elements. For more conceptual information, see Chapter 8, "Accounts and Cost Elements," on page 51.

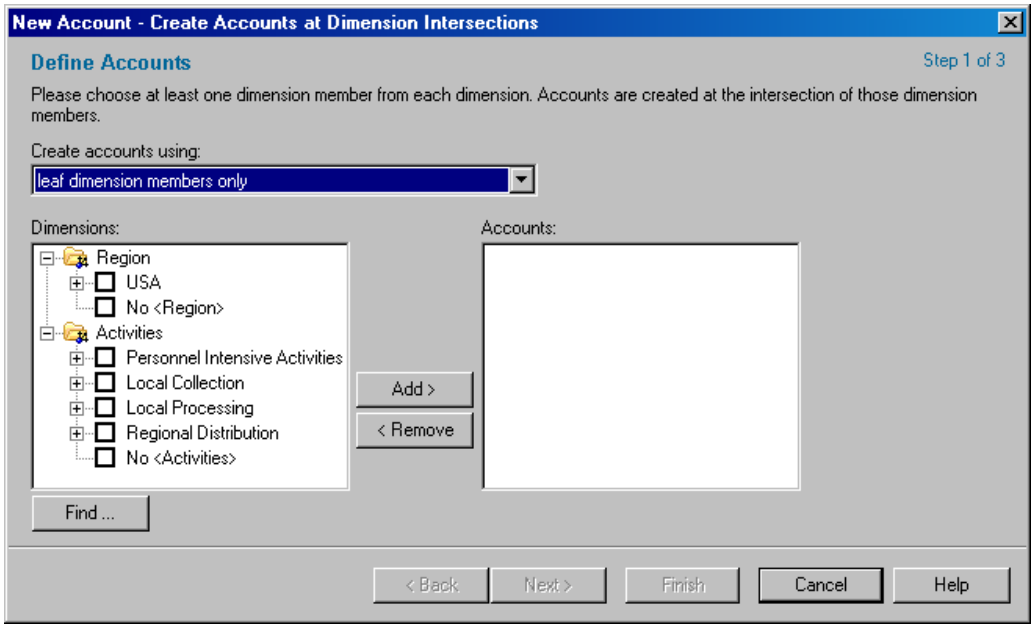
Create an Account

Provide Minimal Information

This task describes the minimal amount of information needed to create an account.

1. If the underlying dimension member does not exist, then create the dimension member.
2. Open a model in Model mode, and select **Model** ⇨ **<name of module>**. For example, **Model** ⇨ **Resource Module**.
3. Select a roll-up account or an account.
4. Select **Edit** ⇨ **New Account**.

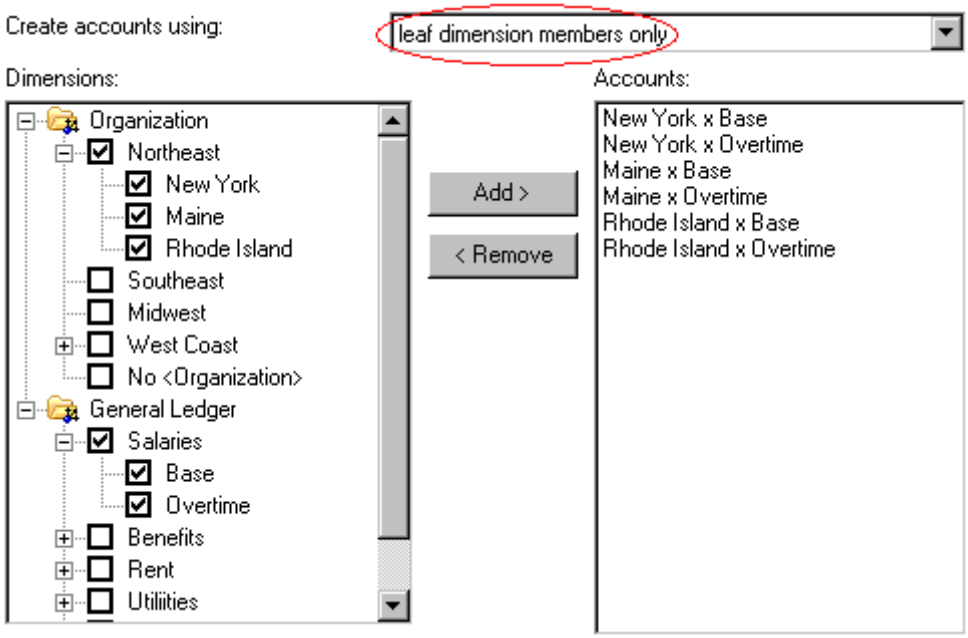
   The New Account Wizard appears.



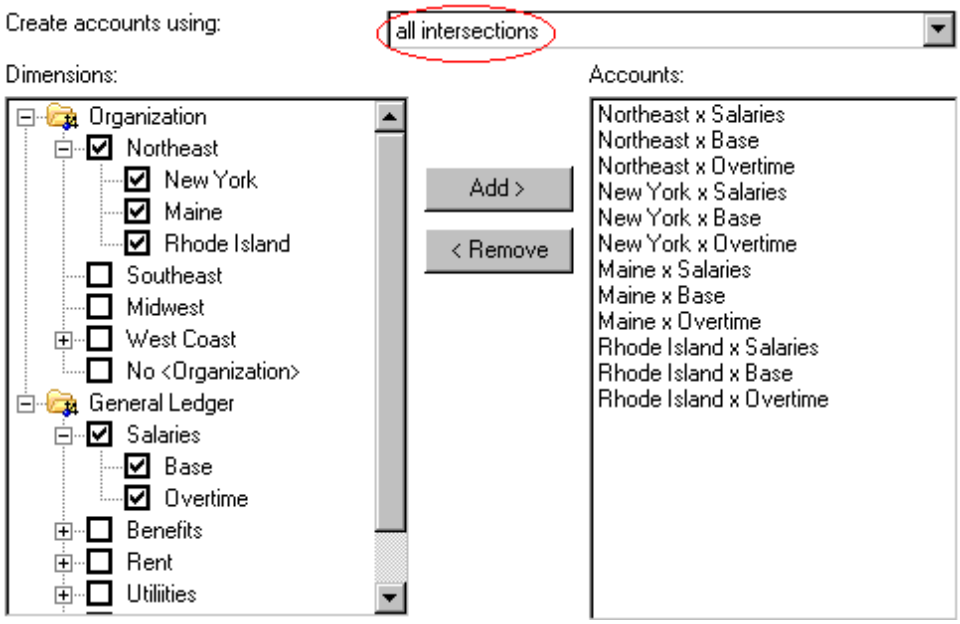
5. From the list of **Dimensions**, select at least two dimensions.
6. From the **Create accounts using** drop-down list, select a value.

   The effect of each value follows:

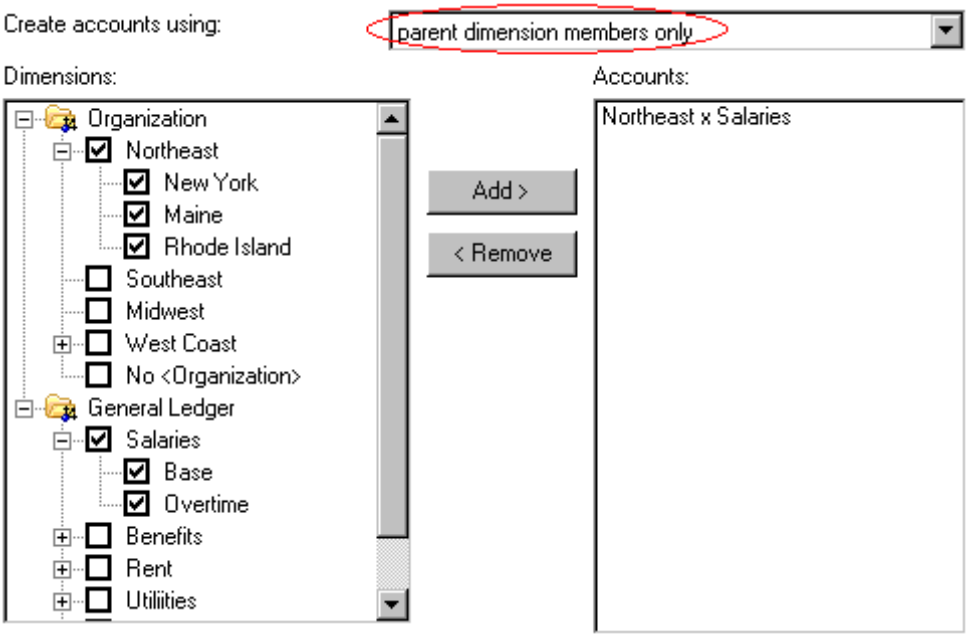
   Selecting **leaf dimension members only** creates accounts only for the intersections of the lowest dimension members in each dimension, as shown:



   Selecting **all intersections** creates accounts for the intersections of all the dimension members in each dimension, as shown:



Selecting **parent dimension members only** creates accounts only for the intersections of the highest dimension members, as shown:



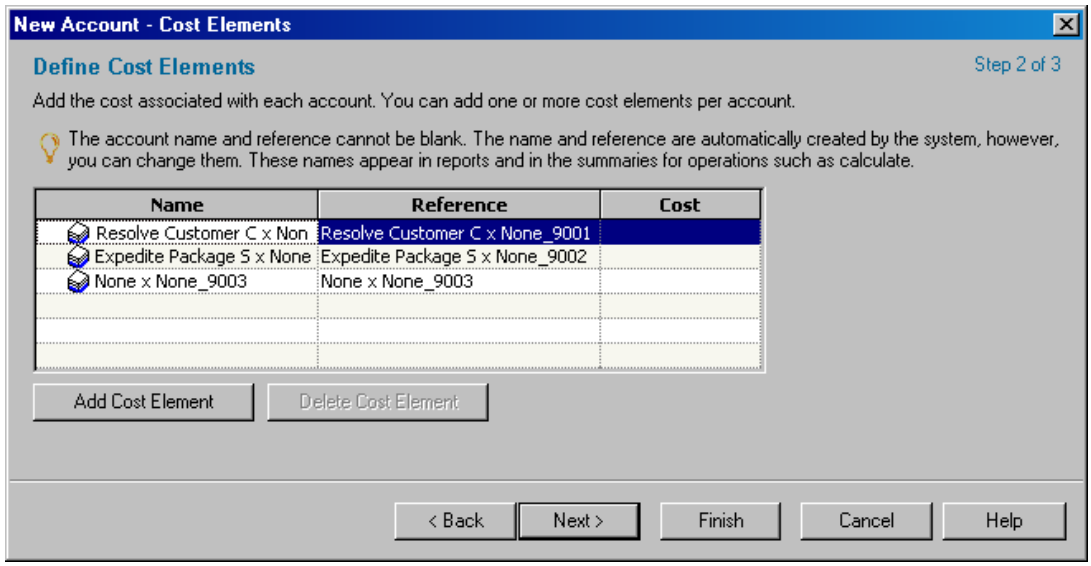
Note: The name shown in the list of **Accounts** is displayed in a module in the **Display Name** column. The display name uniquely identifies an item and is created by SAS Activity-Based Management; you cannot change the display name. However, you can change the account name. You will have the opportunity to change the account name in the next step of the wizard.

At this point, you have specified the minimal information needed to create an account using default information.

7. If you do not want to change an account name, change an account reference, or create cost elements, click **Finish**.

Provide Optional Information

8. Click **Next**(assuming that you did not click **Finish** in step 7).



9. To change the name of an account, click in the **Name** column and type a new name. The name must follow the naming conventions. For information, see the Help.
10. To change the reference of an account, click in the **Reference** column and type a new reference.

    The reference must follow the reference conventions. For information, see the Help.

    Note: At this point in the New Account Wizard, you can create cost elements.
11. Click **Finish**.

Create Cost Elements When an Account Is First Created

1. On the second page of the New Account Wizard, click **Add Cost Element**.

   A new entered cost element with default information appears.

2. To change the name of the cost element, click in the **Name** column and type a new name.

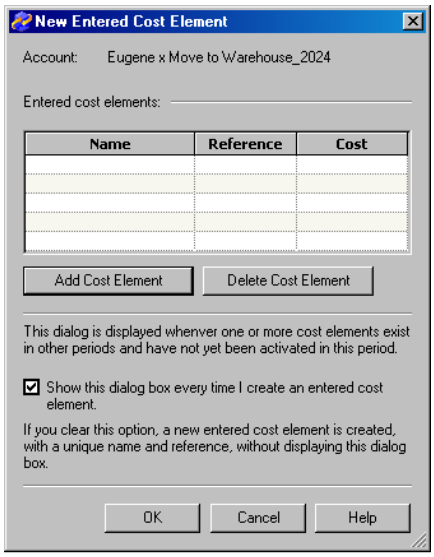
   The name must follow the naming conventions. For information, see the Help.
3. To change the reference of the cost element, click in the **Reference** column and type a new reference.

   The reference must follow the reference conventions. For information, see the Help.
4. To enter the cost of the cost element, click in the **Cost** column and type a cost.

Create an Entered Cost Element on a Module Page

1. In a module, select an account (not a roll-up account).
2. Select **Edit** ⇨ **New Entered Cost Element**.

   The New Entered Cost Element dialog box appears.



3. Click **Add Cost Element**.

   A new row is added to the **Entered cost elements** list. The new row contains default information.
4. Click in the **Name** column and type the name.

   The name must follow the naming conventions. For information, see the Help.
5. Click in the **Reference** column and type the reference.

   A default reference is created from the cost elements name. If you change the reference, the new reference must follow the reference conventions. For information, see the Help.
6. Click in the **Cost** column and type the cost.
7. To delete a cost element, do the following:

   a. From the list of **Entered cost elements**, select an element.

   b. Click **Delete Cost Element**.

Chapter 24

Working with Drivers

Introduction

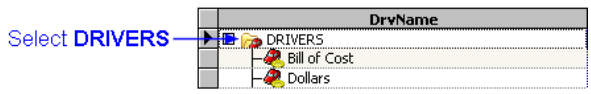
This chapter contains task-based information about working with drivers. For more conceptual information, see Chapter 9, “Drivers,” on page 57.

Create a Driver

1. Open a model in Model mode, and select **Model** ⇨ **Drivers Page**.

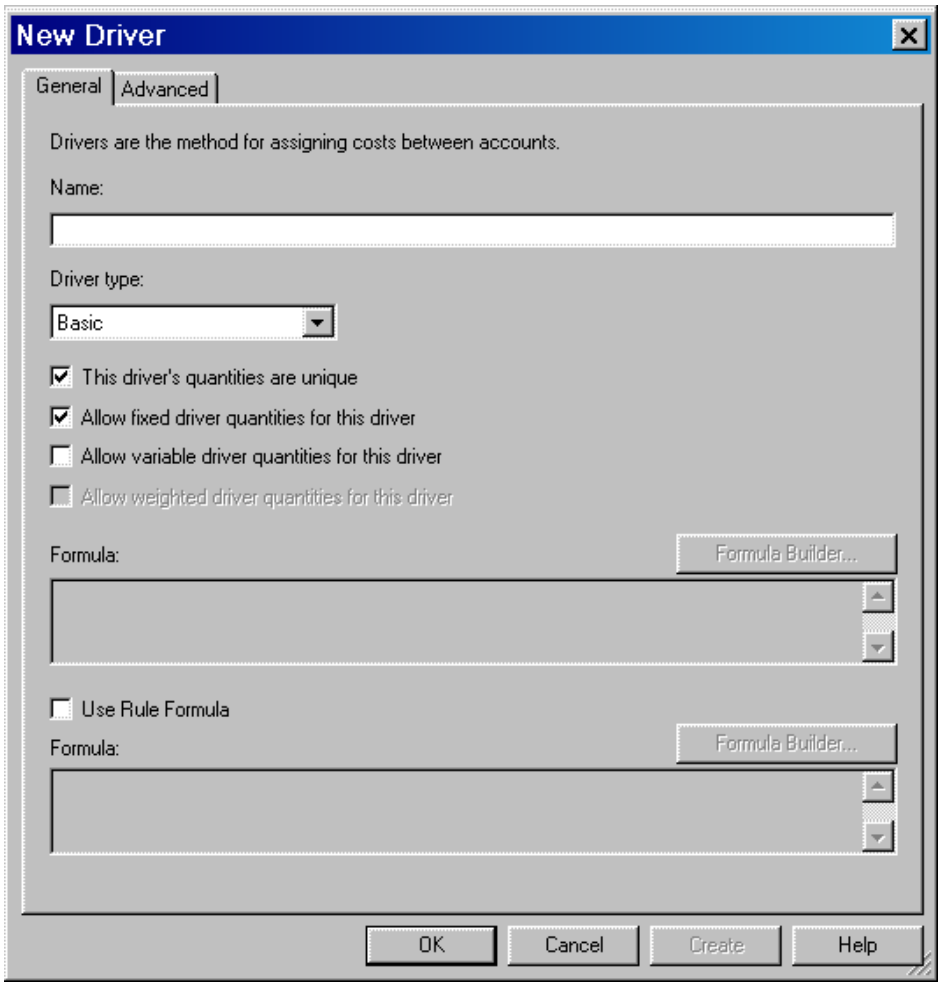
   The Drivers page appears.

2. Select the **DRIVERS** folder.



3. Select **Edit** ⇨ **New Driver**.

   The New Driver dialog box appears.



4. Type the **Name**.

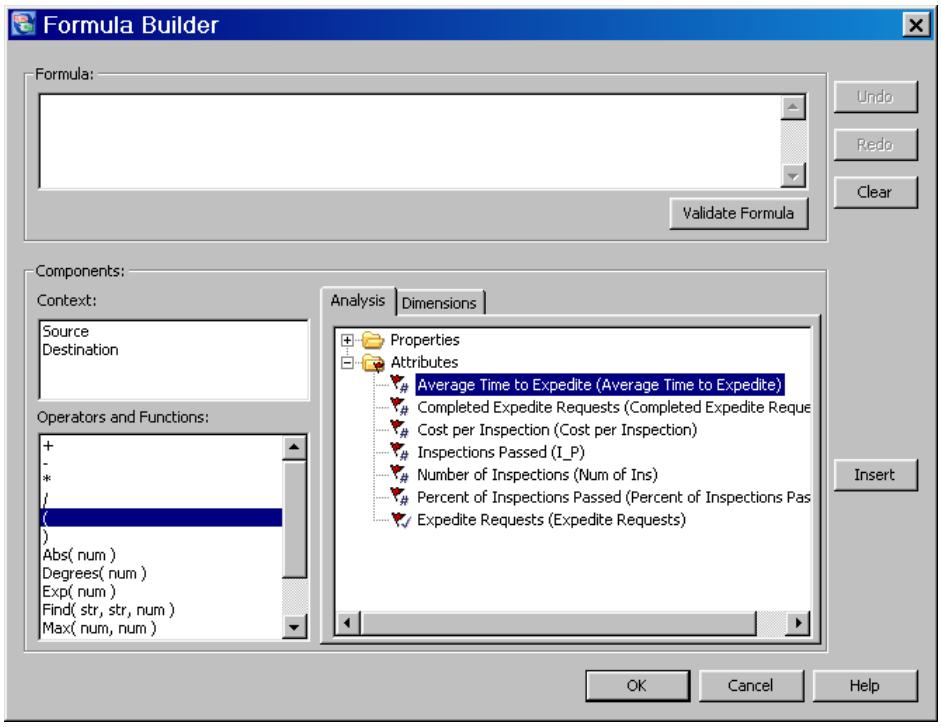
   The name must follow the naming conventions. For information, see the Help.

5. Select the **Driver type**.

6. If you selected a driver type that supports unique driver quantities, then either select or clear the **This driver's quantities are unique** option.

7. If this is a calculated driver, then click **Formula Builder**.

   The Formula Builder dialog box appears.



TIP You can also modify the formula in the **Formula** box.

For information about formulas, see Chapter 12, "Using Formulas," on page 87.

8. Define a formula:

   a. Select an item from the list of **Numeric properties**, from the list of **Operators**, or from the list of **Attributes**.

   b. Click **Insert**.

      The item is added to the formula.

      TIP You can also type the formula.

   c. Continue to insert items until you have built the entire formula.

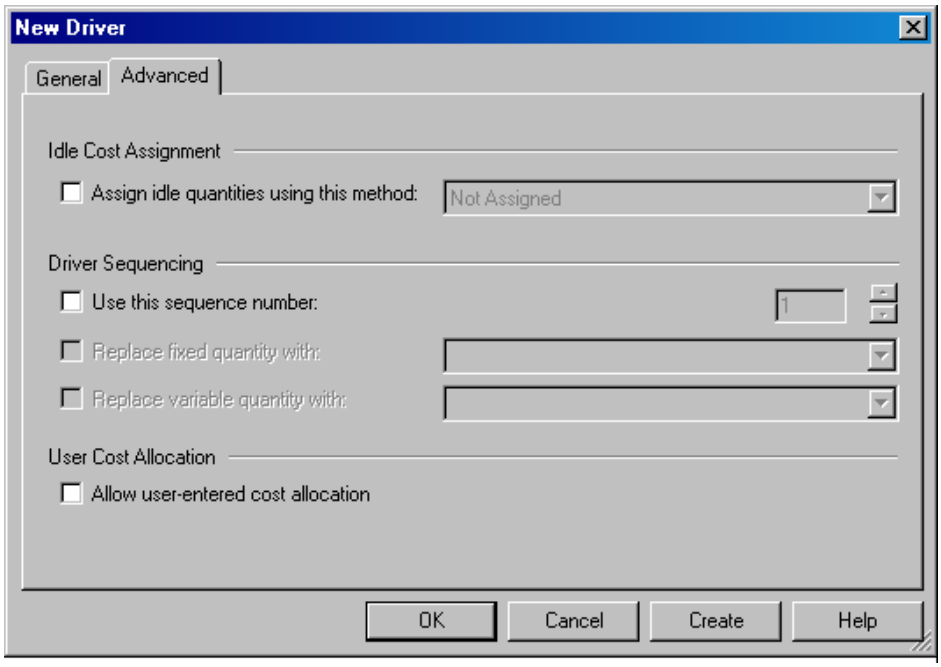
   d. To remove the last item that you inserted, click **Undo**.

   e. To delete the entire formula, click **Clear**.

   f. Click **Test**.

      If no errors are displayed, the syntax of the formula is valid.

9. Click the **Advanced** tab.



10. To specify how idle quantities are handled, select or clear the **Assign idle quantities using this method** option. If you select this option, then select a method from the drop-down list.
11. Specify driver sequencing:
    a. Either select or clear the **Use this sequence number** option. If you select this option, then specify the sequence number in the box provided.
    b. To specify how fixed quantities are handled, select or clear the **Replace fixed quantity with** option. If you select this option, then select a numeric property from the drop-down list on the right.
    c. To specify how variable quantities are handled, select or clear the **Replace variable quantity with** option. If you select this option, then select a numeric property from the drop-down list on the right.
12. To specify how user-entered cost allocation is handled, select or clear the **Allow user-entered cost allocation** option.

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

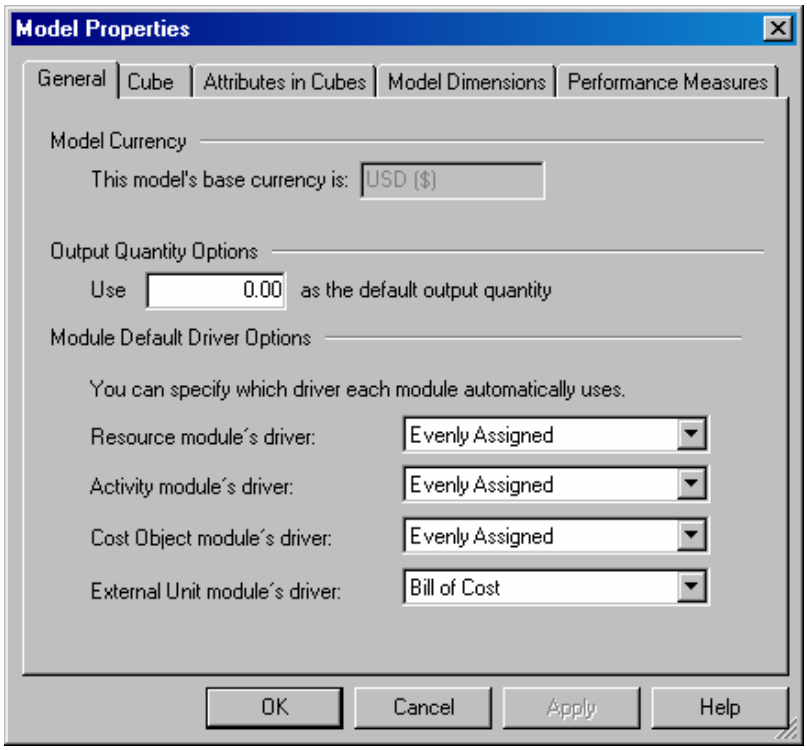
If you are prevented from entering these values, verify that the following are correct:

- The source account's driver is a weighted driver.
- The weighted driver allows fixed driver quantities, variable driver quantities, or both.

Specify the Default Driver

1. Select **Model** ⇨ **Properties**.

   The Model Properties dialog box appears.



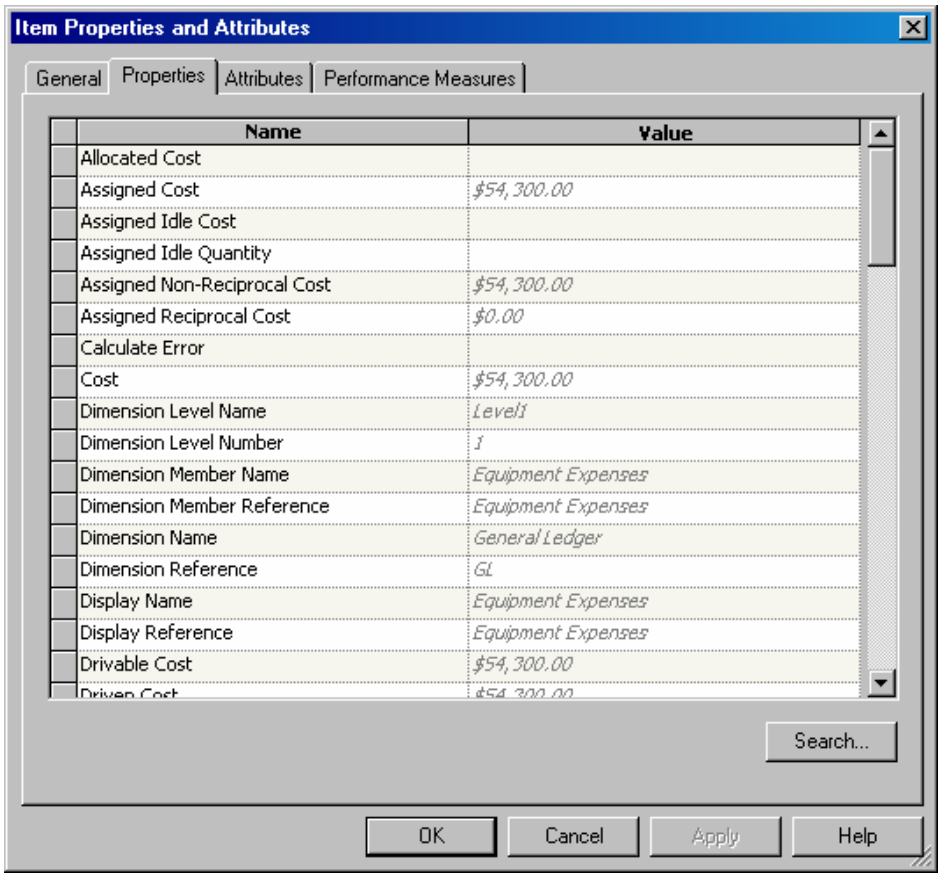
2. Click the **General** tab.
3. In the **Module Default Driver Options** section, select a default driver for each module.

   The drivers that are available for the External Units module include all drivers of type Bill of Cost.

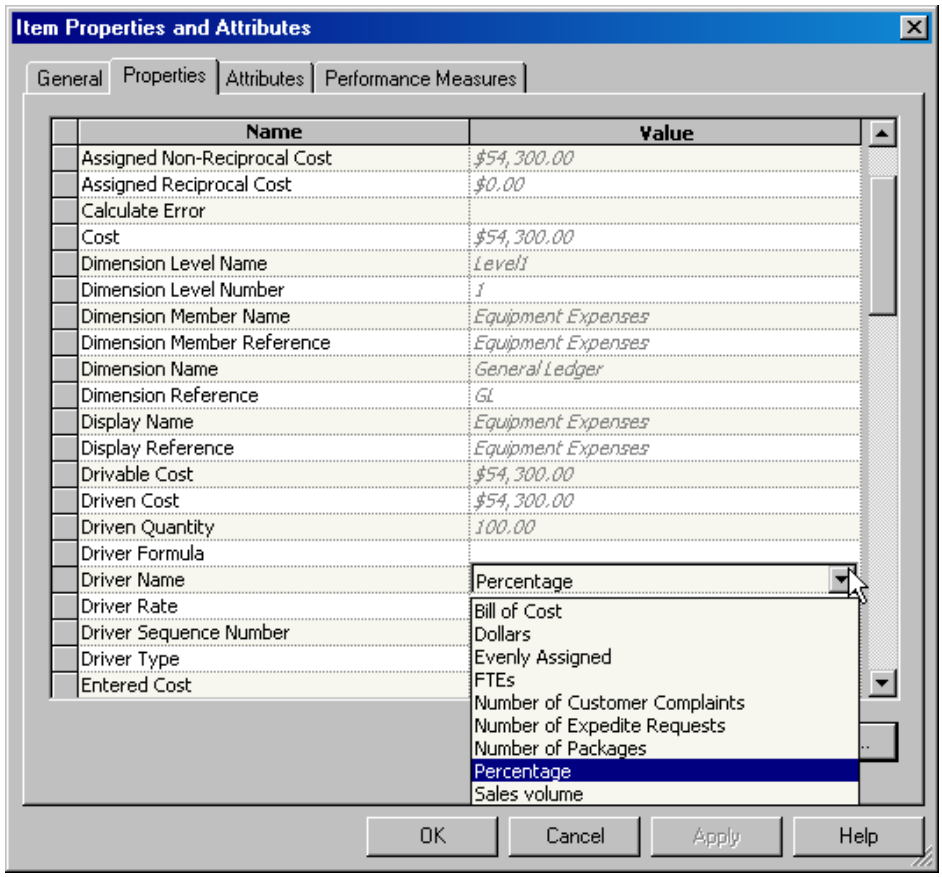
Associate a Driver Using an Account's Item Properties

1. In the Resource module, Activity module, or Cost Object module, select an account.
2. Select **Edit** ⇨ **Item Properties**.

   The Item Properties dialog box appears.

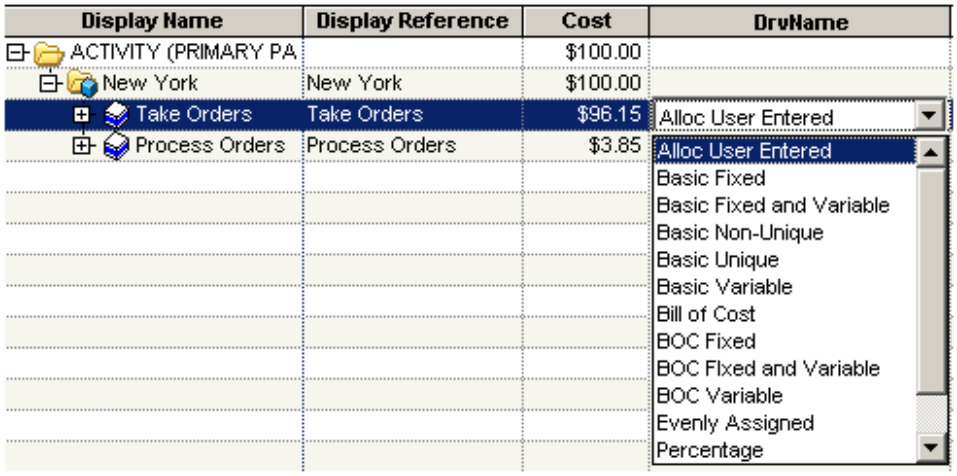


3. Click the **Properties** tab.
4. From the **Properties** list, locate the **Driver Name** property.
5. Click in the **Value** column and select a driver.



Associate a Driver Using a Module's Grid

1. In the Resource module, Activity module, or Cost Object module, add a column to display the **Driver Name** property.
2. Click in the **DrvName** column and select a driver.



Chapter 25

Working with Assignments

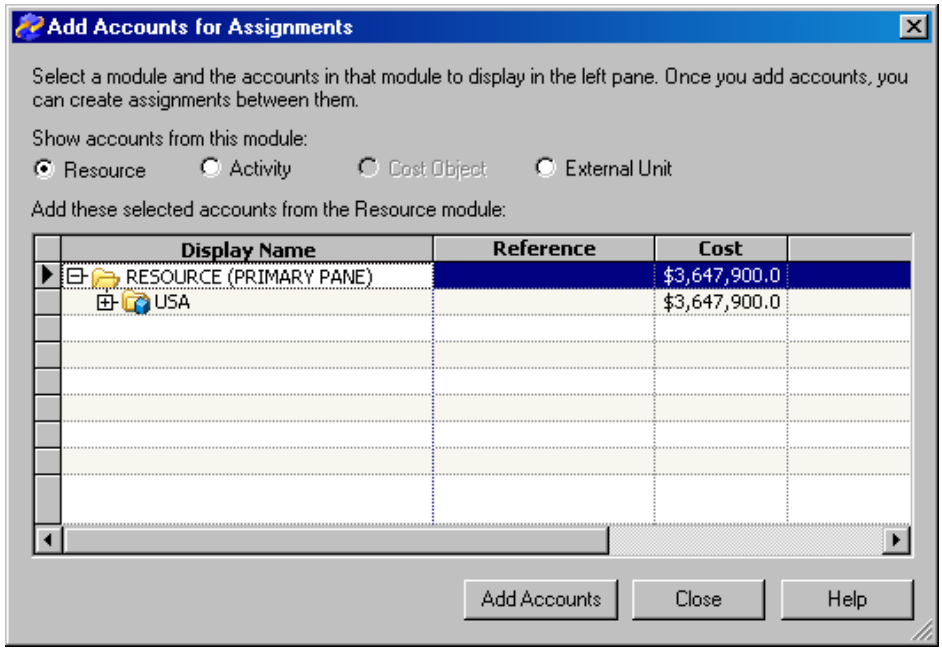
Introduction

This chapter contains task-based information about working with assignments. For more conceptual information, see Chapter 10, “Assignments,” on page 75.

Create an Assignment

1. Decide which assignments panes to show by selecting either:
   - **Model ⇨ Assignments ⇨ Show Left Assignments Pane**
   - **Model ⇨ Assignments ⇨ Show Right Assignments Pane**
   - **Model ⇨ Assignments ⇨ Show Left and Right Assignments Pane**
2. Select either:
   - **Model ⇨ Assignments ⇨ Add Accounts in Left Pane**
   - **Model ⇨ Assignments ⇨ Add Accounts in Right Pane**

   The Add Accounts for Assignments dialog box appears.



3. From **Show accounts from this module**, select an option.
4. From the list of accounts, select an account.

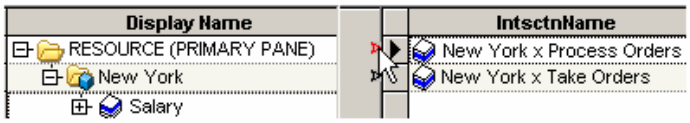
   You can select multiple accounts by using standard Microsoft Windows selection techniques.

   If you select a roll-up account, all accounts within that roll-up account are added.

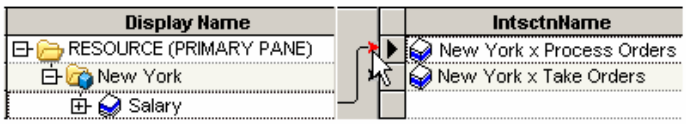
   Note: You cannot select the module roll-up and add all accounts within it.
5. Click **Add Accounts**.

   The accounts are added to the grid.

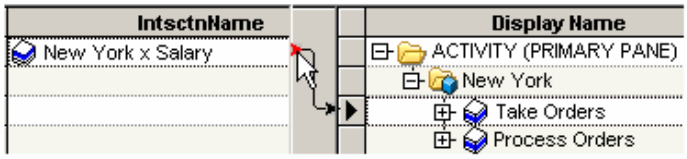
   Note: You can leave this dialog box open to add more accounts individually.
6. Click **Close**.
7. Select the destination account.
8. To assign the cost to an account in the right assignments pane, click the arrowhead to the left of the account.



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.



TIP To quickly create assignments to many accounts, select **Model** ⇨ **Assignments** ⇨ **Assign All Left**, **Assign All Right**, or **Assign All Left and Right**.

Show Only the Source Accounts for 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 the accounts that contribute costs to the selected account.

Show Only the Destination Accounts for 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 the accounts that receive costs from the selected account.

Show the Source Accounts and the Destination Accounts for an Account

1. Select **Model** ⇨ **Assignments** ⇨ **Show Left and Right Assignments Panes**.

   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 contribute costs to the selected account, as well as to the accounts that receive costs from the selected account.

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.

Chapter 26

Working with Attributes

Introduction

This chapter contains task-based information about working with attributes. For more conceptual information, see Chapter 11, “Attributes,” on page 81.

Create an Attribute Folder

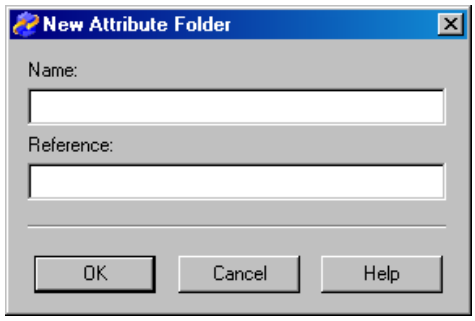
1. Open a model in Model mode, and select **Model** ⇨ **Attributes Page**.

   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. For information, see the Help.

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. For information, see the Help.

Create an Attribute

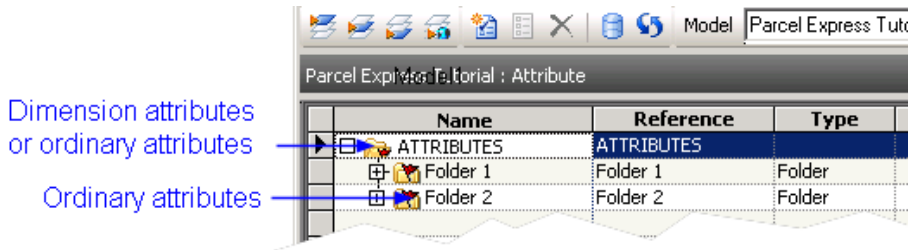
1. Open a model in Model mode, and select **Model** ⇨ **Attributes Page**.

   The Attributes page appears.

2. Select the folder in which the attribute is to go.

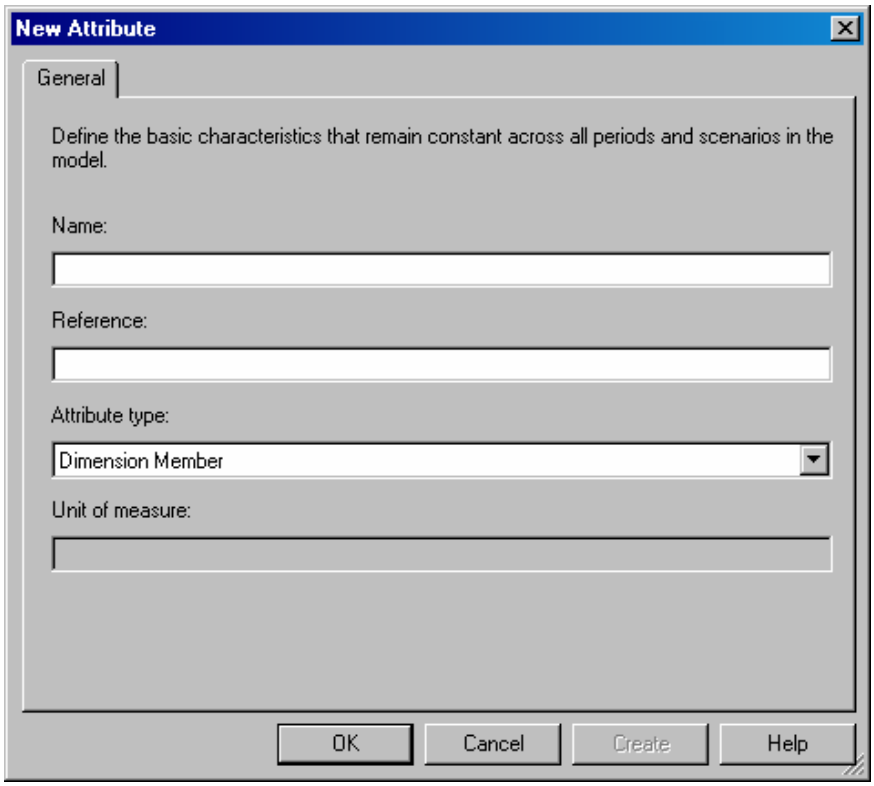
   A dimension attribute can go in the **ATTRIBUTES** folder or in an existing dimension attribute.

   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. For information, see the Help.

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. For information, see the Help.

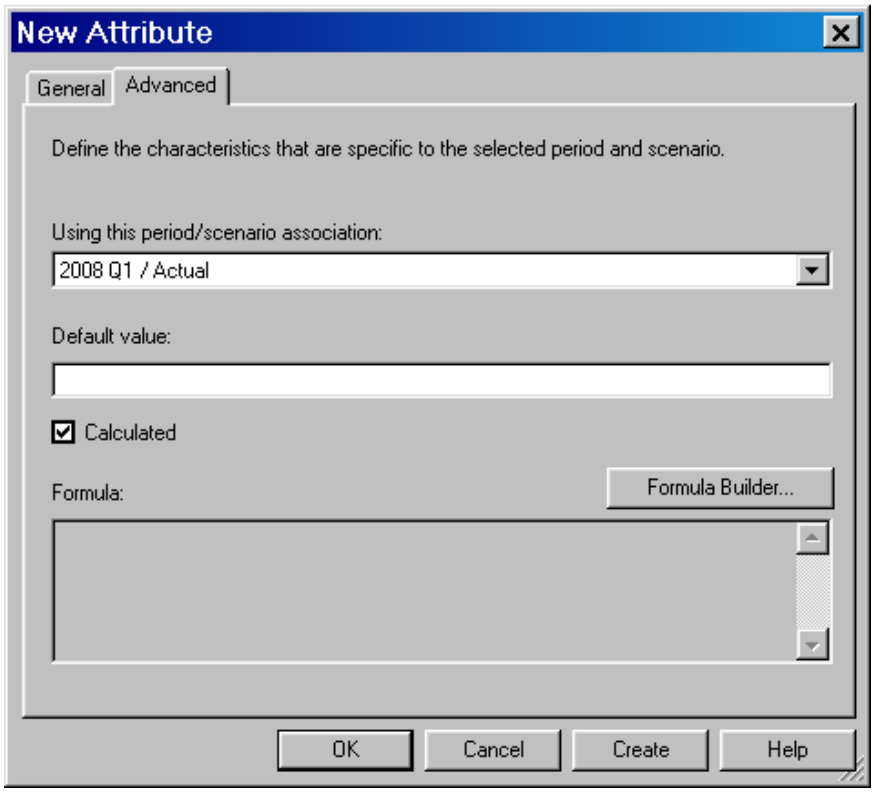
6. Select the **Attribute type**.

   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:

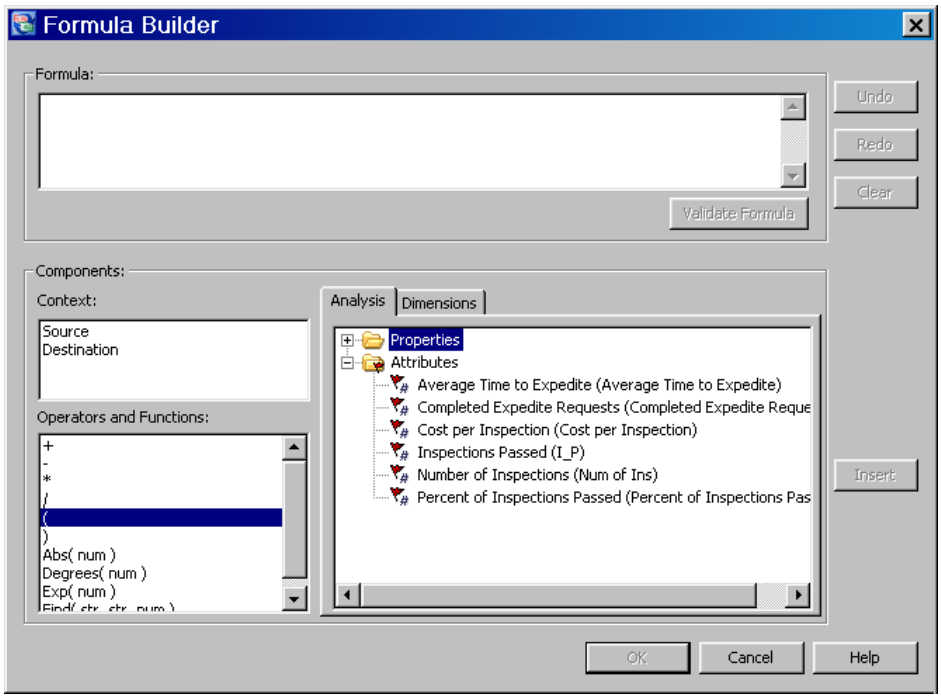
7. Type the **Unit of measure**.

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. For information about formulas, see Chapter 12, "Using Formulas," on page 87.

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**.

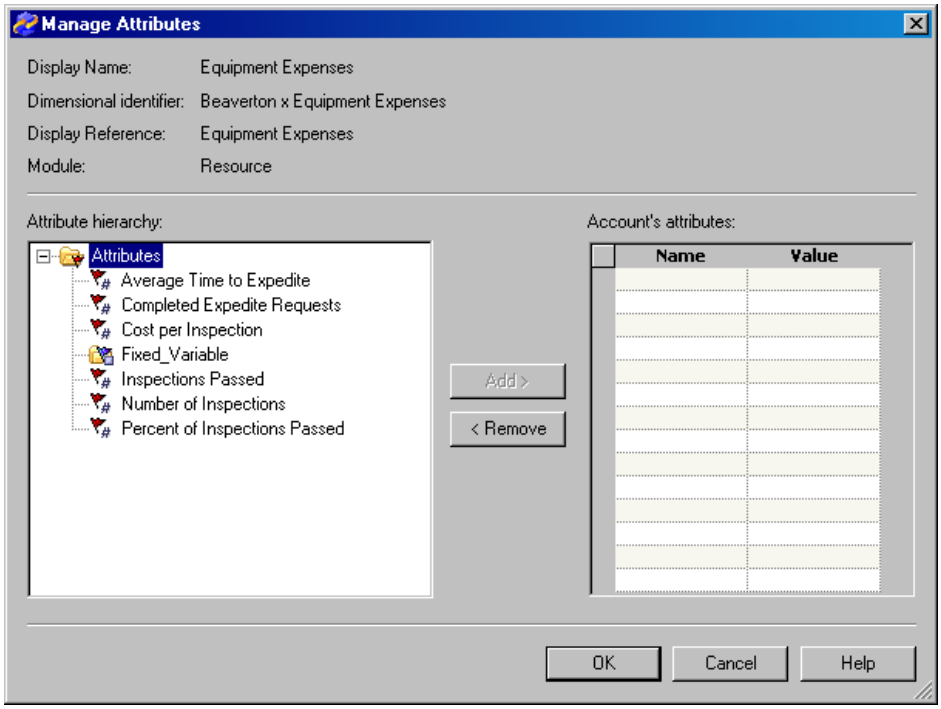
h. Click **Test**.

If no errors are displayed, the syntax of the formula is valid.

Add an Attribute to an Account

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.

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.

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.

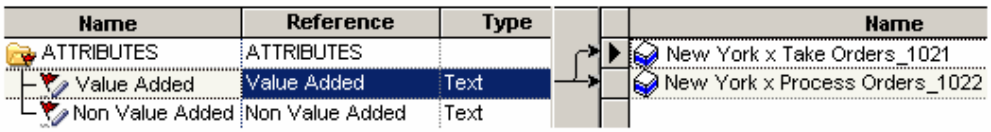
1. Select **Model** ⇨ **Attributes Page**.

   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.

Chapter 27

Working with Currencies

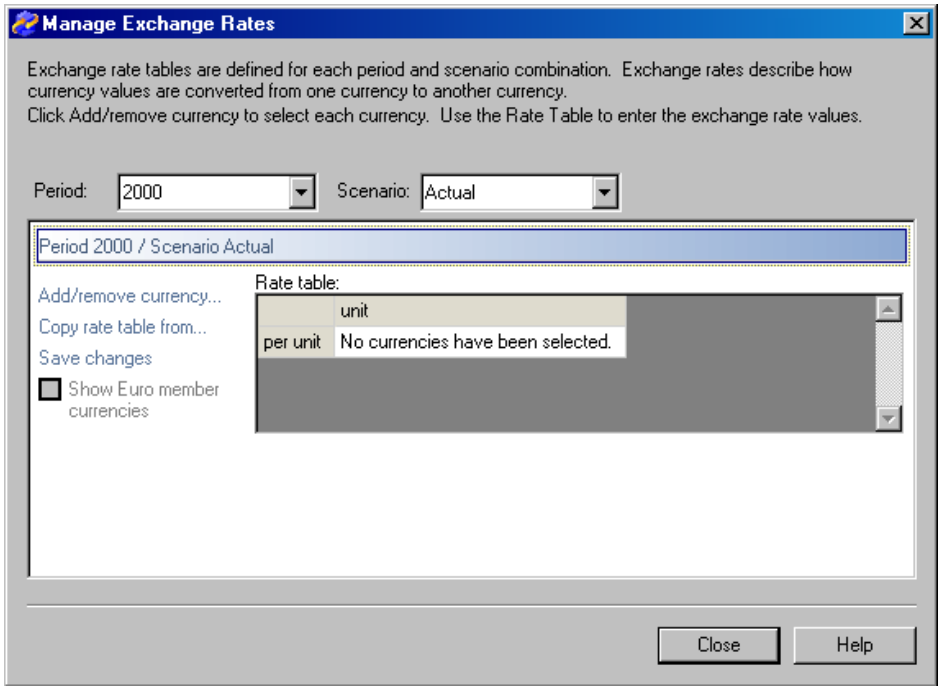
Introduction

This chapter contains task-based information about working with currencies and exchange rates. For more conceptual information, see Chapter 16, “Currencies and Exchange Rates,” on page 105.For more conceptual information, see

Add a Currency

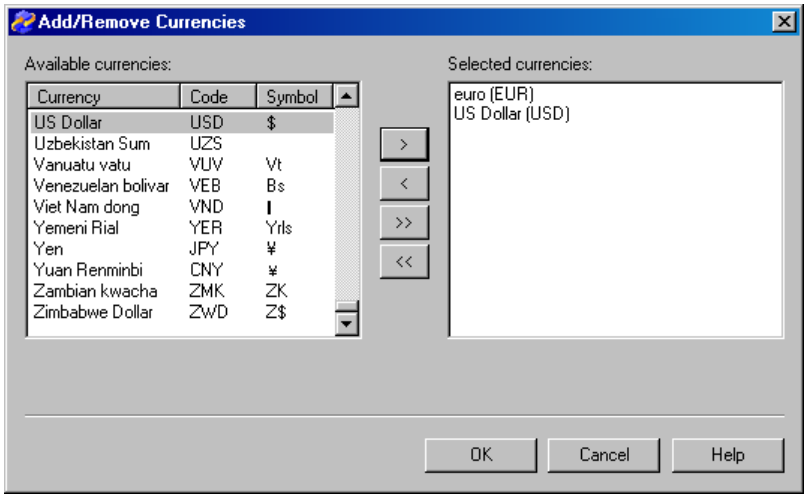
1. Select **Tools** ⇨ **Manage Exchange Rates**.

   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 >>.

Remove 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 **Selected currencies** list, select one or more currencies.

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

4. Click <.

   The currency is removed from the **Available currencies** list.

   Note: To remove all currencies, click <<.

Set Up an Exchange Rates Table

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

   The Manage Exchange Rates dialog box appears.

2. Select a **Period**.

3. Select a **Scenario**.

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

   The Add/Remove Currencies dialog box appears.

5. In the **Rate table**, click on the intersection between two currencies, and type an exchange rate.

6. Repeat the previous step for each exchange rate.

Add Euro Exchange Rates

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

   The Manage Exchange Rates dialog box appears.

2. Click the **Save changes** link.

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

3. Add the currency named **euro**:

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

   The Add/Remove Currencies dialog box appears.

Select the currency named **euro**.

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

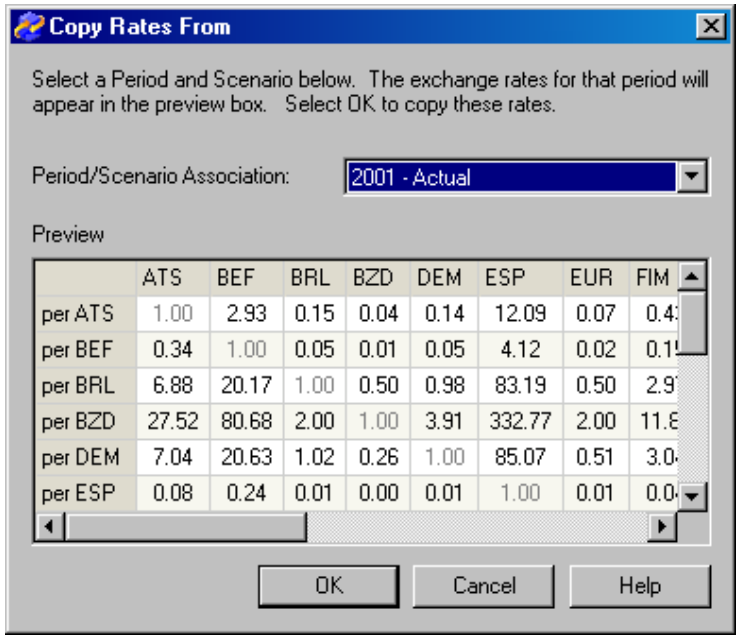
Copy an Exchange Rates Table

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.

Chapter 28

Working with Column Layouts

Introduction

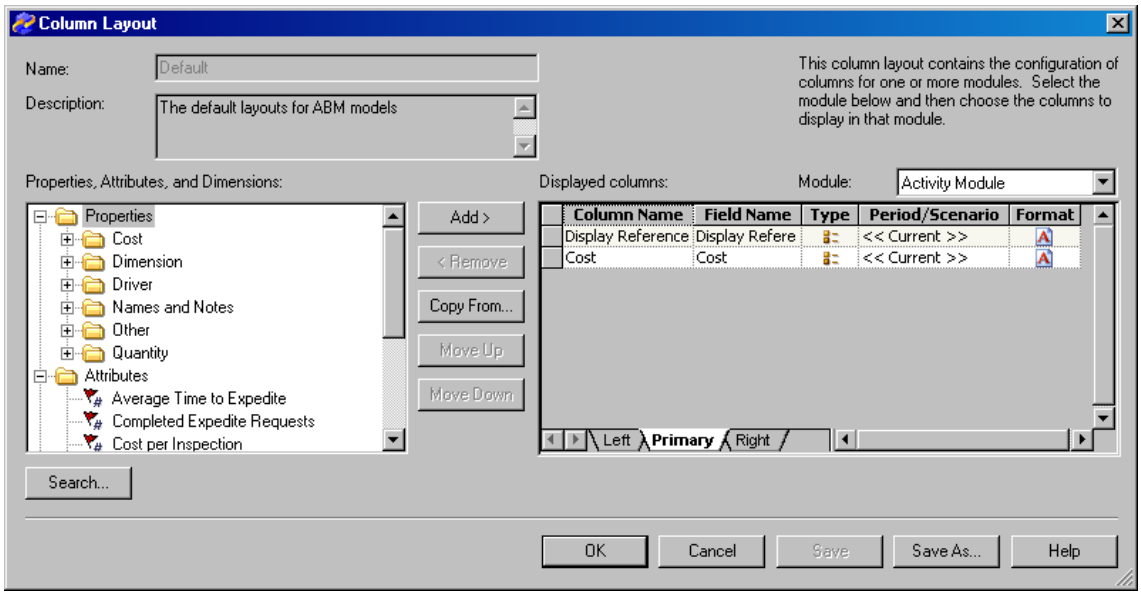
This chapter contains task-based information about working with columns and column layouts. For more conceptual information, see Chapter 15, “Column Layouts,” on page 99.

Add a Column

1. On a module page, select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

   TIP Alternatively, you can double-click a column heading.

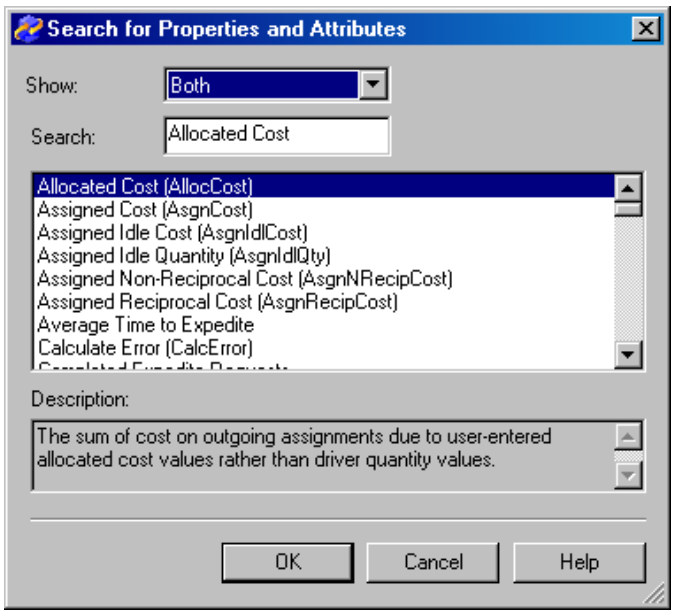
   The Column Layout dialog box appears.



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**) to which you want to add the column.
4. From the list of **Properties, Attributes, and Dimensions**, select an item.
5. 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.

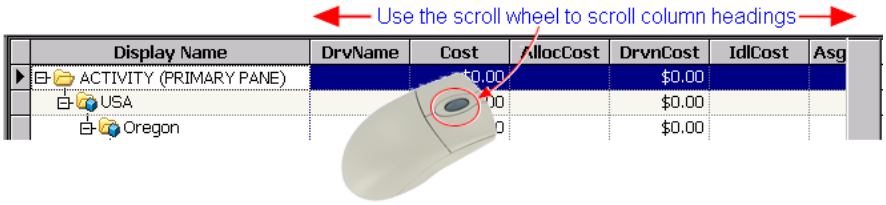
Note: Some items, such as attributes, might not have a description.

e. Click **OK**.

6. Click **Add >**.
7. 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.
8. To select a different period/scenario association, click in the **Period/Scenario** column, and select a period/scenario association.

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.



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 on the following pages: the Resource module page, the Activity module page, the Cost Object module page, and the External Units module page.

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**.

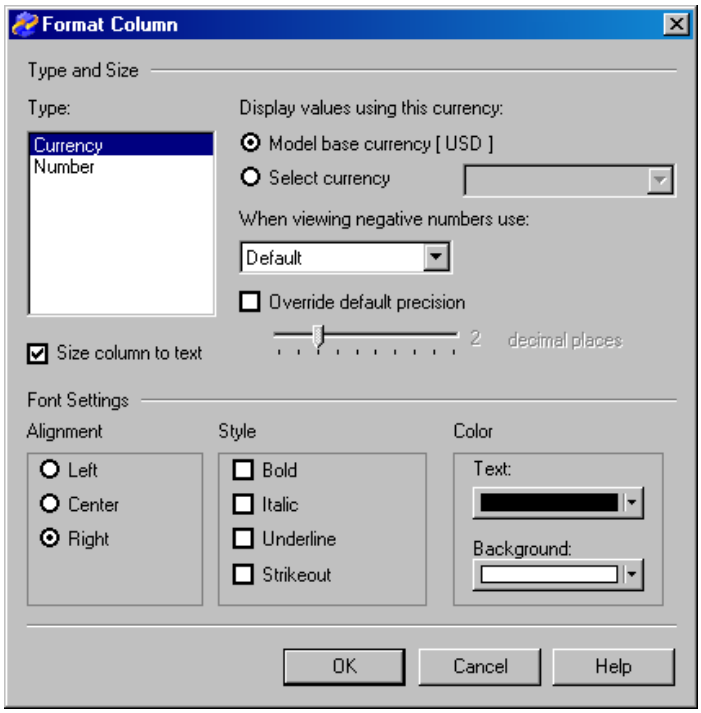
Format a Column

1. On the Resource module page, the Activity module page, the Cost Object module page, or the External Units module page, 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**.

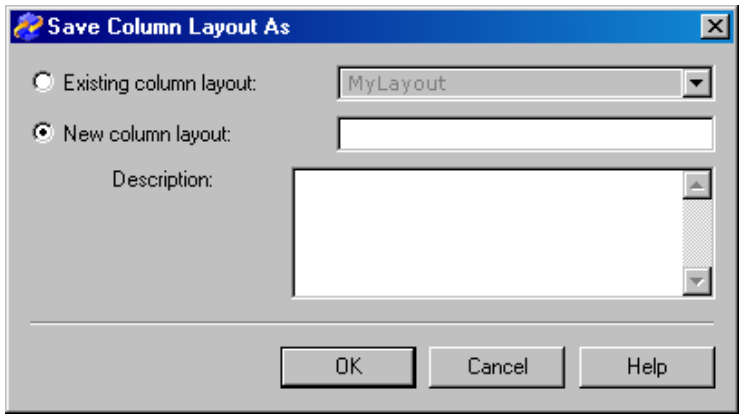
7. To quickly set the width of a column so that the column is wide enough to display all data, select the **Size column to text** option.

8. In the **Font settings**section, select or clear options for **Alignment**, **Style**, and **Color**.

Save a Column Layout

1. To save a column layout on a module page using its current name, select **Model** ⇨ **Column Layout** ⇨ **Save**.
2. To save a column layout using another name, select **Model** ⇨ **Column Layout** ⇨ **Save As**.

   The Save Column Layout As dialog box appears.



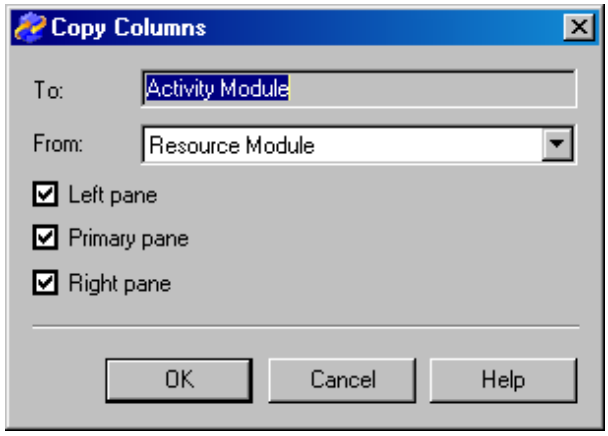
3. To replace an existing column layout, do the following:
   a. Select the **Existing column layout** option.
   b. From the drop-down list on the right, select a column layout.
4. To create a new column layout, do the following:
   a. Select the **New column layout** option.
   b. In the box on the right, type the name of the new column layout.
   c. Type the **Description**.

Copy a Column Layout to another Module

1. On the page to which you want to copy a column layout (the Resource module page, the Activity module page, the Cost Object module page, or the External Units module page), select **Model** ⇨ **Column Layout** ⇨ **Edit Columns**.

   The Column Layout dialog box appears.
2. Click **Copy From**.

   The Copy Columns dialog box appears.

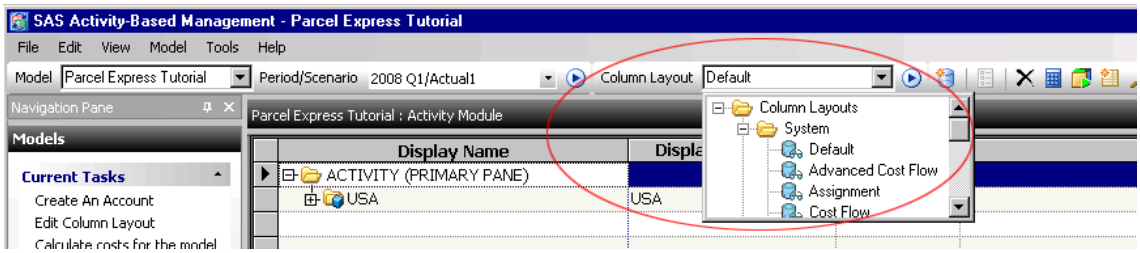


3. Select the module to copy **From**.
4. Select the assignments panes to copy from.

Apply a Saved Column Layout

Note: You can apply a saved column layout to only module pages

1. From the **Column Layout** drop-down list, select a column layout.



2. Click  .

Chapter 29

Working with Querying Contributions

Introduction

The Contributions page provides the fastest and easiest method to view cost flows throughout a model. To query a model on the Contributions page, the model must have been calculated. However, you do not have to generate a cube. Queries are fast because there is no cube to navigate.

To open the Contributions page, do one of the following:

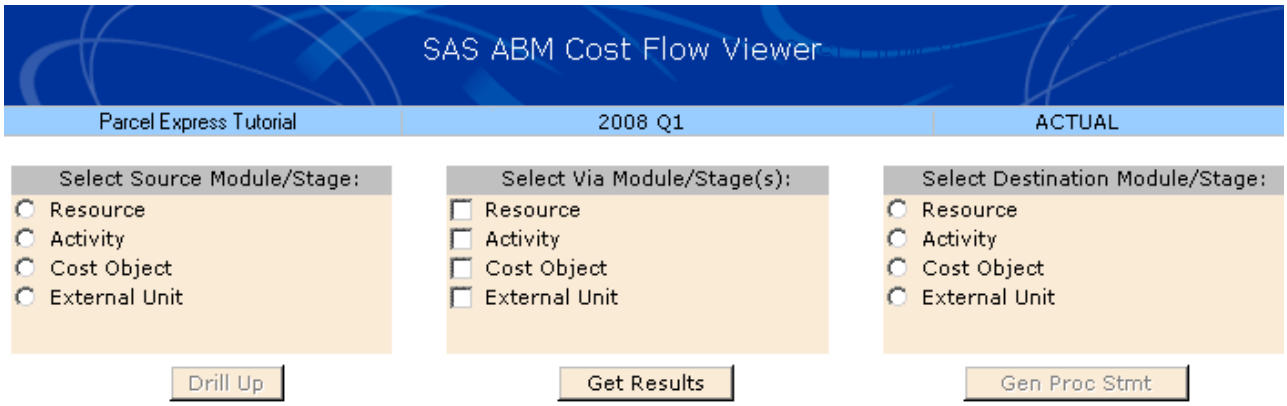
- Click the Contributions tab in the Navigations Pane.

  Note: The Contributions icon appears only if the Contributions Server has been installed.

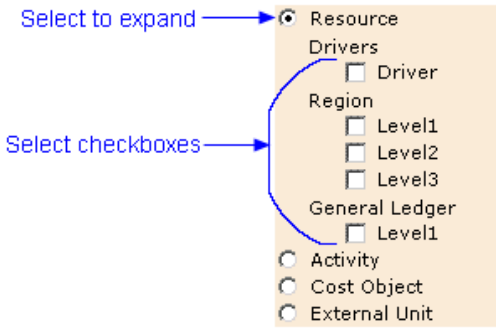
- Or, select **File** ⇨ **New** ⇨ **Contribution Query**

Note: Apache Tomcat must be running on the server.

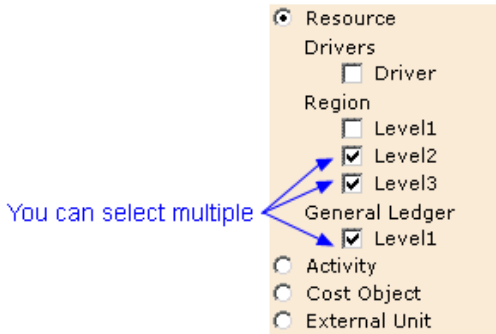


Some Key Things to Know

- Select a module (Resource, Activity, Cost Object, or External Unit) to expand it. Expanding a module does not select any diminsions in it. Select checkboxes to select dimensions in the module.

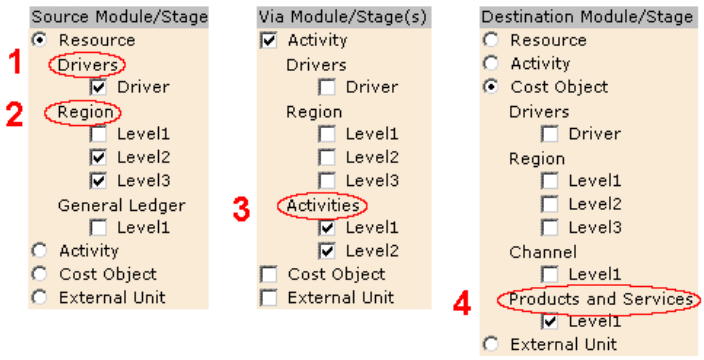


- You can choose only one module for a query, but you can select multiple dimensions in a module.



Note: Only one module is active at a time. Although checkboxes remain selected when a module is collapsed, if a module is collapsed, then it is not active.

- You can select as many as 10 dimensions. You can select any number of levels within a dimension (For the Contributions page, Drivers is considered a dimension). The following picture illustrates how the system counts dimensions. In this picture four dimensions are selected and six dimension levels:



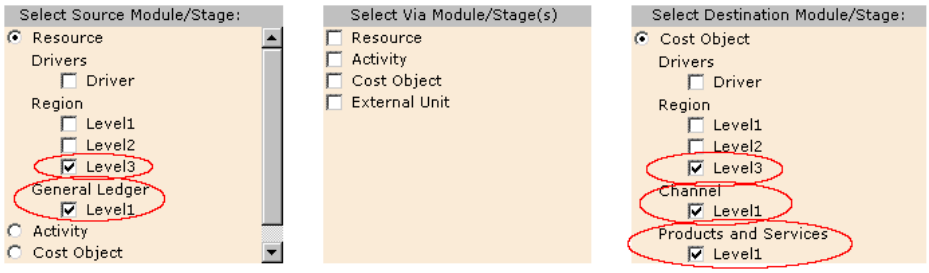
You must select at least one source level and at least one destination level.

- You do not have to select anything in **Via Module/Stage(s)**. If you do not select anything, then all paths from source to destination are included in calculations. Select something in **Via Module/Stage(s)** to restrict the paths that are included.

Query Contributions from Resource to Cost Object

The following example uses data from the Parcel Express Tutorial model to illustrate a typical query using the Contributions page. The particular data is not important. The query shows, by region, the contributions of level 1 General Ledger items (wages, operating expenses, and equipment expenses) to each of the three products by region and channel.

1. For Source Module, the following levels are selected under the **Resource** module:
   - **Level3** (Beaverton, Eugene) under **Region**
   - **Level1** (Wages, Operating Expenses, Equipment Expenses) under **General Ledger**
2. For Destination Module, the following levels are selected under the **Cost Object** module:
   - **Level3** (Beaverton, Eugene) under **Region**
   - **Level1** (Drop Box, Walk In, Commercial Channel) under **Channel**
   - **Level1** (2nd Day Guaranteed, Overnight Express, Standard Ground) under **Products and Services**



3. After clicking **Get Results**, the resulting table shows the contributions of general ledger (wages, operating expenses, and equipment expenses) to each of the three products by region and channel. (The following picture shows the table split in pieces, with one piece on top of the other, to fit better on the page.)

		3:Beaverton 1:None 1:2nd Day Guaran	3:Beaverton 1:None 1:Overnight Expres	3:Beaverton 1:None 1:Standard Grounc	3:Beaverton 1:Drop Box 1:None	3:Beaverton 1:Drop Box 1:2nd Day Guarant
3:Beaverton	1:Wages	334096.90	450652.53	733953.01	14387.71	54837.86
3:Beaverton	1:Operating Expen	48498.77	77840.48	105372.13	754.63	7614.15
3:Beaverton	1:Equipment Exper	10740.49	17585.06	23259.46	325.80	1727.42
3:Eugene	1:Wages	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Operating Expen	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Equipment Exper	0.00	0.00	0.00	0.00	0.00

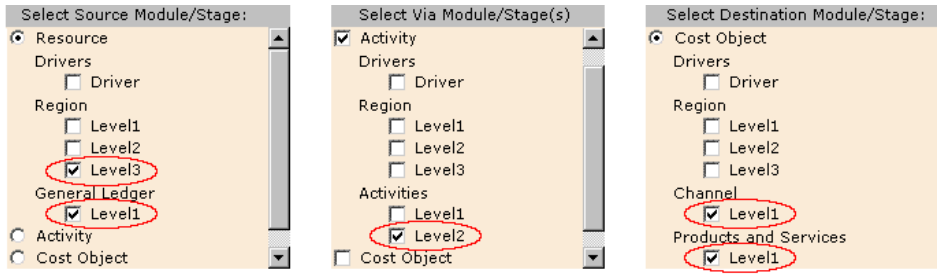
		3:Beaverton 1:Drop Box 1:Overnight Expre	3:Beaverton 1:Drop Box 1:Standard Grounc	3:Beaverton 1:Walk In 1:None	3:Beaverton 1:Walk In 1:2nd Day Guaran	3:Beaverton 1:Walk In 1:Overnight Expres
3:Beaverton	1:Wages	28432.91	132294.68	81530.34	186752.70	271523.56
3:Beaverton	1:Operating Expen	4711.49	18173.92	4276.26	25503.97	44293.78
3:Beaverton	1:Equipment Exper	1082.30	4111.16	1846.20	5839.10	10240.37
3:Eugene	1:Wages	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Operating Expen	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Equipment Exper	0.00	0.00	0.00	0.00	0.00

		3:Beaverton 1:Walk In 1:Standard Grounc	3:Beaverton 1:Commercial Pick 1:None	3:Beaverton 1:Commercial Pick 1:2nd Day Guaran	3:Beaverton 1:Commercial Pick 1:Overnight Expres	3:Beaverton 1:Commercial Pick 1:Standard Grounc
3:Beaverton	1:Wages	459698.70	23979.51	119556.03	181816.30	203687.27
3:Beaverton	1:Operating Expen	62121.46	1257.72	16799.40	30467.46	28314.35
3:Beaverton	1:Equipment Exper	14183.29	543.00	3786.49	6967.08	6362.79
3:Eugene	1:Wages	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Operating Expen	0.00	0.00	0.00	0.00	0.00
3:Eugene	1:Equipment Exper	0.00	0.00	0.00	0.00	0.00

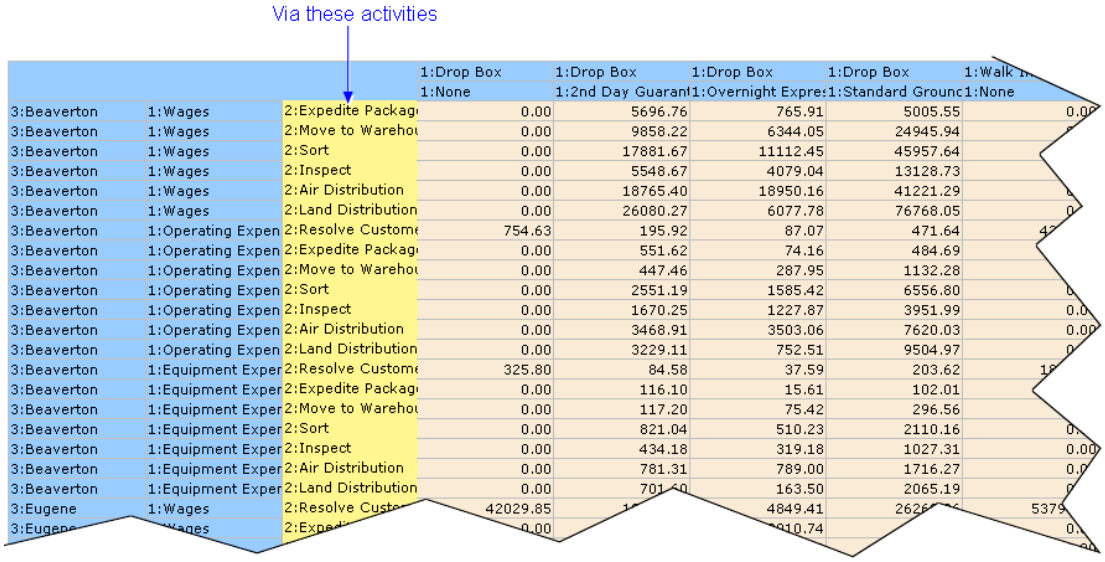
Query Contributions Via the Activity Module

The following example shows the selection of cost flows from a source to a target via an intermediary module or stage. The query shows the general ledger contributions by activity to each of the three products by channel.

1. For Source Module, the following levels are selected under the **Resource** module:
   - **Level3** (Beaverton, Eugene) under **Region**
   - **Level1** (Wages, Operating Expenses, and Equipment Expenses) under **General Ledger**
2. For Via Module, the following level is selected under the **Activity** module:
   - **Level2** (Expedite Package Shipments, Move to Warehouse, and so on) under **Activities**
3. For Destination Module, the following levels are selected under the **Cost Object** module:
   - **Level1** (Drop Box, Walk In, Commercial Channel) under **Channel**
   - **Level1** (2nd Day Guaranteed, Overnight Express, Standard Ground) under **Products and Services**



4. After clicking**Get Results**, the resulting table shows the general ledger contributions by activity (activities in the via column are shown in yellow) to each of the three products by channel. (Because it is large, only part of the table is shown here.)

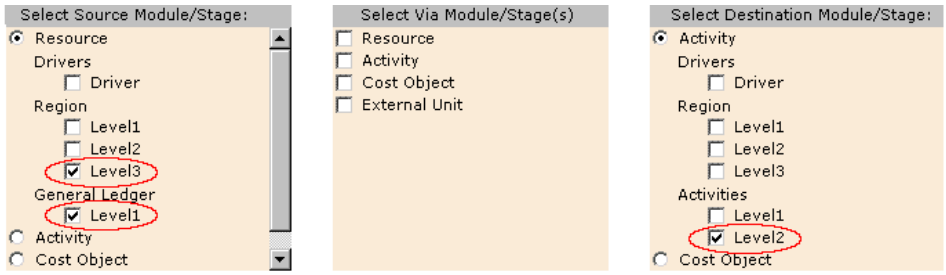


			1:Drop Box 1:None	1:Drop Box 1:2nd Day Guaran	1:Drop Box 1:Overnight Expre	1:Drop Box 1:Standard Grounc	1:Walk 1:None
3:Beaverton	1:Wages	2:Expedite Packag	0.00	5696.76	765.91	5005.55	0.00
3:Beaverton	1:Wages	2:Move to Wareho	0.00	9858.22	6344.05	24945.94	
3:Beaverton	1:Wages	2:Sort	0.00	17881.67	11112.45	45957.64	
3:Beaverton	1:Wages	2:Inspect	0.00	5548.67	4079.04	13128.73	
3:Beaverton	1:Wages	2:Air Distribution	0.00	18765.40	18950.16	41221.29	
3:Beaverton	1:Wages	2:Land Distribution	0.00	26080.27	6077.78	76768.05	
3:Beaverton	1:Operating Expen	2:Resolve Custome	754.63	195.92	87.07	471.64	
3:Beaverton	1:Operating Expen	2:Expedite Packag	0.00	551.62	74.16	484.69	
3:Beaverton	1:Operating Expen	2:Move to Wareho	0.00	447.46	287.95	1132.28	
3:Beaverton	1:Operating Expen	2:Sort	0.00	2551.19	1585.42	6556.80	
3:Beaverton	1:Operating Expen	2:Inspect	0.00	1670.25	1227.87	3951.99	0.00
3:Beaverton	1:Operating Expen	2:Air Distribution	0.00	3468.91	3503.06	7620.03	0.00
3:Beaverton	1:Operating Expen	2:Land Distribution	0.00	3229.11	752.51	9504.97	
3:Beaverton	1:Equipment Exper	2:Resolve Custome	325.80	84.58	37.59	203.62	
3:Beaverton	1:Equipment Exper	2:Expedite Packag	0.00	116.10	15.61	102.01	
3:Beaverton	1:Equipment Exper	2:Move to Wareho	0.00	117.20	75.42	296.56	
3:Beaverton	1:Equipment Exper	2:Sort	0.00	821.04	510.23	2110.16	
3:Beaverton	1:Equipment Exper	2:Inspect	0.00	434.18	319.18	1027.31	0.00
3:Beaverton	1:Equipment Exper	2:Air Distribution	0.00	781.31	789.00	1716.27	
3:Beaverton	1:Equipment Exper	2:Land Distribution	0.00	701.60	163.50	2065.19	
3:Eugene	1:Wages	2:Resolve Custo	42029.85		4849.41	2626	5379
3:Eugene	1:Wages	2:Expedi	0.00		910.74		

Query Contributions from Resource to Activity Module

The destination module does not need to be a cost object. The query shows the contributions, by region, of general ledger items to activities

1. For Source Module, the following levels are selected under the **Resource** module:
   - **Level3** (Beaverton, Eugene) under **Region**
   - **Level1** (Wages, Operating Expenses, and Equipment Expenses) under **General Ledger**
2. For Destination Module, the following level is selected under the **Activity** module:
   - **Level2** (Expedite Package Shipments, Move to Warehouse, and so on) under **Activities**



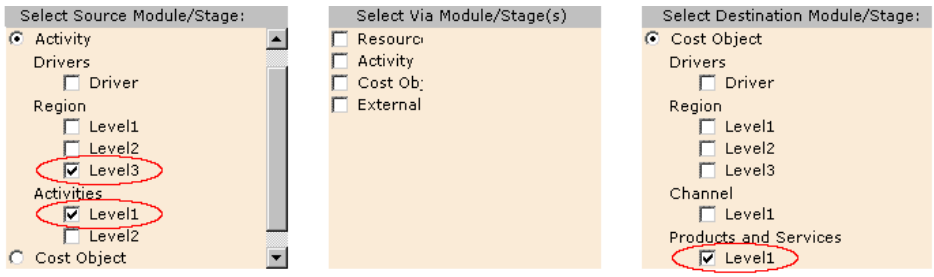
3. After clicking**Get Results**, the resulting table shows the contributions, by region, of general ledger items to activities.

		2:Resolve Custom	2:Expedite Packag	2:Move to Wareho	2:Sort	2:Inspect	2:Air Distribution	2:Land Distribution
3:Beaverton	1:Wages	119897.56	79931.71	319726.83	577517.64	183084.68	687030.62	729769.95
3:Beaverton	1:Operating Expen	6288.62	7739.84	14512.20	82394.70	55111.93	127002.26	90355.85
3:Beaverton	1:Equipment Exper	2715.00	1629.00	3801.00	26516.86	14326.19	28604.80	19632.06
3:Eugene	1:Wages	112640.00	56320.00	281600.00	546512.59	150186.67	0.00	1222352.59
3:Eugene	1:Operating Expen	7452.50	8710.71	66782.14	152727.86	65330.36	0.00	247577.86
3:Eugene	1:Equipment Exper	1900.00	760.00	3040.00	19039.41	9905.33	0.00	34239.41

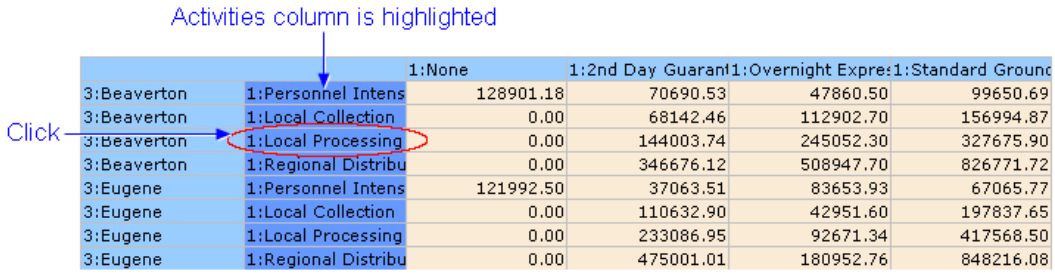
Drilling Down to a Lower Level

The table that is generated by a query is not static. If you display a dimension level that has additional levels under it, you can click the generated table to display the next level of detail.

1. For Source Module, the following levels are selected under the **Activity** module:
   - **Level3** (Beaverton, Eugene) under **Region**
   - **Level1** (Personnel Intensive Activities, Local Collection, Local Processing, and Regional Distribution) under **Activities**
2. For Destination Module, the following level is selected under the **Cost Object** module:
   - **Level1** (2nd Day Guaranteed, Overnight Express, and Standard Ground) under **Products and Services**



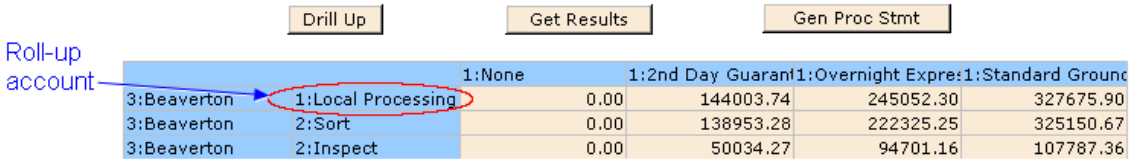
3. After clicking **Get Results**, the resulting table shows the contributions of activities, by region, to products and services. Note that the Activities column is highlighted.



		1:None	1:2nd Day Guaran	1:Overnight Expre	1:Standard Ground
3:Beaverton	1:Personnel Intens	128901.18	70690.53	47860.50	99650.69
3:Beaverton	1:Local Collection	0.00	68142.46	112902.70	156994.87
3:Beaverton	1:Local Processing	0.00	144003.74	245052.30	327675.90
3:Beaverton	1:Regional Distribu	0.00	346676.12	508947.70	826771.72
3:Eugene	1:Personnel Intens	121992.50	37063.51	83653.93	67065.77
3:Eugene	1:Local Collection	0.00	110632.90	42951.60	197837.65
3:Eugene	1:Local Processing	0.00	233086.95	92671.34	417568.50
3:Eugene	1:Regional Distribu	0.00	475001.01	180952.76	848216.08

4. Click **1:Local Processing** in the Activities column to go to the next level of detail.

   Initially, activities were displayed at Level1. By clicking, you drill down to Level2 activities. Notice that the entire table is replaced by the Level2 activities. Also notice that the roll-up account **1:Local Processing** is displayed.



		1:None	1:2nd Day Guaran	1:Overnight Expre	1:Standard Ground
3:Beaverton	1:Local Processing	0.00	144003.74	245052.30	327675.90
3:Beaverton	2:Sort	0.00	138953.28	222325.25	325150.67
3:Beaverton	2:Inspect	0.00	50034.27	94701.16	107787.36

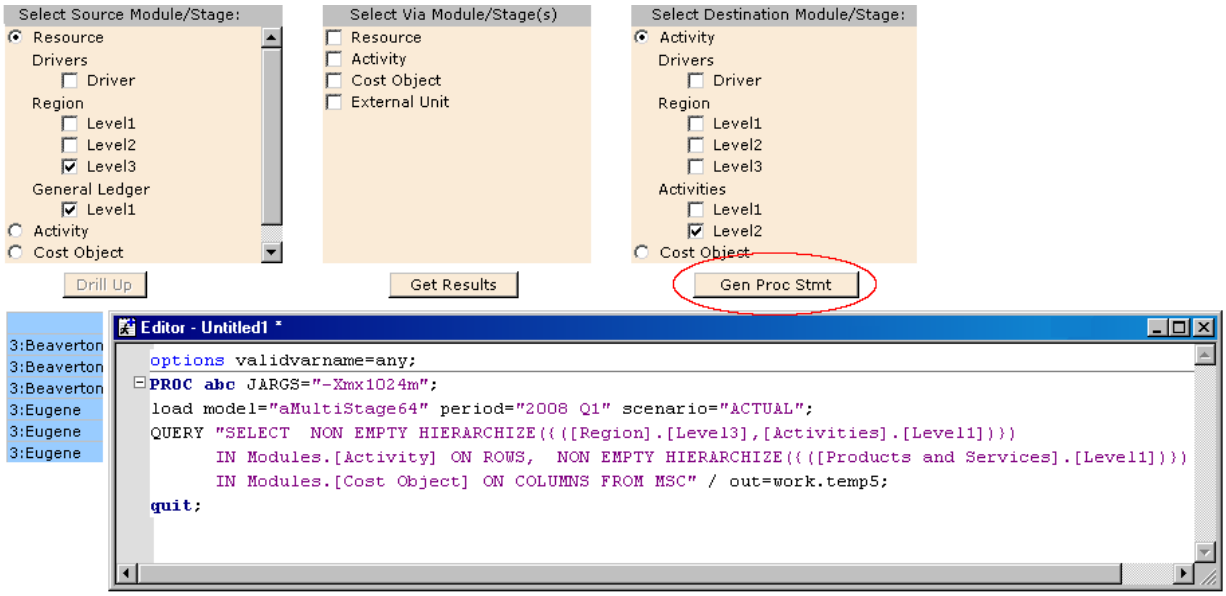
5. Click **Drill Up** to return to the previous table.

		1:None	1:2nd Day Guaran	1:Overnight Expre	1:Standard Ground
3:Beaverton	1:Personnel Intens	128901.18	70690.53	47860.50	99650.69
3:Beaverton	1:Local Collection	0.00	68142.46	112902.70	156994.87
3:Beaverton	1:Local Processing	0.00	144003.74	245052.30	327675.90
3:Beaverton	1:Regional Distribu	0.00	346676.12	508947.70	826771.72
3:Eugene	1:Personnel Intens	121992.50	37063.51	83653.93	67065.77
3:Eugene	1:Local Collection	0.00	110632.90	42951.60	197837.65
3:Eugene	1:Local Processing	0.00	233086.95	92671.34	417568.50
3:Eugene	1:Regional Distribu	0.00	475001.01	180952.76	848216.08

Using theABC Procedure

SAS Activity-Based Management has externalized, in the form of the ABC procedure, the processing that it uses internally to calculate a model and query a cube. The ABC procedure allows you to create SAS programs to query model data outside of SAS Activity-Based Management.

The easiest method to create a program that uses the ABC procedure is to use the Contributions page to perform a query. Then click **Gen Proc Stmt** to copy (to the clipboard) the PROC ABC statement that the Contributions page itself used for the query. For example, assuming that you have performed the query described above, click **Gen Proc Stmt** and paste into the SAS editor (or any ASCII editor). You will see results similar to the following:



You can modify the program in any way that you want before you run it.

Note: The statement that is copied to the clipboard is the last query that you executed not any query that you are in the process of formulating. After formulating a query, you must click **Get Results** before clicking the **Gen Proc Stmt** button to copy the new query to the clipboard.

Note: If you want to run the ABC Procedure in a SAS session outside of SAS Activity-Based Management, then you must set the CLASSPATH variable for the following JAR fies:

- sas.solutions.abm.contributions.jar
- commons-logging.jar
- log4j.jar
- sas.antlr.jar

Chapter 30

Working with Reports

Introduction

This chapter contains task-based information about working with reports. For more conceptual information, see Chapter 14, “Reporting Model Data,” on page 95.

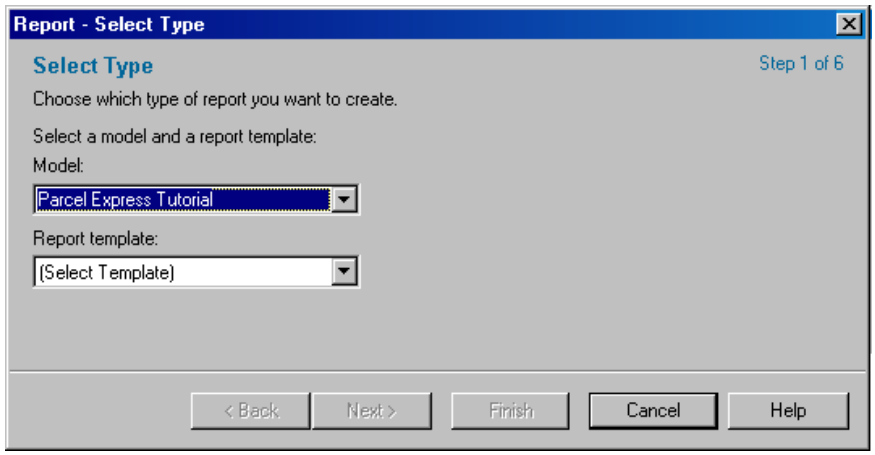
Start Reports Mode

Click the Reports tab in the Navigation Pane.

Create a Report

1. Select **File** ⇨ **New** ⇨ **Report**.

   The Report Wizard appears.

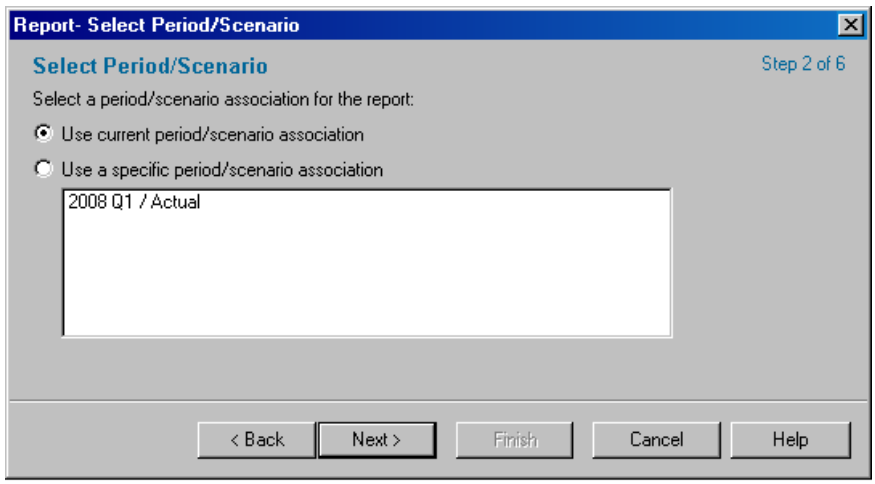


Note: To create a report you do not have to generate a cube. However, to create the following reports you must have already generated the Fact table for the model:

- Resource Contributions
- Destination Furthest
- Profit and Loss (Resource Contribution)

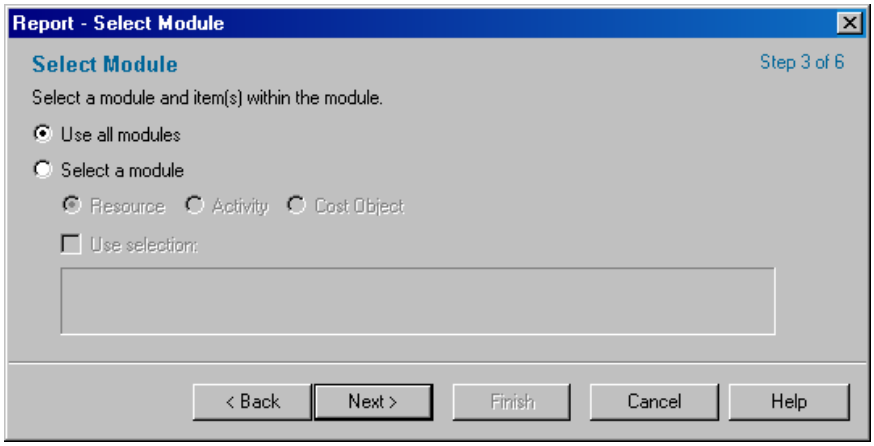
2. Select a **Model**.
3. Select a **Report template**.

   For information about what data each report template contains, see the Help.
4. Click **Next**.

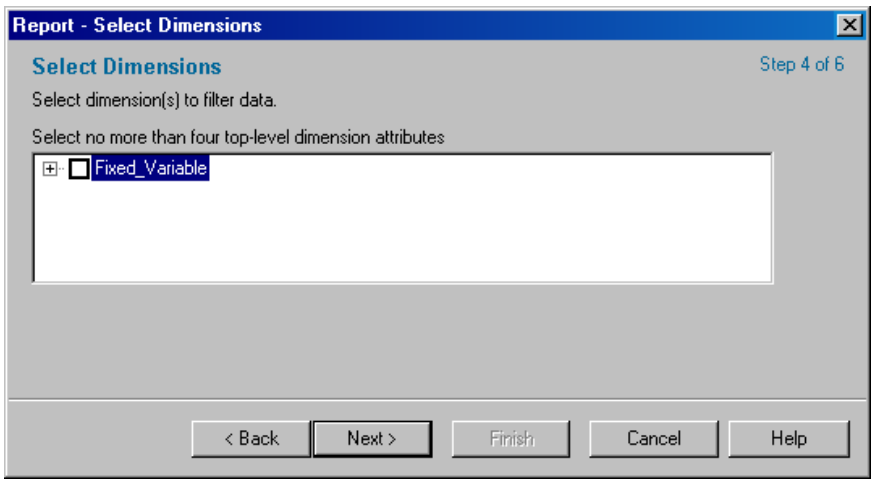


5. To use the data in the period/scenario association for a model that is open, select the **Use current period/scenario association** option.

   If you select this option and a model is not open, you will be prompted to select a period/scenario association at the end of the wizard.
6. To use the data in another period/scenario association, select the **Use a specific period/scenario association** option, and select the period/scenario association from the list.
7. Click **Next**.

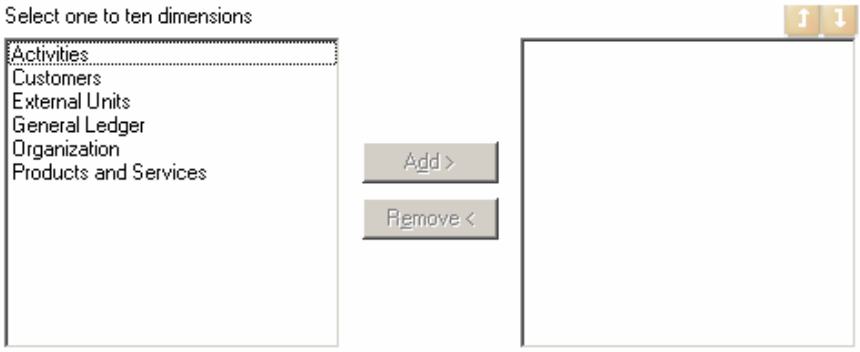


8. To include data from all the modules and all the accounts, select the **Use all modules** option.
9. To limit data to a specific module, select the **Select a module** option, and select a module.
10. To limit data to specific accounts, select the **Use selection** check box, then select one or more accounts from the list.
11. Click **Next**.



Note: The content of this step varies depending on the report you are creating. However, in each variation of content, you are selecting dimensions to include in the report.

12. From the list, expand the hierarchy and select the check box next to each dimension to include in the report.
13. If the wizard presents the dimension attributes as shown, do the following:



a. From the left list, select a dimension.

b. Click **Add >**.

The dimension moves to the right list.

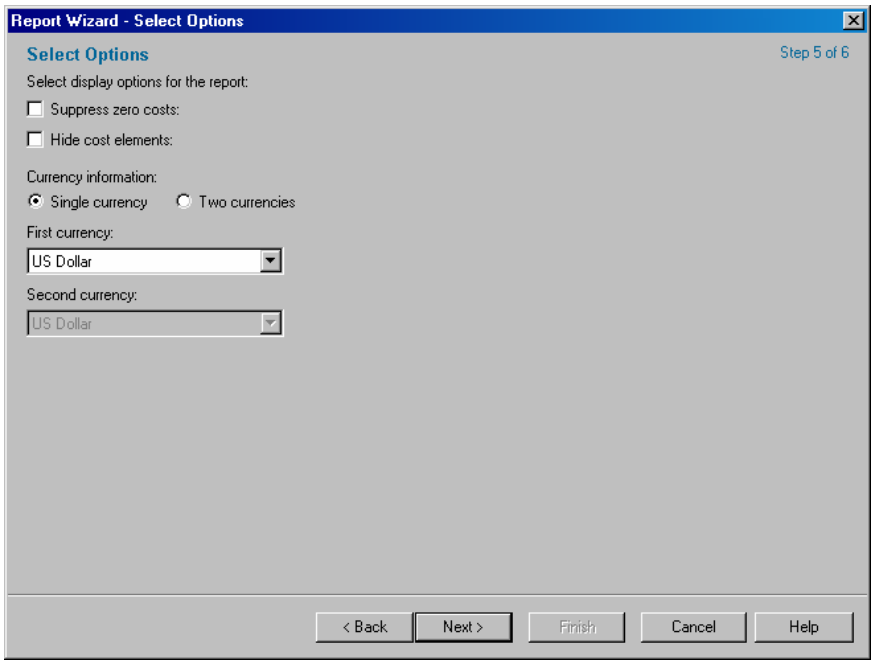
The order that you select dimensions affects the order of the results in the report.

c. To reorder the dimensions in the right list, select a dimension and click  or  .

d. To remove a dimension from the right list, select the dimension and click **Remove <**.

Next, you will select other options for the report. The available options depend on the report.

14. Click **Next**.



15. To exclude accounts that contain zero costs, select the **Suppress zero costs** option.

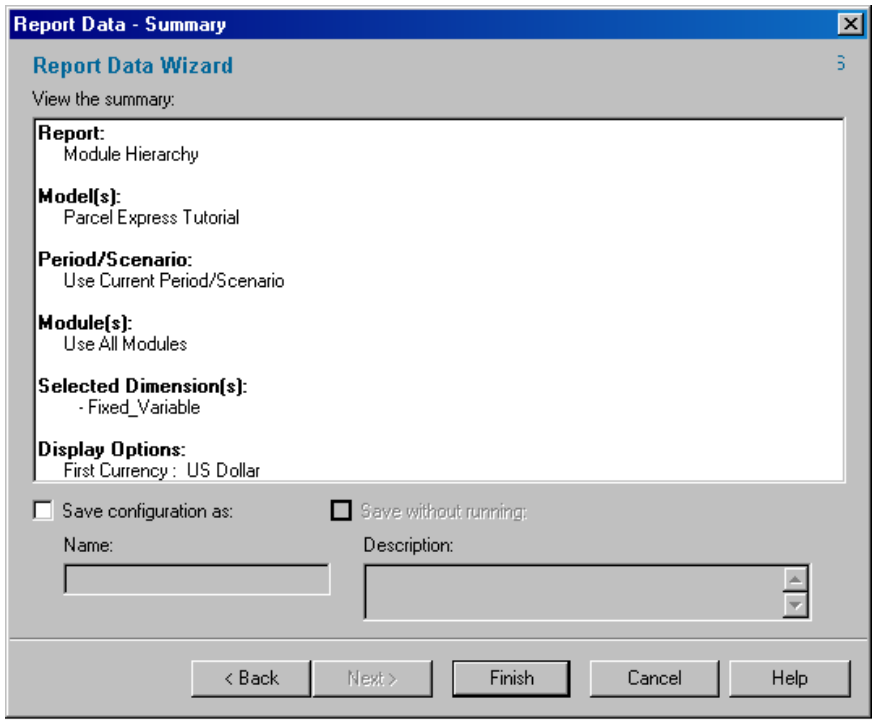
16. To exclude cost elements, select the **Hide cost elements** option.

17. To include a single currency, select the **Single currency** option, and select the **First currency**.

18. To include a second currency, select the **Two currencies** option, and select the **Second currency**.

    you will review the report summary and create the report.

19. Click **Next**.



20. Review the report summary.

21. If you need to change any information, click **Back** until you reach the page 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.

22. 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. Type the **Name**.

    c. Type the **Description**.

23. To rename an existing report configuration, do the following:

    a. Select the **Update existing configuration** option.

    b. Type the new **Name**.

    c. Type the **Description**.

24. Click **Finish**.

    If you selected the option to use the current period/scenario association in the second step of the Report Wizard and the model is not open, you will be prompted to select a period/scenario association now.

Open a Report Without a Report Already Open

If you start Reports mode without a report already open, you can open a report from the Reports workspace.

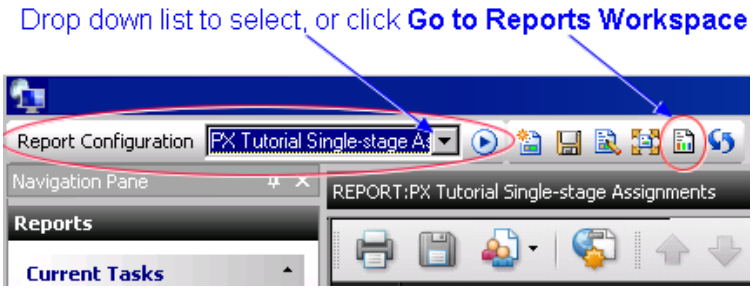
The lists of **Folders** and **Report Configurations** correspond to the Reports branch of the server area in Workspace Manager.

1. From the list of **Folders**, select a folder.
2. From the list of **Report Configurations**, click a configuration.

Open a Report with a Report Already Open

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

- Select a report configuration from the drop-down list at the top of the Reports workspace.
- Click the **Go to Reports Workspace** icon in the Reports workspace toolbar, and then select a report configuration.



Change a Report Configuration

1. Open a report configuration.
2. Select **Reports** ⇨ **Configure**.

   The settings you selected the last time that you used the Report Wizard for this report configuration appear.
3. Move through the pages of the Report Wizard to change the report configuration.

   Note: If you have Read-only permission to a model, then you do not have the ability to configure.

Save a Report Configuration

1. Open a report configuration.
2. Select **Reports > Save Configuration**.

   The last page of the Report Wizard appears, in which you specify the name of the report configuration.
3. Select the **Save configuration as** option.
4. Type the **Name**.
5. Type the **Description**.
6. Click **Finish**.

Export a Report

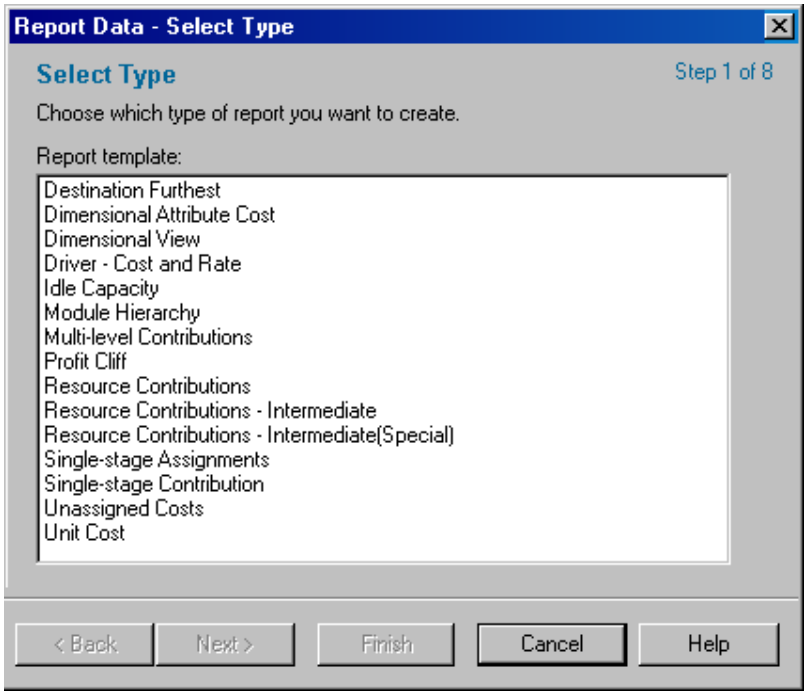
You can export report data from within SAS Activity-Based Management or from inside SAS Enterprise Guide. For information about how to use SAS Enterprise Guide, see the SAS Enterprise Guide Help.

The report data will be exported to a database.

1. Select **File** ⇨ **Export Report Data**.

   You do not need to start Reports mode; you can export report data in any mode.

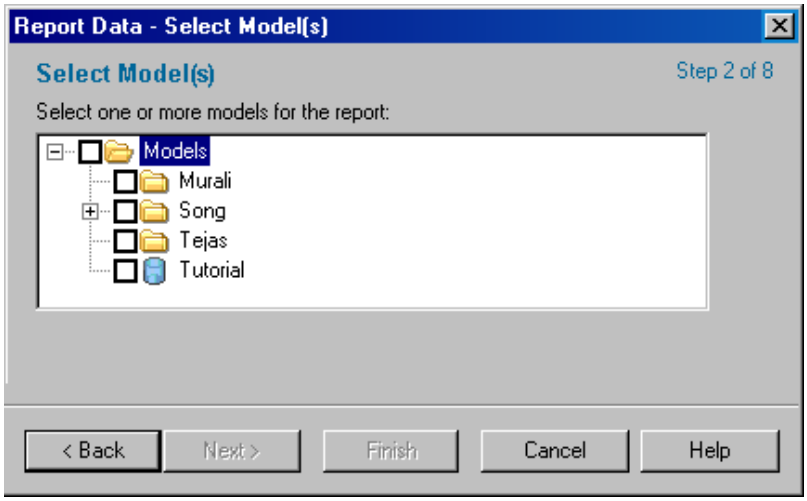
   The Report Data Wizard appears.



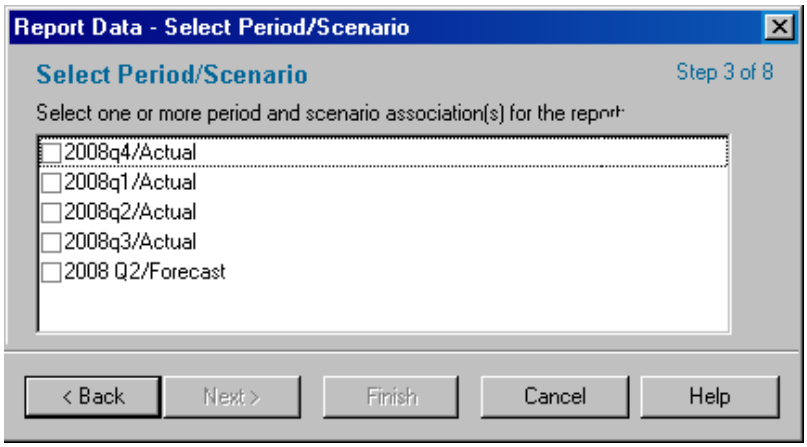
2. Select a **Report template**.

For information about what data each report template contains, see the Help.

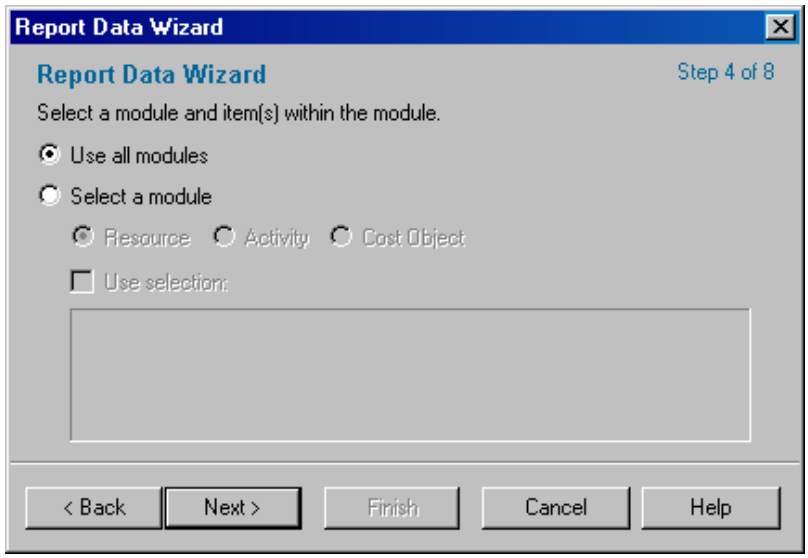
3. Click **Next**.



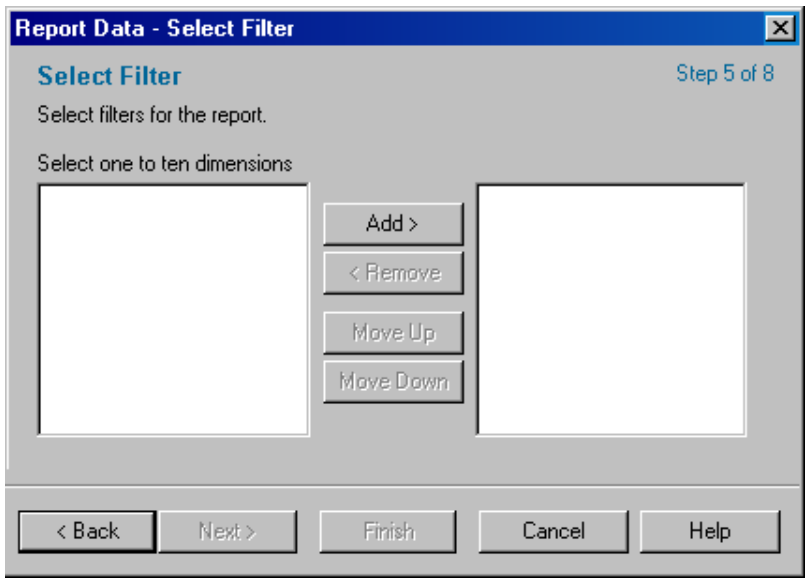
4. Select one or more **Models**.
5. Click **Next**.



6. Select one or more **Period(s)/Scenario(s)**.
7. Click **Next**.

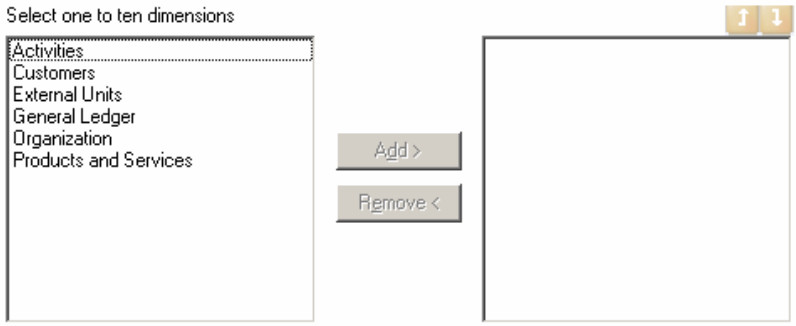


8. To include data from all the modules and all the accounts, select the **Use all modules** option.
9. To limit data to a specific module, select the **Select a module** option, and select a module.
10. To limit data to specific accounts, select the **Use selection** check box, then select one or more accounts from the list.
11. Click **Next**.



Note: The content of this step varies depending on the report you are creating. However, in each variation of content, you are selecting dimensions to include in the report.

12. If the wizard presents a list, then expand the hierarchy and select the check box next to each dimension to include in the report.
13. If the wizard presents the dimensions as shown, do the following:



a. From the left list, select a dimension.

b. Click **Add >**.

The dimension moves to the right list.

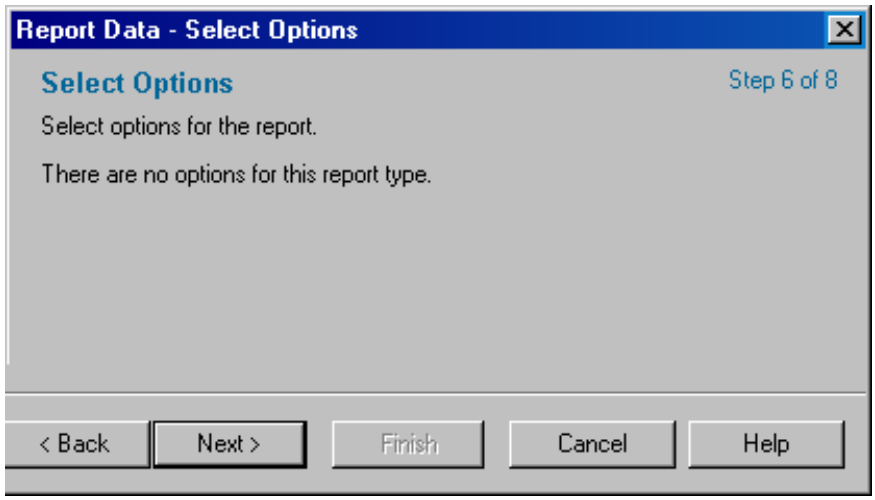
The order that you select dimensions affects the order of the results in the report.

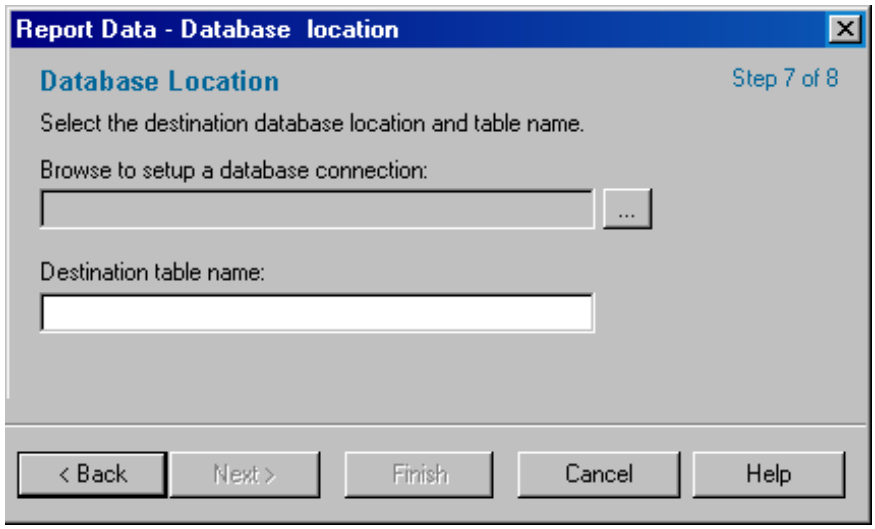
c. To reorder the dimensions in the right list, select a dimension and click  or  .

d. To remove a dimension from the right list, select the dimension and click **Remove <**.

Next, you will select other options for the report. The available options depend on the report.

14. Click **Next**.



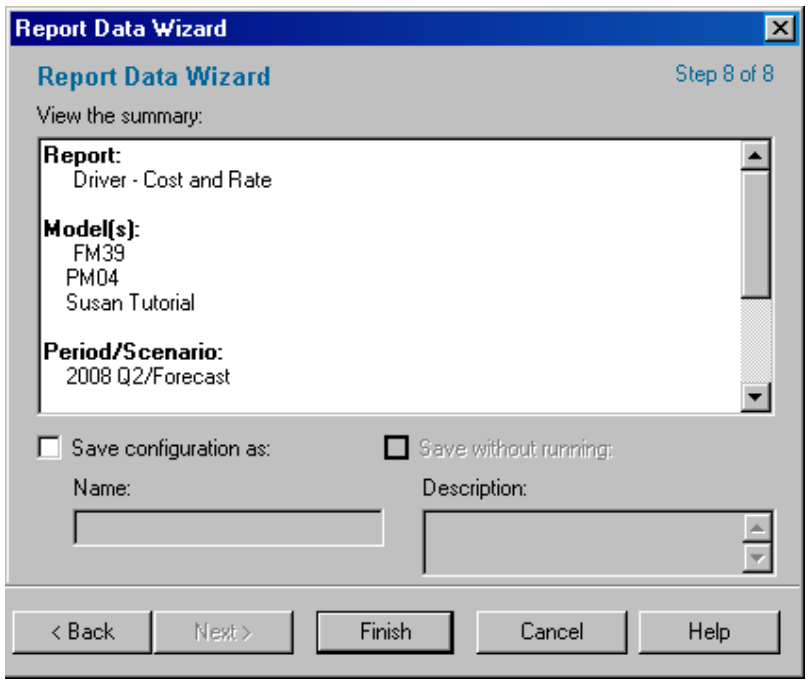


15. Click **...**. (to the right of **Browse to setup a database connection**)

    The Data Link Properties dialog box appears. For information about this dialog box, see the Microsoft Help.

16. Type the **Destination table name**.

17. Click **Next**.



18. Review the export summary.

19. If you need to change any information, click **Back** until you reach the page 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.

20. 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**.

21. To rename an existing export configuration, do the following:

a. Select the **Update existing configuration** option.

b. Type the new **Name**.

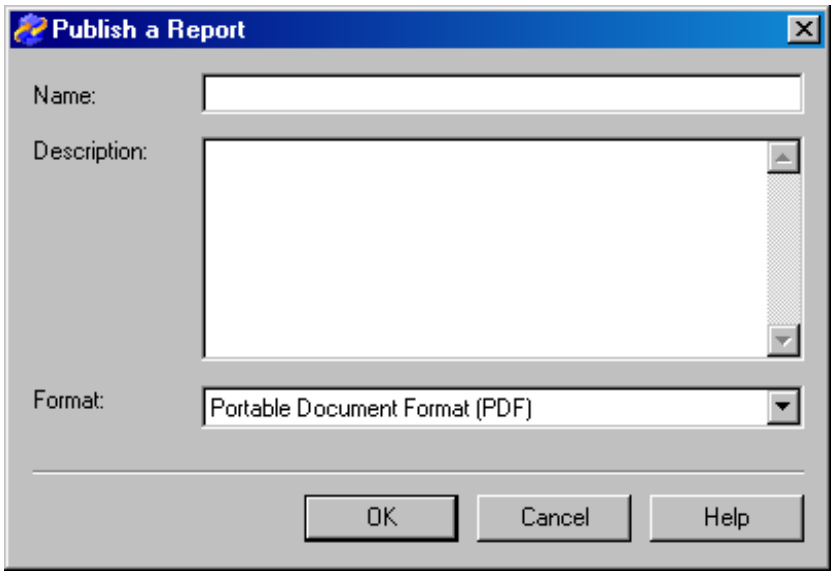
c. Type the **Description**.

22. Click **Finish**.

Publish a Report

1. Open a report in Reports mode.
2. Open a report configuration.
3. Select **Reports** ⇨ **Publish**.

   The Publish a Report dialog box appears.



4. Type the **Name**.

   The name must follow the naming conventions. For information, see the Help.

5. Type the **Description**.
6. Select a **Format**.

   Note: If you have Read-only permission to a model, then you do not have the ability to publish.

Delete a Report Configuration

1. Go to the Reports workspace.
2. Select a report configuration.
3. Click **Delete**.

   Note: Depending on your permissions, you may not have the ability to delete a report configuration.

Chapter 31

Working with Cube Generation

Overview of Cube Generation

Generating a cube involves two tasks:

Task 1: Create a cube configuration
From a single model you can generate multiple cubes. A cube configuration remembers your choices for a particular cube. A model can have multiple cube configurations, each one specifying a single cube, a fact table, or both.

To create a cube configuration, go to the Workspace Manager or to the Resource, Activity, or Cost Object module and select **File** ⇨ **New** ⇨ **Cube Configuration** .

Task 2: Generate a cube

You generate cubes by specifying which of the cube configurations belonging to a model you want to use, and which periods you want to include. If a period has not already been calculated, you are asked if you want to calculate it as part of cube generation. For each cube configuration that you select, a cube and a fact table are generated, or only a fact table.

To generate cubes for a model, open the model and select **Model** ⇨ **Generate Cubes** .

Task 1. Create a Cube Configuration

Overview

You must create a cube configuration before you can generate a cube. A cube configuration holds the options that you select for generating a cube and a fact table. After creating a cube configuration, you use it to generate a cube and a fact table, or a fact table only.

When creating a multi-stage contribution cube, you can choose what to include in the cube so that you can create custom cubes for specialized purposes. From a single model, you can create multiple multi-stage contribution cubes. By selecting what to include in a cube, you can create smaller and faster customized cubes. A cube configuration remembers your choices for a particular cube.

You can create a cube configuration for each of the four types of cubes:

- Single-Stage Contribution Cube
- Resource Contribution Cube
- 6.3 Compatible Multi-Stage Contribution Cube
- Custom Multi-Stage Contribution Cube

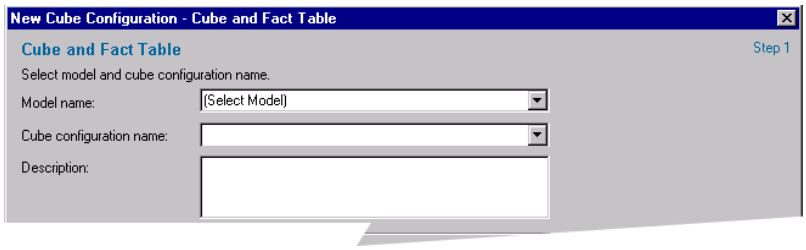
To create a cube configuration, invoke the cube configuration wizard by going to the Workspace Manager or the Resource, Activity, or Cost Object module and selecting **File** ⇨ **New** ⇨ **Cube Configuration** .

Cube Configuration for a Single-Stage Contribution Cube

To create a cube configuration for a single-stage contribution cube, go to the Workspace Manager or the Resource, Activity, or Cost Object module and select **File** ⇨ **New** ⇨ **Cube Configuration** to open the New Cube Configuration wizard. Click **Next** to progress to the each subsequent page of the wizard.

Step 1: General Options

1. Select a model, and name the cube configuration



Model name

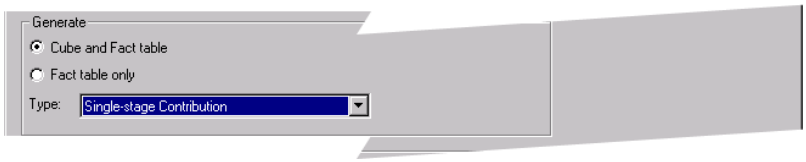
Select the model whose cube and fact table, or fact table only, you want to generate. A model can have more than one cube configuration because you can generate different cubes from the same model.

Cube configuration name

You will use the cube configuration later to generate cubes. In this step of the wizard, you assign a name to the cube configuration.

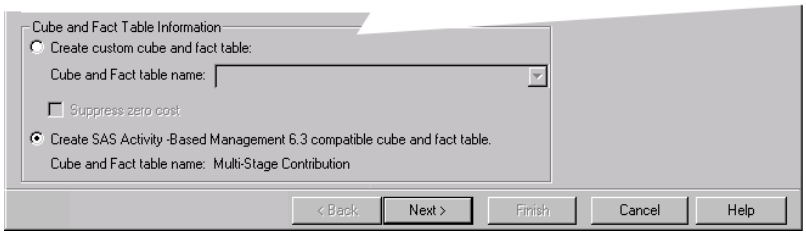
2. Select whether to generate both a cube and a fact table, or only a fact table; and Select **Single-Stage Contribution** as the type of cube to generate.

   A single-stage contribution cube analyzes the cost contributions from one assignment-level back.



   If you choose to generate only a fact table, you can use it to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft Analysis Services.

3. Select to create a **6.3 compatible cube**.

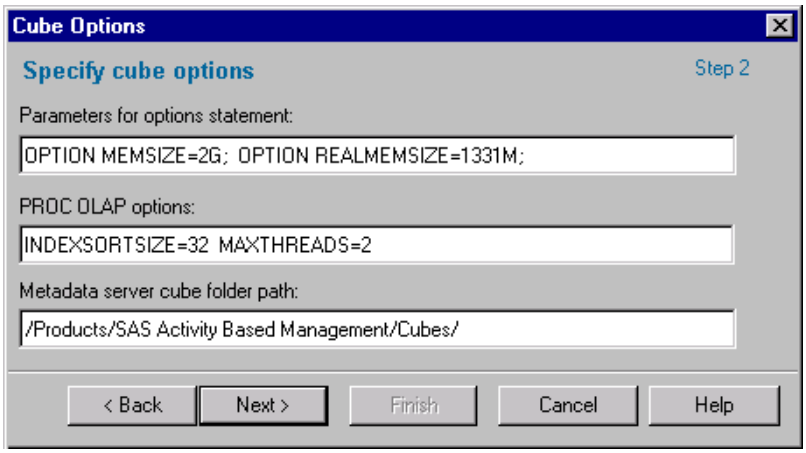


Note: When you select single-stage contribution, you can generate only a 6.3 compatible cube because the option to generate a custom cube is disabled.

Step 2: Specify Cube Options

This step allows you to specify options for cube generation. The options available depend on whether you are using SAS OLAP or Microsoft Analysis Services to display cubes. The following picture shows the options for SAS OLAP cubes.

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.



Specify options to be used in generating the cube.

Paramaters 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 Activity-Based 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*.

Note: The following options are not supported: DRILLTHROUGH_TABLE=, WORKPATH=, DESCRIPTION=, IGNORE_MISSING_DIMKEYS=.

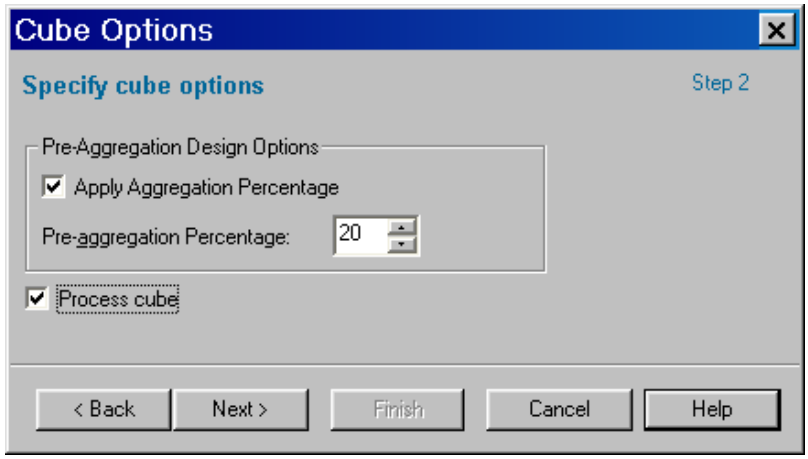
Metadata server cube folder path

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 Activity Based 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.

The following picture shows the options for Microsoft Analysis Services cubes.



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. Therefore, 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.

Step 2: Review Your Selections

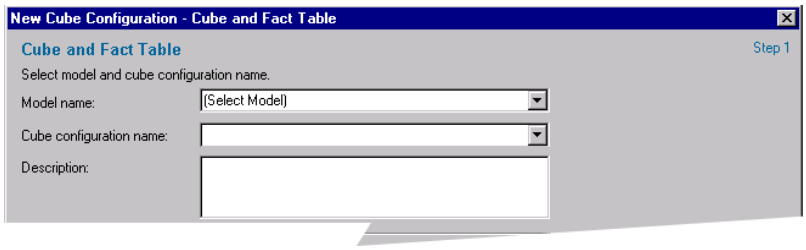
1. Review your selections.
2. Click **Finish** to create the cube configuration.

Cube Configuration for a Resource Contribution Cube

To create a cube configuration for a resource contribution cube, go to the Workspace Manager or the Resource, Activity, or Cost Object module and select **File** ⇨ **New** ⇨ **Cube Configuration** to open the New Cube Configuration wizard. Click **Next** to progress to the each subsequent page of the wizard.

Step 1: General Options

1. Select a model, and name the cube configuration



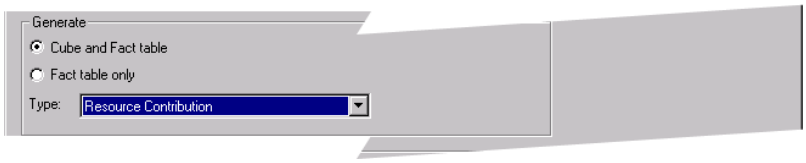
Model name

Select the model whose cube and fact table, or fact table only, you want to generate. A model can have more than one cube configuration because you can generate different cubes from the same model.

Cube configuration name

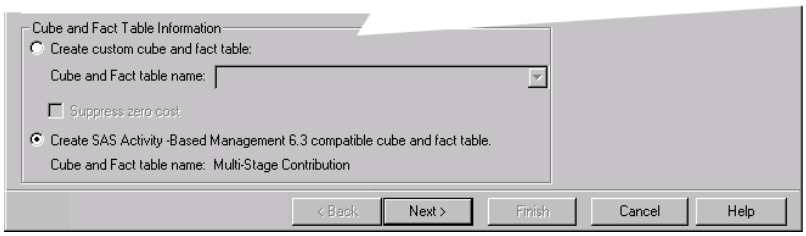
You will use the cube configuration later to generate cubes. In this step of the wizard, assign a name to the cube configuration.

2. Select whether to generate both a cube and a fact table, or only a fact table; and select **Resource Contribution** as the type of cube.



If you choose to generate only a fact table, you can use it to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft Analysis Services.

3. Select to create a **6.3 compatible cube**.

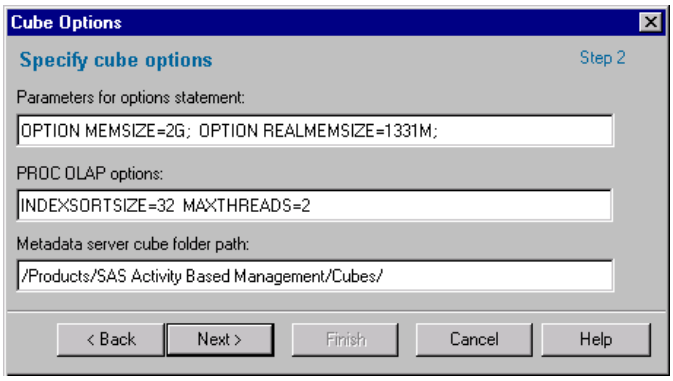


Note: When you select resource contribution, you can generate only a 6.3 compatible cube because the option to generate a custom cube is disabled.

Step 2: Specify Cube Options

This step allows you to specify options for cube generation. The options available depend on whether you are using SAS OLAP or Microsoft Analysis Services to display cubes. The following picture shows the options for SAS OLAP cubes.

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.



Specify options to be used in generating the cube.

Paramaters for options statement

Specify SAS System options here to be used in generating the OLAP cube.

For available system options that can be specified with the options statement, see "SAS System Options" in the *SAS 9.2 Language Reference: Dictionary*.

PROC OLAP options

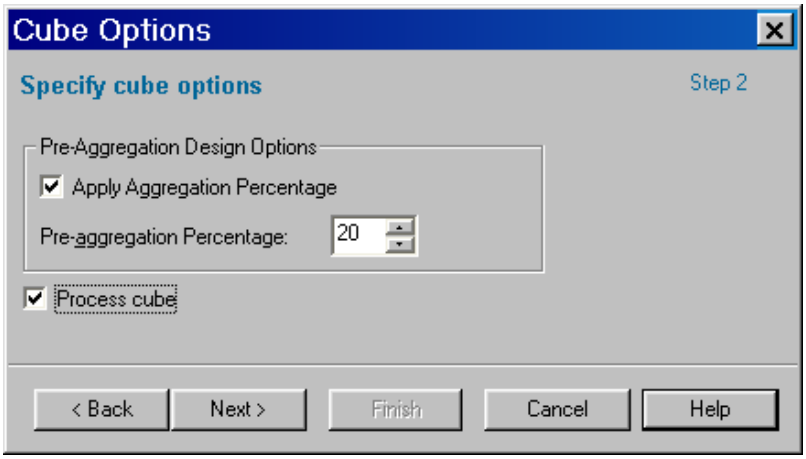
SAS Activity-Based Management uses the SAS OLAP procedure to generate cubes. If you want other than the default options to be used, you can specify them here.

For available PROC OLAP options, see "The OLAP Procedure" in the *SAS OLAP Server: User's Guide*.

Metadata server cube folder path

If you want to override the default location for the generated cube, you can specify a path on the metadata server.

The following picture shows the options for Microsoft Analysis Services cubes.



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. Therefore, 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.

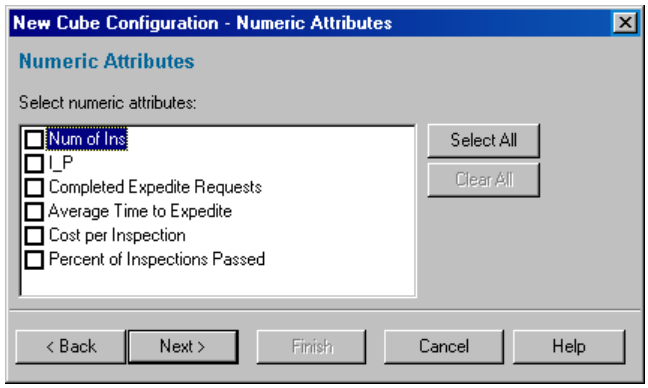
Process cube

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.

Step 3: Select Numeric Attributes for Inclusion

Note : The following picture is only a sample. The attributes listed depend on the attributes in your model.



Note : To select the numeric attributes that are included by default in the cube configurations for a model:

1. In model mode, select **Model** ⇨ **Properties** .
2. Select the **Attributes in Cube** tab.
3. Select the attributes that you want to include by default in a cube configuration for that model.

Attributes that you select in the Model Properties dialog are automatically checked in the cube configuration to be included in the generated cube. However, you can uncheck the attributes in the cube configuration and select others.

Step 4: Review Your Selections

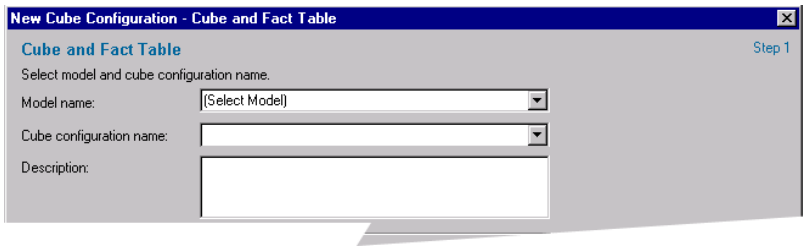
1. Review your selections.
2. Click **Finish** to create the cube configuration.

Cube Configuration for a 6.3 Compatible Multi-Stage Contribution Cube

To create a cube configuration for a 6.3 compatible multi-stage contribution cube, go to the Workspace Manager or the Resource, Activity, or Cost Object module and select **File** ⇨ **New** ⇨ **Cube Configuration** to open the New Cube Configuration wizard. Click **Next** to progress to the each subsequent page of the wizard.

Step 1: General Options

1. Select a model, and name the cube configuration



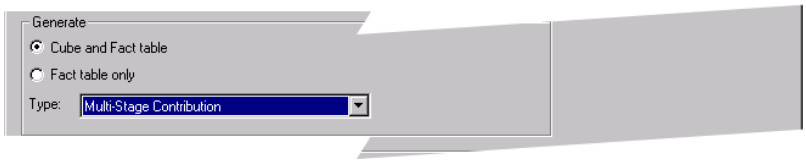
Model name

Select the model whose cube and fact table, or fact table only, you want to generate. A model can have more than one cube configuration because you can generate different cubes from the same model.

Cube configuration name

You will use the cube configuration later to generate cubes. In this step of the wizard, assign a name to the cube configuration.

2. Select whether to generate both a cube and a fact table, or only a fact table.

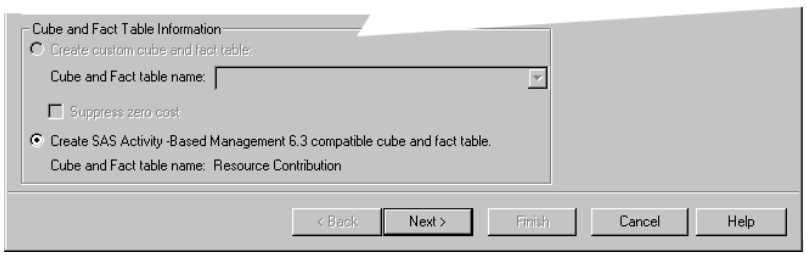


If you choose to generate only a fact table, you can use it to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft Analysis Services.

3. Select **Multi-Stage Contribution** as the type of cube.

A multi-stage contribution cube analyzes the cost contributions to or from accounts that have a Stages attribute.

4. Select to create a 6.3 compatible cube and fact table



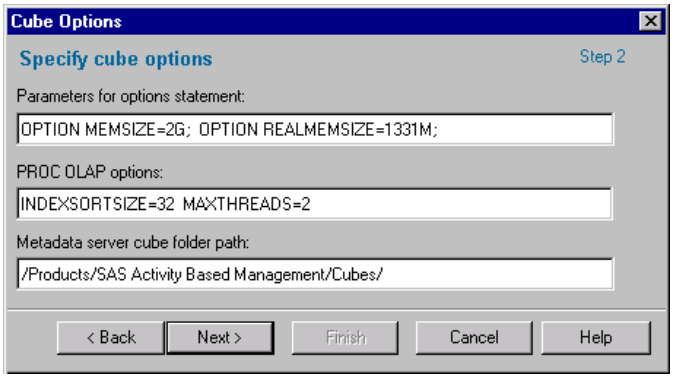
When you select this option for a multi-stage contribution cube, you do not have as many choices for what to include in the cube as you do for a custom cube. SAS Activity-Based Management makes the choices for you and the cube is generated exactly as it was in the previous release. The name of the cube is automatically assigned as: Multi-Stage Contribution.

Note: You can have only one 6.3-compatible cube configuration of a given type for example, multi-stage contribution per model. This is because you can have only one 6.3-compatible cubeof a given type per model. If you create a second cube configuration for a 6.3-compatible cube, then the first cube configuration is de-activated (detached from the model). You are issued an informational message when this happens.

Step 2: Specify Cube Options

This step allows you to specify options for cube generation. The options available depend on whether you are using SAS OLAP or Microsoft Analysis Services to display cubes. The following picture shows the options for SAS OLAP cubes.

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.



Specify options to be used in generating the cube.

Paramaters for options statement

Specify SAS System options here to be used in generating the OLAP cube.

For available system options that can be specified with the options statement, see "SAS System Options" in the *SAS 9.2 Language Reference: Dictionary*.

PROC OLAP options

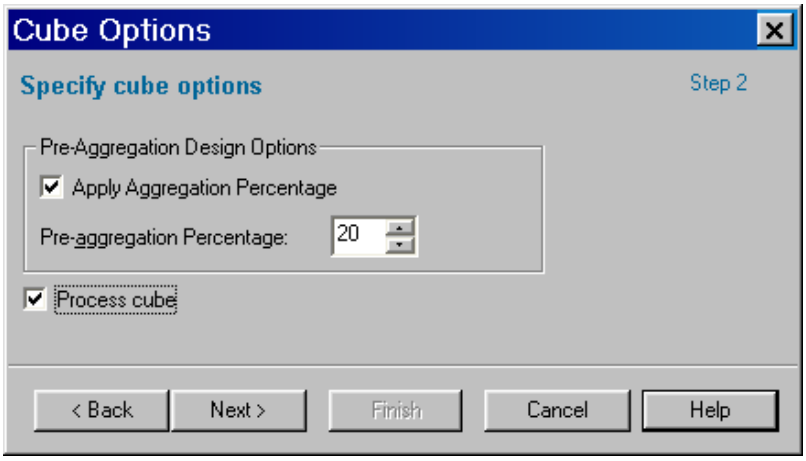
SAS Activity-Based Management uses the SAS OLAP procedure to generate cubes. If you want other than the default options to be used, you can specify them here.

For available PROC OLAP options, see "The OLAP Procedure" in the *SAS OLAP Server: User's Guide*.

Metadata server cube folder path

If you want to override the default location for the generated cube, you can specify a path on the metadata server.

The following picture shows the options for Microsoft Analysis Services cubes.



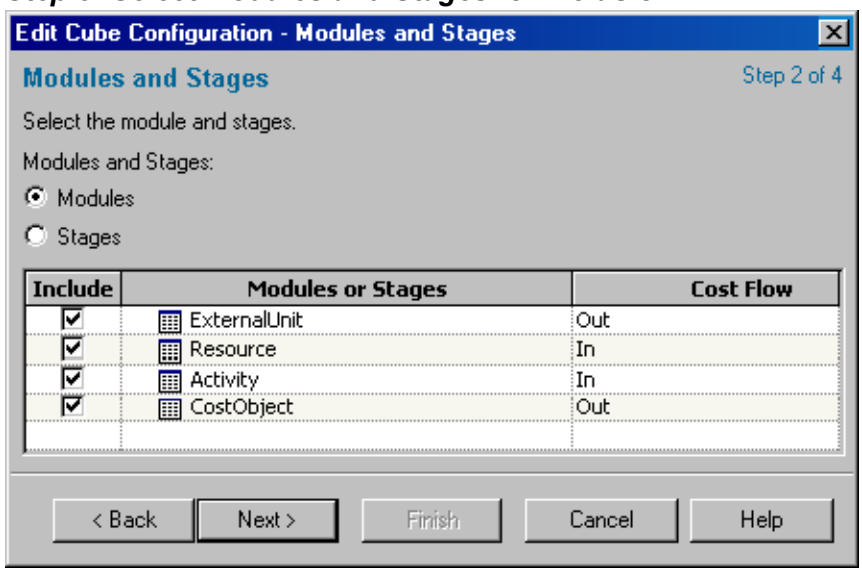
Pre-aggregation Percentage

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. Therefore, 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.

Step 3: Select Modules and Stages for Inclusion



Modules and Stages

Modules: Each module defines a single stage (one stage per module) in the following order:

1. External Units
2. Resource
3. Activity
4. Cost Object

Stages: Each stage is defined by a dimension member attribute in a dimension attribute named Stages.

Include

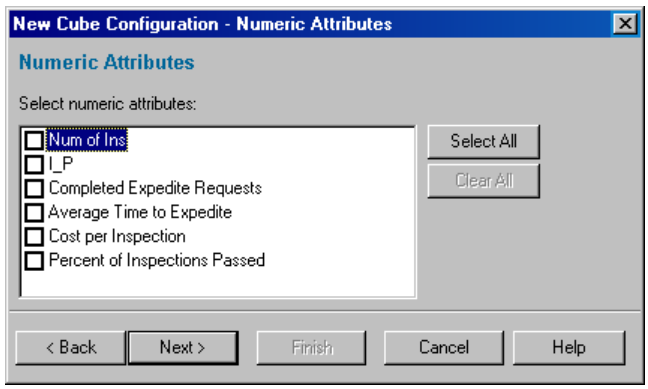
Check this box to include the module or stages in the cube.

Cost Flow

Select whether to include cost flows into or out of the selected module or stage.For information on this topic, see "Select Cost Flow: In or Out" on page 232.

Step 4: Select Numeric Attributes for Inclusion

Note: The following picture is only a sample. The attributes listed depend on the attributes in your model.



Note : You can select the numeric attributes that are included by default in the cube configurations for a model.

1. In model mode, select **Model** ⇨ **Properties** .
2. Select the **Attributes in Cube** tab.
3. Select the attributes that you want to have included by default in a cube configuration for that model.

Attributes that you select in the Model Properties dialog are automatically checked in the cube configuration for inclusion in the generated cube. However, you can uncheck the attribute in the cube configuration and select other attributes.

Step 5: Review Your Selections

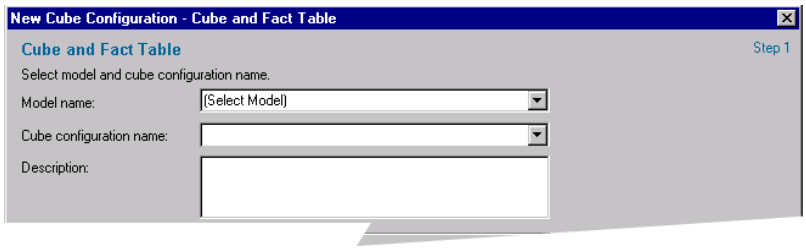
1. Review your selections.
2. Click **Finish** to create the cube configuration.

Cube Configuration for a Custom Multi-Stage Contribution Cube

To create a cube configuration for a custom multi-stage contribution cube, go to the Workspace Manager or the Resource, Activity, or Cost Object module and select **File** ⇨ **New** ⇨ **Cube Configuration** to open the New Cube Configuration wizard. Click **Next** to progress to the each subsequent page of the wizard.

Step 1: General Options

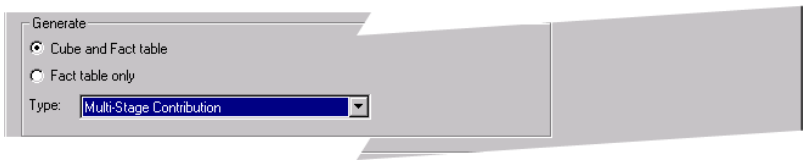
1. Select a model and name the cube configuration



Model name
Select the model whose cube and fact table, or fact table only, you want to generate. A model can have more than one cube configuration because you can generate different cubes from the same model.

Cube configuration name
You will use the cube configuration later to generate cubes. In this step of the wizard, assign a name to the cube configuration.

2. Select whether to generate both a cube and a fact table, or only a fact table

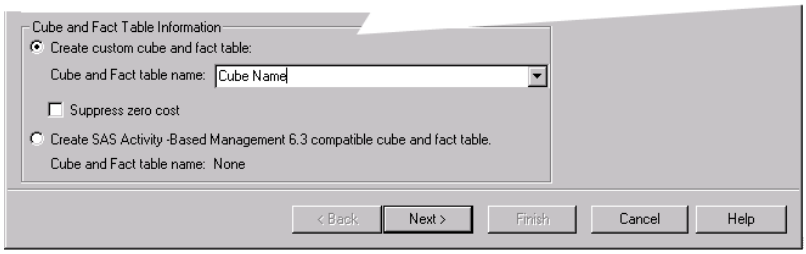


If you choose to generate only a fact table, you can use it to generate a customized cube in another application, such as SAS OLAP Cube Studio or Microsoft Analysis Services.

3. Select **Multi-Stage Contribution** as the type of cube to generate.

A multi-stage contribution cube analyzes the cost contributions to or from accounts that have a Stage attribute.

4. Select the option to create a custom cube and fact table



5. Name the cube and fact table.

Because you can generate different multi-stage contribution cubes from the same model, you must assign a name to each cube to identify it.

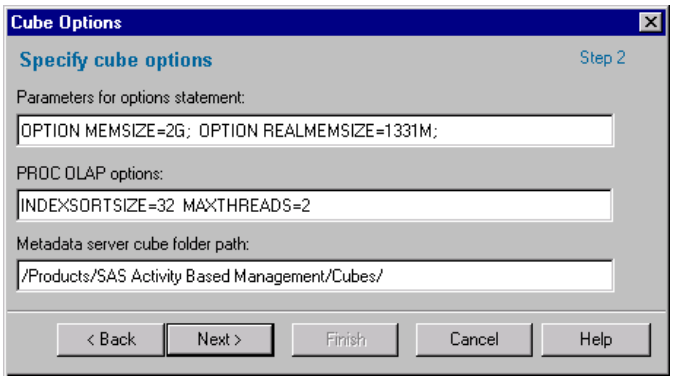
Note: The name is also assigned to the fact table. If you generate only a fact table, then it is the fact table name.

Note: The name that you specify here is for display purposes inside SAS Activity-Based Management. Custom, multi-stage contribution cubes are automatically assigned an internal name, such as , on your OLAP server. You do not see this internal name unless you go outside of SAS Activity-Based Management.

Step 2: Specify Cube Options

This step allows you to specify options for cube generation. The options available depend on whether you are using SAS OLAP or Microsoft Analysis Services to display cubes. The following picture shows the options for SAS OLAP cubes.

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.



Specify options to be used in generating the cube.

Paramaters for options statement

Specify SAS System options here to be used in generating the OLAP cube.

For available system options that can be specified with the options statement, see "SAS System Options" in the *SAS 9.2 Language Reference: Dictionary*.

PROC OLAP options

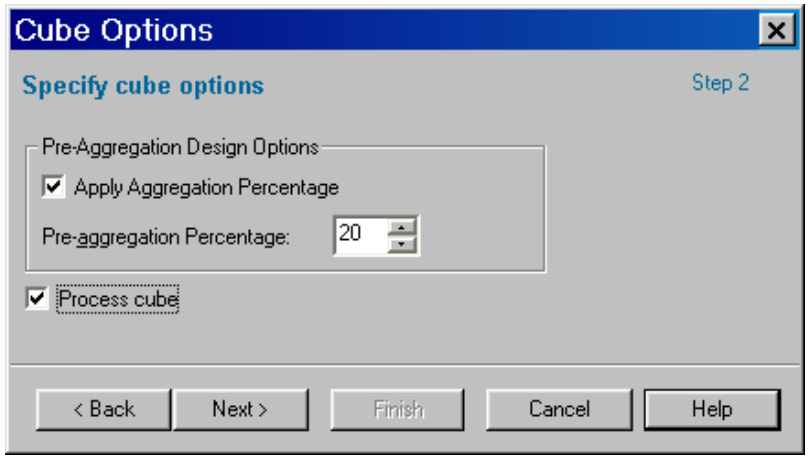
SAS Activity-Based Management uses the SAS OLAP procedure to generate cubes. If you want other than the default options to be used, you can specify them here.

For available PROC OLAP options, see "The OLAP Procedure" in the *SAS OLAP Server: User's Guide*.

Metadata server cube folder path

If you want to override the default location for the generated cube, you can specify a path on the metadata server.

The following picture shows the options for Microsoft Analysis Services cubes.



Pre-aggregation Percentage

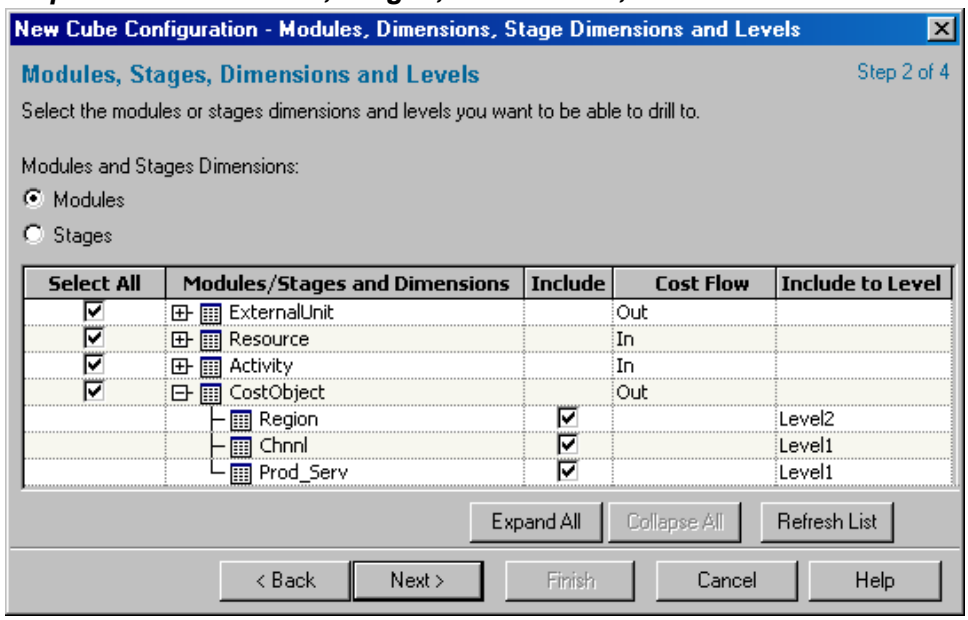
Pre-aggregating numeric data results in faster performance when you work with a cube in the OLAP analyzer. But, for large models, pre-aggregating data can take a long time, which increases cube-generation time. Therefore, the more pre-aggregation you do, the faster you can navigate a cube, but the longer 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.

Step 3: Select Modules, Stages, Dimensions, and Levels



Modules and Stages

Modules : Each module defines a single stage (one stage per module) in the following order:

1. External Units
2. Resource
3. Activity
4. Cost Object

Stages: Each stage is defined by a dimension member attribute in a dimension attribute named Stages.

Select All

If you select **Modules** , then check this box to select all dimensions in a module for inclusion in the cube. Selecting this check box turns on the Include flag for each individual dimension.

If you select **Stages** , then check this box to include the stages in the cube.

Include

If you selected **Modules** , then check this box to include the selected dimension in the cube.

If you selected **Stages** , then this column is not applicable.

Cost Flow

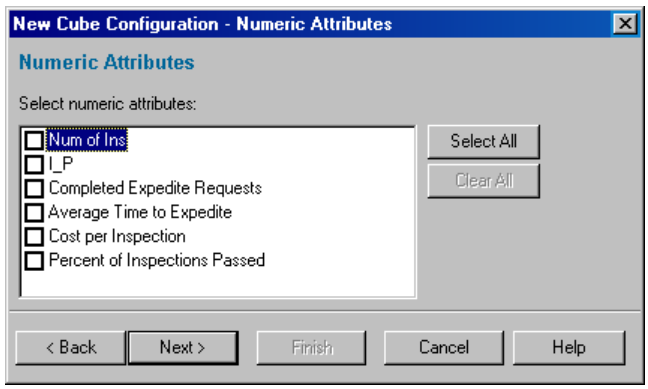
Select whether to include cost flows into or out of the selected module or stage. For information on this topic, see "Select Cost Flow: In or Out" on page 232.

Include to Level

For dimensions with multiple levels, specify how many levels you want to include in the cube for drill down.

Step 4: Select Numeric Attributes for Inclusion

Note: The following picture is only a sample. The attributes that are listed depend on the attributes in the model.



Note: To select the numeric attributes that are included by default in a model's cubes, do the following:

1. In model mode, select **Model** ⇨ **Properties** .
2. Select the **Attributes in Cube** tab.
3. Select the attributes that you want to include by default in a cube configuration for that model.

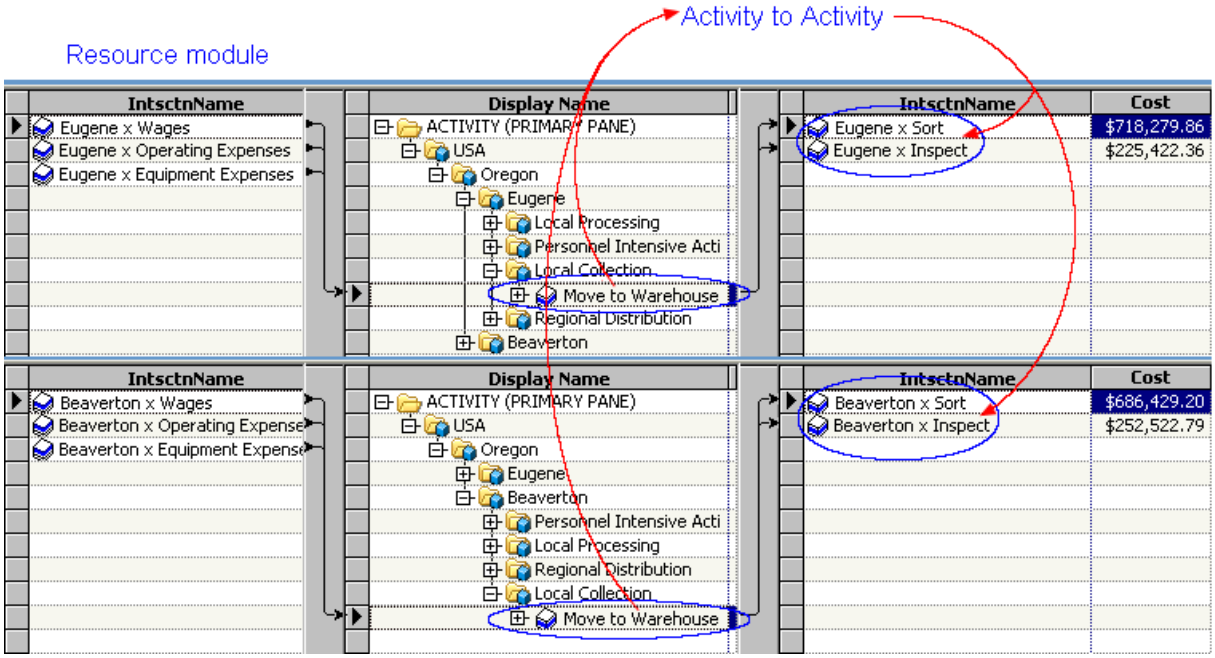
Attributes that you select in Model Properties are automatically checked in the cube configuration to be included in the generated cube. However, you can uncheck the attributes in the cube configuration and select other attributes.

Step 5: Review Your Selections

1. Review your selections.
2. Click **Finish** to create the cube configuration.

Select Cost Flow: In or Out

With some models, it makes a difference in cube generation whether you choose to show costs flowing into a module/stage or out of a module/stage. It makes a difference when the model has assignments from accounts within one module/stage to other accounts within the same module/stage. In the Parcel Express Tutorial model, for example, there are assignments from activities to activities. One such example is shown in the following figure.



When you generate a cube, you must specify what cost assignments you want to appear in the cube for each module/stage. In the case of the current example, you must choose assignments into the Activity module (Move to Warehouse), or out of the Activity Module (Sort and Inspect). You cannot choose both because that would result in double-counting final costs (part of Sort and Inspect costs are Move to Warehouse costs).

The following figure shows the results of showing costs flowing into the Activity module. Notice that Move to Warehouse has non-zero cost because it receives assignments from the Resource module. Inspect and Sort also have non-zero costs because they also receive assignments from the Resource module.

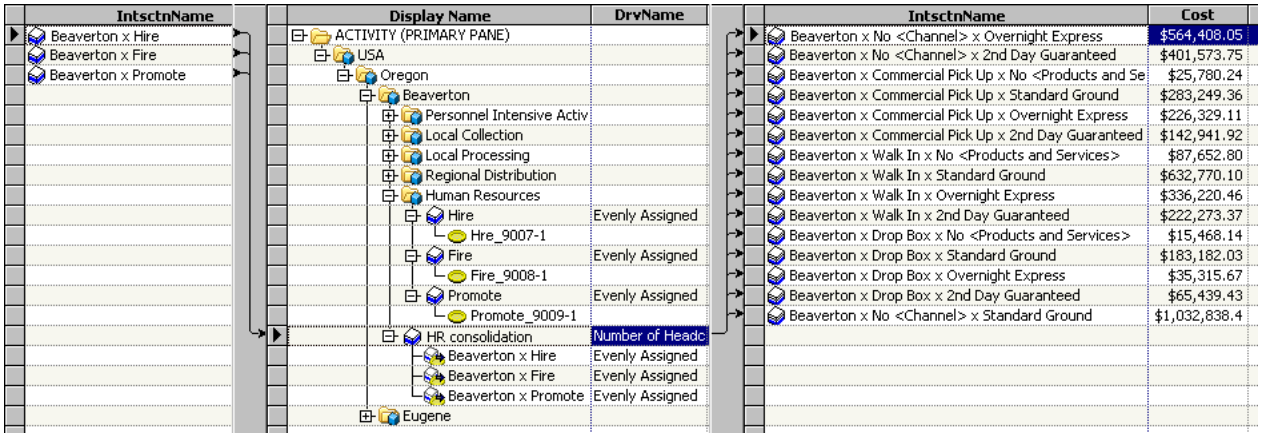
All_Products_and_Services		All
MeasuresLevel		Cost
Level1	*Level2*	
All		.
Local Collection	**Local Collection**	689,462.17
	Local Collection (direct)	.
	Move to Warehouse	689,462.17
Local Processing	**Local Processing**	770,596.55
	Inspect	364,282.53
	Local Processing (direct)	.
	Sort	406,314.02
None		208,393.20
Personnel Intensive Activities		405,984.94
Regional Distribution		1,781,856.34

By contrast, the following figure illustrates the results of showing costs flowing out of the Activity module. Notice that Move to Warehouse now has zero cost because it has no assignments out of the Activity module into the Cost Object module. Its assignments are entirely within the Activity module.

All_Products_and_Services		All
MeasuresLevel		Cost
Level1	Level2	
All		.
Local Collection	**Local Collection**	.
	Local Collection (direct)	.
	Move to Warehouse	.
Local Processing	**Local Processing**	55,349.66
	Inspect	55,349.66
	Local Processing (direct)	.
	Sort	.
None		208,393.20
Personnel Intensive Activities		405,984.94
Regional Distribution		3,186,565.40

If a model has no cost assignments from accounts within a module/stage to other accounts within the same module/stage, then it makes no difference whether you show costs flowing into or out of a module/stage. (Every cost into a module/stage is also a cost out of the module/stage). If you define multiple stages so that there are no assignments within a single stage, you do not have to make the choice of showing cost flows in or out.

You might, however, in some cases want to make assignments within a single stage or module. One common reason is to consolidate cost assignments. For example, suppose a model has a Human Resources dimension member with three cost accounts: Hire, Fire, and Promote. Although for some purposes you want to know how much expense is devoted to each HR activity, for other purposes you care about only the total HR expense. To see the total expense, you can create an account whose sole purpose is to consolidate all of the HR expenses as shown in the following figure. Here, the HR consolidation activity receives assignments from each individual HR activity: Hire, Fire, Promote. And, in turn, the HR consolidation activity assigns its costs to all of the products.



The following figure illustrates the results of choosing cost flows into the Activity module. Notice that Human Resources has costs coming in, but HR Consolidation has zero costs because it has assignments going out of the module only.

All_Products_and_Services		All
	MeasuresLevel	Cost
Level1	*Level2*	
All		3 856 893,20
HR Consolidation		.
Human Resources	**Human Resources**	600.00
	Fire	200.00
	Hire	200.00
	Human Resources (direct)	.
	Promote	200.00
Local Collection		689,462.17
Local Processing		770,596.55
None		208,393.20
Personnel Intensive Activities		405,984.94
Regional Distribution		1,781,856.34

By contrast, the following figure illustrates the results of choosing cost flows out of the Activity module. Notice that HR Consolidation now has costs flowing out of the module, and Human Resources doesn t show costs because all of its costs are consolidated under HR Consolidation.

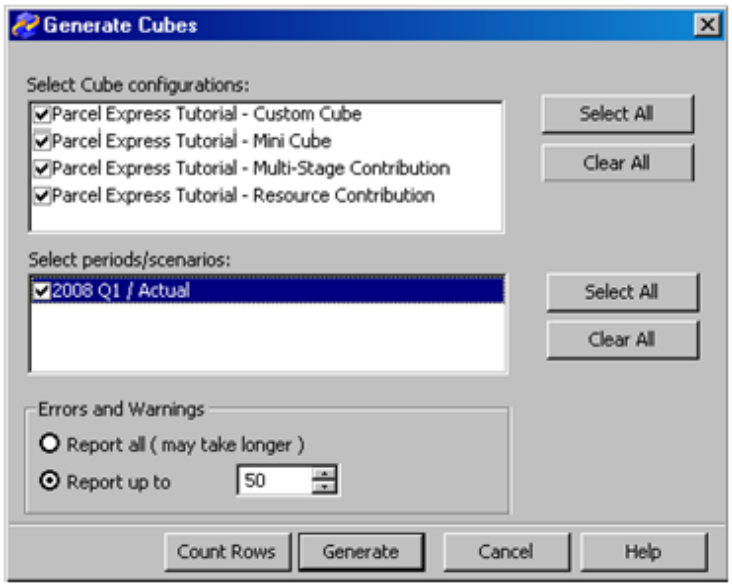
All_Products_and_Services		All
	MeasuresLevel	Cost
Level1	*Level2*	
All		3 856 893,20
HR Consolidation	**HR Consolidation**	600,00
	HR Consolidation (direct)	600,00
Human Resources		.
Local Collection		.
Local Processing		55,349.66
None		208,393.20
Personnel Intensive Activities		405,984.94
Regional Distribution		3,186,565.40

Task 2. Generate Cubes

After you create cube configurations, you can use them to generate cubes.

1. In Model mode, select **Model** ⇨ **Generate Cubes** .

   The Generate Cubes dialog box opens. The dialog box lists all of the cube configurations for the model (the following picture is just a sample).

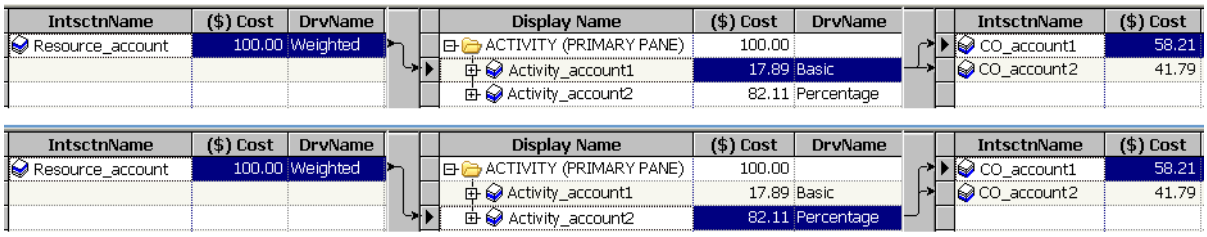


2. Select the cube configurations that you want to use. Each cube configuration that you choose generates a cube and a fact table, or only a fact table.
3. Select the periods and scenarios that you want to include in the cube.
4. Click **Generate**.

Postscript: Detailed Example of a Calculation

Introduction

This postscript describes what the system does to calculate the cost flows between accounts for the following simple example:



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.

To summarize the calculation process for this example:

- Resource_account assigns costs to Activity_account1 and Activity_account2 using a weighted driver.

  The weighted driver allows both fixed driver quantities (DQF and DWF) and variable driver quantities (DQV and DWV).

  The driver also allows user-entered cost allocation (DrvAllocCost).

- Activity_account1 assigns costs to CO_account1 and CO_account2 using a basic driver.

  This basic driver allows both fixed driver quantities (DQF) and variable driver quantities (DQV) — but not DWF and DWV because it is not a weighted driver.

- Activity_account2 also assigns costs to CO_account1 and CO_account2 but using a percentage driver.

To anticipate, the process by which the system determines the cost flow from Resource_account to Activity_account1 and Activity_account2 is the following:

1. The user enters driver quantities (DQF, DQV, DWF, DWV).

   The subsequent steps are all performed by the system.

2. Calculate Dest.TDQ for each assignment path.
3. Calculate DrvQtyCalc for each assignment path.
4. Calculate TDQCalc as the sum of all DrvQtyCalc.
5. Assign TDQCalc to the Source.TDQ.
6. Calculate Source.Allocated Cost as the sum of all DrvAllocCost.
7. Calculate DrivableCost as (Cost - Allocated Cost).
8. Calculate DriverRate as (DrivableCost / TDQ).
9. Calculate Received Driven Cost as (DriverRate x DrvQtyCalc) for each assignment path.
10. Calculate DriverCost as (Received Driven Cost + DrvAllocCost) for each assignment path.

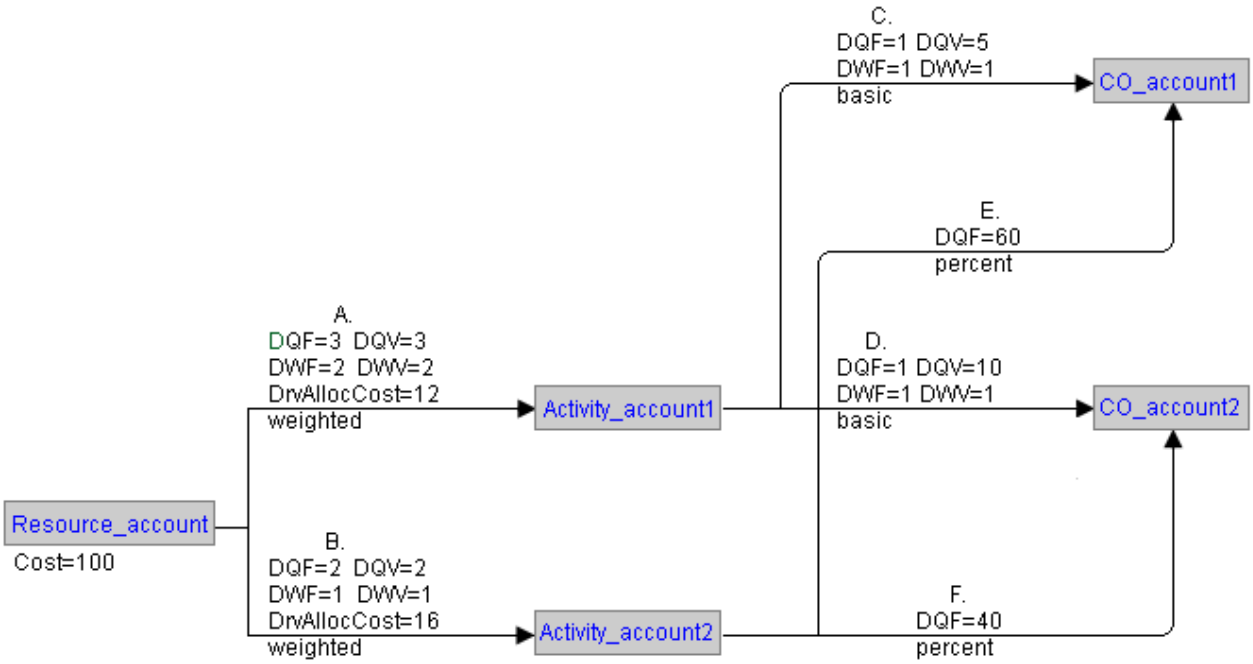
    Next, the preceding steps are repeated—this time to calculate the cost flows from Activity_account1 and Activity_account2 to CO_account1 and CO_account2.

11. Calculate Dest.TDQ for Cost Object Accounts and Calculate DrvQtyCalc.
12. Calculate TDQCalc and TDQ.
13. DrivableCost = Cost - Allocated Cost.
14. DriverRate = DrivableCost / TDQ.
15. Received Driven Cost = DriverRate x DrvQtyCalc.
16. Calculate Cost for Cost Object Accounts by adding different assignment paths.

Step 1– User Enters Driver Quantities (DQF, DQV, DWF, DWV)

The driver quantities for this example are shown in the following picture.

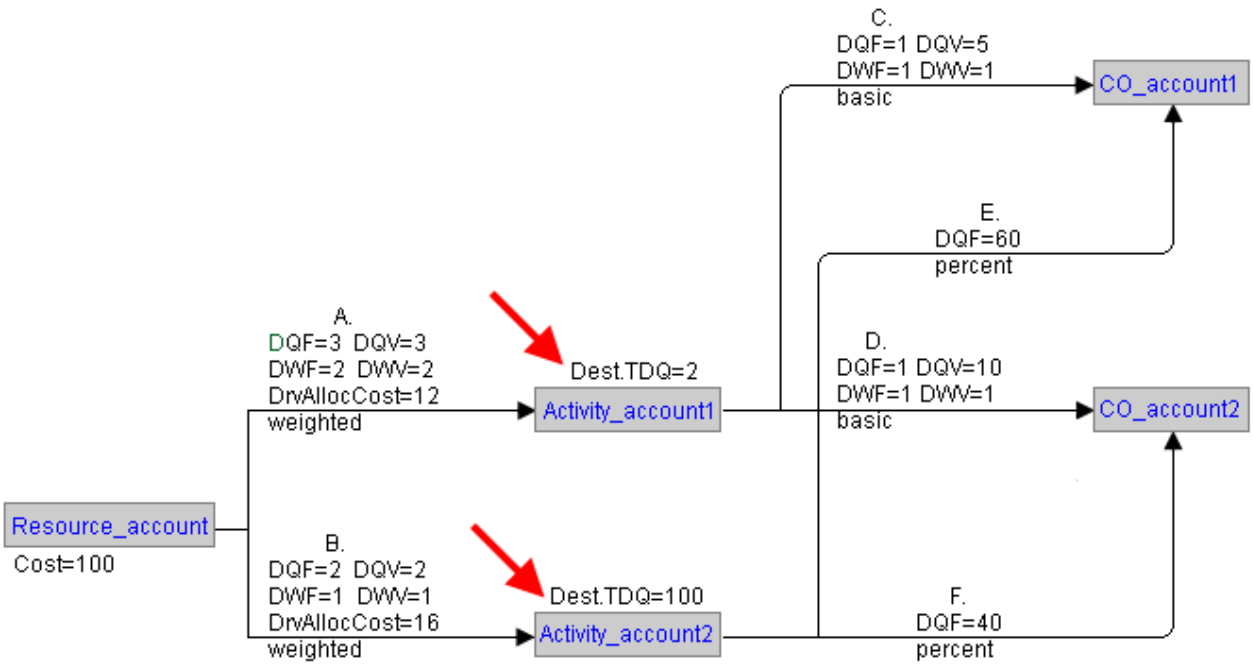
Note: For non-weighted drivers, DWF and DWV default to 1.



Here are the user-entered driver quanties:

- DQF (Driver Quantity Fixed)
- DQV (Driver Quantity Variable)
- DWF (Driver Weighted Quantity Fixed)
- DWV (Driver Weighted Quantity Variable)

Step 2–Calculate Dest.TDQ for Each Assignment Path



Dest.TDQ for Activity_account1 = DQF to CO_account1 + DQF to CO_account2

Dest.TDQ for Activity_account1 = 1 + 1

Dest.TDQ for Activity_account1 = 2

Dest.TDQ for Activity_account2 = DQF to CO_account1 + DQF to CO_account2

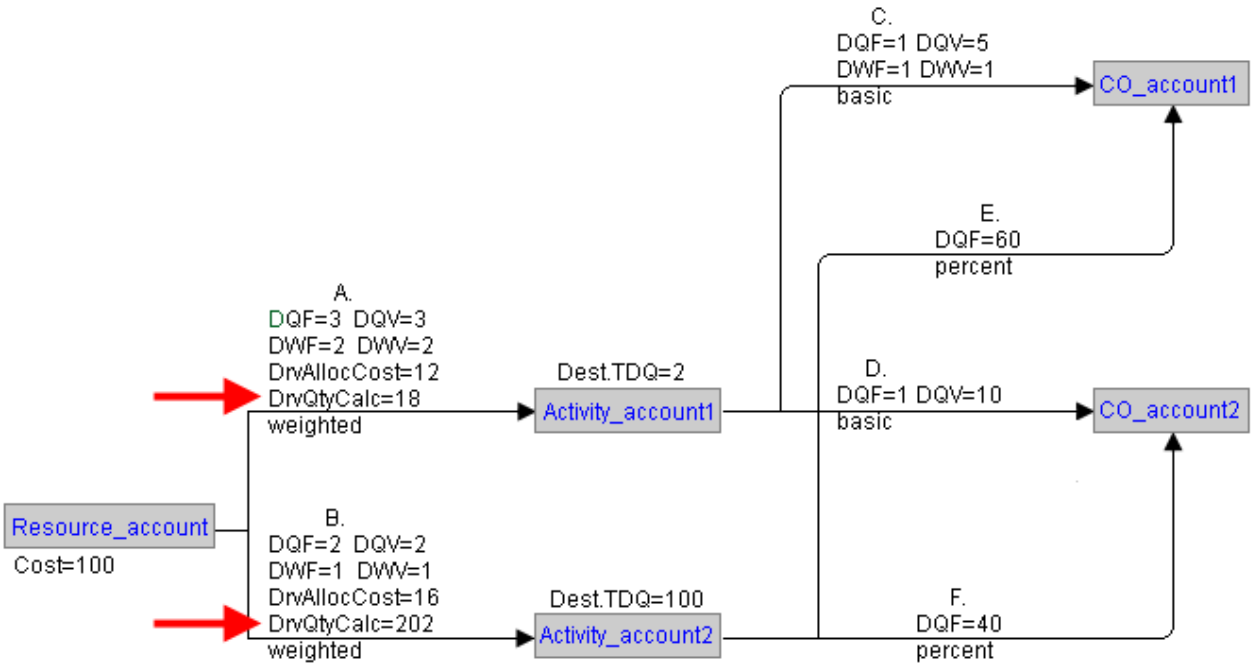
Dest.TDQ for Activity_account2 = 60 + 40

Dest.TDQ for Activity_account2 = 100

Step 3–Calculate DrvQtyCalc for Each Assignment Path

DrvQtyCalc is calculated according to the following formula:

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



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

DrvQtyCalc for Activity_account1 = (3 x 2) + (3 x 2 x 2)

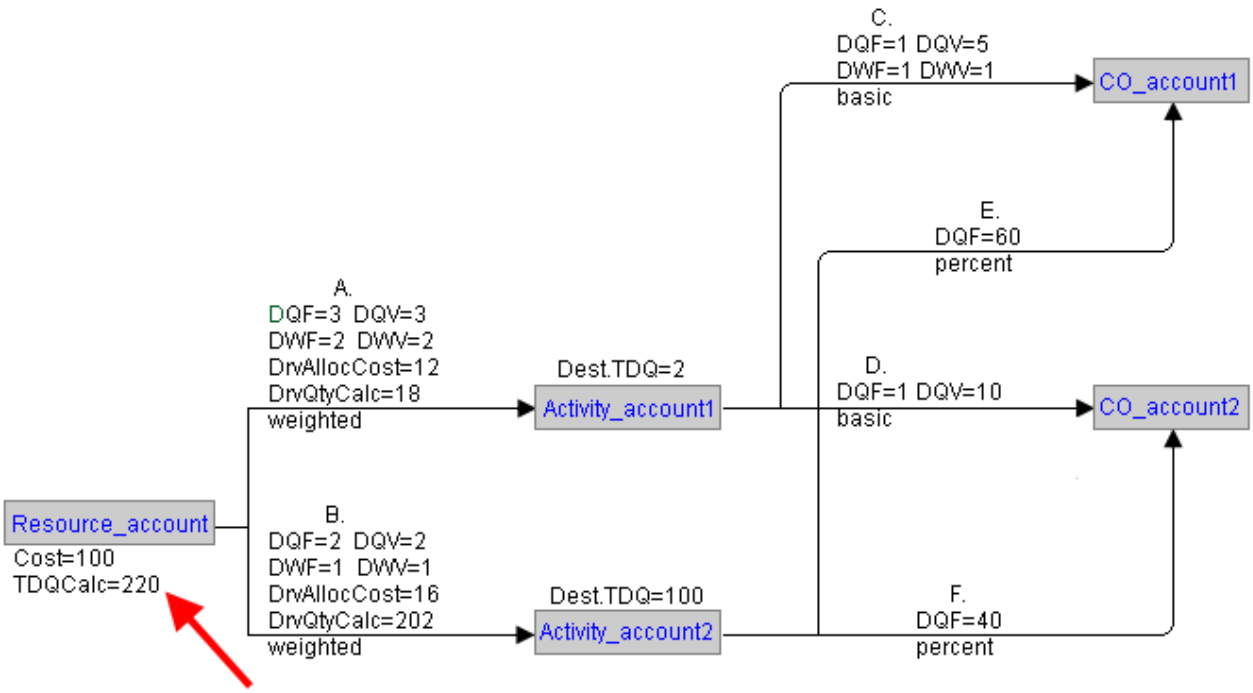
DrvQtyCalc for Activity_account1=18

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

DrvQtyCalc for Activity_account2 = (2 x 1) + (2 x 1 x 100)

DrvQtyCalc for Activity_account2 = 202

Step 4–Calculate TDQCalc as the Sum of all DrvQtyCalc



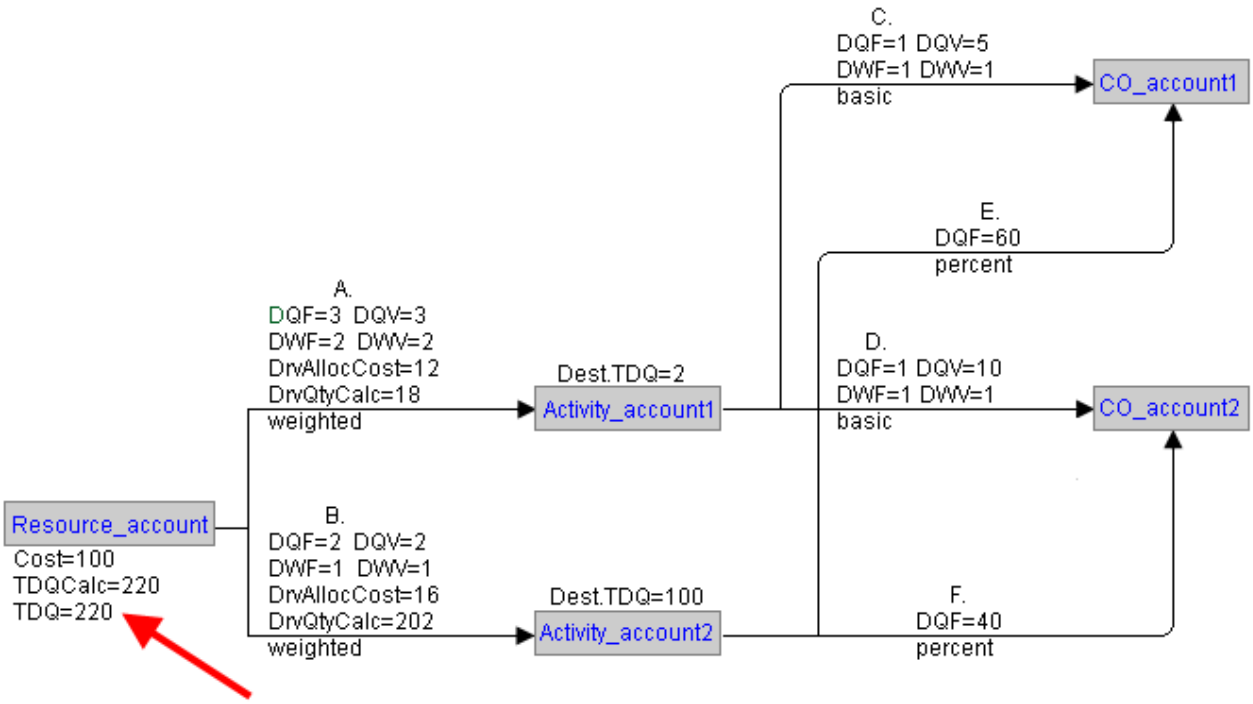
TDQCalc is the sum of DrvQtyCalc for all outgoing assignment paths.

TDQCalc = (DrvQtyCalc for Activity_account1) + (DrvQtyCalc for Activity_account2)

TDQCalc = 18 + 202

TDQCalc=220

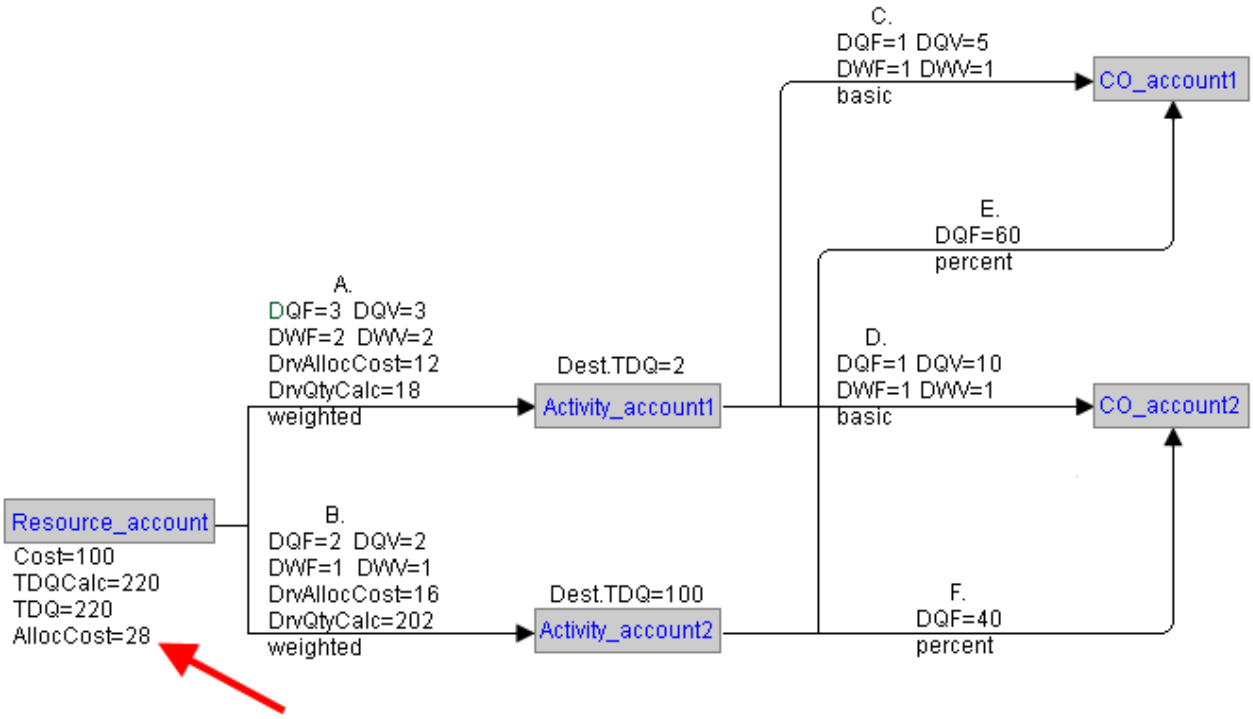
Step 5–Assign TDQCalc to the Source.TDQ



Source.TDQ = TDQCalc

Source.TDQ = 220

Step 6–Calculate Source.Allocated Cost as the Sum of all DrvAllocCost



Source Allocated Cost = Sum of DrvAllocCost for all outgoing assignment paths

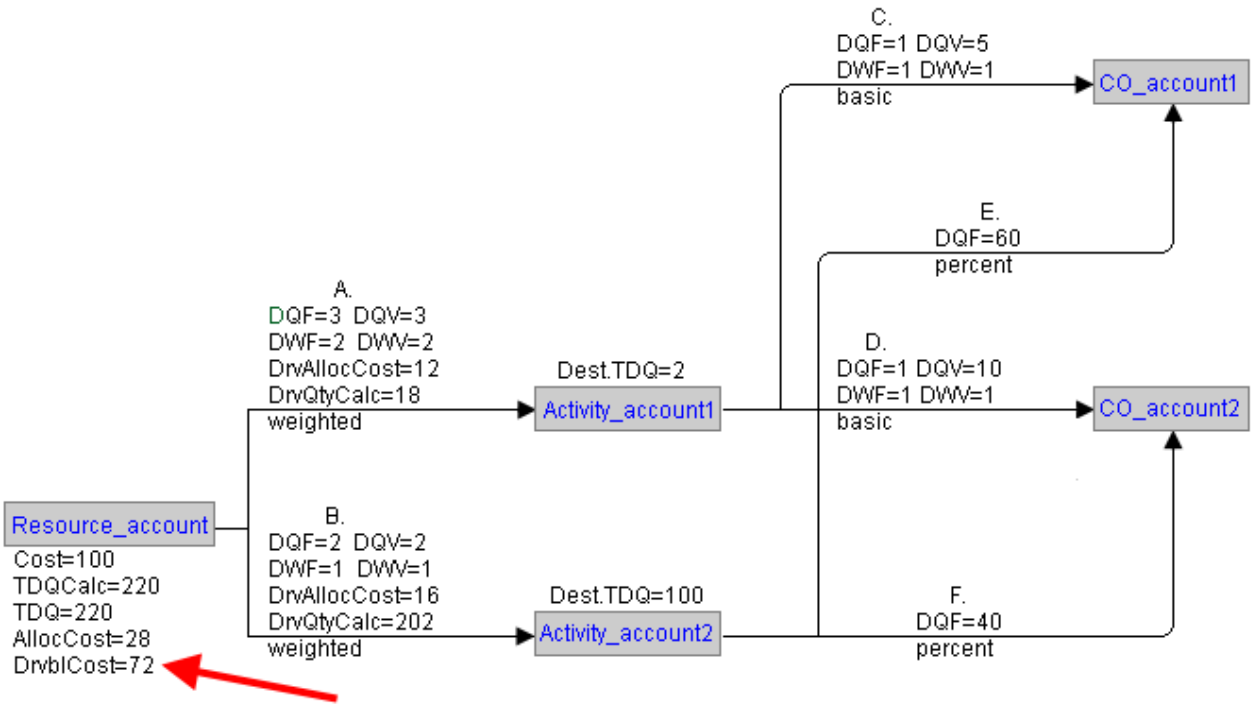
DrvAllocCost of assignment path from Resource_account to Activity_account1 = 12

DrvAllocCost of assignment path from Resource_account to Activity_account2 = 16

Source Allocated Cost = 12 + 16

Source Allocated Cost = 28

Step 7–Calculate DrivableCost as (Cost - Allocated Cost)

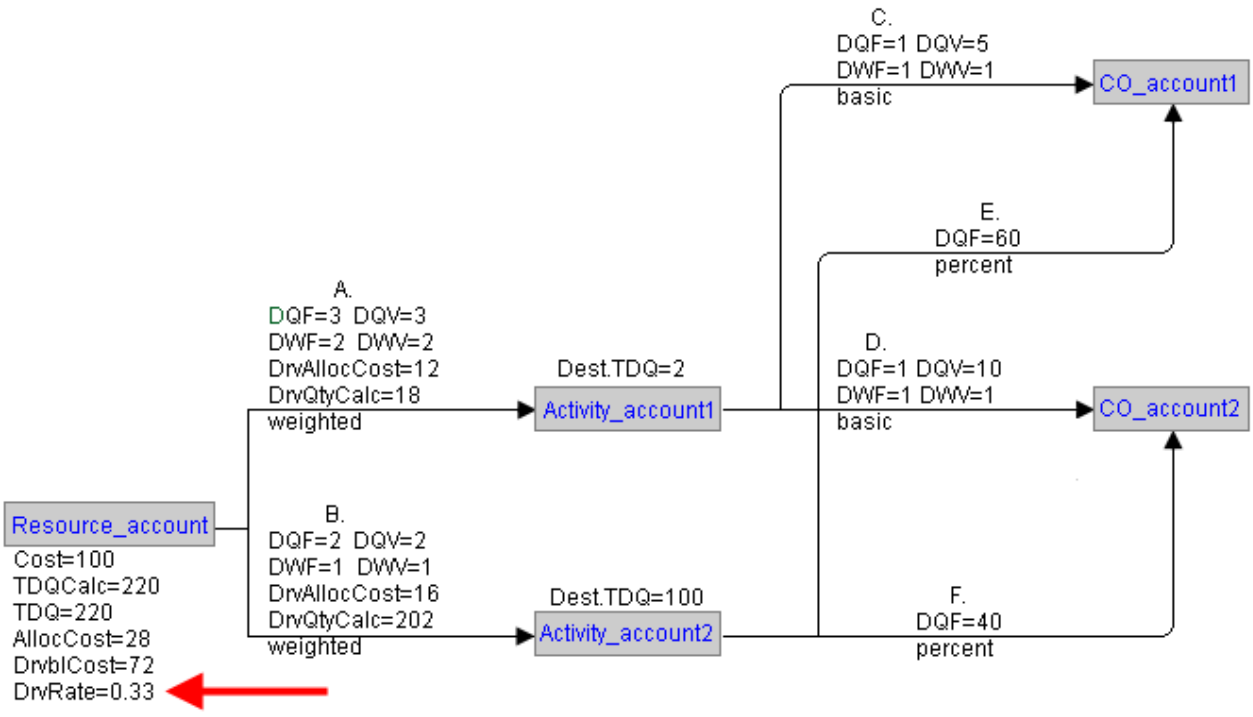


Drivable Cost = Cost - Allocated Cost

Drivable Cost = 100 - 28

Drivable Cost = 72

Step 8–Calculate DriverRate as (DrivableCost / TDQ)

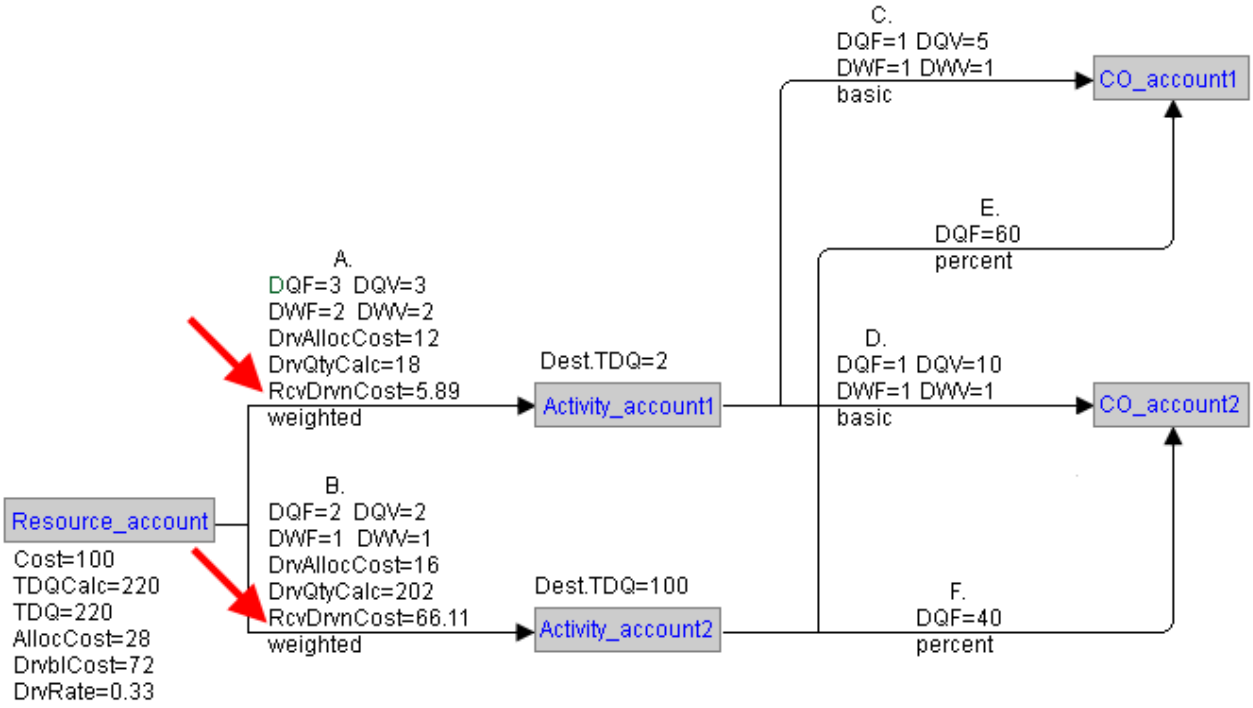


DriverRate = DrivableCost / TDQ

DriverRate = 72 / 220

DriverRate = 0.33 (rounded up)

Step 9–Calculate Received Driven Cost as (DriverRate x DrvQtyCalc) for Each Assignment Path



RcvDrvnCost for Resource_account to Activity_account1

RcvDrvnCost = DriverRate x DrvQtyCalc

RcvDrvnCost = 0.33 x 18

RcvDrvnCost = 5.89

RcvDrvnCost for Resource_account to Activity_account2

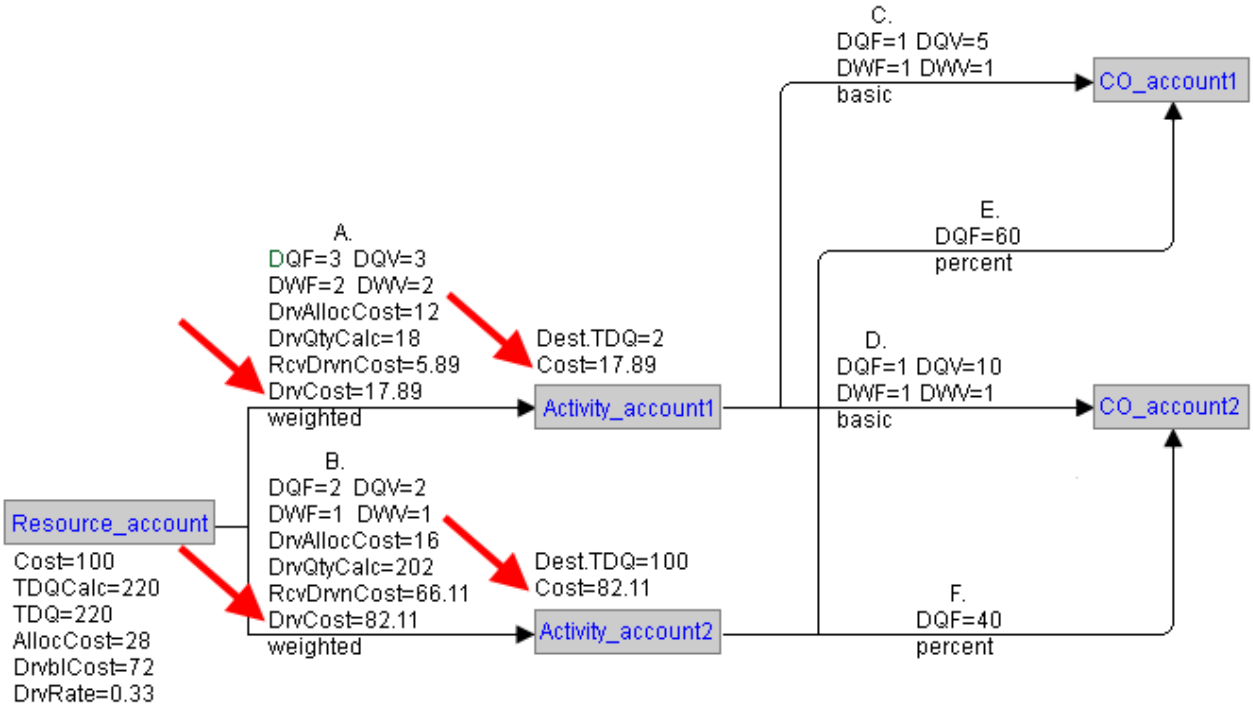
RcvDrvnCost = DriverRate x DrvQtyCalc

RcvDrvnCost = 0.33 x 202

RcvDrvnCost = 66.11

Step 10–Calculate DriverCost as (Received Driven Cost + DrvAllocCost)

Do this for each assignment path.



Calculate Driver Cost for assignment path from Resource_account to Activity_account1

DriverCost = Received Driven Cost + DrvAllocCost

DriverCost = 5.89 + 12

DriverCost = 17.89

So, Cost=17.89 for Activity_account1

Calculate Driver Cost for assignment path from Resource_account to Activity_account2

DriverCost = Received Driven Cost + DrvAllocCost

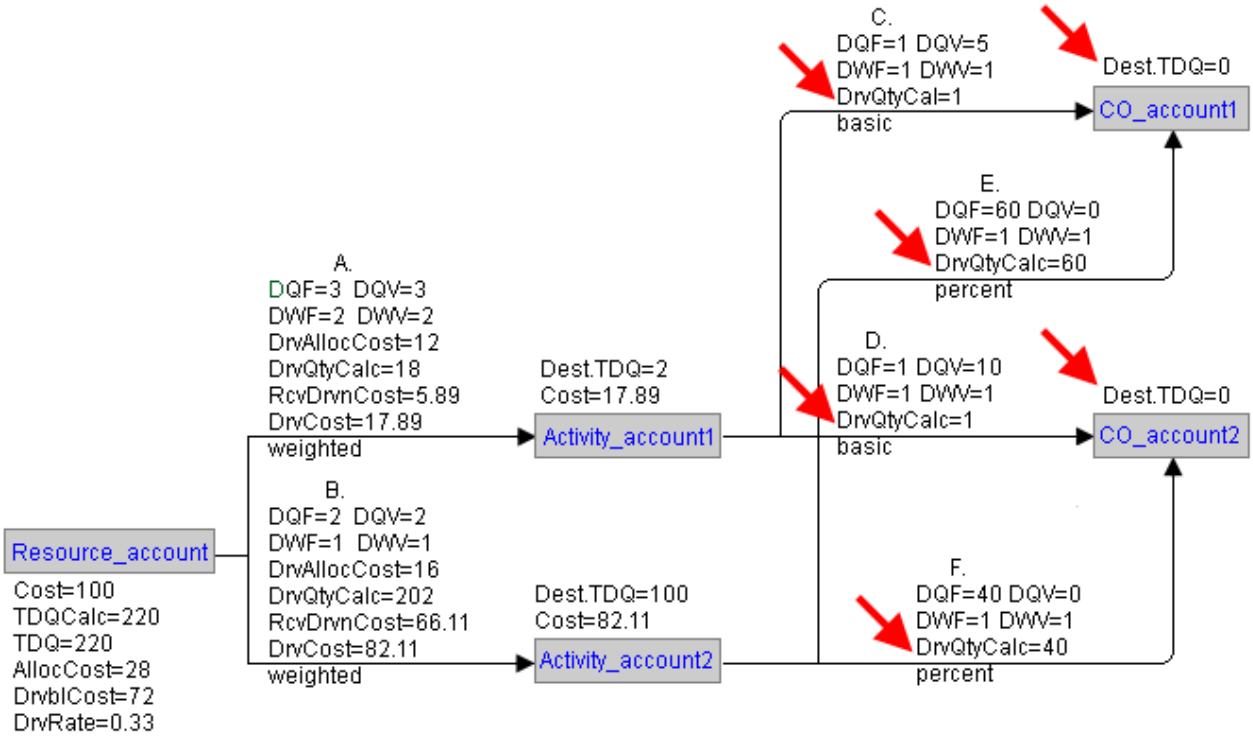
DriverCost = 66.11 + 16

DriverCost = 82.11

So, Cost=82.11 for Activity_account2

Step 11–Calculate Dest.TDQ for Cost Object Accounts and Calculate DrvQtyCalc

Next the system calculates the cost flows from the Activity accounts to the Cost Object accounts.The first step is to calculate Dest.TDQ for the Cost Object accounts and then calculate DrvQtyCalc.



Dest.TDQ for CO_account1 = 0 because there are no costs flowing out of this Cost Object account.

Dest.TDQ for CO_account2 = 0 because there are no costs flowing out of this Cost Object account either.

Note: The Percentage driver that flows costs from Activity_account2 does not allow variable driver quantities. So, the value of its DQV is null, and for calculations a null is counted as zero (DQV=0).

Note: Neither the Percentage driver nor the Basic driver that flow costs from Activity_account2 is a weighted driver. For non-weighted drivers, DWF and DWV default to 1.

Now, remember that the general formula for calculating DrvQtyCalc is the following:

```
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)

DrvQtyCalc from Activity_account1 to CO_account1=1

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

DrvQtyCalc from Activity_account1 to CO_account2 = (1 x 1) + (10 x 1 x 0)

DrvQtyCalc from Activity_account1 to CO_account2=1

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

DrvQtyCalc from Activity_account2 to CO_account1 = (60 x 1) + (0 x 1 x 0)

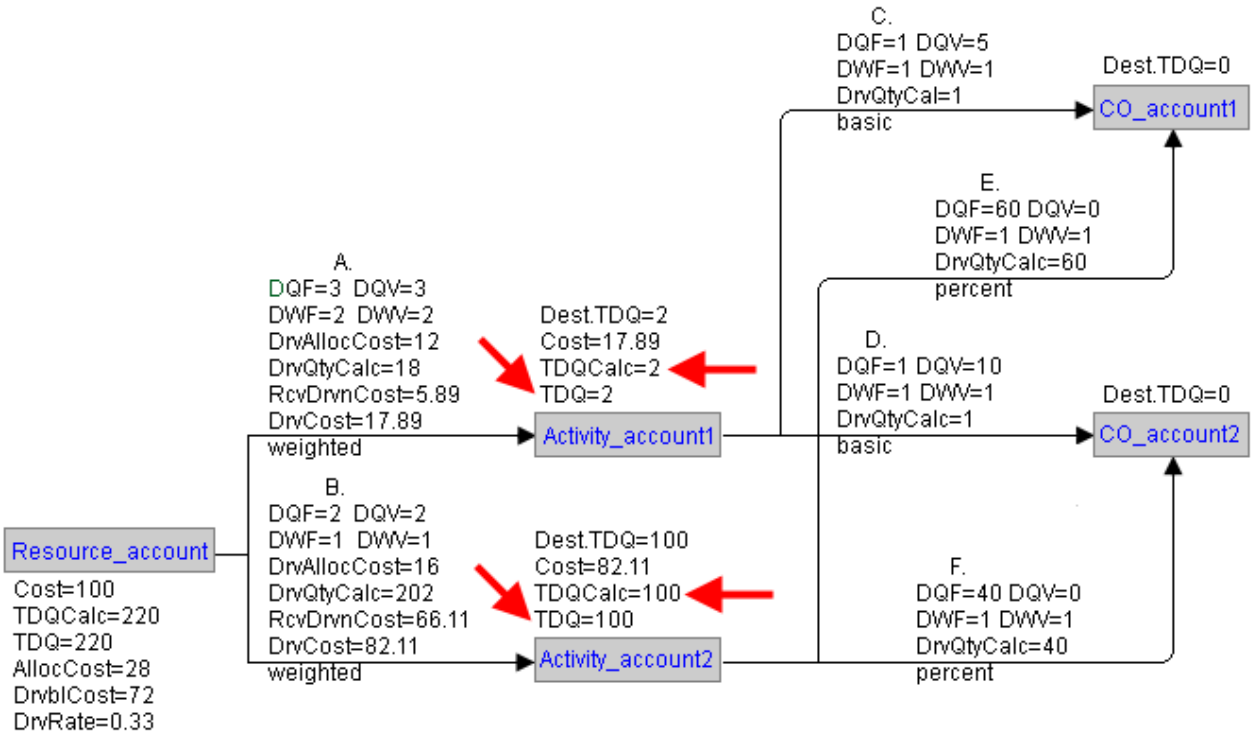
DrvQtyCalc from Activity_account2 to CO_account1=60

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

DrvQtyCalc from Activity_account2 to CO_account2 = (40 x 1) + (0 x 1 x 0)

DrvQtyCalc from Activity_account2 to CO_account2=40

Step 12–Calculate TDQCalc and TDQ



TDQCalc is the sum of all DrvQtyCalc

TDQCalc for Activity_account1 = 1 + 1

TDQCalc for Activity_account1 = 2

TDQCalc for Activity_account2 = 60 + 40

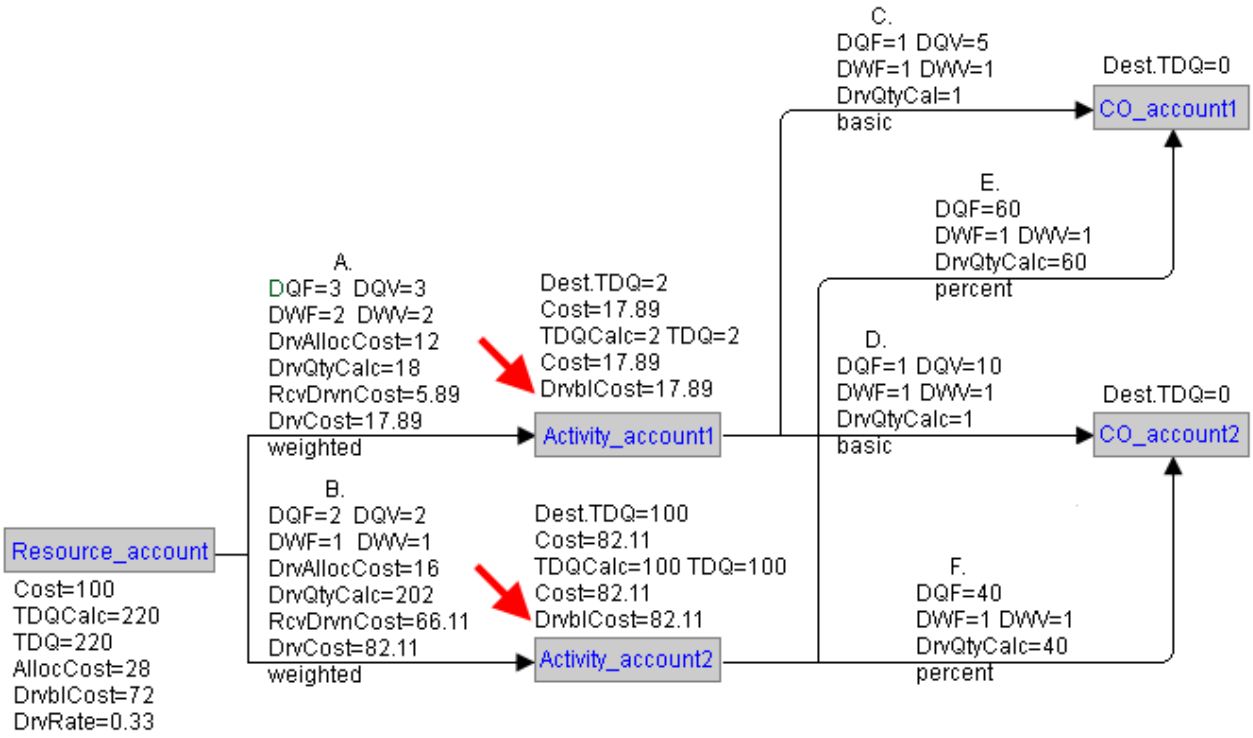
TDQCalc for Activity_account2 = 100

TDQCalc is assigned to Source.TDQ

Source.TDQ for Activity_account1=2

Source.TDQ for Activity_account2=100

Step 13–DrivableCost = Cost - Allocated Cost



DrivableCost = Cost - Allocated Cost

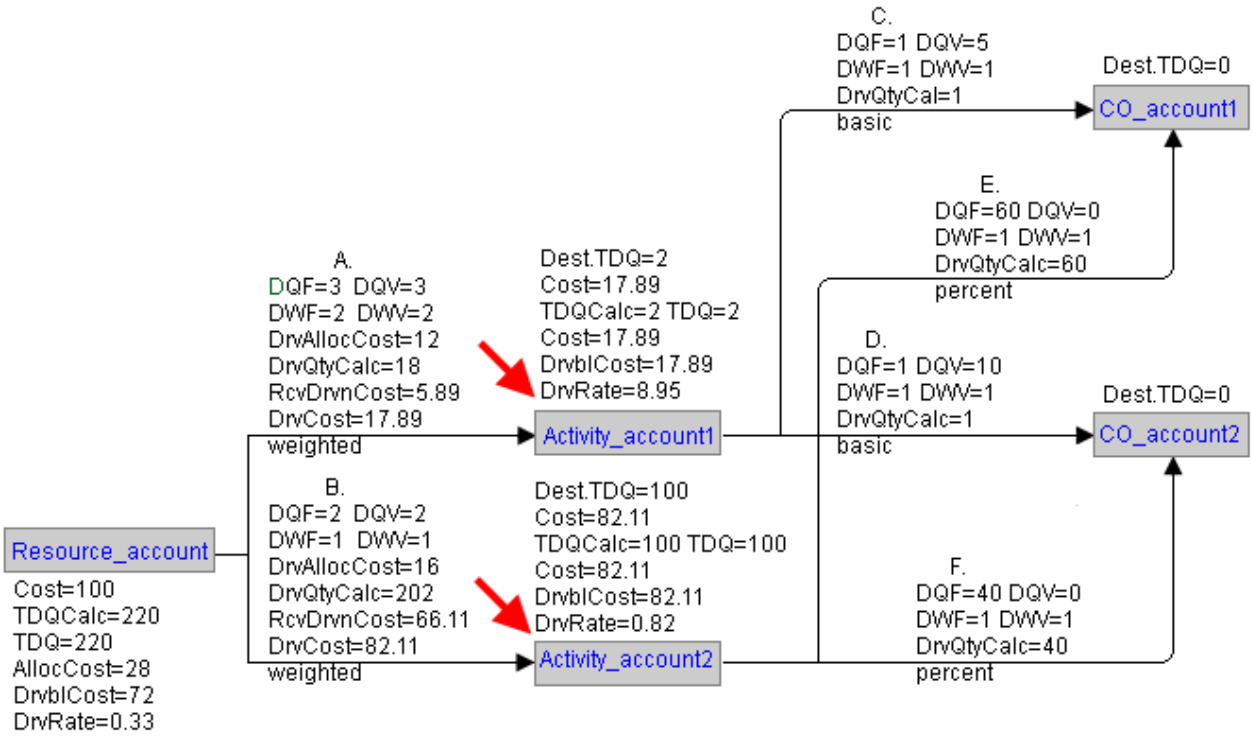
DrivableCost for Activity_account1 = 17.89 - 0

DrivableCost for Activity_account1 = 17.89

DrivableCost for Activity_account2 = 82.11 - 0

DrivableCost for Activity_account2 = 82.11

Step 14–DriverRate = DrivableCost / TDQ



DriverRate = DrivableCost / TDQ

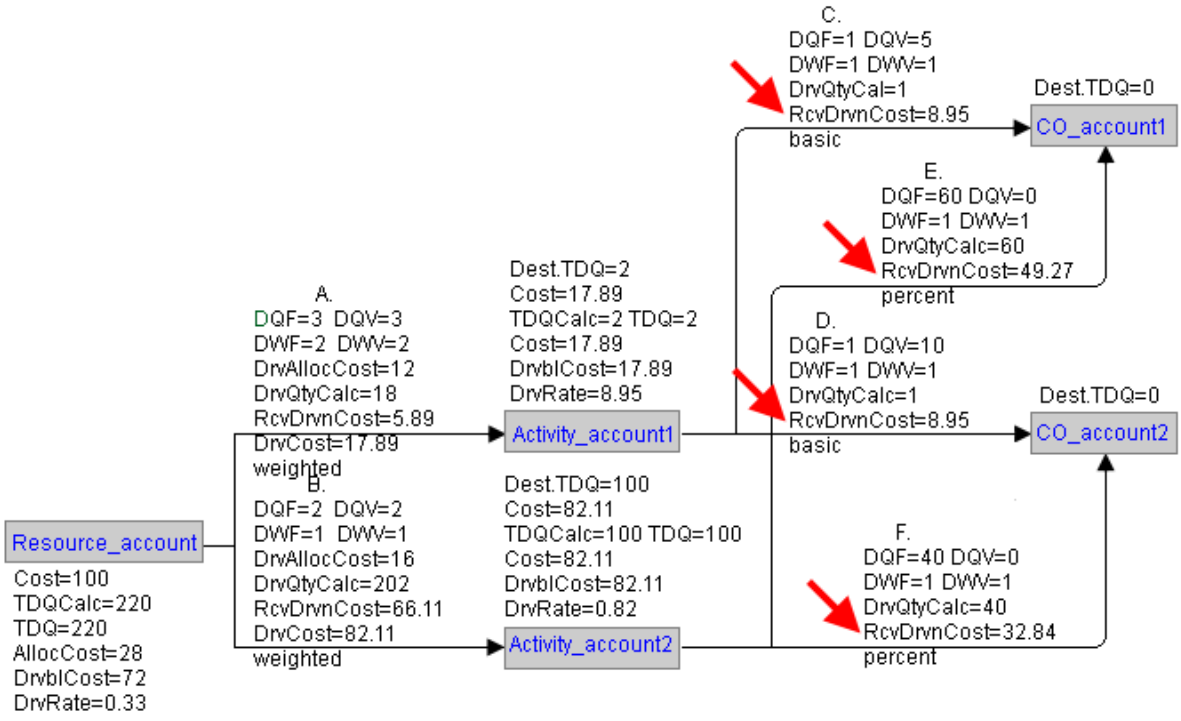
DriverRate for Activity_account1= 17.89 / 2

DriverRate for Activity_account1 = 8.95 (rounded up)

DriverRate for Activity_account2 = 82.11 / 100

DriverRate for Activity_account2 = .82 (rounded down)

Step 15–Received Driven Cost = DriverRate x DrvQtyCalc



Received Driven Cost = DriverRate x DrvQtyCalc

Received Driven Cost for Activity_account1 to CO_account1 = 8.95 x 1

Received Driven Cost for Activity_account1 to CO_account1 = 8.95

Received Driven Cost for Activity_account1 to CO_account2 = 8.95 x 1

Received Driven Cost for Activity_account1 to CO_account2 = 8.95

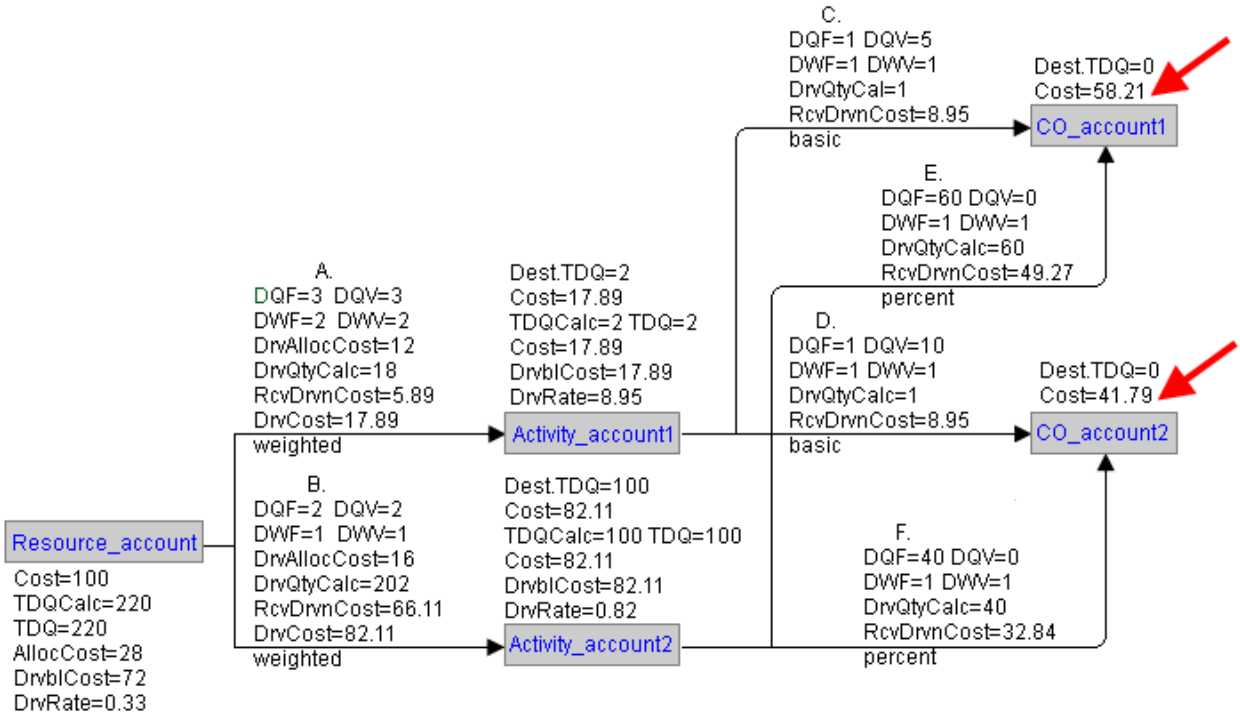
Received Driven Cost for Activity_account2 to CO_account1 = .82 x 60

Received Driven Cost for Activity_account2 to CO_account1 = 49.27

Received Driven Cost for Activity_account2 to CO_account2 = .82 x 40

Received Driven Cost for Activity_account2 to CO_account2 = 32.84

Step 16–Calculate Cost for Cost Object Accounts



Cost for destination account = Sum of individual Received Driven Costs

Cost for Activity_account1= 8.95 + 49.27

Cost for Activity_account1 = 58.21

Cost for Activity_account2 =8.95 + 32.84

Cost for Activity_account2 = 41.79

Chapter 32

Working with OLAP

Introduction

This chapter contains task-based information about working with OLAP. For more conceptual information, see Chapter 13, “Analyzing Model Data with OLAP,” on page 89.

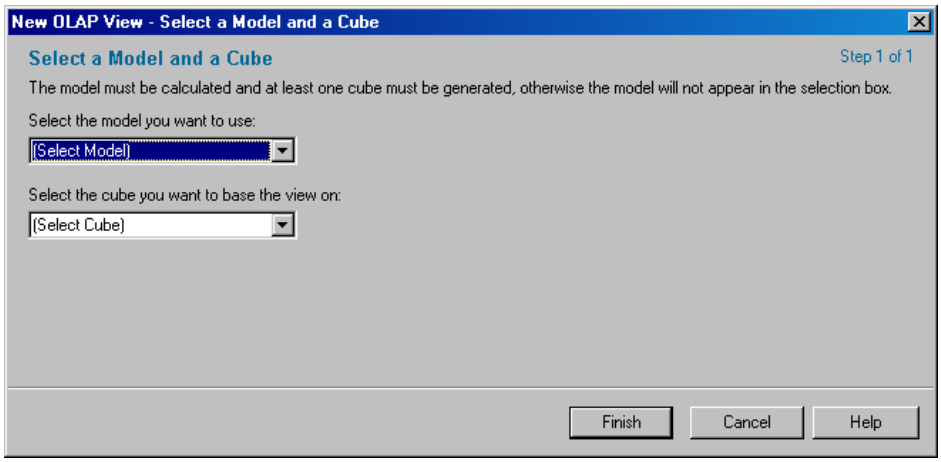
Start OLAP Mode

Click the Analysis tab in the Navigation Pane.

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.

Save an OLAP View

1. On the OLAP page, select **OLAP** ⇨ **Save View**.

   The Save OLAP View As dialog box appears.
2. Type the **Name**.

   The name must follow the naming conventions. For information, see the Help.
3. Type the **Description**.

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**.

Open an OLAP View without an OLAP View Already Open

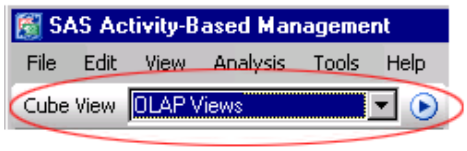
If you start OLAP mode without an OLAP view open, you can open an OLAP view from the Analysis workspace. The list of **Folders** and **OLAP Views** corresponds to the OLAP branch of the server area in Workspace Manager.

1. From the list of **Folders**, select a folder.
2. From the list of **OLAP Views**, click a view.

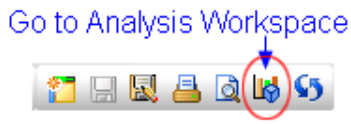
Open an OLAP View with an OLAP View Already Open

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.



Delete an OLAP View

1. Go to the Analysis workspace.
2. Select an OLAP view.
3. Click **Delete**.

Note: Depending on your permissions, you may not have the ability to delete an OLAP view..

Chapter 33

Working with Importing and Exporting Data

Introduction

This chapter contains task-based information about working with importing data and exporting data. For more conceptual information, see Chapter 17, "Importing Data and Exporting Data," on page 107.

Import Model Data from a Database with the Import Data Wizard

Before attempting to import data from a database, see "Connect to a Database" on page 278, which provides information about preparing to connect to the most common databases.

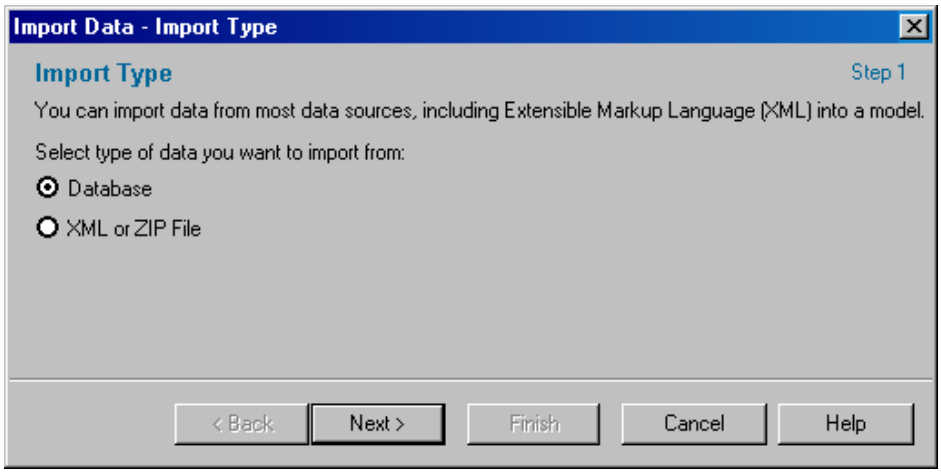
Note: You can perform this task without first opening a model.

1. Create a database or a database view that matches the data schema.

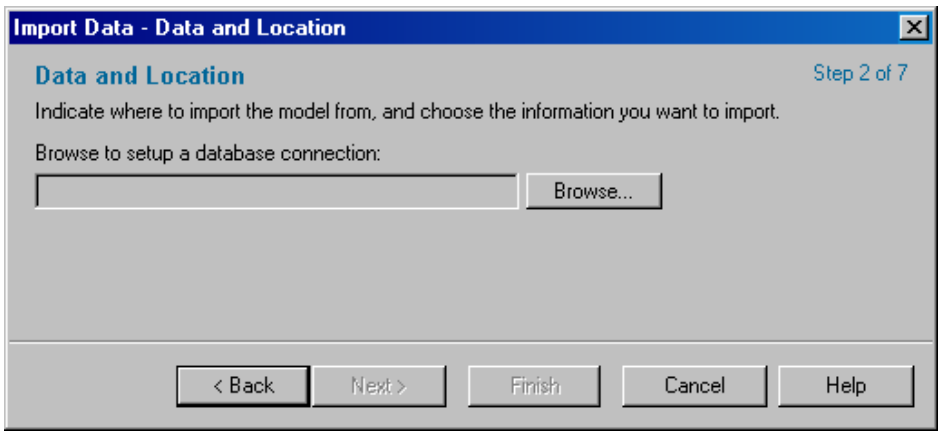
For information about the data schema, see the Help.

2. In SAS Activity-Based Management, verify or create the period/scenario associations.
3. Select **File** ⇨ **Import** ⇨ **Model Data**.

   The Import Data Wizard appears.



4. Select the **Database** option.
5. Click **Next**.



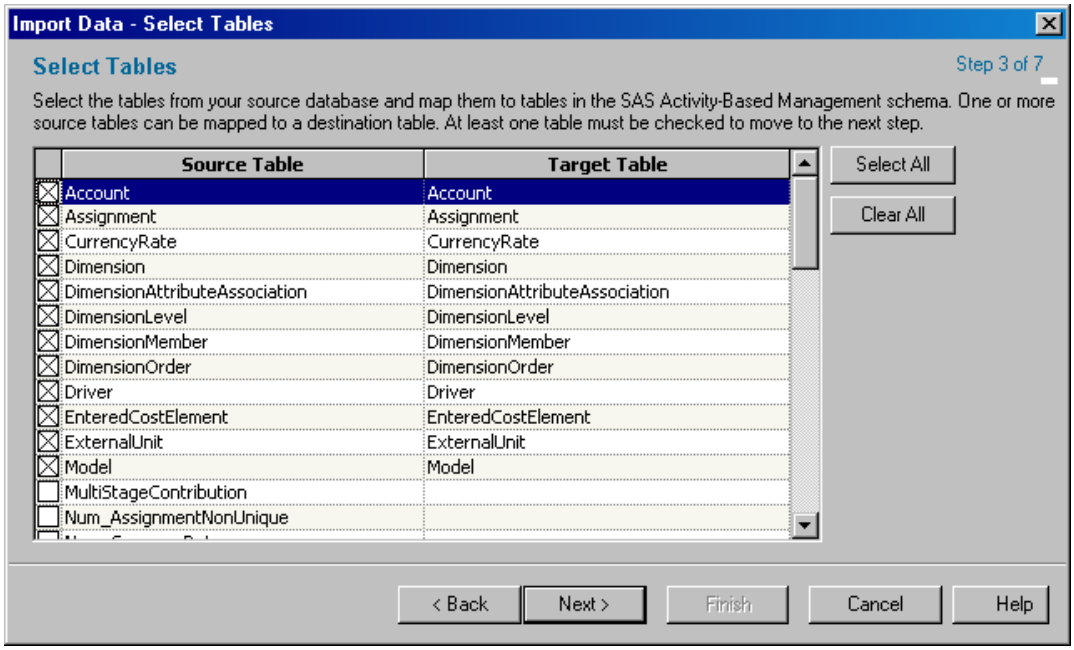
6. Click 

.

   The Data Link Properties dialog box appears. For information about this dialog box, see the Microsoft Help. For information about the task of creating a connection string to a specific database, see the appropriate section at the end of this chapter.

   Next you will specify which tables to import. You will specify which imported tables (source tables) map to which tables in SAS Activity-Based Management.

7. Click **Next**.

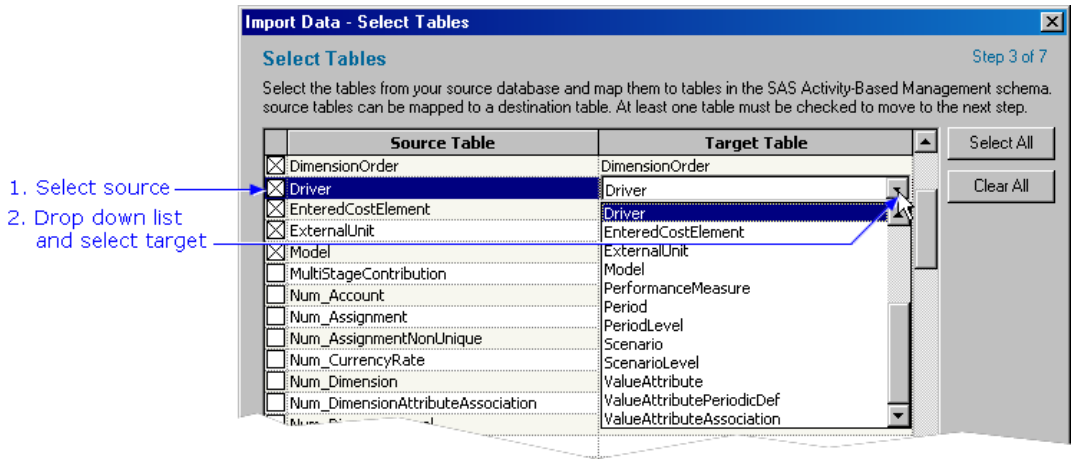


If you have named the source tables with the same names that are used in SAS Activity-Based Management, the wizard automatically creates the mappings and selects the tables to be imported. If you have not used the same names, then you must specify which source table maps to which SAS Activity-Based Management table.

8. To select a table to import, select the check box to the left of the table name in the **Source Table** column.

   You can select as many tables as needed.

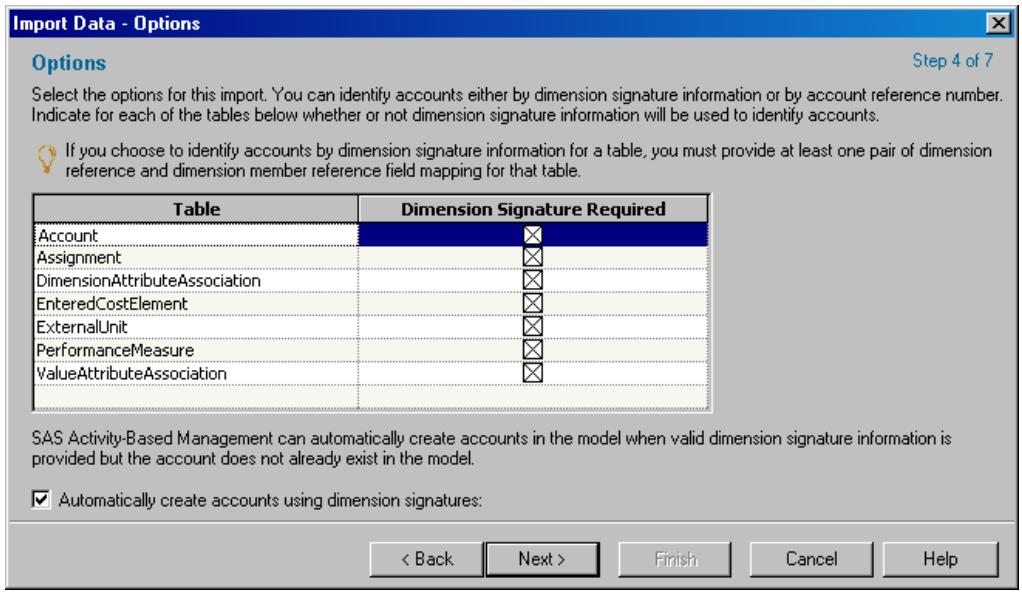
9. To map a **Source Table**, click in the **Target Table** column to the right of the source table, and select a SAS Activity-Based Management table from the drop-down list.



10. Repeat steps 8 and 9 to map every table that you want to import.

    Next, you will specify how accounts will be 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.

11. Click **Next**.



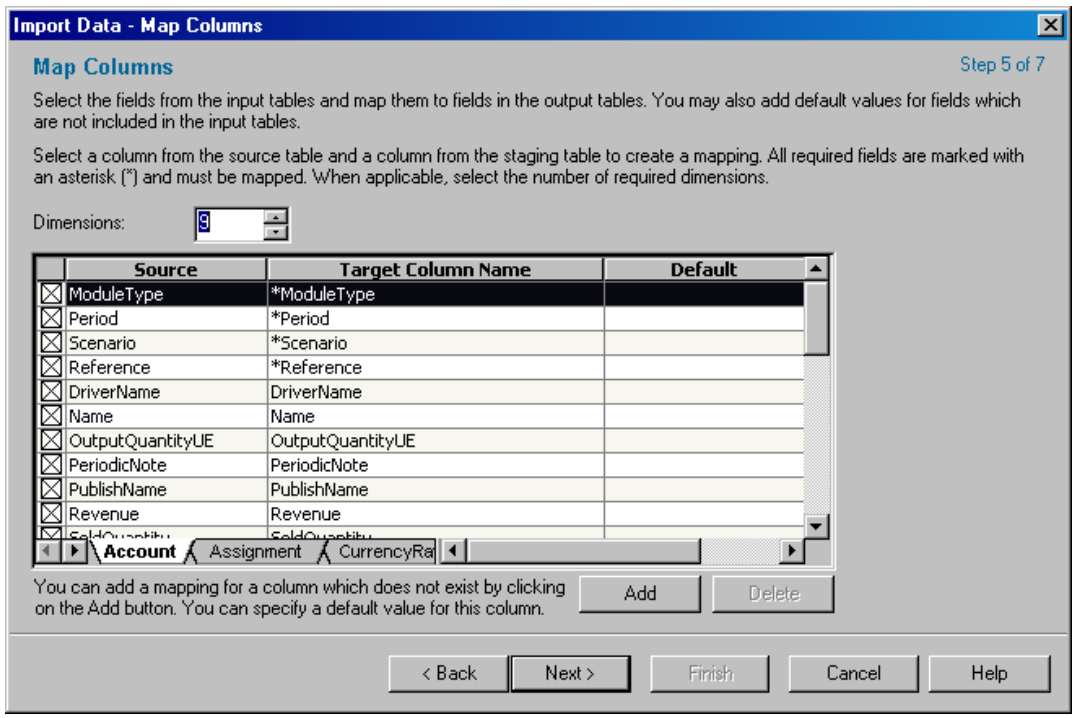
12. To identify accounts by the dimension signature, select the check box in the **Dimension Signature Required** column to the right of each source **Table**.

    If you clear the check box, accounts will be identified by their references.

13. To have the wizard automatically create accounts for all valid dimension signatures, select the **Automatically create accounts using dimension signatures** option.

    Next, you will specify which table columns to import. You will specify which imported table columns map to which table columns in SAS Activity-Based Management.

14. Click **Next**.



If you have named the source table columns with the same column names that are used in SAS Activity-Based 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 Activity-Based Management table column.

15. 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 (*).

16. If you chose in the last 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 Activity-Based 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.

17. 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 Activity-Based 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.

18. If a source table does not contain a column that you want to create in the model, and you want to assign a value to this column, do the following:

    a. Click **Add**.

       A new row appears.

    b. Click in the **Source** column, and select a column name from the drop-down list.

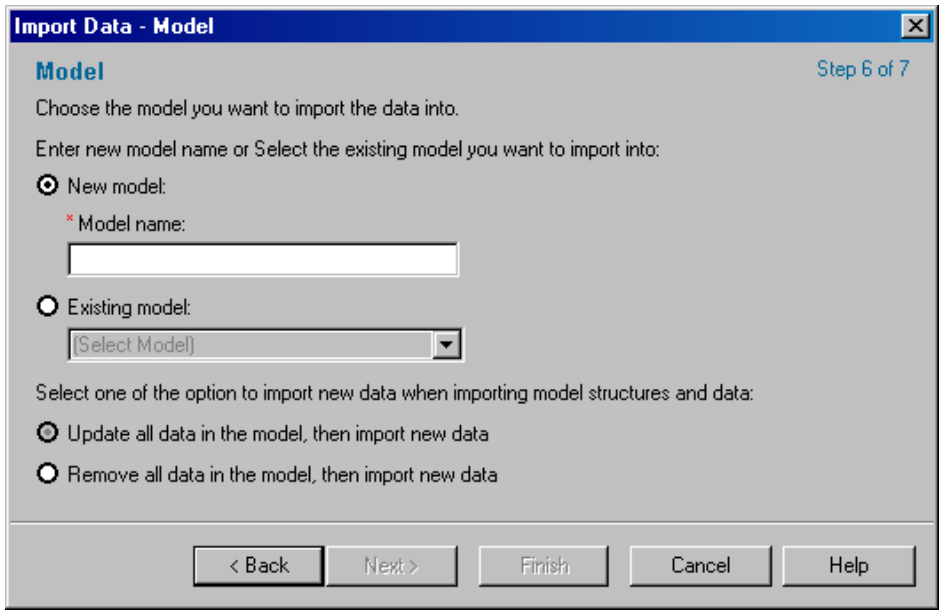
    c. Type a **Default** value.

       For example, if the source table does not contain a column to map to the Period column in SAS Activity-Based Management, and you want to create a period named MyPeriod in the model, then add a column named <None>. Map this new column to the **Target Column Name** Period, and specify the **Default** value as MyPeriod.

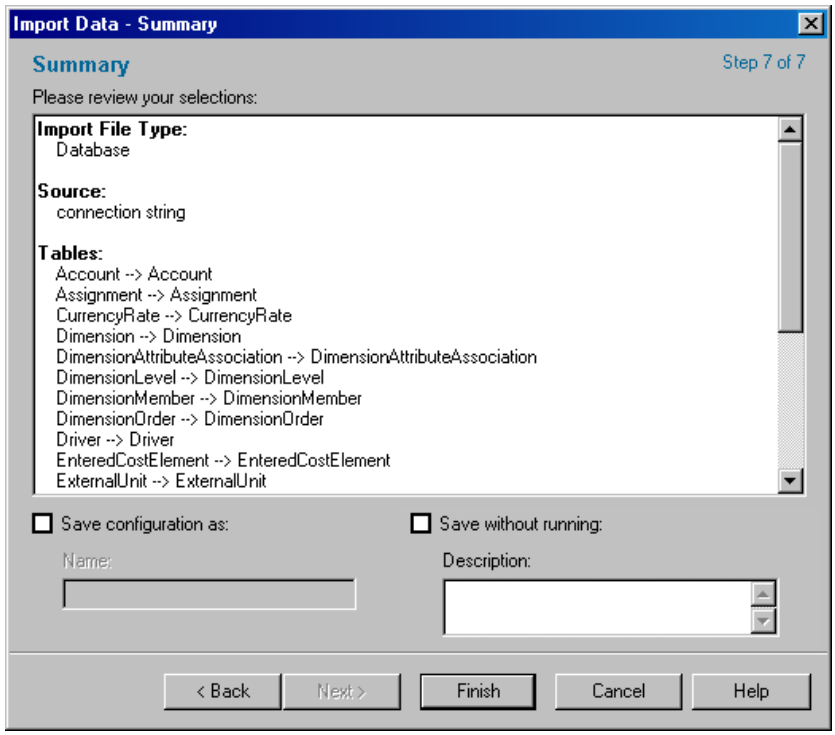
19. Repeat steps 15 through 18 for every table column that you want to import.

20. Click the appropriate tab at the bottom of the grid to map the columns for another table.

21. Click **Next**.



22. 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**.
23. If you want to import the database into an existing model, do the following:
    a. Select the **Existing model** option.
    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. To remove all existing data, select the **Remove all data in the model, then import new data** option.
24. Click **Next**.



25. Review the import summary.
26. 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.
27. 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**.
28. To rename an existing import configuration, do the following:
    a. Select the **Update existing configuration** option.
    b. Type the new **Name**.
    c. Type the **Description**.
29. Click **Finish**.

Import Model Data from an XML File with the Import Data Wizard

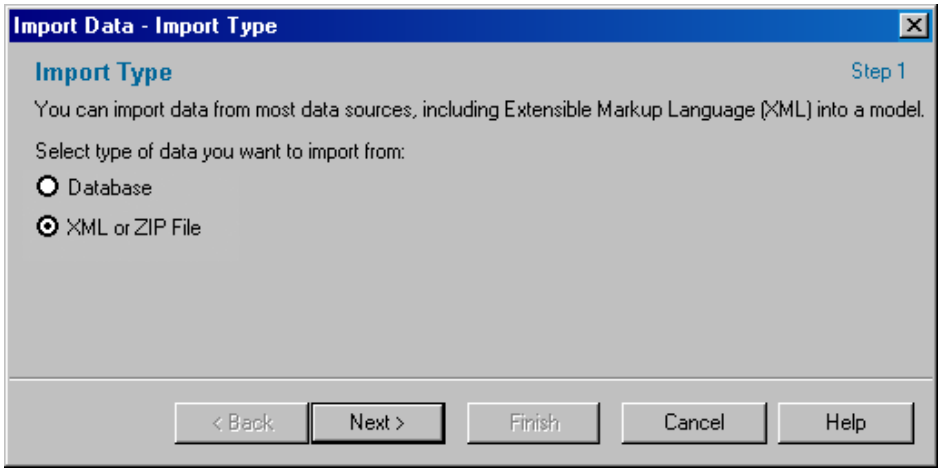
Note: You can perform this task without first opening a model.

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

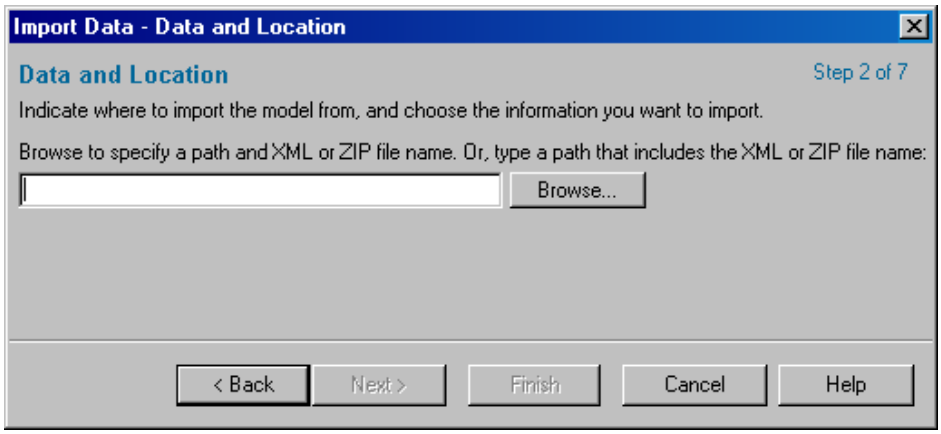
import model data from an XML file to incrementally update an existing model or to combine several models into a single model. Whatever data is contained by an existing model is removed and replaced by the model data in the imported XML file.

1. Select **File** ⇨ **Import** ⇨ **Model Data**.

   The Import Data Wizard appears.

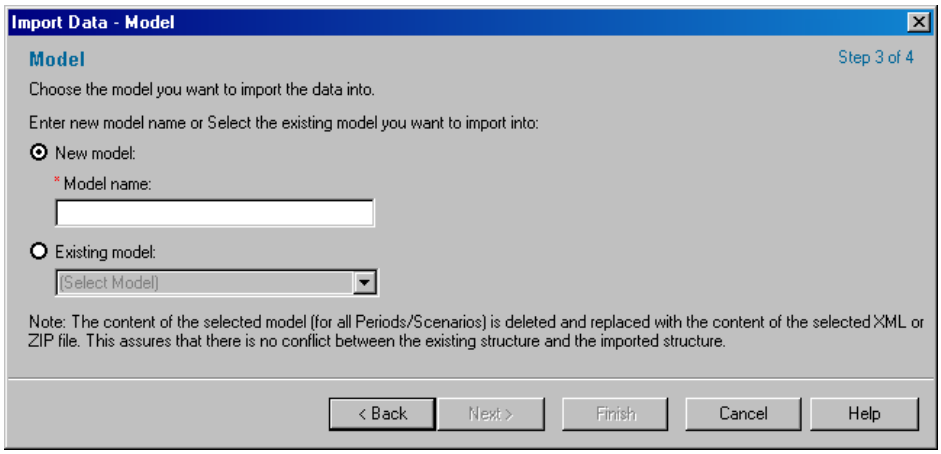


2. Select the **XML File** option.
3. Click **Next**.



4. Type the absolute pathname to the XML file. Or, click 

.
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.

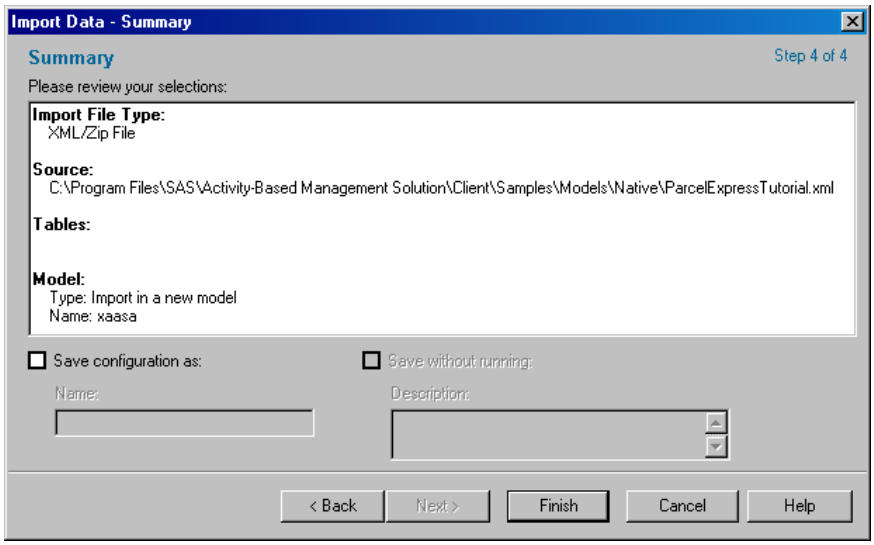
   b. Type the **Model name**.

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.

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. To rename an existing import configuration, do the following:
    a. Select the **Update existing configuration** option.
    b. Type the new **Name**.
    c. Type the **Description**.
13. Click **Finish**.

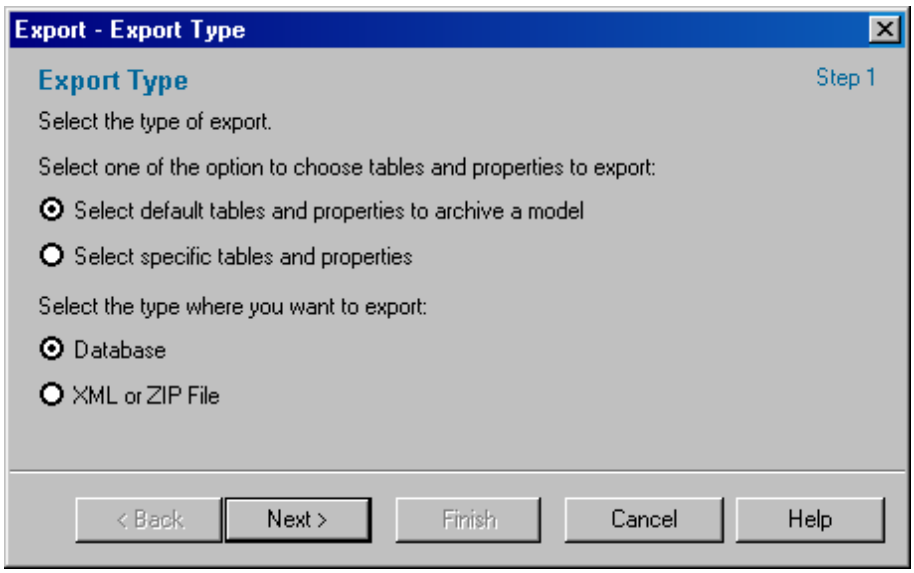
Archive a Model to a Database with the Export Wizard

Before attempting to archive a model to a database, see "Connect to a Database" on page 278, which provides information about preparing to connect to the most common databases.

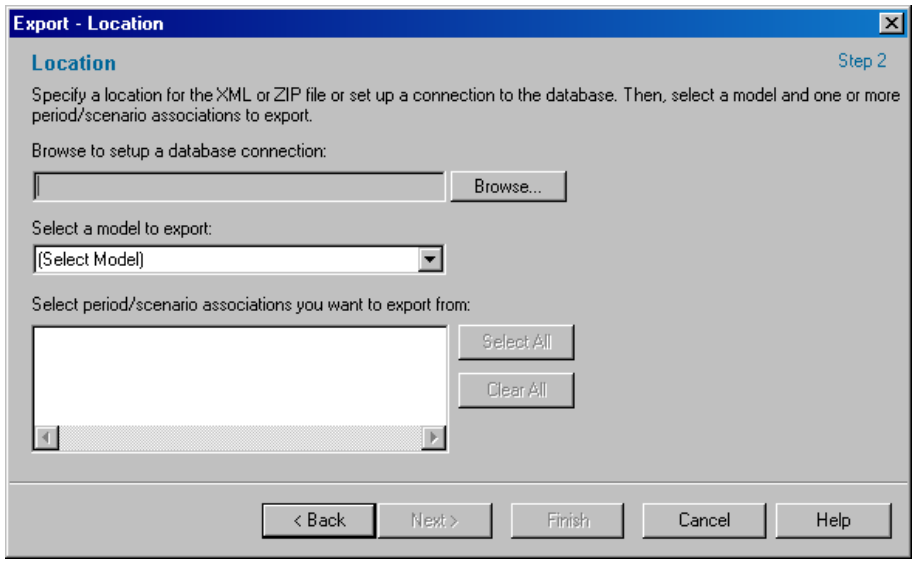
Note: You can perform this task without first opening a model.

1. Verify that the model and an empty database are 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 **Database** option.
5. Click **Next**.

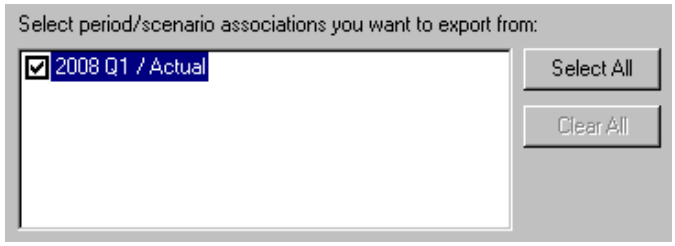


6. Click Browse....

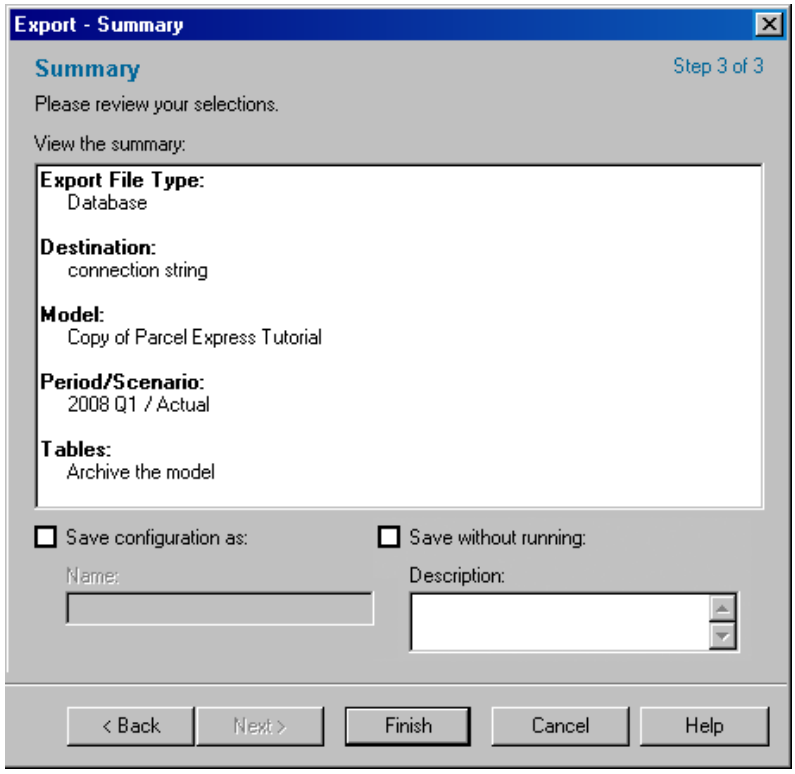
   The Data Link Properties dialog box appears. For information about this dialog box, see the Microsoft Help. For information about the task of creating a connection string to a specific database, see the appropriate section at the end of this chapter.

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. To rename an existing export configuration, do the following:
    a. Select the **Update existing configuration** option.
    b. Type the new **Name**.
    c. Type the **Description**.
14. Click **Finish**.

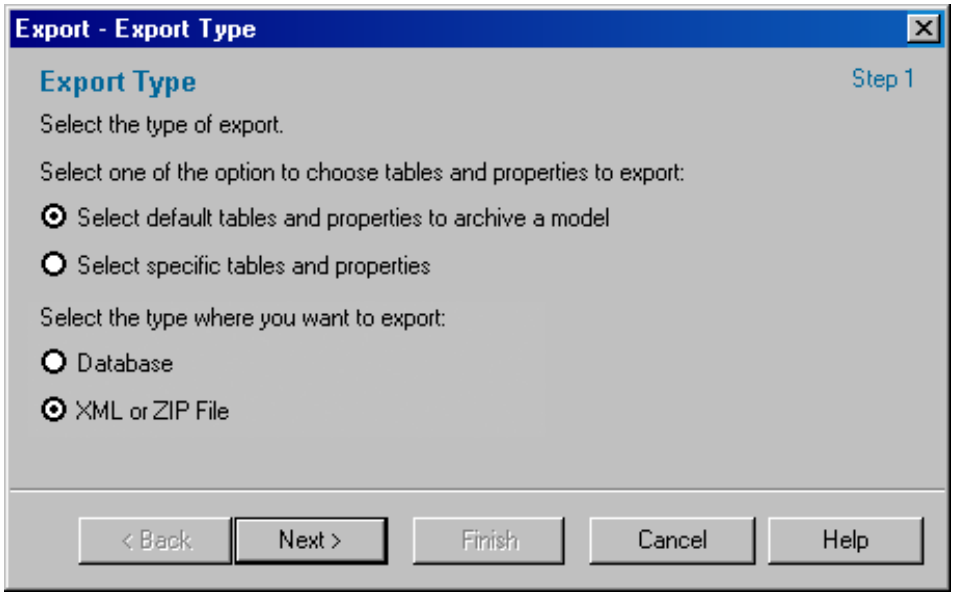
Archive a Model to an XML File with the Export Wizard

Note: You can perform this task without first opening a model.

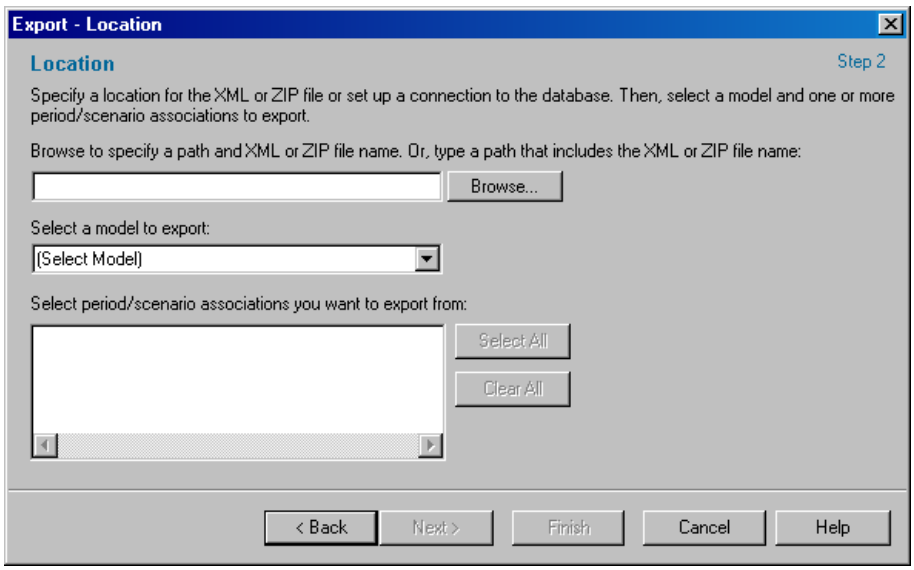
1. Verify that the model is ready.

2. Select **File** ⇨ **Export Model Data**.

   The Export Wizard appears.

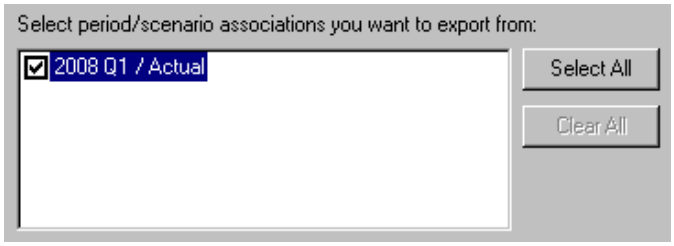


3. Select the **Select default tables and properties to archive a model** option.
4. Select the **XML File** option.
5. Click **Next**.

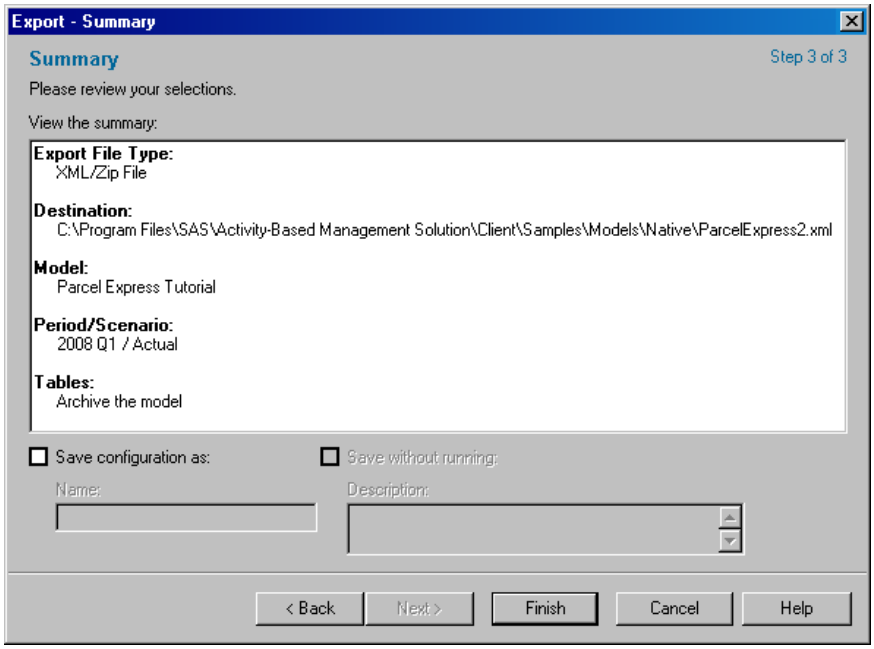


6. Type the absolute pathname to the XML file. Or, click 

.
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. To rename an existing export configuration, do the following:
    a. Select the **Update existing configuration** option.
    b. Type the new **Name**.
    c. Type the **Description**.
14. Click **Finish**.

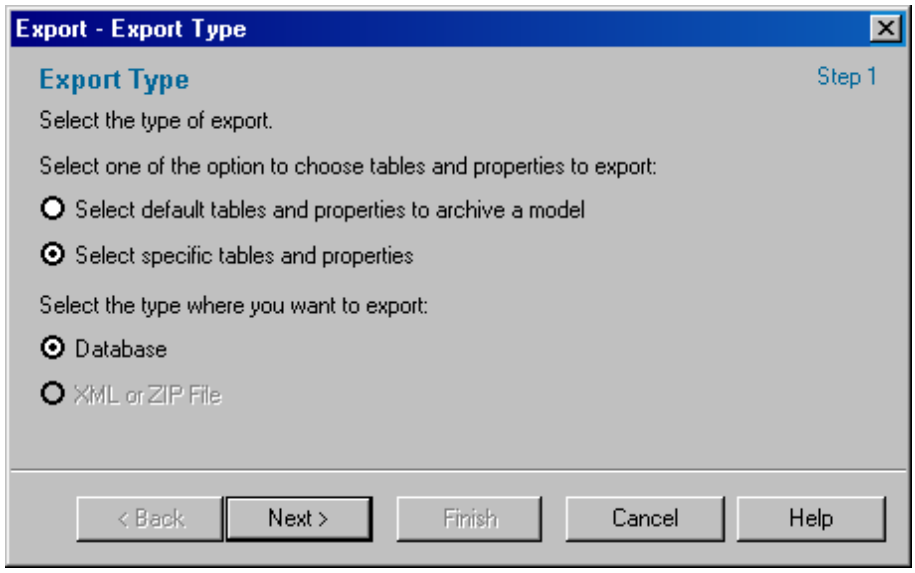
Export Model Data to a Database with the Export Wizard

Before attempting to export data to a database, see "Connect to a Database" on page 278, which provides information about preparing to connect to the most common databases.

Note: You can perform this task without first opening a model.

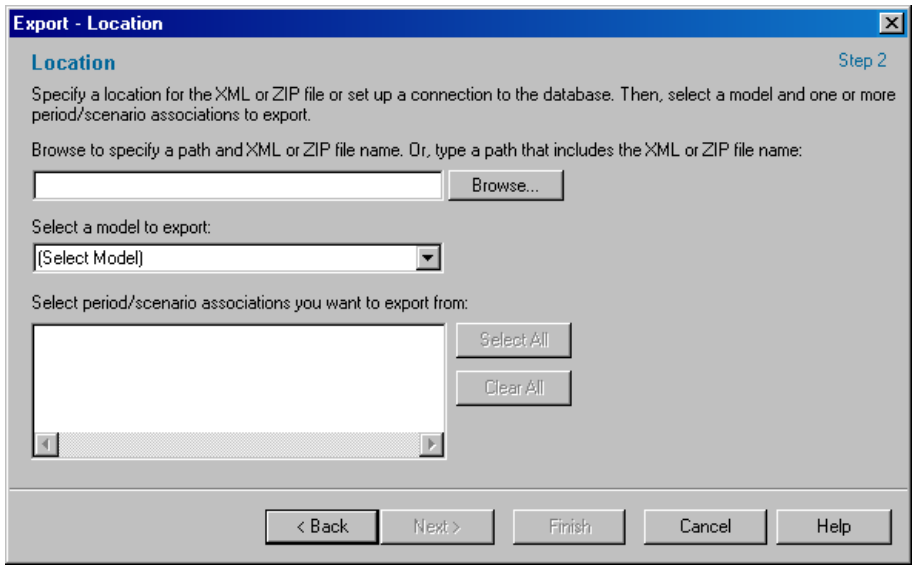
1. Verify that the model is ready.
2. Select **File** ⇨ **Export Model Data**.

   The Export Wizard appears.



3. Select the **Select specific tables and properties** option.

   Notice that the only export type available is **Database**.
4. Click **Next**.

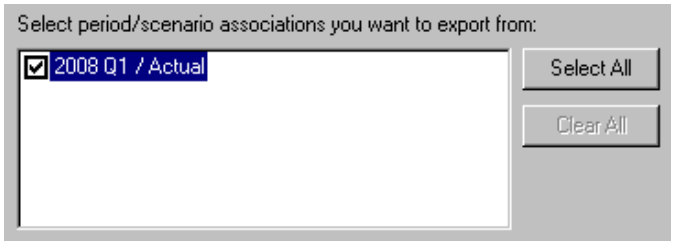


5. Click Browse...

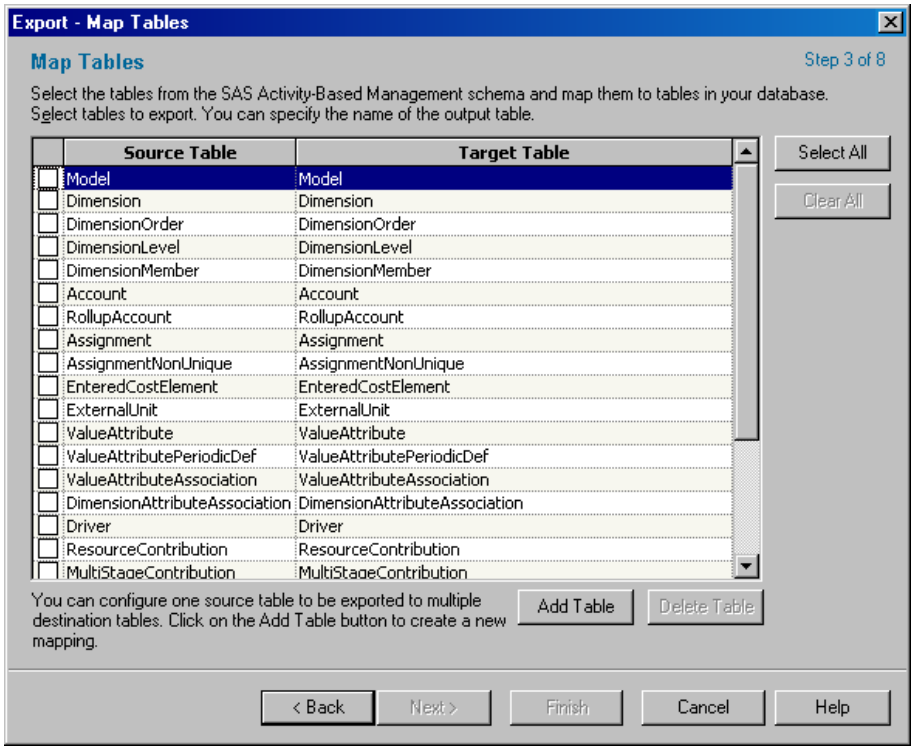
   The Data Link Properties dialog box appears. For information about this dialog box, see the Microsoft Help. For information about the task of creating a connection string to a specific database, see the appropriate section at the end of this chapter.

6. From the **Select a model to export**drop-down list, select a model.

7. From the **Select period/scenario associations you want to export from** list, select the check box next to one or more period/scenario associations.



8. Click **Next**.



9. 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.

10. 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.

11. To map a source table to more than one target table, do the following:

    a. Click **Add Table**.

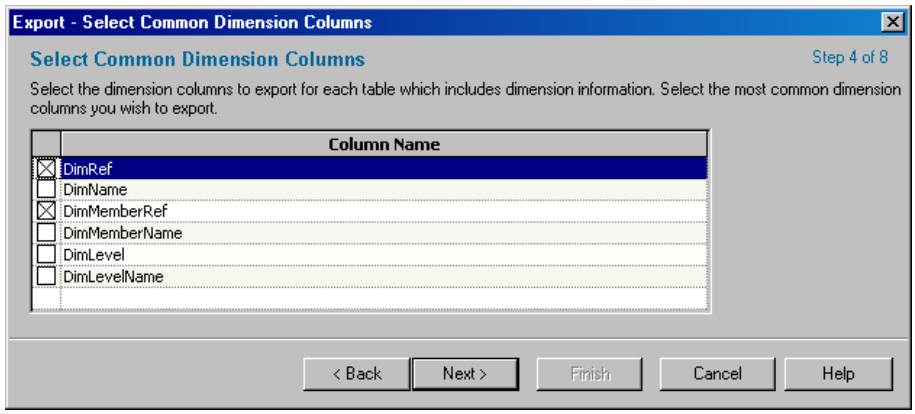
       A new row is added with default information.

    b. Click in the **Source Table** column, and select a SAS Activity-Based Management table from the drop-down list.

12. Repeat steps 9 and 10 for every table that you want to export.

    Next, you will select dimensions to export for each table that contains dimension information.

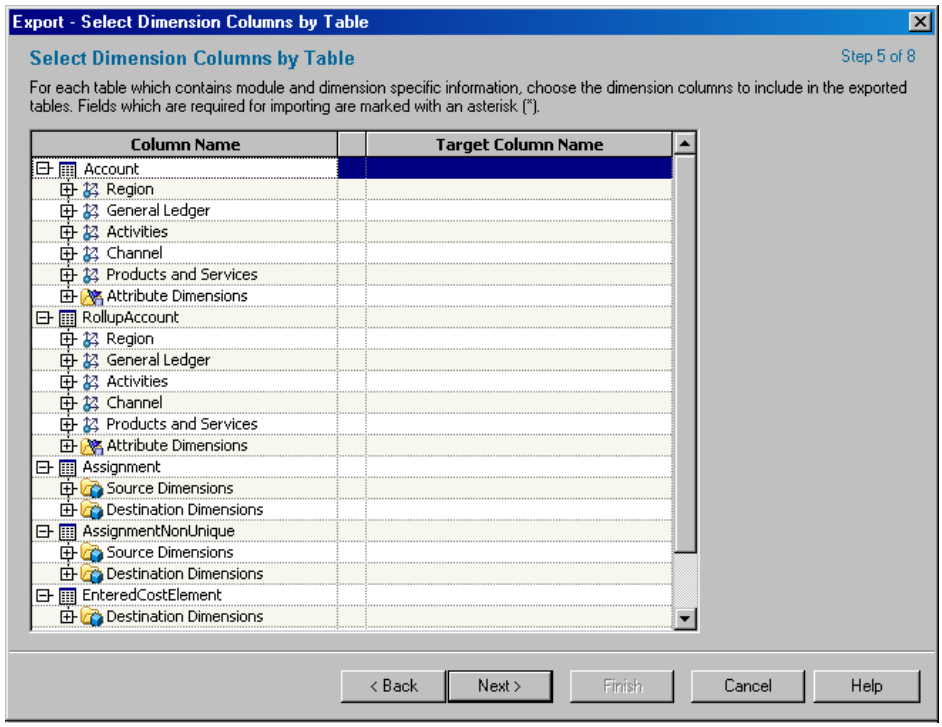
13. Click **Next**.



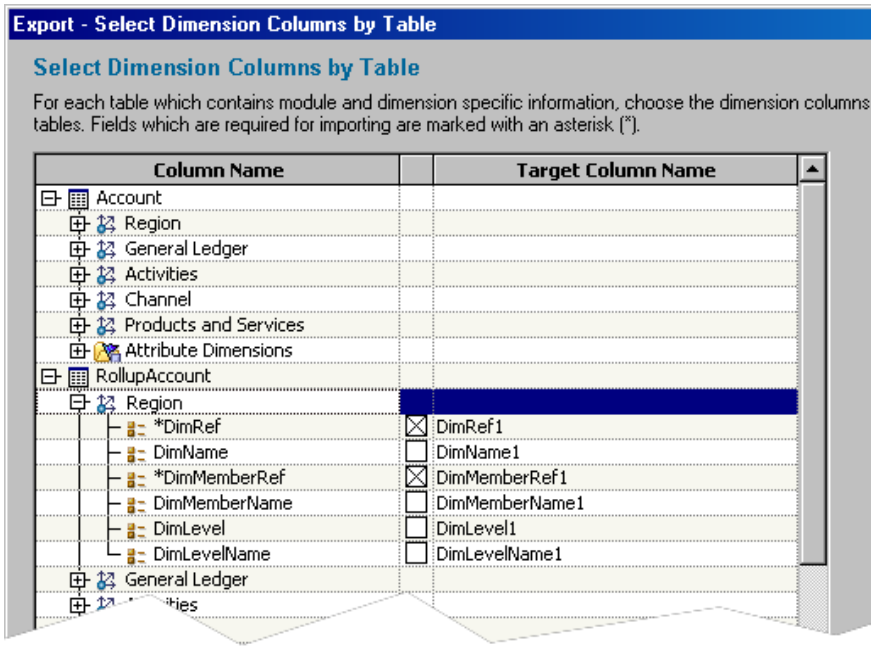
14. To select dimensions within tables to export, select the check box to the left of the dimension in the **Column Name** column.

    Next, you will select dimensions and module information to export for each table that contains dimension information or module information.

15. Click **Next**.



16. Expand each table to review which columns are automatically selected to export.



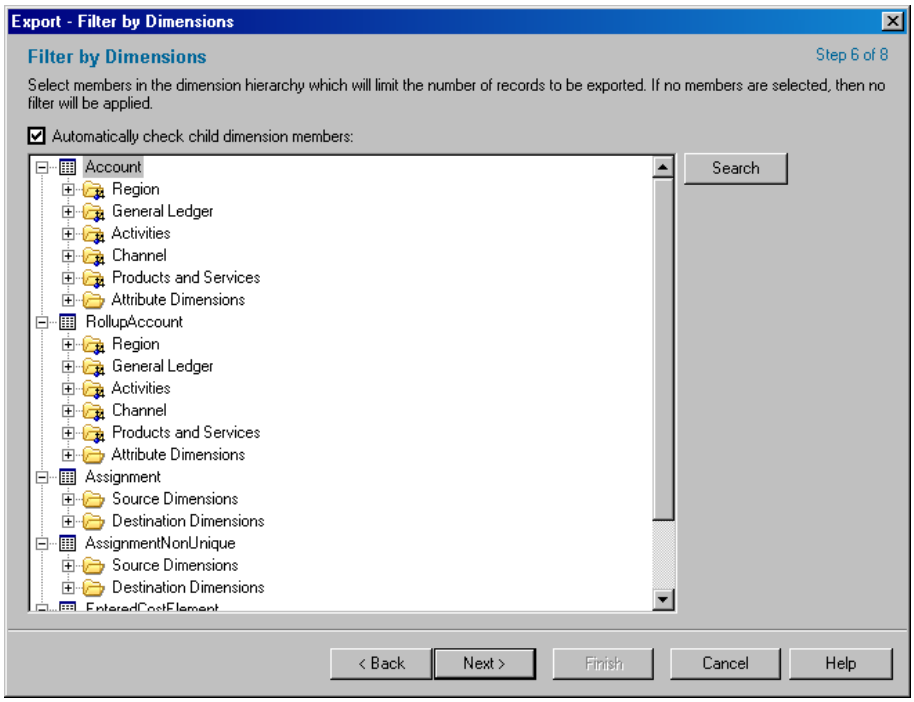
17. To select a column to export, select the check box to the left of the **Target Column Name**.

    Notice that columns marked with an asterisk (*) must be exported.

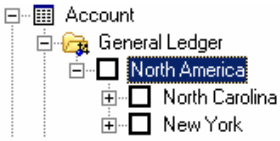
18. To change the name of an exported column, click in the **Target Column Name**column to the right of a column name, and type a new name.

    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.

19. Click **Next**.

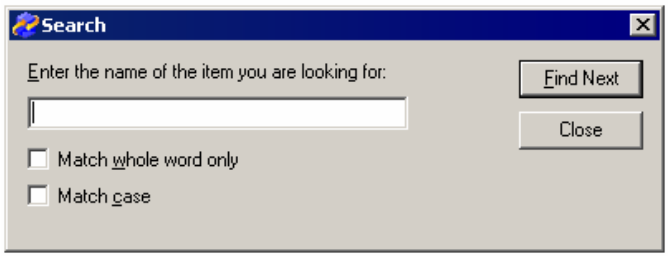


20. Expand each table to view the dimension members.



21. If you want to select all dimension members when a dimension is selected, select the **Automatically check child dimension members** option.
22. To select a dimension member, select the check box to the left of the dimension member.
23. To search for a dimension member, do the following:
    a. Click **Search**.

       The Search dialog box appears.



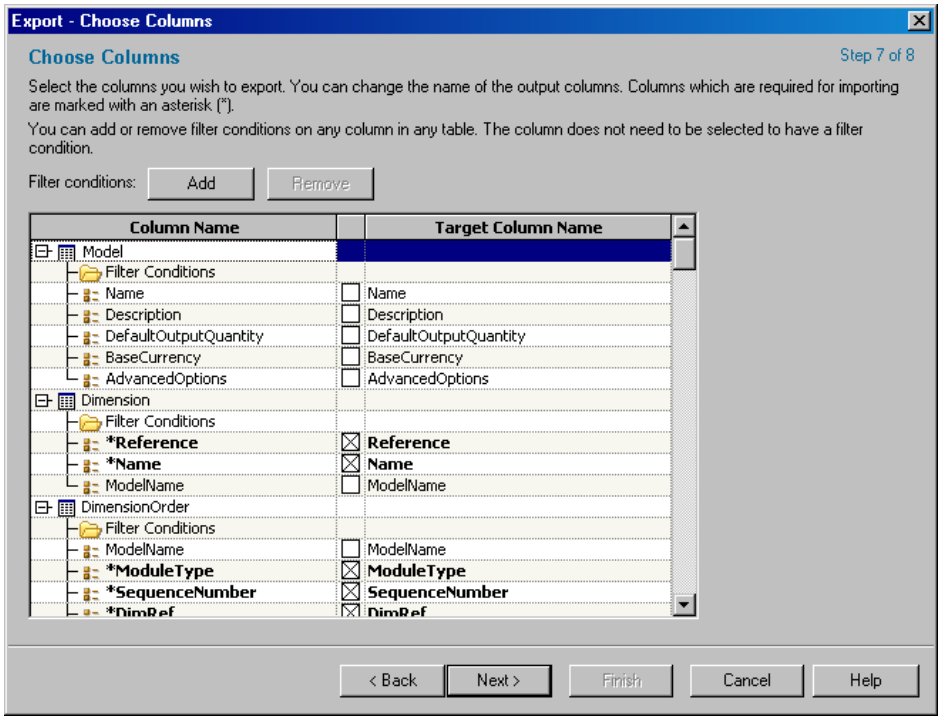
    b. In the **Enter the name of the item you are looking for** box, type the name of the item.

       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.
24. Repeat steps 20 through 22 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.
25. Click **Next**.



26. To select a column, select the check box to the left of the target column name.

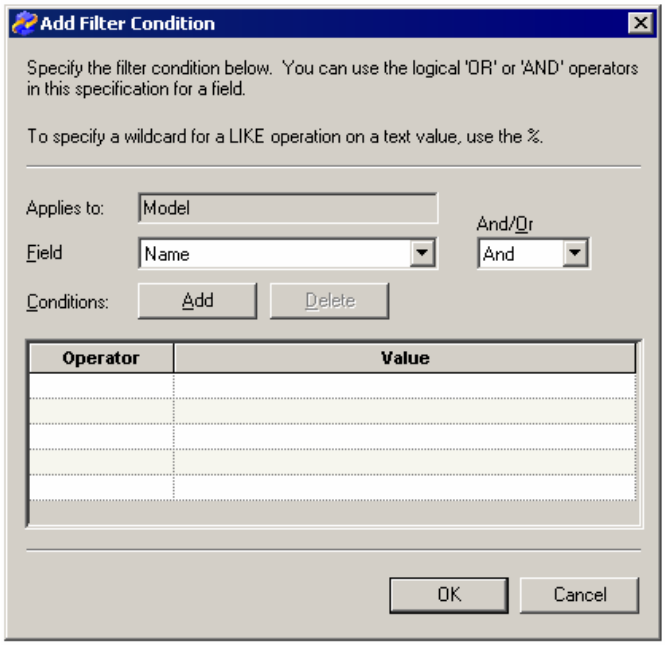
    Notice that columns marked with an asterisk (*) must be exported.

27. 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.

28. 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.

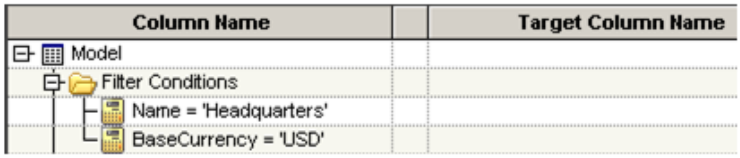
d. Click **Add**.

The field appears in the list.

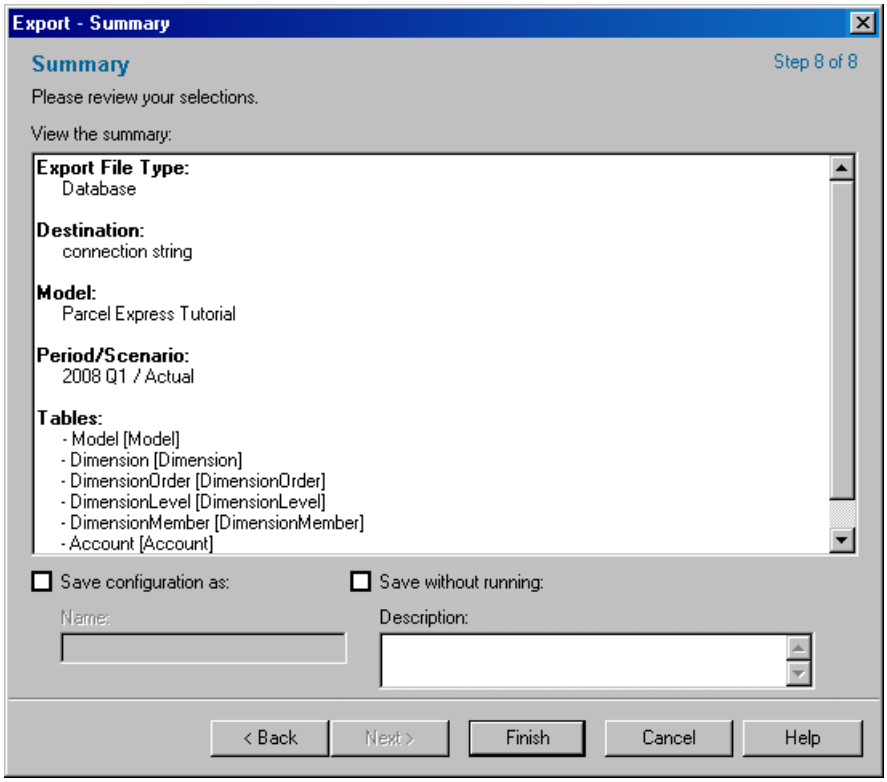
e. Select an **Operator** from the drop-down list.

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:



29. Click **Next**.



30. Review the export summary.

31. 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.

32. 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**.
33. To rename an existing export configuration, do the following:
    a. Select the **Update existing configuration** option.
    b. Type the new **Name**.
    c. Type the **Description**.
34. Click **Finish**.

Connect to a Database

About Connecting to a Database

In order to import or export from a database, you must connect to the database. You can connect using either:

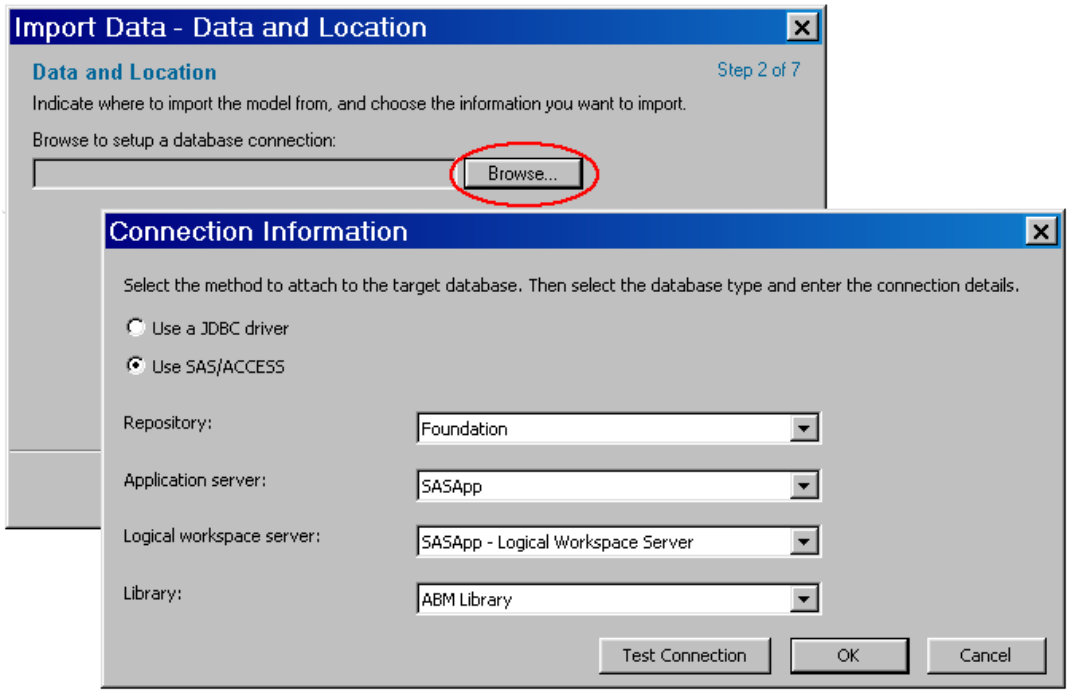
- SAS/ACCESS
- JDBC drivers

Following are details concerning these two methods of connecting to a database.

How to Access the Connection Information Dialog

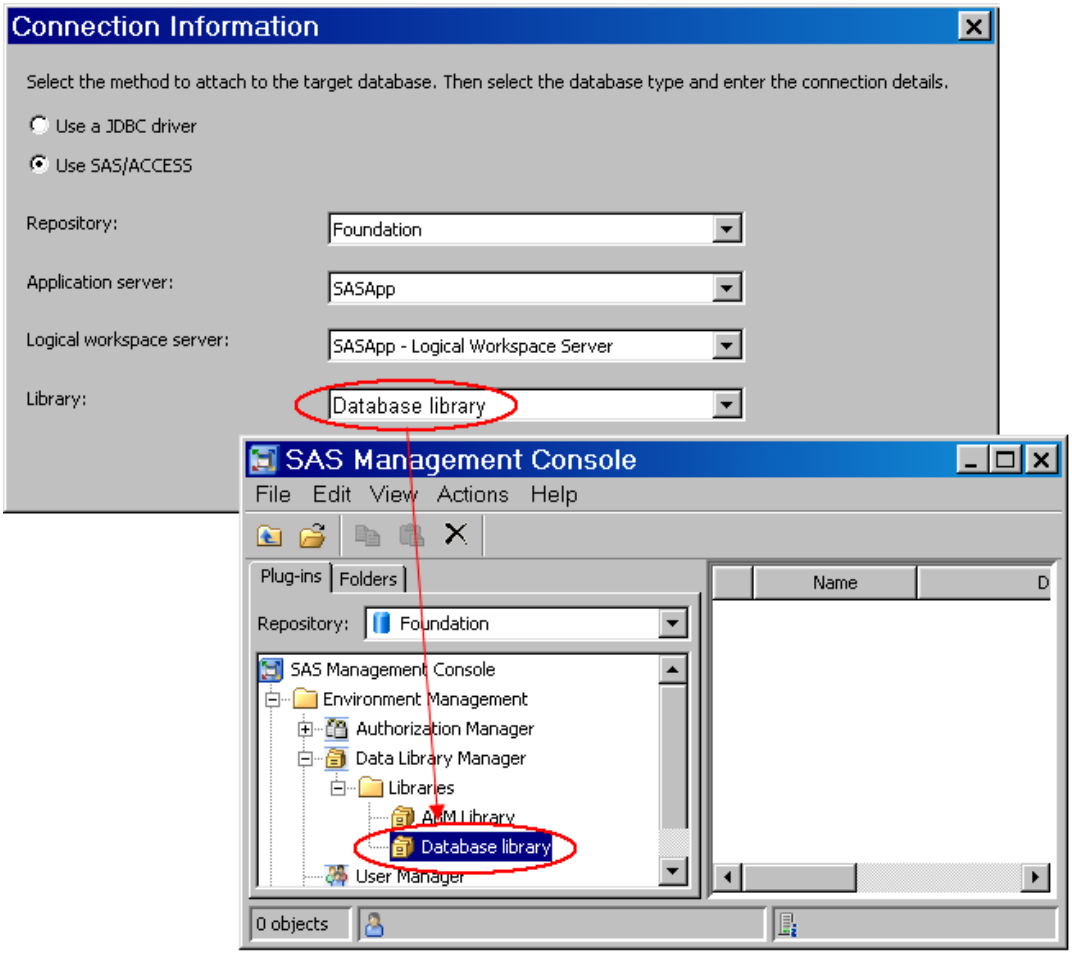
To access the Connection Information dialog, click the Browse button on the following dialogs:

- Import Data (**File** ⇨ **Import** ⇨ **Model Data**)
- Export (**File** ⇨ **Export** ⇨ **Model Data**)
- Report Data (**File** ⇨ **Export** ⇨ **Report Data**)



SAS/ACCESS

Before using SAS/ACCESS, you must first use SAS Management Console to create a library that points to the target database. The library should have all the required READ/WRITE permissions for the user of SAS Activitiy-Based Management.

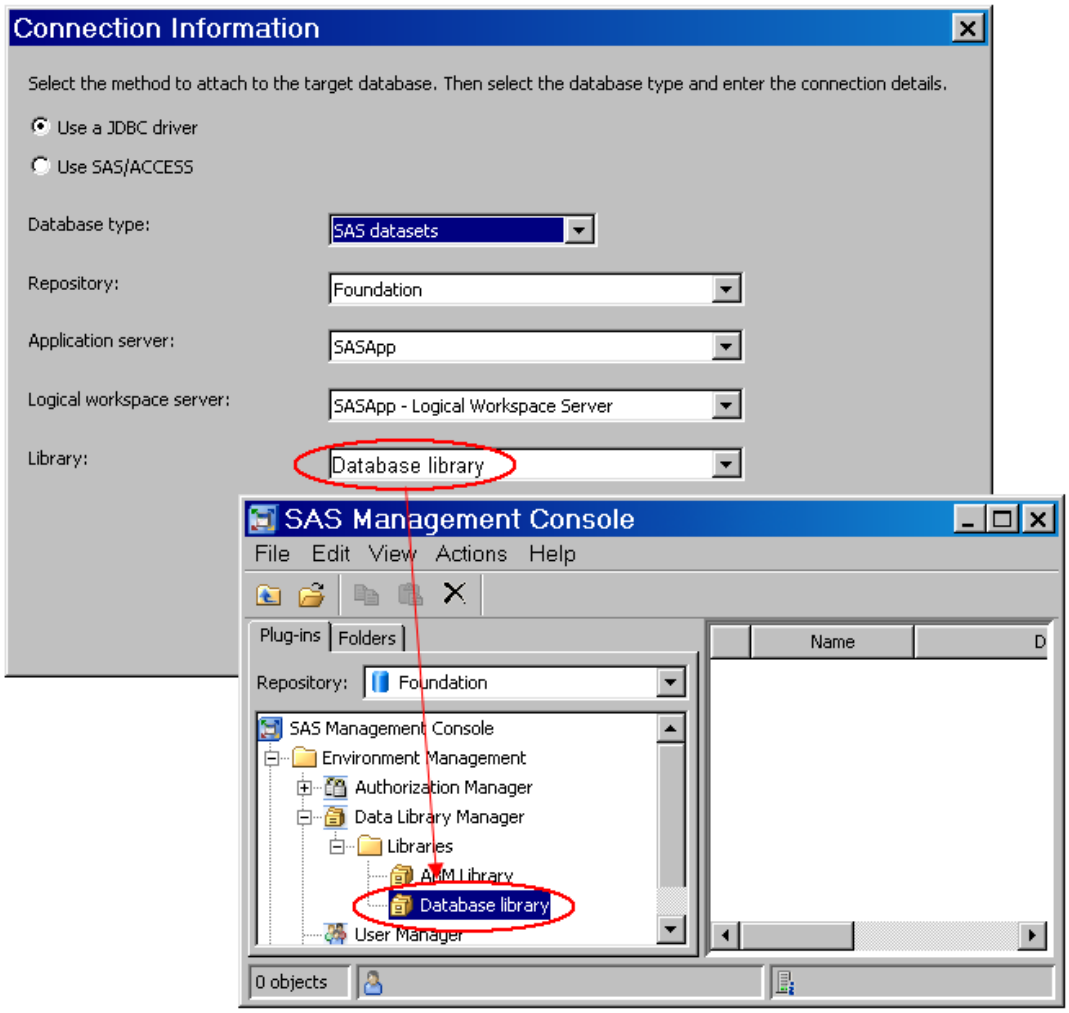


Notes:

- When you use SAS Management Console to create a library pointing to any database on your Metadata Server, you must specify the maximum text length as an advanced option. The maximum text length should be longer than the longest column in your target database table used for import or export. For more information, see the topic on "Specifying the Maximum Text Length" in the SAS Activity-Based Management online help.
- When you use SAS Management Console to create a library for Microsoft Excel on your Metadata Server, you must specify SCAN_TEXT=NO as an advanced option. For more information, see the topic on "Specifying SCAN_TEXT=NO as Advanced Option" in the SAS Activity-Based Management online help. This option is only required for Microsoft Excel.
- When you create a library to export to Microsoft Excel, always use a clean Excel file. The system cannot remove the existing data from Excel before doing the export.

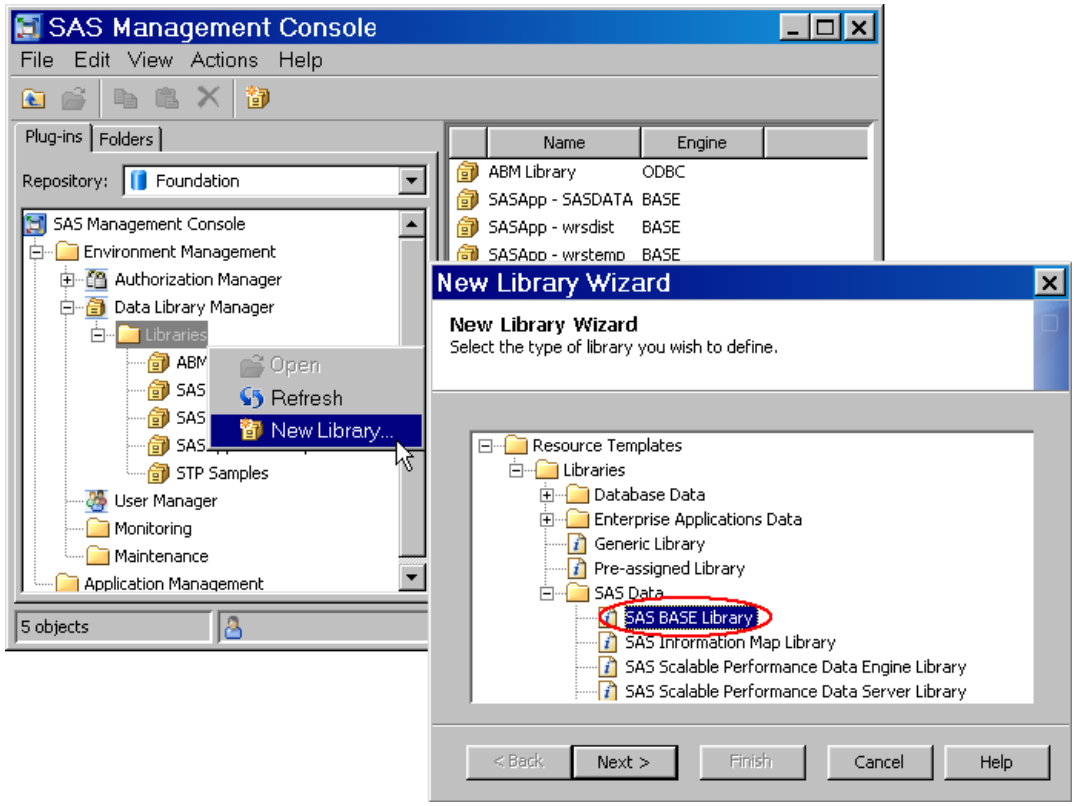
Using a JDBC Driver with SAS Datasets

Before importing from or exporting to SAS datasets, you must create a SAS Base library.



To create the library:

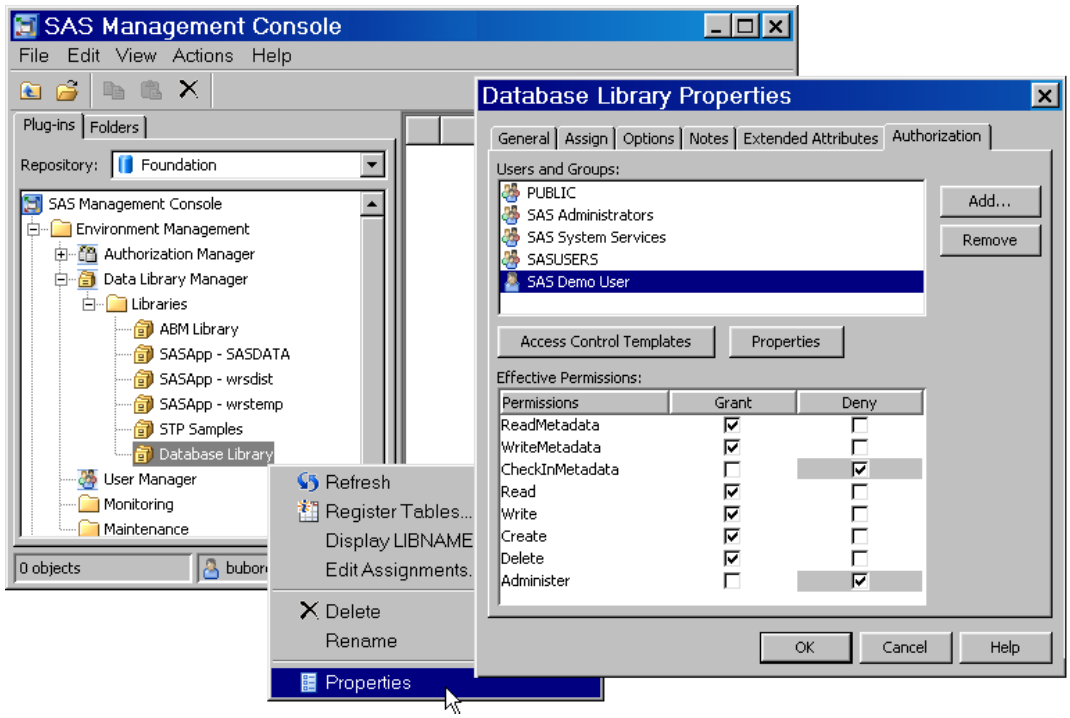
1. Open SAS Management Console connected to your Metadata Server.
2. On the **Plug-ins** tab, right-click **Libraries** (under **Environment Management** ⇨ **Data Library Manager**) and select **New Library**.
3. Select **SAS Base Library**.



The SAS Activity-Based Management user who wants to import from or export to SAS datasets must have the appropriate permissions for the SAS Base Library that you created.

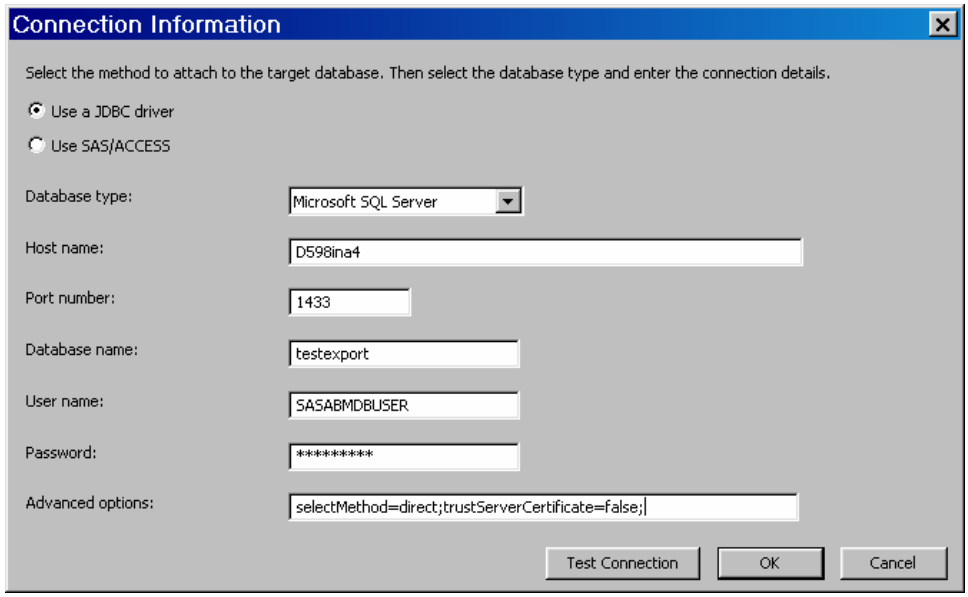
To set the library permissions:

1. Right-click the SAS Base Library that you created, and then select **Properties**.
2. Select the user of SAS Activity-Based Management who will be importing from or exporting to SAS datasets.
3. Set the permissions as shown in the following picture.



Using a JDBC 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** is the MachineName where the SAS Activity-Based Management database is installed.
- **Port** is the TCP port where the SAS Activity-Based Management database instance is running.

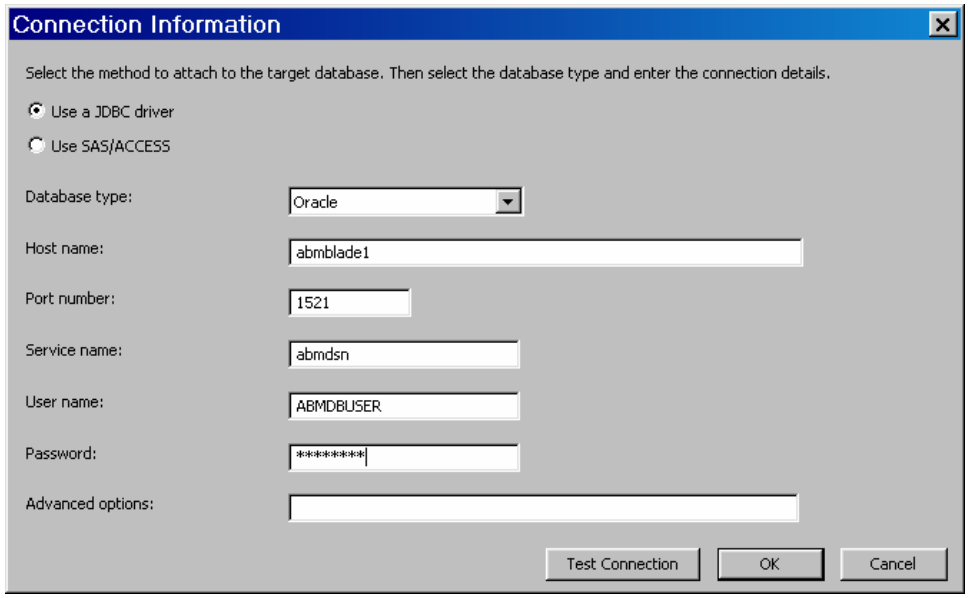
If the database is on a Named Instance, then go to the SQL Server Configuration manager to check the TCP port on which it is running. Named Instances run on a dynamic TCP port.

- If you specify advanced options, separate multiple options with a semicolon. For example:

```
selectMethod=direct;trustServerCertificate=false;
```

Using a JDBC Driver with Oracle

The following picture shows an example of connecting to an Oracle database.



The following are some considerations to keep in mind when connecting to an Oracle database:

- **Host name** is the MachineName where the Oracle client is installed.
- **Port** is the TCP port where the SAS Activity-Based Management database instance is running.
- 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).

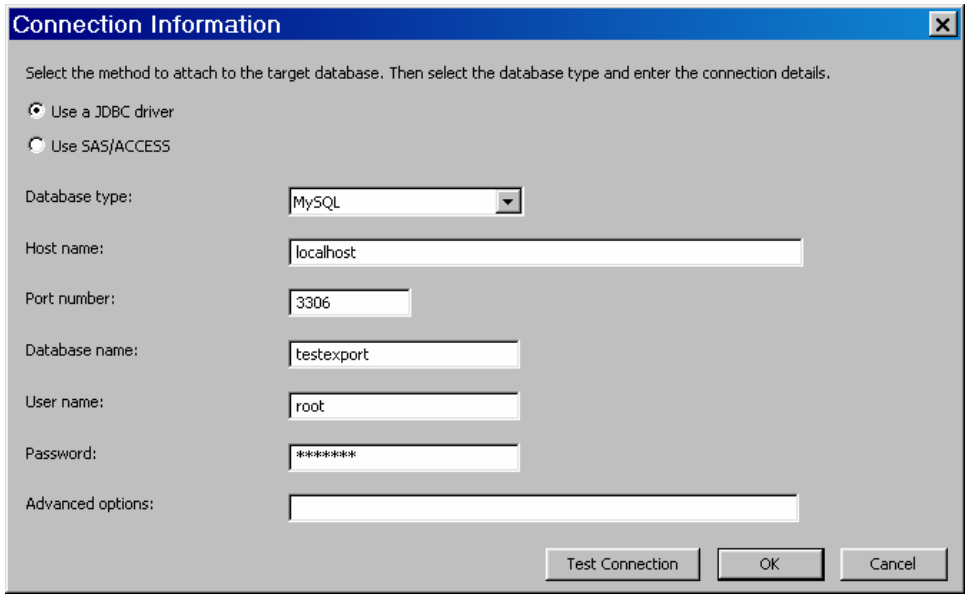
  This file is at the same location as **`tnsnames.ora`**. For example:

```
C:\Oracle\product\11.1.0\client_1\network\admin\sqlnet.ora
```

.

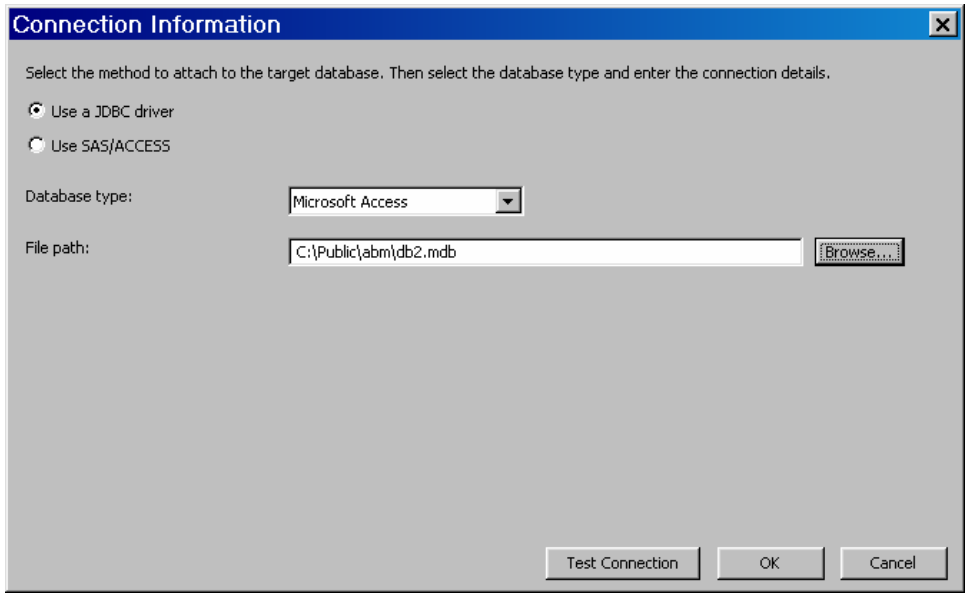
Using a JDBC Driver with MySQL

The following picture shows an example of connecting to a MySQL database



Using a JDBC Driver with Microsoft Access

The following picture shows an example of connecting to Microsoft Access. The file path that you enter should have READ/WRITE permission.



Notes:

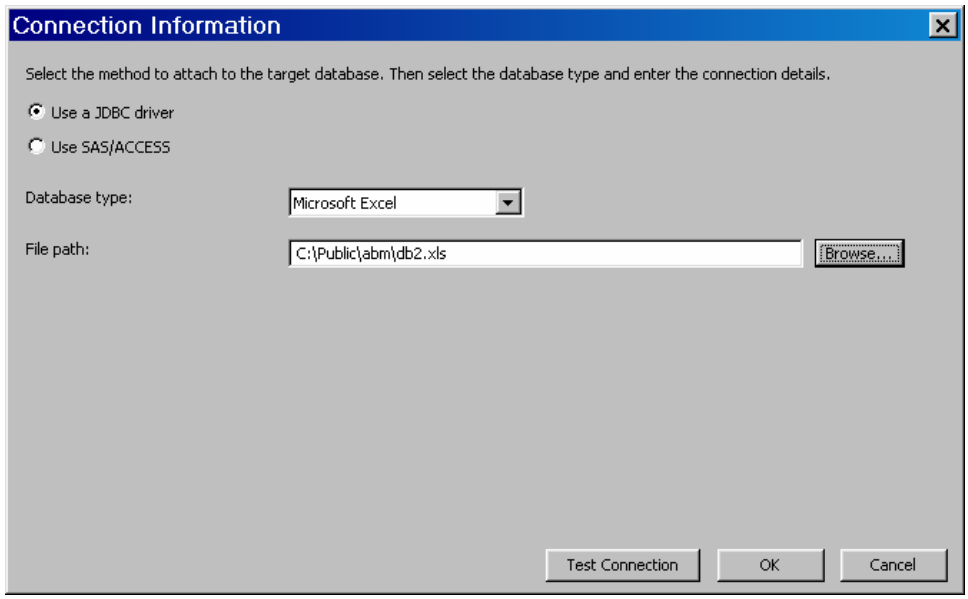
- The JDBC driver for Microsoft Access must use a bridge to ODBC. This means that you must install the appropriate ODBC driver on the SAS Activity-Based Management

server machine. For more information, see the topic on “Installing the ODBC Driver for Microsoft Access and Excel” in the SAS Activity-Based Management online help.

- The directory that you specify for importing from or exporting to must have the proper permissions. For more information, see the topic on “Permissions for Microsoft Access and Microsoft Excel” in the SAS Activity-Based Management online help.
- If SAS Activity-Based Management client and server are on same machine, then you can use a local path. For example: c:\accessexport\test.mdb. If they are on different machine, then you should use the network path. For example: \\d12345\accessexport \test.mdb.

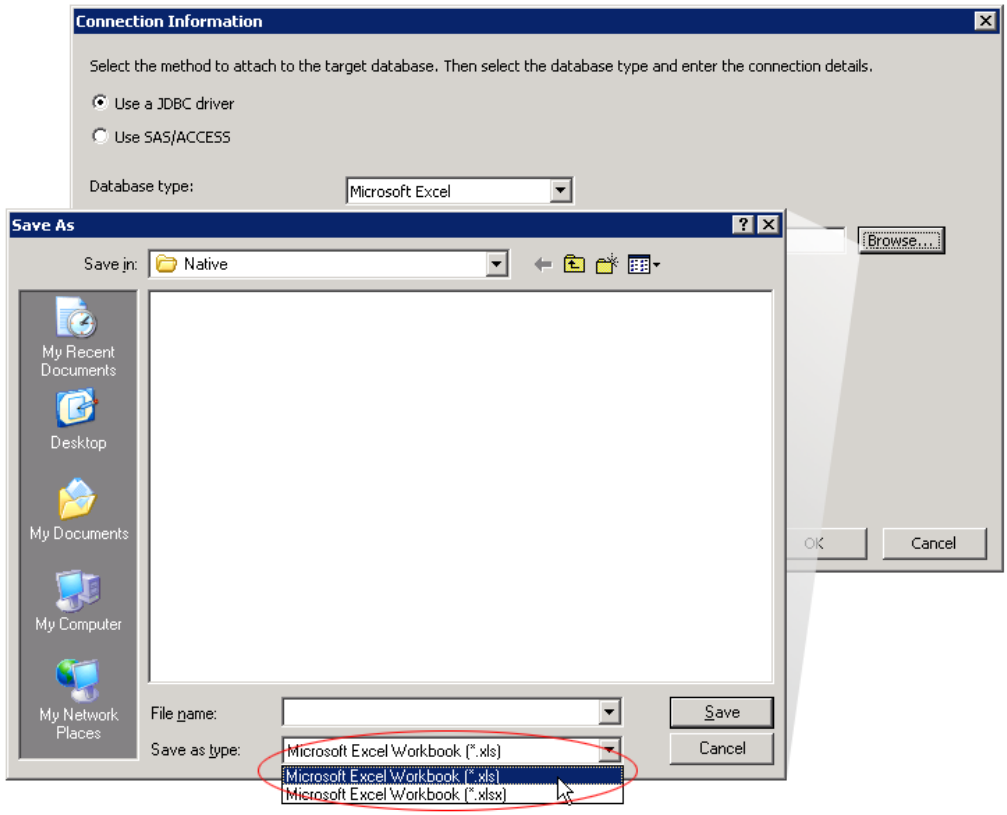
Using a JDBC Driver with Microsoft Excel

The following picture shows an example of connecting to Microsoft Excel.



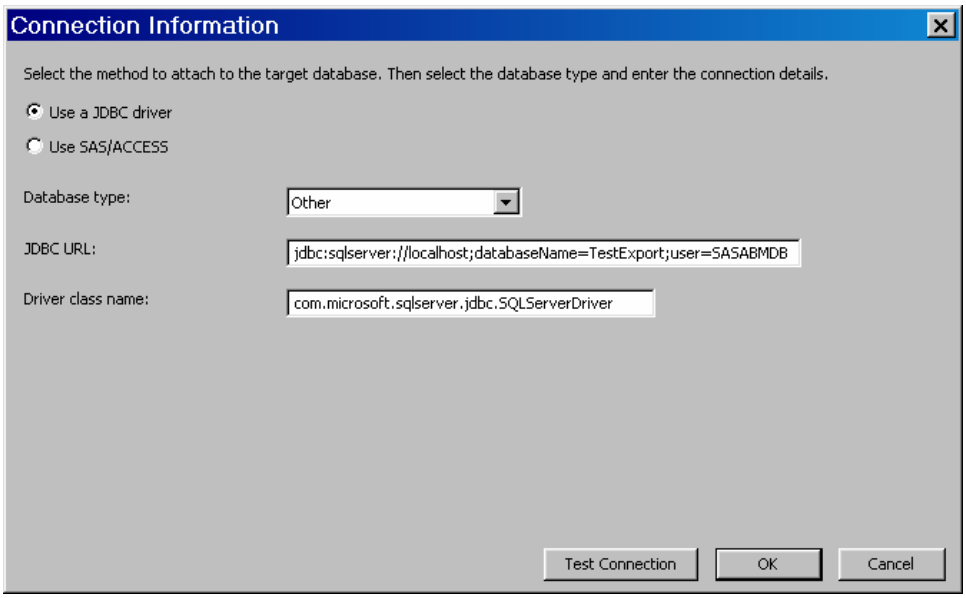
Notes:

- The JDBC driver for Microsoft Excel must use a bridge to ODBC. This means that you must install the appropriate ODBC driver on the SAS Activity-Based Management server machine. For more information, see the topic on “Installing the ODBC Driver for Microsoft Access and Excel” in the SAS Activity-Based Management online help.
- The directory that you specify for importing from or exporting to must have the proper permissions. For more information, see the topic on “Permissions for Microsoft Access and Microsoft Excel” in the SAS Activity-Based Management online help.
- If SAS Activity-Based Management client and server are on same machine, then you can use a local path. For example: c:\excelexport\test.xls. If they are on different machine, then you should use the network path. For example: \\d12345\excelexport \test.xls.
- You can save the Excel file as either *.xls(2003 format) or *.xlsx (2007 format). When you export to Microsoft Excel, always use a clean Excel file. The system cannot remove existing data from Excel before doing the export.



Using a JDBC Driver with Another Database

In order to use a JDBC driver to connect to any other database, you must enter a JDBC connection string to connect to the target database and a JDBC driver class name. (When you connect to one of the previously mentioned databases, SAS Activity-Based Management takes care of creating the JDBC connection string and specifying the driver class name).



Part 4

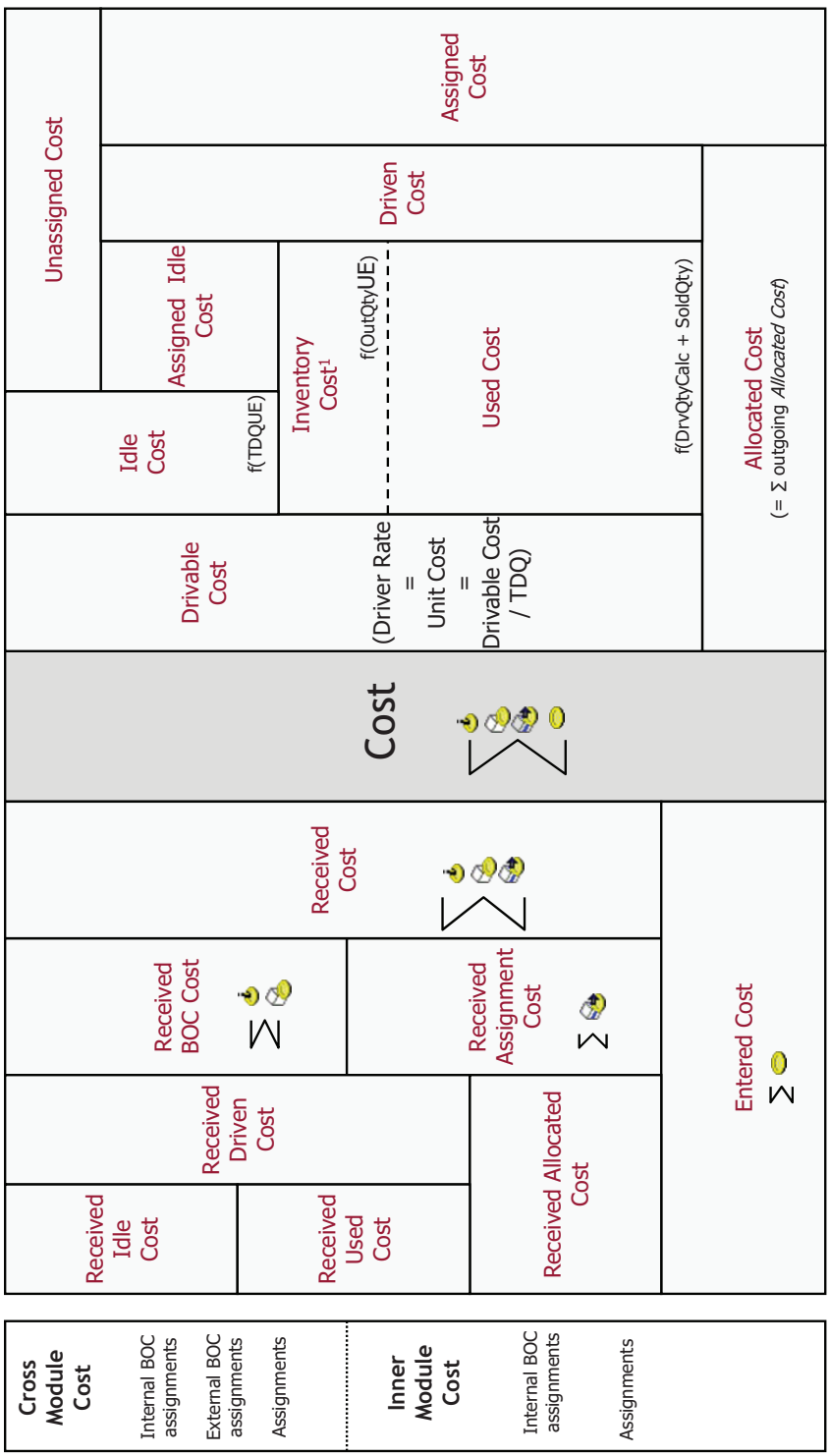
Appendices

Appendix 1

Properties

Property Relationship Diagrams

Account Properties (non-reciprocal)



Assumption: There are different "assignment" methodologies that can be used to assign costs — BOC method or the Assignment method.
Assigned Costs: represent all assignment methodologies used to flow costs from an account — BOC and Assignment costs leaving an account. From the account point of view, "what costs am I assigning out?"
Received Costs: represent all assignment methodologies that are used to contribute costs to an account. As entered costs are not part of the assignment methodology they are not included. From the account point of view, "what 'Assigned Costs' am I receiving in?"

Footnotes
[1] *Inventory Cost* represents an adjustment to reflect difference between *Sold Quantity* and *Output Quantity*.

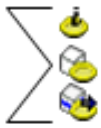
Legend

Received BOC Cost =  = sum of received internal unit cost elements and received external unit cost elements.

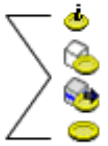
Received Assignment Cost =  = sum of received assigned cost elements.

Note: The received assigned cost elements can come from the Resource, Activity, or Cost Object module.

Entered Cost =  = sum of user entered cost elements.

Received Cost =  = sum of received internal unit cost elements and received external unit cost elements and received assigned cost elements. Note: The received assigned cost elements can come from the Resource, Activity, or Cost Object module.

Note: The received assigned cost elements can come from the Resource, Activity, or Cost Object module.

Cost =  = sum of received internal unit cost elements and received external unit cost elements and received assigned cost elements and user entered cost elements.

Cost Properties on the Source Account Side

```
Drivable Cost = (Cost - Allocated Cost)

Driver Rate = Drivable Cost/TDQ

Drivable Cost = (Used Cost + Inventory 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 Relationship on the Source Account Side

```
Driver Rate = Cost/TDQ (or TDQUE)

Used Cost = UsedQty*Driver Rate

IdleCost = IdleQty*Driver Rate

IdlePercentage = IdleQty/TDQ
```

Costs on the Destination Account Side

```
(Received Driven Cost + Received Allocated Cost) = (Received BOC
Cost + Received Assignment Cost) = Received Cost
```

```
(Received Cost + Entered Cost) = Cost
```

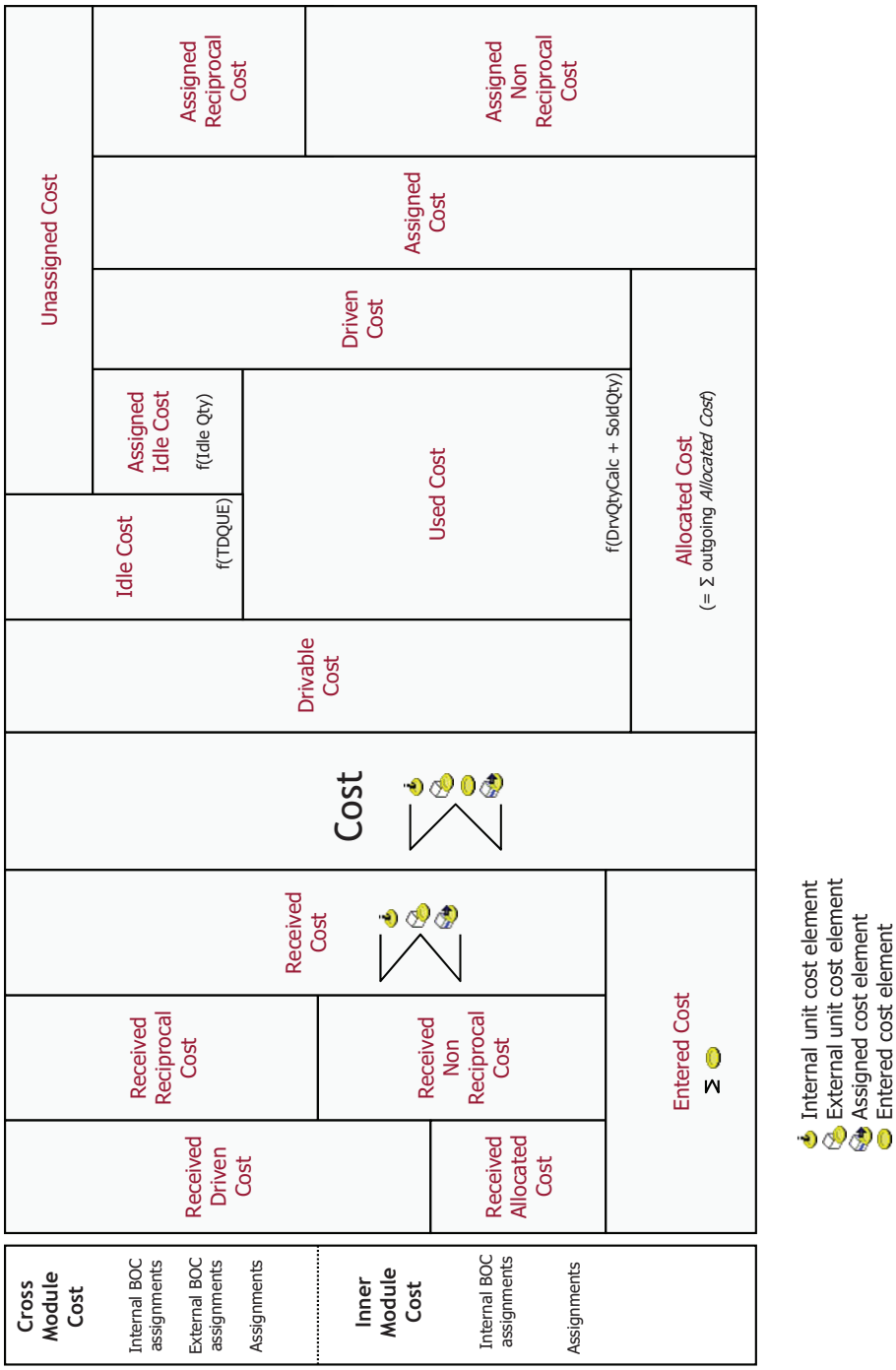
Costs on the Assignment Path

```
Driver Driven Cost = (Driver Used Cost + Driver Idle Cost)

Driver Cost = (Driver Driven Cost + Driver Allocated Cost)
```

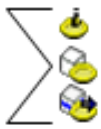
Cost and Quantity Relationships on the Assignment Path

```
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)
```

Account Properties (reciprocal)

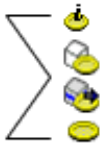


Legend

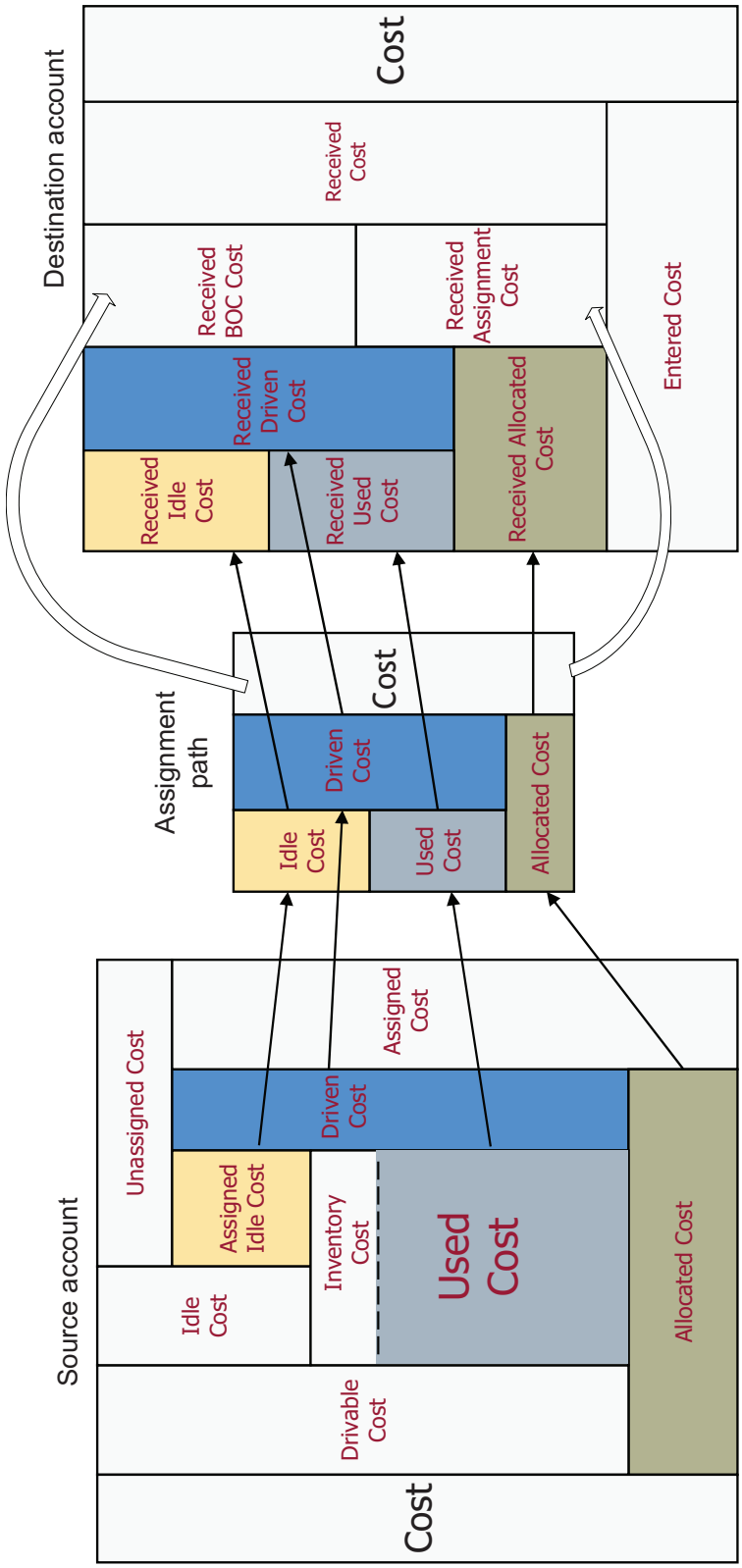
Received Cost =  = sum of received internal unit cost elements and received external unit cost elements and received assigned cost elements. Note: The received assigned cost elements can come from the Resource, Activity, or Cost Object module.

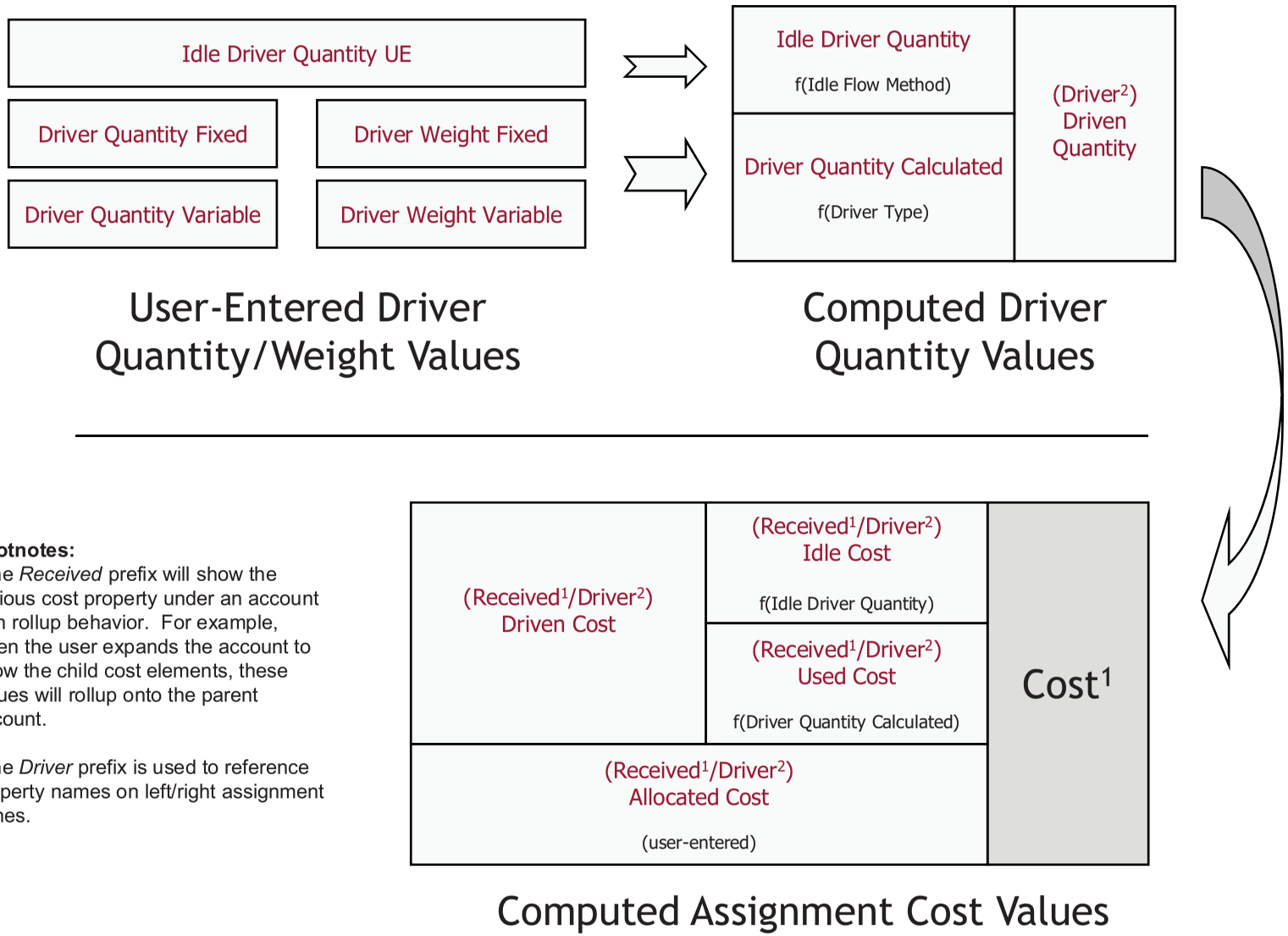
Note: The received assigned cost elements can come from the Resource, Activity, or Cost Object module.

Entered Cost =  = sum of user entered cost elements.

Cost =  = sum of received internal unit cost elements and received external unit cost elements and received assigned cost elements and user entered cost elements.

Cost Flows (non-reciprocal)

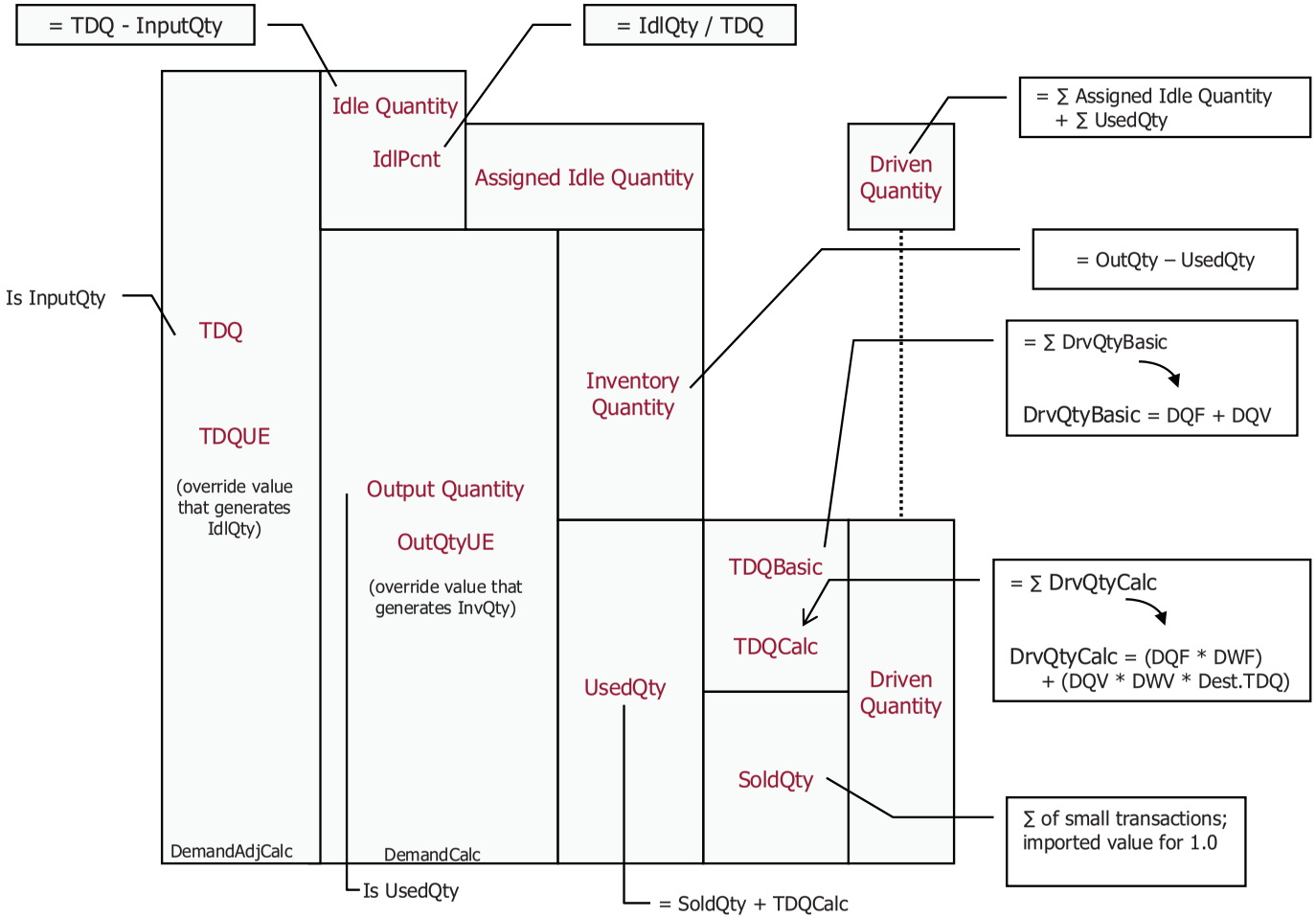




Footnotes:

[1]The *Received* prefix will show the various cost property under an account with rollup behavior. For example, when the user expands the account to show the child cost elements, these values will rollup onto the parent account.

[2]The *Driver* prefix is used to reference property names on left/right assignment panes.



- IdleQty implies IdleQtyCalc. If we need to introduce the notion of IdleWeight then we need to revisit this property. IdleQty implies a weight of 1 if weighted drivers are used.
- DQ represents the quantities used to flow cost using both assignment methodologies: BOCs as well as Assignment paths. There are two types of DQs: DriverQuantityFixed (DQ_F) and DriverQuantityVariable (DQ_V).
- DW represents Driver Weight. There are two types of DWs: DriverWeightFixed (DW_F) and DriverWeightVariable (DW_V).

Quantities on the Assignment Path

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

DQBasic = DQF + DQV* Dest.TDQ

Idle Driver Quantity = Idle Driver Quantity UE

Cost and Quantity Relationships on the Assignment Path

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 Source Account Side

UsedQty = (TDQCalc + SoldQty)

DrivenQty = ∑ Assigned Idle Quantity + ∑ Used Qty

Inventory Qty = (OutputQty - UsedQty)

IdleQty = (TDQUE – OutputQty) (TDQUE and OutputQtyUE override respective quantities)

UnassignedQty = (OutputQty – UsedQty + UnassignedIdleQty)

If **TDQUE > OutputQty** then **Positive Idle**

If **TDQUE < OutputQty/OutputQtyUE** then **Negative Idle**

Cost and Quantity Relationships on the Source Account Side

Driver Rate = Cost/TDQ (or TDQUE)

Used Cost = UsedQty*Driver Rate

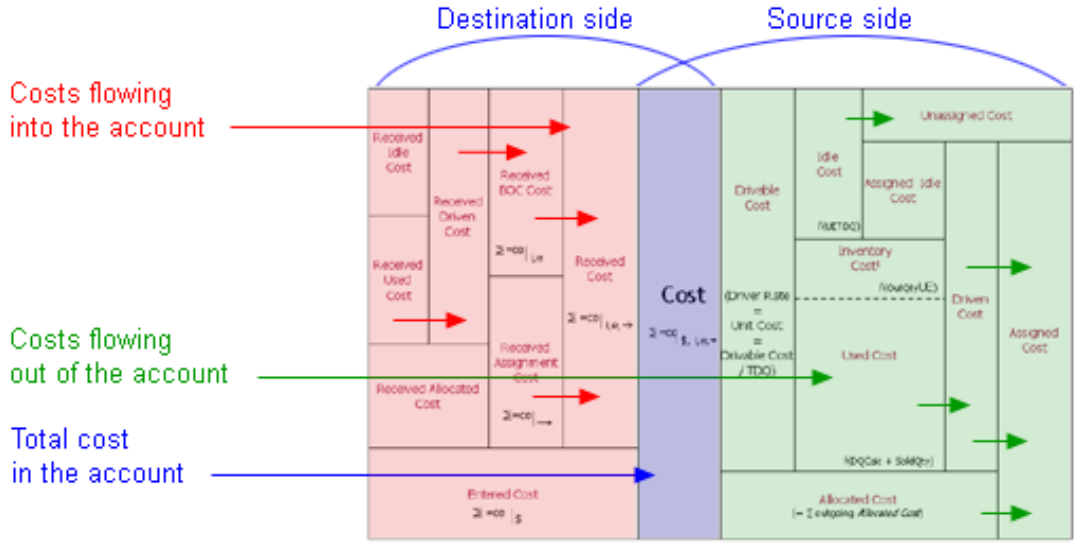
IdleCost = IdleQty*Driver Rate

IdlePercentage = IdleQty/TDQ

How to Read these Diagrams

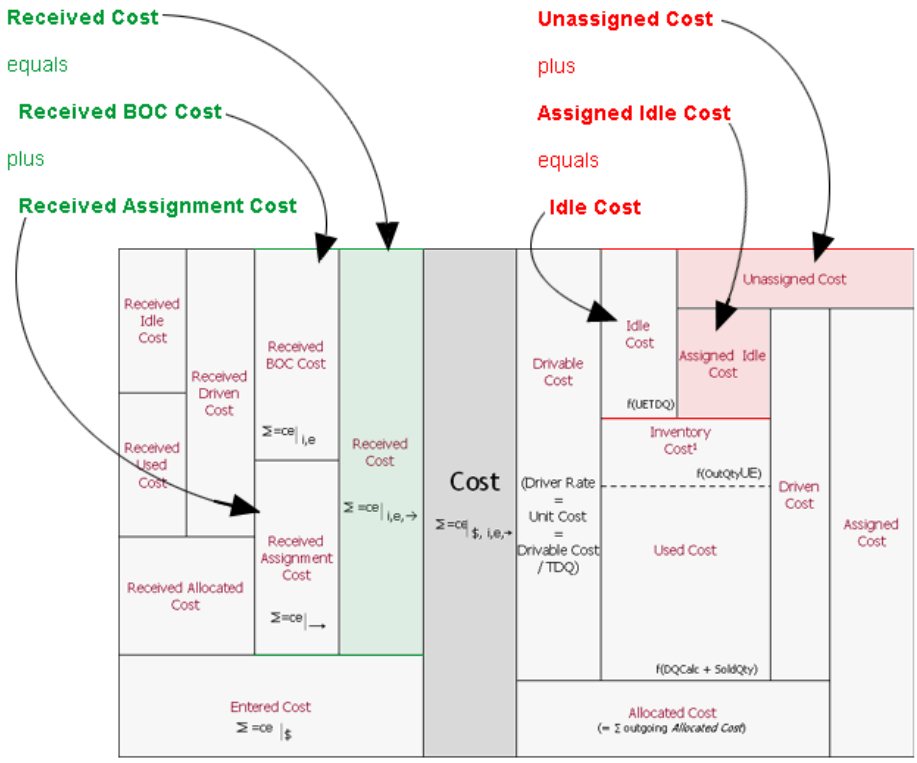
A Single Account

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:



A Slightly Unintuitive Case

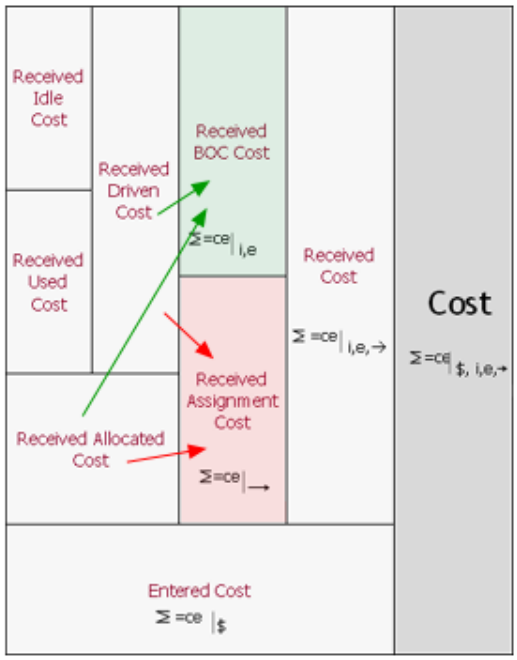
As the following picture shows:

- **Received BOC Cost** = some amount of **Received Driven Cost** plus some amount of **Received Allocated Cost** .

- **Received Assignment Cost** = some amount of **Received Driven Cost** plus some amount of **Received Allocated Cost** .

Received BOC Cost
equals
some **Received Driven Cost**
plus
some **Received Allocated Cost**

Received Assignment Cost
equals
some **Received Driven Cost**
plus
some **Received Allocated Cost**



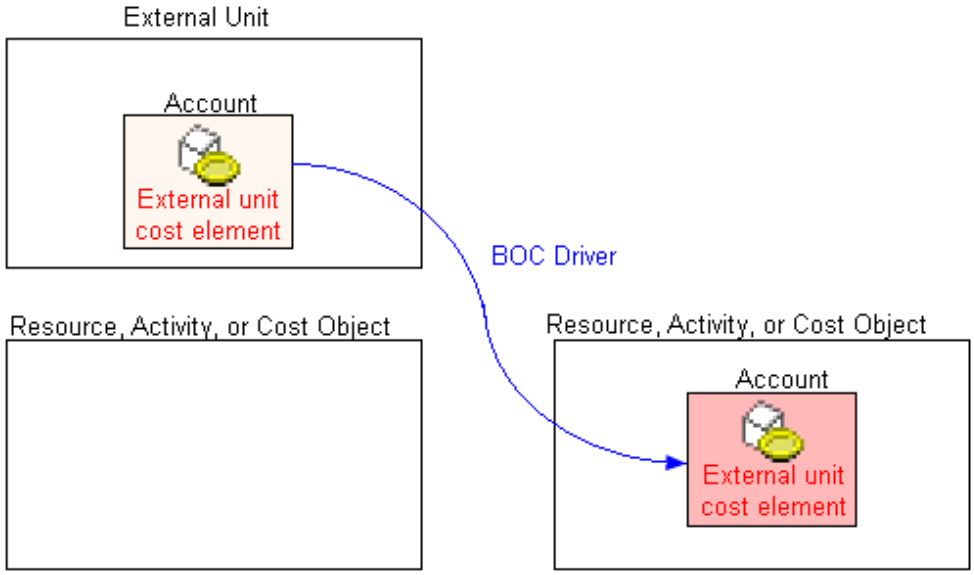
Types of Costs

The diagrams show both entered costs and cost flows from one account to another.

- An entered cost is one that you enter directly into an account. It does not come from another account.
- There are three types of cost flows from one account to another:
  - External BOC
  - Internal BOC
  - Assignment

External BOC

An External BOC assignment flows from an account in an External Unit to an account in the Resource, Activity, or Cost Object module. All assignments from an External Unit use a Bill of Cost (BOC) driver.



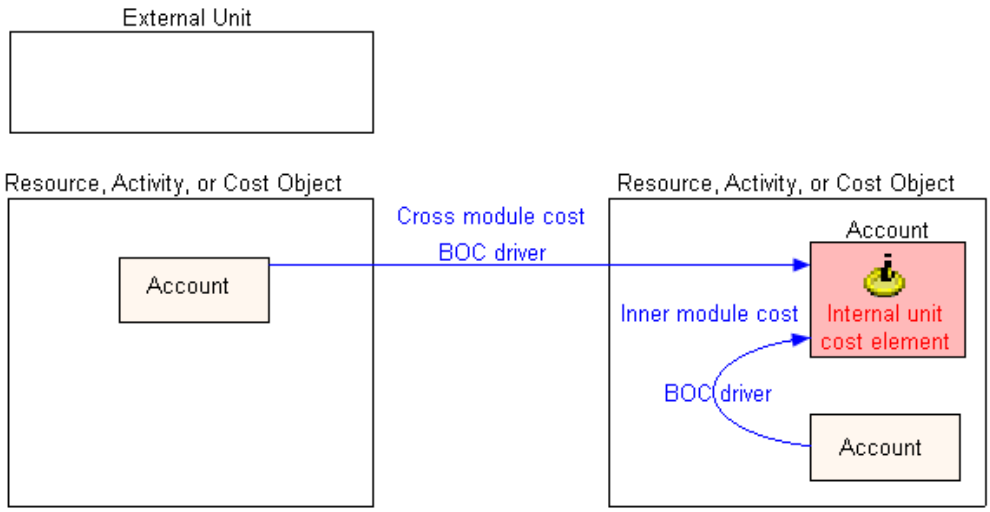
Internal BOC

An Internal BOC assignment is an assignment that uses a Bill of Costs (BOC) driver and flows from an account in the Resource, Activity, or Cost Object module to an account that is also in the Resource, Activity, or Cost Object module.

The assignments can be either of two kinds:

- **Cross module** (Resource-to-Activity, Resource-to-Cost Object, or Activity-to-Cost Object)
- **Inner module** (Resource-to-Resource, Activity-to-Activity, or Cost Object-to-Cost Object)

The word "internal" in the expression "Internal BOC" means that the assignment does not come from an External Unit.



Assignment

An Assignment is a cost flow from an account in the Resource, Activity, or Cost Unit module to an account that is also in the Resource, Activity, of Cost Object module. It uses a driver other than a Bill of Costs (BOC) driver. That is, it uses one of the following drivers: Basic, Calculated, Evenly Assigned, Percentage, Sales Volume, or Weighted.

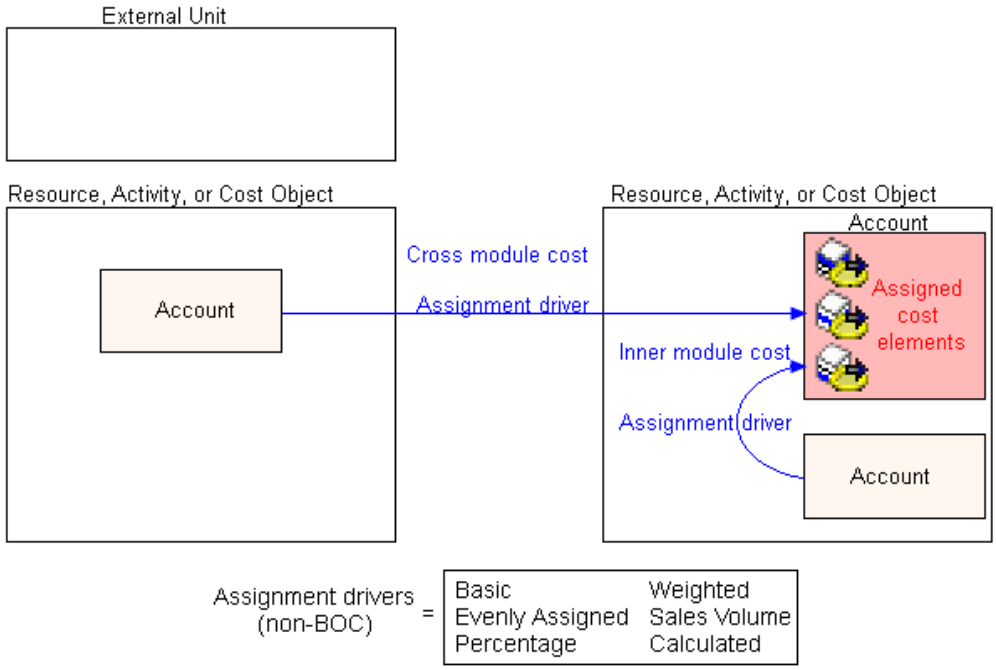
Note: The same word is used in both a general sense and in a specific sense.

In the **general** sense, an "assignment" is any cost flow from one account in any module to another account in any module.

In the **specific** sense, an "Assignment" (generally written with an uppercase "A"), is a cost flow from one account that is not in an External Unit to another account (that is also not in an External Unit), and it uses a non-BOC driver.

The assignments can be either of two kinds:

- **Cross module** (Resource-to-Activity, Resource-to-Cost Object, or Activity-to-Cost Object)
- **Inner module** (Resource-to-Resource, Activity-to-Activity, or Cost Object-to-Cost Object)

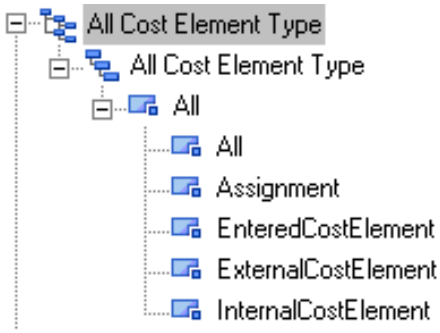


(However, in fact, Basic = Weighted = BOC)

Cost Element Types

In a Single-stage Contribution cube you can display costs by cost element type. Here are the types available that are available:

- Assignment
- Entered Cost Element
- External Cost Element
- Internal Cost Element



The following table shows the reason for the cost for each type of cost element.

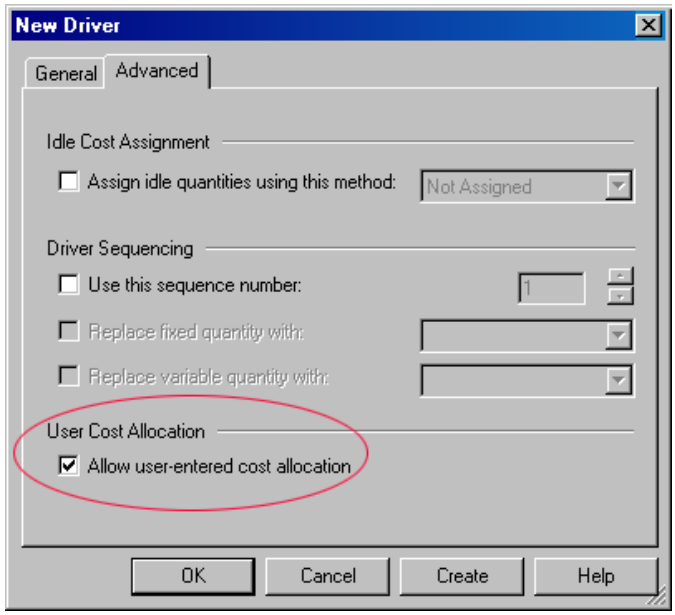
Note: In this case, the cost of a cost element is considered from the destination-side point of view. It is the cost resulting from some source.

Cost element type	Cost results from
Assignment 	Basic, Calculated, Evenly Assigned, Percentage, Sales Volume, or Weighted driver from an account in the Resource, Activity, or Cost Object module
Entered Cost Element 	User allocated cost
External Cost Element 	Bill of Costs driver from an external unit account
Internal Cost Element 	Bill of Costs driver from an account in the Resource, Activity, or Cost Object module

Allocated Cost (AllocCost)

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

When you add this property to a column in a grid, the default format is Currency.

This is an 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Assigned Reciprocal Cost on page 309
- Received Cost on page 329

Assigned Idle Cost (AsgnIdlCost)

The sum of costs on outgoing assignments that is specifically because of the source account's idle flow behavior.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.

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 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Cost on page 310
- Received Reciprocal Cost on page 330

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.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Cost (Cost)

The calculated cost of a cost element or account.

When you add this property to a column in a grid, the default format is Currency.

You can change this value only for entered cost elements. All other values of this property are generated by the system. You cannot change these values.

This is an 8-byte, floating-point number.

See Also

- Assigned Reciprocal Cost on page 309
- Received Cost on page 329

Dimension Level Name (DimLevelName)

The name of the level for the dimension that completes an intersection.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Dimension Level Number (DimLevelNumber) on page 310

Dimension Level Number (DimLevelNum)

The depth in the dimension hierarchy of the dimension that completes an intersection.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

This is a 4-byte integer value.

See Also

- "Dimension Level Number (DimLevelNum)" on page 310
- Dimension Level Name (DimLevelName) on page 310

Dimension Member Name (DimMemName)

The name of the dimension member that defines an account (dimensional intersection).

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Dimension Member Reference (DimeMemberReference) on page 311

Dimension Member Reference (DimMemRef)

The reference of the dimension member that defines an account (dimensional intersection).

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Dimension Member Name (DimMemberName) on page 311

Dimension Name (DimName)

The name of the dimension that completes an intersection.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Dimension Reference (DimReference) on page 312

Dimension Reference (DimRef)

The reference of the dimension that completes an intersection.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Dimension Name (DimName) on page 311

Display Name (Display Name)

The name for the displayed item.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value. However, you can change Name .

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Display Reference on page 312

Display Reference (Display Reference)

The reference for the displayed item.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value. However, you can change Reference .

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data , this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Display Name on page 312

Drivable Cost (DrvblCost)

The remainder of an account's cost that can be assigned by its driver after subtracting user-allocated costs.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Driven Cost (DrvnCost)

The amount of cost that is driven by the account's driver, including both Used Cost and Assigned Idle Cost.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Driven Quantity (DrvnQty)

The sum of the driver quantity on an account's outgoing assignments, including both Used Quantity and Assigned Idle Quantity.

When you add this property to a column in a grid, the default format is Number.

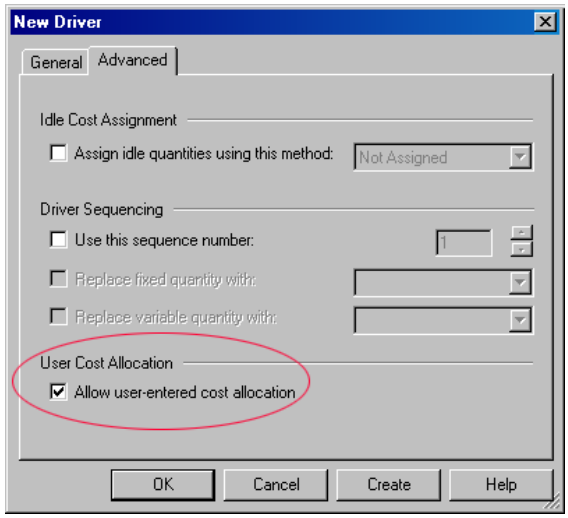
The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.



When you add this property to a column in a grid, the default format is Currency.

This is an 8-byte, floating-point number.

Driver Cost (DrvCost)

The cost that is flowing out of or flowing into an account through an assignment .

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Cost on page 310

Driver Driven Cost (DrvDrvnCost)

The assigned cost that is flowing on an assignment path, including bothUsed Cost and Assigned Idle Cost .

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Cost on page 310

Driver Driven Quantity (DrvDrvnQty)

The assigned quantity that is flowing on an assignment path, including both Used Quantity and Assigned Idle Quantity.

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 8-byte, floating-point number.

Driver Formula (DrvFormula)

The formula that is attached to a source account's driver.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

This is a large text-based value.

When you import or export model data, this value must translate into the TEXT or NTEXT data types in Microsoft SQL Server.

See Also

Formula on page 319

Driver Idle Cost (DrvIdlCost)

A source account's Idle Cost. The assigned idle cost that is flowing on an assignment path.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Driver Cost on page 314

Driver Name (DrvName)

The name of the driver that is associated with an account.

When you add this property to a column in a grid, the default format is Text.

You can change this value only on accounts and external units.

The maximum length is 64 alphanumeric Unicode characters.

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.

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 8-byte, floating-point number.

Driver Quantity Basic (DrvQtyBasic)

The sum of the fixed and variable quantities that flows from one account to another.

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 8-byte, floating-point number.

See Also

- Driver Quantity Calculated on page 316
- Driver Quantity Fixed on page 317
- Driver Quantity Variable on page 317

Driver Quantity Calculated (DrvQtyCalc)

The calculated quantity that flows from one account to another.

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 8-byte, floating-point number.

See Also

- Driver Quantity Basic on page 316
- Driver Quantity Fixed on page 317
- Driver Quantity Variable on page 317

Driver Quantity Fixed (DQF)

The user-entered fixed quantity that flows from one account to another.

When you add this property to a column in a grid, the default format is Number.

You can change this value only on assigned cost elements with a driver that allows fixed driver quantities.

This is an 8-byte, floating-point number.

See Also

- Driver Quantity Basic on page 316
- Driver Quantity Calculated on page 316
- Driver Quantity Variable on page 317

Driver Quantity Variable (DQV)

The user-entered variable quantity that flows from one account to another.

When you add this property to a column in a grid, the default format is Number.

You can change this value only on assigned cost elements with a driver that allows variable driver quantities.

This is an 8-byte, floating-point number.

See Also

- Driver Quantity Basic on page 316
- Driver Quantity Calculated on page 316
- Driver Quantity Fixed on page 317

Driver Rate (DrvRate)

The calculated price per unit for assigned costs. The calculated cost per driver quantity unit for assignments. It is calculated according to the following formula:

```
Driver Rate = Drivable Cost / TDQ
```

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Driver Sequence Number (DrvSeq)

The sequence number that is associated with a source account's driver.

When you add this property to a column in a grid, the default format is Number.

This property is read-only in a grid. You cannot change this value in a grid.

This is a 4-byte integer value.

Driver Type (DrvType)

The type of driver: Basic, Bill of Cost, Calculated, Evenly Assigned, Percentage, Sales Volume, or Weighted. The default driver is Basic.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Driver Weight Fixed (DWF)

The numeric factor that modifies a fixed driver quantity. The default weight is 1.

When you add this property to a column in a grid, the default format is Number.

You can change this value only on assigned cost elements with a driver that allows fixed driver weights.

This is an 8-byte, floating-point number.

See Also

Driver Weight Variable on page 319

Driver Weight Variable (DWV)

The numeric factor that modifies a variable driver quantity. The default weight is 1.

When you add this property to a column in a grid, the default format is Number.

You can change this value only on assigned cost elements with a driver that allows variable driver weights.

This is an 8-byte, floating-point number.

See Also

Driver Weight Fixed on page 318

Entered Cost (EntCost)

The user-entered cost for a cost element or the calculated total of all account entered cost elements for an .

When you add this property to a column in a grid, the default format is Currency.

This is an 8-byte, floating-point number.

See Also

Cost on page 310

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.

When you add this property to a column in a grid, the default format is Text.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Variable Driver Quantity Override on page 338

Formula

The formula for a calculated driver or calculated attribute.

When you add this property to a column in a grid, the default format is Text.

This is a large text-based value.

When you import or export model data, this value must translate into the TEXT or NTEXT data types in Microsoft SQL Server.

See Also

Driver Formula on page 315

Has Assignments (HasAsgn)

Indicates whether an account receives costs from another account.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has Attributes (HasAttr)

Indicates whether an item has at least one attribute.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has BOC (HasBOC)

Indicates whether an account has at least one bill of costs.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has Entered Cost (HasEntCost)

Indicates whether an account has at least one user-entered cost element.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has Idle Cost (HasIdlCost)

Indicates whether an item has a non-zero idle cost.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has Notes (HasNotes)

Indicates whether an item has a Periodic Note.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

Has Used Cost (HasUsedCost)

Indicates whether an item has a non-zero used cost.

When you add this property to a column in a grid, the default format is Checkbox.

The value of this property is generated by the system. You cannot change this value.

This is a Boolean value.

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
```

This remaining cost can be assigned using a specific idle cost flow method (driver, percentage, user-entered, or evenly).

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Cost on page 310

- Idle Quantity on page 323
- Idle Percentage on page 323

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.

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 8-byte, floating-point number.

See Also

- Idle Driver Quantity UE on page 322
- Idle Flow Method on page 322

Idle Driver Quantity UE (IdlQtyEU)

The optional, user-entered idle driver quantity value for assigned cost elements and internal unit cost elements.

This value is used to drive idle costs that are based on source account's idle flow behavior.

When you add this property to a column in a grid, the default format is Number.

This is an 8-byte, floating-point number.

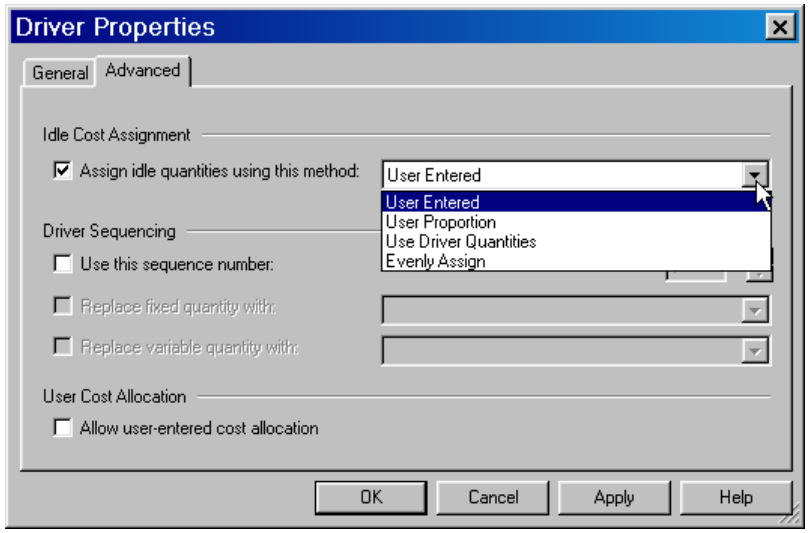
See Also

- Idle Driver Quantity on page 322
- Idle Flow Method on page 322

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.

When you add this property to a column in a grid, the default format is Text.

This property is read-only in a grid. You cannot change this value in a grid.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

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.

When you add this property to a column in a grid, the default format is Percentage.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Idle Cost on page 321
- Idle Quantity on page 323

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
```

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 8-byte, floating-point number.

See Also

- Idle Cost on page 321
- Idle Percentage on page 323

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.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Intersection Reference on page 324

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.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Intersection Name on page 324

Inventory Cost (InvCost)

The difference between the cost that is represented by the output quantity and the cost that is consumed.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Inventory Quantity on page 325

Inventory Quantity (InvQty)

The difference between the output quantity and what is consumed.

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 8-byte, floating-point number.

See Also

Inventory Cost on page 325

Module Type (ModType)

The type of module: Resource, Activity, Cost Object, External Units, or Profitability. (The Profitability value supports the Sales Volume driver.)

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

Name (Name)

The user-entered or default name of an item.

When you add this property to a column in a grid, the default format is Text.

You can change this value for any item except external unit cost elements, internal unit cost elements, and assignments, which are generated by the system.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Reference on page 331

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.

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 8-byte, floating-point number.

Output Quantity UE (OutQtyUE)

The user-entered output quantity.

When you add this property to a column in a grid, the default format is Number.

You can change this value only on accounts.

This is an 8-byte, floating-point number.

See Also

Output Quantity on page 326

Periodic Note (PerNote)

The period or scenario information that is attached to an item.

When you add this property to a column in a grid, the default format is Text.

The maximum length is 32,000 alphanumeric Unicode characters.

See Also

Has Notes on page 321

Profit (Profit)

The calculated difference between revenue and cost.

When you add this property to a column in a grid, the default format is Currency.

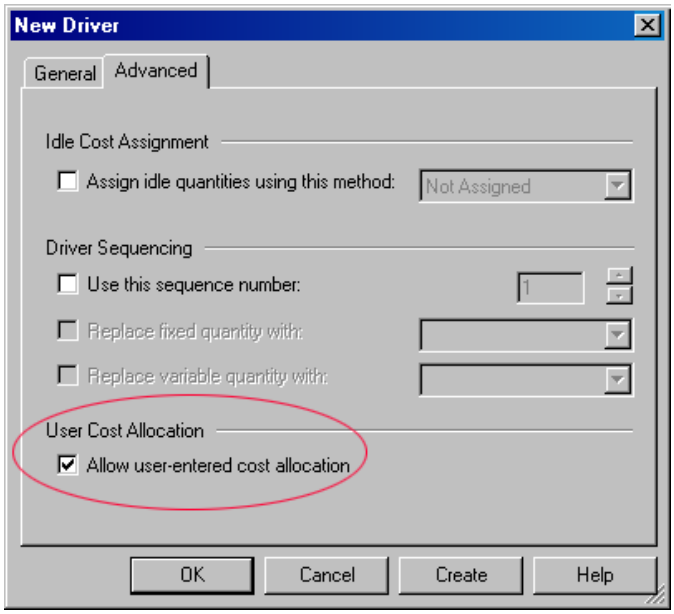
The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.



When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Received Assignment Cost (RcvAsgnCost)

The calculated cost of all assigned cost elements within an account.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

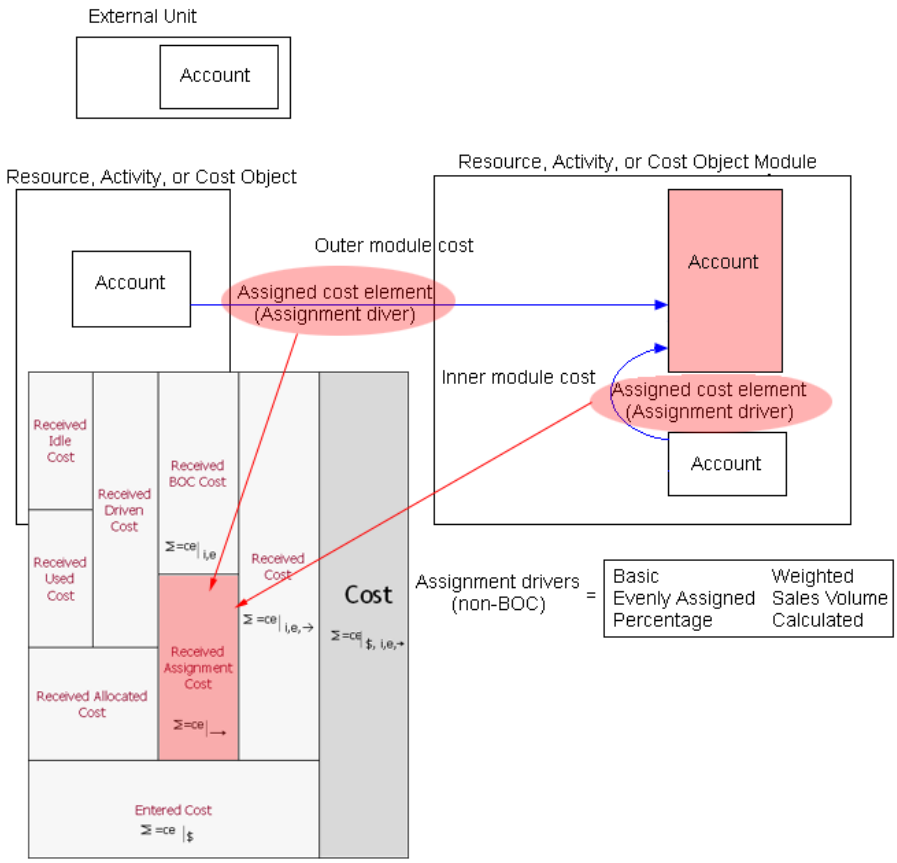
Assignment costs result from non-BOC driver assignments. They satisfy the following two conditions:

- The contributing account is in the Resource, Activity, or Cost Unit module.
- The driver is a non-BOC driver (i.e., one of the following: Basic, Calculated, Evenly Assigned, Percentage, Sales Volume, or Weighted).

The Received Assignment Cost is the sum of:

Cross module assignments
From an account in one module (Resource, Activity, or Cost Unit) to an account in a different module.

Inner module assignments
From an account in one module (Resource, Activity, or Cost Unit) to an account in the same module.



This is an 8-byte, floating-point number.

Received BOC Cost (RcvBOCCost)

The calculated cost of all cost elements that use the bill of cost driver within an account.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This property of an account is the sum of:

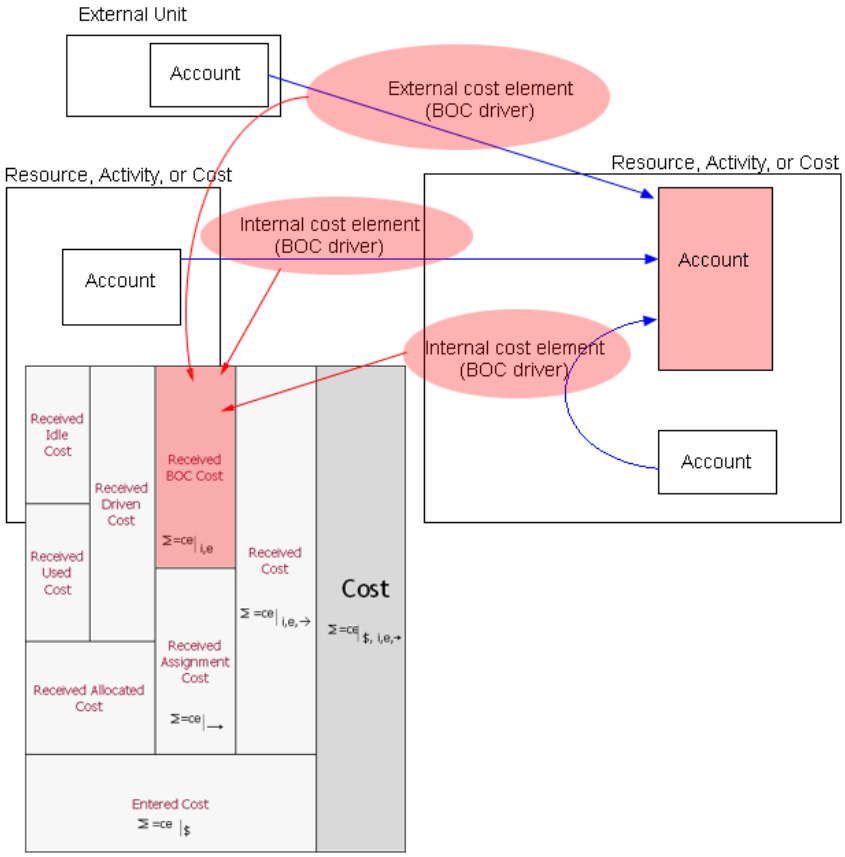
External Unit cost elements
Assignments, using a BOC driver, to that account from accounts in an External Unit.

All assignments from an External Unit use Bill of Costs (BOC) drivers.

Internal Unit cost elements
Assignments, using a BOC driver, to that account from accounts in the Resource, Activity, or Cost Object modules. Such assignments can be either:

Cross Module
From an account in one module to an account in a different module. Inner Module

Inner Module
From an account in one module to an account in the same module.



This is an 8-byte, floating-point number.

Received Cost (RcvCost)

The calculated cost that is received by an account from all assignments.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency .

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Received Reciprocal Cost on page 330

Received Reciprocal Cost (RcvRecipCost)

The calculated cost that an account receives from other accounts in the same .

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Received Non Reciprocal Cost on page 330

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.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

Reference (Reference)

The user-entered or default label that uniquely identifies an item.

When you add this property to a column in a grid, the default format is Text.

You can change this value for any item except external unit cost elements, internal unit cost elements, and assignments, which are generated by the system.

The maximum length is 64 alphanumeric Unicode characters.

When you mport or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Name on page 325

Revenue (Revenue)

The user-entered revenue.

When you add this property to a column in a grid, the default format is Currency.

This is an 8-byte, floating-point number.

Sold Quantity (SoldQty)

The user-entered number of units sold.

When you add this property to a column in a grid, the default format is Number.

This is an 8-byte, floating-point number.

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.

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 8-byte, floating-point number.

See Also

- Total Driver Quantity Basic (TDQBasic) on page 332
- Total Driver Quantity Calculated (TDQCalculated) on page 332

Total Driver Quantity Basic (TDQBasic)

The calculated quantity of all outgoing Basic driver quantities (Driver Quantity Basic) for an account.

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 8-byte, floating-point number.

See Also

- Total Driver Quantity (TDQ) on page 332
- Total Driver Quantity Calculated (TDQCalculated) on page 332
- Total Driver Quantity UE (TDQUE) on page 333

Total Driver Quantity Calculated (TDQCalc)

The calculated quantity of all outgoing Calculated driver quantities (Driver Quantity Calculated) for an account.

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 8-byte, floating-point number.

See Also

- Total Driver Quantity (TDQ on page 332

- "Total Driver Quantity UE (TDQUE)" on page 333
- Total Driver Quantity Basic (TDQBasic on page 332
- Total Driver Quantit UE (TDQUE) on page 333

Total Driver Quantity UE (TDQUE)

The user-entered total driver quantity.

When you add this property to a column in a grid, the default format is Number.

You cannot change this value for accounts with a driver type of Evenly Assigned, Sales Volume, or Percentage.

This is an 8-byte, floating-point number.

See Also

- Total Driver Quantity (TDQ on page 332
- "Total Driver Quantity UE (TDQUE)" on page 333
- Total Driver Quantity Basic (TDQBasic on page 332
- Total Driver Quantity Calculated (TDQCalculated) on page 332

Type (Type)

The type of item.

When you add this property to a column in a grid, the default format is Text.

The value of this property is generated by the system. You cannot change this value.

This is a 4-byte integer value.

Unassigned Cost (UnAsgnCost)

The calculated cost that does not flow out of an account.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Assigned Cost on page 308
- Unassigned Quantity on page 334

Unassigned Quantity (UnAsgnQty)

The calculated number of driver quantity units that do not cause cost on an account's outgoing assignments.

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 8-byte, floating-point number.

See Also

Unassigned Cost on page 333

Unique Driver Quantities (UniqDvrQty)

Indicates whether a driver quantity is not shared by assignments from other accounts.

When you add this property to a column in a grid, the default format is Checkbox.

This property is read-only in a grid. You cannot change this value in a grid.

This is a Boolean value.

Unit Cost (UnitCost)

The calculated cost of one unit. It is calculated according to the following formula:

```
Unit Cost = Cost/OutputQty
```

Unit Cost is used to drive Sales Volume Cost and External Bills of Cost.

When you add this property to a column in a grid, the default format is Currency.

You can change this value only for external unit cost elements. All other values for this property are generated by the system and you cannot change these values.

This is an 8-byte, floating-point number.

Unit Of Measure (UoM)

The user-entered name for the unit of measure for the output of an account.

When you add this property to a column in a grid, the default format is Text.

The maximum length is 64 alphanumeric Unicode characters.

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. It is calculated according to the following formula:

```
Unit Profit = Profit/SoldQty
```

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Unit Cost on page 334
- Unit Revenue on page 335

Unit Revenue (UnitRevenue)

The calculated revenue for a sold unit. It is calculated according to the following formula:

```
Unit Revenue = Revenue/SoldQty
```

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

- Unit Cost on page 334
- Unit Profit on page 335

Use Fixed Quantities (UseFixQty)

Indicates that Driver Quantity Fixed is editable on an assignment whose source account uses the fixed quantity driver.

When you add this property to a column in a grid, the default format is Checkbox.

This property is read-only in a grid. You cannot change this value in a grid.

This is a Boolean value.

See Also

- Use Variable Quantities on page 336
- Use Weighted Quantities on page 336

Use Variable Quantities (UseVarQty)

Indicates that Driver Quantity Variable is editable on an assignment whose source account uses the variable quantity driver.

When you add this property to a column in a grid, the default format is Checkbox.

This property is read-only in a grid. You cannot change this value in a grid.

This is a Boolean value.

See Also

- Use Fixed Quantities on page 335
- Use Weighted Quantities on page 336

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.

When you add this property to a column in a grid, the default format is Checkbox.

This property is read-only in a grid. You cannot change this value in a grid.

This is a Boolean value.

See Also

- Use Fixed Quantities on page 335
- Use Variable Quantities on page 336

Used Cost (UsedCost)

The calculated used cost for outgoing driver quantities and sold quantities for an account.

When you add this property to a column in a grid, the default format is Currency.

The value of this property is generated by the system. You cannot change this value.

This is an 8-byte, floating-point number.

See Also

Used Quantity on page 337

Used Quantity (UsedQty)

The calculated amount for outgoing driver quantities and sold quantities for an account.

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 8-byte, floating-point number.

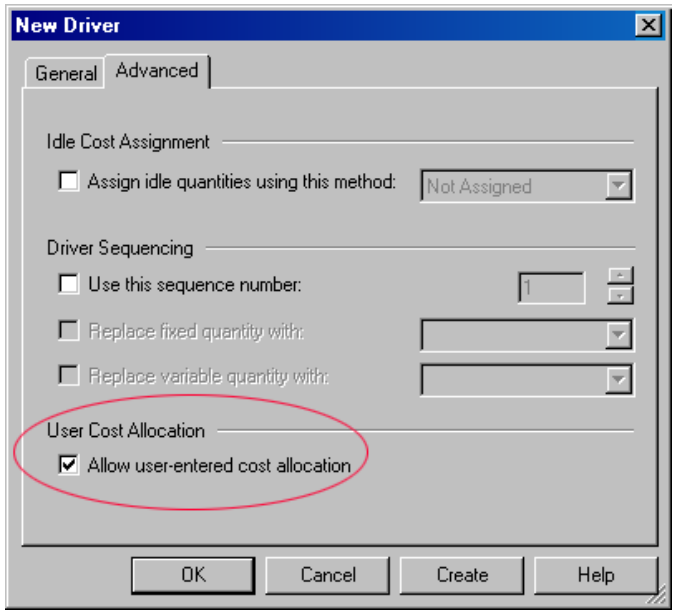
See Also

Used Cost on page 336

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**:

When you add this property to a column in a grid, the default format is Checkbox.

This is a Boolean value.

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.

When you add this property to a column in a grid, the default format is Text.

The maximum length is 64 alphanumeric Unicode characters.

When you import or export model data, this value must translate into a CHAR, NCHAR, VARCHAR, or NVARCHAR.

See Also

Fixed Driver Quantity Override on page 319

Your Turn

We welcome your feedback.

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

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn

sas® THE POWER TO KNOW®

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2009 SAS Institute Inc. All rights reserved. 518177_1US.0109