# Contents

## Part 1  Getting Started

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Activity-Based Management</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Activity-Based Costing</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Working in SAS Activity-Based Management</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Connecting to a Server</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Model Availability</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Saving Data</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Organization of Features</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Task Bars</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Status Bar</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Refreshing Information</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Canceling a Server Task</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>The Work Flow</td>
<td>12</td>
</tr>
</tbody>
</table>

## Part 2  Modeling Concepts

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Models</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Structural Information</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Guidelines for Creating the Model Structure</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Guidelines for Sharing Models, Configurations, and Data</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Calculating Costs</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>Periods and Scenarios</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Setting Up Periods and Scenarios</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Deleting Periods or Scenarios</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Guidelines for Creating Periods and Scenarios</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Period/Scenario Associations</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Periods and Scenarios in OLAP Analysis</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>Dimensions</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Dimension Levels</td>
<td>31</td>
</tr>
<tr>
<td>Dimension Members</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>Types of Dimensions</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Guidelines for Creating Dimensions</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Example: Multiple Dimensions for Better Analysis</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

**Chapter 7 Modules**

<table>
<thead>
<tr>
<th>Introduction</th>
<th>39</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>40</td>
</tr>
<tr>
<td>References</td>
<td>40</td>
</tr>
<tr>
<td>The Cost Object Module</td>
<td>40</td>
</tr>
<tr>
<td>The Activity Module</td>
<td>42</td>
</tr>
<tr>
<td>The Resource Module</td>
<td>43</td>
</tr>
<tr>
<td>The External Units Module</td>
<td>44</td>
</tr>
<tr>
<td>Guidelines for Creating the Modules</td>
<td>44</td>
</tr>
</tbody>
</table>

**Chapter 8 Accounts and Cost Elements**

<table>
<thead>
<tr>
<th>Accounts</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Elements</td>
<td>50</td>
</tr>
<tr>
<td>Validating a Model</td>
<td>50</td>
</tr>
</tbody>
</table>

**Chapter 9 Drivers**

<table>
<thead>
<tr>
<th>Introduction</th>
<th>53</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of Drivers</td>
<td>54</td>
</tr>
<tr>
<td>Default Driver</td>
<td>58</td>
</tr>
<tr>
<td>Types of Driver Quantities</td>
<td>58</td>
</tr>
<tr>
<td>Driver Sequencing</td>
<td>63</td>
</tr>
<tr>
<td>User-Entered Cost Allocation</td>
<td>64</td>
</tr>
<tr>
<td>Guidelines for Creating Drivers</td>
<td>65</td>
</tr>
</tbody>
</table>

**Chapter 10 Assignments**

<table>
<thead>
<tr>
<th>Introduction</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment Paths</td>
<td>67</td>
</tr>
<tr>
<td>Assignments Panes</td>
<td>68</td>
</tr>
<tr>
<td>Bills of Costs</td>
<td>68</td>
</tr>
<tr>
<td>Reciprocal Costing</td>
<td>71</td>
</tr>
<tr>
<td>Guidelines for Creating Assignments</td>
<td>72</td>
</tr>
</tbody>
</table>

**Chapter 11 Attributes**

<table>
<thead>
<tr>
<th>Introduction</th>
<th>73</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean Attributes</td>
<td>73</td>
</tr>
<tr>
<td>Dimension Attributes, Dimension Member Attributes, and Dimension Value Attributes</td>
<td>74</td>
</tr>
<tr>
<td>Numeric Attributes</td>
<td>75</td>
</tr>
<tr>
<td>Calculated Attributes</td>
<td>76</td>
</tr>
<tr>
<td>Text Attributes</td>
<td>76</td>
</tr>
<tr>
<td>Attribute Folders</td>
<td>77</td>
</tr>
<tr>
<td>Stages Attributes</td>
<td>77</td>
</tr>
</tbody>
</table>
Chapter 20   Working with Models  133
Introduction  133
Create a Model  133
Start Model Mode  142
Open a Model without a Model Already Open  142
Open a Model with a Model Already Open  142
Calculate Costs  143
Validate a Model  144

Chapter 21   Working with Periods and Scenarios  147
Introduction  147
Create a Period  147
Create a Scenario  149
Manage Period Level Names  150
Manage Scenario Level Names  152
Create a Period/Scenario Association  153
Copy Period/Scenario Data  154
Publish or Unpublish a Period/Scenario Association  154

Chapter 22   Working with Dimensions  155
Introduction  155
Create a Dimension Member  155
Delete a Dimension Member  157

Chapter 23   Working with Accounts and Cost Elements  159
Introduction  159
Create an Account  159
Create Cost Elements When an Account Is First Created  164
Create an Entered Cost Element on a Module Page  164

Chapter 24   Working with Drivers  167
Introduction  167
Create a Driver  167
Specify the Weights for a Weighted Driver  170
Specify the Default Driver  171
Associate a Driver Using an Account’s Item Properties  172
Associate a Driver Using a Module’s Grid  173

Chapter 25   Working with Assignments  175
Introduction  175
Create an Assignment  175
Show Only the Source Accounts for an Account  177
Show Only the Destination Accounts for an Account  177
Show the Source Accounts and the Destination Accounts for an Account  177
Go to an Account  178

Chapter 26   Working with Attributes  179
Introduction  179
Create an Attribute Folder  179
Chapter 31  Working with Importing Data and Exporting Data  229
Introduction  229
Import Model Data from a Database with the Import Data Wizard  229
Import Model Data from an XML File with the Import Data Wizard  239
Archive a Model to a Database with the Export Wizard  243
Archive a Model to an XML File with the Export Wizard  246
Export Model Data to a Database with the Export Wizard  250
Connect to a Database  263

Index  281
PART 1

Getting Started

Chapter 1. Overview  3
Chapter 2. Working in SAS Activity-Based Management  7
Chapter 3. Workspaces and Workspace Manager  13
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**

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:

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

**Licenses**

When a SAS Activity-Based Management administrator created your SAS Activity-Based Management account, you were assigned either a Designer license or a Viewer license. The SAS Activity-Based Management administrator also created roles for your organization and assigned one or more roles to you.

Only the SAS Activity-Based Management features that your combination of license and roles (with model permissions) allows you to use are available to you. If you do not know your license or your role, ask your SAS Activity-Based Management administrator.

**Designer**

With a Designer license, you can view and set up models and use SAS Activity-Based Management to import data. With a role that has Read and Write permissions, you can use SAS Activity-Based Management to calculate costs, to generate cubes, to set up reports, to run reports, and so on.
Viewer

With a Viewer license, you can use the models that have been set up by someone with a Designer license.
Connecting to a Server

Before you can use SAS Activity-Based Management, you must connect to a SAS Activity-Based Management server. When you specify the server name, you need to only specify the server name, not a protocol such as HTTP://. However, if your organization requires a secure connection, then you might need to specify a protocol such as HTTPS://.

Note: Your organization might have different requirements.

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 Activity-Based Management server.

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 your license and the status of the period/scenario association.)
- Unlike a desktop application in which your changes can be canceled, changes cannot be canceled.
  Because changes are saved for multiple users, your individual changes are immediately merged with other users' changes and cannot be separated. You can cancel a dialog box or wizard, but once you click **OK**, your changes are saved in the model.
- Unlike a desktop application, which stores a copy of your data in memory so that you can undo your changes, SAS 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

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

- **Home page, including Workspace Manager**
  
  On the Home page, you access shared information in Workspace Manager. You can start frequently used tasks. For more information, see Chapter 3, “Workspaces and Workspace Manager.”

- **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.”

- **Reports mode**
  
  In Reports mode, you open or create a report configuration. For more information, see Chapter 14, “Reporting Model Data.”

### Changing Modes

To change modes, use the links at the top of the window.

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 and/or a different period/scenario association. When you first change to another mode, you can select a model and a period/scenario association.

Mode Home Pages

Each mode contains a home page and one or more related pages. Each home page shows you a subset of a branch of Workspace Manager.

For example, the Model mode home page shows you the Models branch of Workspace Manager:

However, mode pages are different from Workspace Manager. On a mode page, you cannot edit the information for an item, and you cannot reorganize the folder structure in the left pane. However, you can sort the items in the right pane.

Task Bars

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

For example, the taskbar on the Resource module page contains these operations:
On pages other than the mode home pages, you can hide the taskbar in order to display more data.

**Status Bar**

The status bar, which is at the bottom of the window, displays informational messages, such as whether the current period/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 (يدة), 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

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.
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

Workspace Manager, which is located on the Home page, is your personalized view of a workspace. 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: 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

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’ roles.

To each role, 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 role, users who are members of that role cannot see the item.

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

Access Conflicts between Roles and Licenses

The access that is granted to a user based on a role sometimes conflicts with the access that is granted by a license. In that case, the most restrictive access applies.

For example, suppose that a user is a Viewer but the user is a member of a role that allows Read and Write access. In that case, the user is restricted to the access of a Viewer, which is Read-only access. Similarly, suppose that a user is a Designer but the user is a member of a role that allows Read-only access. In that case, the user is restricted to Read-only access, even though a Designer usually has Read and Write access.
Modeling Concepts

Chapter 4............ Models 19
Chapter 5............ Periods and Scenarios 25
Chapter 6............ Dimensions 31
Chapter 7............ Modules 39
Chapter 8............ Accounts and Cost Elements 47
Chapter 9............ Drivers 53
Chapter 10.......... Assignments 67
Chapter 11.......... Attributes 73
Chapter 12.......... Using Equations with Calculated Drivers and Calculated Attributes 79
Chapter 13.......... Analyzing Model Data with OLAP 81
Chapter 14.......... Reporting Model Data 87
Chapter 15.......... Column Layouts 91
Chapter 16.......... Currencies and Exchange Rates 97
Chapter 17.......... Importing Data and Exporting Data 99
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.”

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.”
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.”
- assignment paths that assign resource expenses to activities, and activity costs to products
  For more information, see Chapter 10, “Assignments.”
- 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.”
- 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

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.”

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
- the licenses that can use 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 home page 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 OLAP home page 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

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.”

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.”
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.
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

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.”

**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

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 “Status of Associations.”

*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.”)

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.
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.”

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.
As you drill down into the COST OBJECT 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.”

---

**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.”

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.”

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.”

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.”

---

**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:

![Diagram showing a single, structural dimension for North America, with sub-dimensions for North Carolina and New York, each with sub-dimensions for Personnel, Operating Expenses, and Equipment.](image)
In the Resource module, the structure looks like the following figure (with example values entered):

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE (PRIMARY PANE)</td>
<td>$5,000.00</td>
</tr>
<tr>
<td>North America</td>
<td>$5,000.00</td>
</tr>
<tr>
<td>North Carolina</td>
<td>$3,000.00</td>
</tr>
<tr>
<td>Personnel</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>Personnel_1017-1</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>Operating Expenses</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>Equipment</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>Equipment_1019-1</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>New York</td>
<td>$6,000.00</td>
</tr>
<tr>
<td>Personnel</td>
<td>$2,000.00</td>
</tr>
<tr>
<td>Personnel_1026-1</td>
<td>$2,000.00</td>
</tr>
<tr>
<td>Operating Expenses</td>
<td>$2,000.00</td>
</tr>
<tr>
<td>Equipment</td>
<td>$2,000.00</td>
</tr>
<tr>
<td>Equipment_1028-1</td>
<td>$2,000.00</td>
</tr>
</tbody>
</table>

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:
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, which are presented in Chapter 8, “Accounts and Cost Elements.”

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:

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Display Reference</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE (PRIMARY PANE)</td>
<td></td>
<td>$2,362,543.00</td>
</tr>
<tr>
<td>PERSONNEL</td>
<td>PER</td>
<td>$57,600.00</td>
</tr>
<tr>
<td>BUSINESS OPERATIONS</td>
<td>BUS</td>
<td>$1,573,543.00</td>
</tr>
<tr>
<td>INFORMATION SERVICES OPER</td>
<td>IS</td>
<td>$721,200.00</td>
</tr>
</tbody>
</table>

You can add or remove columns from the grid and specify how information in the columns appears. 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.”

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.

Note: The Display Name 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

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:

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Line</td>
<td>Shoes</td>
</tr>
<tr>
<td>Product Group</td>
<td>Summer Sandals</td>
</tr>
<tr>
<td>Product</td>
<td>Leather Weave</td>
</tr>
<tr>
<td>SKU (if needed)</td>
<td>Tan576830</td>
</tr>
</tbody>
</table>

Customers Dimension

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

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Group</td>
<td>Retail</td>
</tr>
<tr>
<td>Customer Category</td>
<td>Discount</td>
</tr>
<tr>
<td>Specific Customer</td>
<td>Big Mart</td>
</tr>
<tr>
<td>Location (if needed)</td>
<td>Phoenix</td>
</tr>
</tbody>
</table>

Channels Dimension

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

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Group</td>
<td>Traditional</td>
</tr>
<tr>
<td>Channel Category</td>
<td>Wholesale</td>
</tr>
<tr>
<td>Channel</td>
<td>Catalog</td>
</tr>
<tr>
<td>Channel Details (if needed)</td>
<td>Targeted Mail</td>
</tr>
</tbody>
</table>
The Activity Module

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:

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro Process</td>
<td>Receive Packages</td>
</tr>
<tr>
<td>Process</td>
<td>Collect by Region</td>
</tr>
<tr>
<td>Activity</td>
<td>Unload Trucks</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facility</td>
<td>North Carolina</td>
</tr>
<tr>
<td>Function</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>Department</td>
<td>Inspection</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Dimension Level Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL Account Group</td>
<td>Wages, Salaries, and Benefits</td>
</tr>
<tr>
<td>GL Account</td>
<td>Salaries and Wages</td>
</tr>
</tbody>
</table>

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.”

**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.”

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.”)

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.
Accounts

Accounts 47
Creating and Deleting Accounts 48
Guidelines for Creating Accounts 48
Guidelines for Creating Cost Object Accounts 49
Guidelines for Creating Activity Accounts 49
Guidelines for Creating Resource Accounts 49
Cost Elements 50
Entered Cost Element 50
Assigned Cost Element 50
Internal Unit Cost Element 50
External Unit Cost Element 50
Validating a Model 50

Accounts

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY (PRIMARY PANE)</td>
<td>All</td>
</tr>
<tr>
<td>Administration</td>
<td>Administration</td>
</tr>
<tr>
<td>Dept-2000</td>
<td></td>
</tr>
<tr>
<td>Support Web Site</td>
<td>2000-Support Web Site</td>
</tr>
<tr>
<td>Trade Show Activities</td>
<td>2000-Attend Trade Show</td>
</tr>
<tr>
<td>Preliminary RFQ</td>
<td>2000-Preliminary RFQ</td>
</tr>
<tr>
<td>Salary - Sale Manage</td>
<td>2000-Salary - Sale Mana</td>
</tr>
<tr>
<td>Salary - Direct Sales</td>
<td>2000-Salary - Direct Sale</td>
</tr>
<tr>
<td>Salary - Trade Show</td>
<td>2000-Salary - Advertisin</td>
</tr>
<tr>
<td>Deprec &amp; Facilities</td>
<td>2000-Deprec &amp; Facilities</td>
</tr>
<tr>
<td>Sales Call</td>
<td>2000-Sales Call</td>
</tr>
<tr>
<td>Take Order</td>
<td>2000-Take Order</td>
</tr>
<tr>
<td>Process Returns</td>
<td>2000-Process Returns</td>
</tr>
<tr>
<td>Engineering</td>
<td>Dept-4000</td>
</tr>
<tr>
<td>Accounting</td>
<td>Dept-5000</td>
</tr>
<tr>
<td>Production</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Ungrouped Accounts</th>
<th>Grouped Accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magazine Advertising Inserts</td>
<td>Advertising</td>
</tr>
<tr>
<td>Direct Mail Advertising</td>
<td>Print</td>
</tr>
<tr>
<td>In-Store Advertising Circulars</td>
<td>Magazine Inserts</td>
</tr>
<tr>
<td>Radio Advertising</td>
<td>Direct Mail</td>
</tr>
<tr>
<td>TV Advertising</td>
<td>In-Store Circulars</td>
</tr>
<tr>
<td>Yellow Pages Advertising</td>
<td>Radio</td>
</tr>
<tr>
<td></td>
<td>TV</td>
</tr>
<tr>
<td></td>
<td>Yellow Pages</td>
</tr>
</tbody>
</table>

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 advertising 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.”

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

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

Calculated Driver

A calculated driver (named Calculated) includes several factors that represent the relationship between accounts. The factors are determined by an equation that you define. An equation that includes numeric information about the destination accounts produces a calculated driver quantity from multiple variables. For information about equations and functions, see Chapter 12, “Using Equations with Calculated Drivers and Calculated Attributes.”

An equation for a calculated driver is not periodic; the equation 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 equation 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:

\[(\text{OutputQuantity}/\text{InventoryTurns}) \times \text{CuFtPerUnit}\]

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

\[(\text{OutputQuantity}/\text{InventoryTurns}) \times \text{Cost}\]

**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:

\[\text{NumberOfOrdersProcessed} \times \text{OrderComplexityByCustomer}\]

For `OrderComplexityByCustomer`, a different value could be assigned to each customer, and the same equation could be used for all customers.

**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’s not practical to determine the actual number of hours that clerical personnel spend on various tasks per year. The cost to determine the actual number of hours outweighs the benefit. To approximate the annual number of hours, management asks the clerical personnel to record their actual hours for one week. From these recordings, a percentage is assigned to each task. For this organization, a percentage is reasonably accurate and acceptable.

### Sales Volume Driver

A sales volume driver (named Sales Volume) is associated with an account that is the source of a profitability assignment.

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

### 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” later in this chapter.

**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

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 account’s unit cost is multiplied by the driver quantity (the property Driver Quantity Fixed (abbreviated DQF)). For information about source accounts and destination accounts, see Chapter 10, “Assignments.” 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:

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE PRI</td>
<td>$100.00</td>
</tr>
<tr>
<td>New York</td>
<td>$100.00</td>
</tr>
<tr>
<td>Salary</td>
<td>$100.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>InstrName</th>
<th>Cost</th>
<th>DQF</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York x Take Orders</td>
<td>$66.67</td>
<td>2.00</td>
</tr>
<tr>
<td>New York x Process Orders</td>
<td>$33.33</td>
<td>1.00</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Account</th>
<th>Unit Cost x DQF</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take Orders</td>
<td>$33.33 x 2.00</td>
<td>$66.67</td>
</tr>
<tr>
<td>Process Orders</td>
<td>$33.33 x 1.00</td>
<td>$33.33</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Account</th>
<th>DQV x OutQtyUE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take Orders</td>
<td>10.00 x 10.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Process Orders</td>
<td>3.00 x 10.00</td>
<td>30.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>130.00</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Account</th>
<th>Unit Cost x DQV x OutQtyUE</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take Orders</td>
<td>$00.7692 x 10.00 x 10.00</td>
<td>$76.92</td>
</tr>
<tr>
<td>Process Orders</td>
<td>$00.7692 x 3.00 x 10.00</td>
<td>$23.08</td>
</tr>
</tbody>
</table>

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:
The driver quantity for **Salary** is 133.00, which is calculated as follows:

<table>
<thead>
<tr>
<th>Account</th>
<th>(DQF) +</th>
<th>(DQV x OutQtyUE)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take Orders</td>
<td>2.00</td>
<td>10.00 x 10.00</td>
<td>102.00</td>
</tr>
<tr>
<td>Process Orders</td>
<td>1.00</td>
<td>3.00 x 10.00</td>
<td>31.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>133.00</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Account</th>
<th>(Unit Cost x DQF) +</th>
<th>(Unit Cost x DQV x OutQtyUE)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take Orders</td>
<td>$00.7519 x 2.00</td>
<td>$00.7519 x 10.00 x 10.00</td>
<td>$76.69</td>
</tr>
<tr>
<td>Process Orders</td>
<td>$00.7519 x 1.00</td>
<td>$00.7519 x 3.00 x 10.00</td>
<td>$23.31</td>
</tr>
</tbody>
</table>

*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

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:
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:

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:

<table>
<thead>
<tr>
<th>Display Name</th>
<th>IdlQty</th>
<th>TDQUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production</td>
<td>2.00</td>
<td>10.00</td>
</tr>
<tr>
<td>Stamping Parts</td>
<td>2.00</td>
<td>10.00</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>InstructionName</th>
<th>DQF</th>
<th>IdlQtyUE</th>
<th>IdlDrvQty</th>
<th>DrvIdlCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America x Front Fender</td>
<td>5.00</td>
<td>1.00</td>
<td>0.50</td>
<td>$50.00</td>
</tr>
<tr>
<td>North America x Rear Fender</td>
<td>3.00</td>
<td>2.00</td>
<td>1.50</td>
<td>$150.00</td>
</tr>
</tbody>
</table>

The system-generated value for the idle driver quantity (the column IdlDrvQty) corresponds to your entry. After calculating the costs, the idle driver 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.

<table>
<thead>
<tr>
<th>InstructionName</th>
<th>DQF</th>
<th>IdlQtyUE</th>
<th>IdlDrvQty</th>
<th>DrvIdlCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America x Front Fender</td>
<td>5.00</td>
<td>1.00</td>
<td>0.50</td>
<td>$50.00</td>
</tr>
<tr>
<td>North America x Rear Fender</td>
<td>3.00</td>
<td>3.00</td>
<td>1.50</td>
<td>$150.00</td>
</tr>
</tbody>
</table>

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)).

<table>
<thead>
<tr>
<th>InstrctnName</th>
<th>DQF</th>
<th>IdlQtｙUE</th>
<th>IdlDrvQty</th>
<th>DrvIdlCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America x Front Fender</td>
<td>5.00</td>
<td></td>
<td>1.25</td>
<td>$125.00</td>
</tr>
<tr>
<td>North America x Rear Fender</td>
<td>3.00</td>
<td></td>
<td>0.75</td>
<td>$75.00</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>InstrctnName</th>
<th>DQF</th>
<th>IdlQtｙUE</th>
<th>IdlDrvQty</th>
<th>DrvIdlCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America x Front Fender</td>
<td>5.00</td>
<td></td>
<td>1.00</td>
<td>$100.00</td>
</tr>
<tr>
<td>North America x Rear Fender</td>
<td>3.00</td>
<td></td>
<td>1.00</td>
<td>$100.00</td>
</tr>
</tbody>
</table>

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

*Driver sequencing* changes driver quantities to a property’s assigned costs from a previous calculation. As the following shows, the driver quantities that are used in the second calculation are based on the costs from the first calculation.
Driver 1 (D1): Percentage driver set for 25% for each account. Sequence #1.
Driver 2 (D2): Percentage driver set for 75% for each account. Sequence #2.

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:

  Account A → Account B → Account C
  Sequence 1    Sequence 2

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

---

## User-Entered Cost Allocation

User-entered cost allocation enables you to assign a specific cost to an assignment path.

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
Drivers

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.
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.”)

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, the Activity module page, and the 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.

In both the two-pane view and the three-pane view, costs flow from the left to the right. The space between the panes is called the *splitter bar*. Within the splitter bar, lines and arrows indicate the assignments between accounts.

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

### Bills of Costs

When you use the bill of costs driver to create an assignment between accounts, you create a bill of costs between accounts. A *bill of costs* (BOC) provides an easy way to add material costs and unit costs directly to accounts.

Use a bill of costs for these circumstances:

- **Costs outside of the 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.
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 driver.

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

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 driver quantities, see Chapter 9, “Drivers.”

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 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.)
Reciprocal Costing

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:

The equations:

A = 1,500 + .10B
B = 2,500 + .25A
X = .5A
Y = .25A + .9B

The results:

A = $1,794.87
B = $2,948.72
X = $897.44
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.
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.

Boolean Attributes

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

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Cost</th>
<th>Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST OBJECT (PRIMARY FANE)</td>
<td>$3,647,900.0</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>$3,647,900.0</td>
<td></td>
</tr>
<tr>
<td>Oregon</td>
<td>$3,647,900.0</td>
<td></td>
</tr>
<tr>
<td>Beaverton</td>
<td>$1,930,900.0</td>
<td></td>
</tr>
<tr>
<td>Drop Box</td>
<td>$15,468.16</td>
<td></td>
</tr>
<tr>
<td>3 Day Ground</td>
<td>$3.00</td>
<td></td>
</tr>
<tr>
<td>Overnight Express</td>
<td>$3.00</td>
<td></td>
</tr>
<tr>
<td>Standard Ground</td>
<td>$3.00</td>
<td></td>
</tr>
<tr>
<td>No &lt;Per WS&gt;</td>
<td>$5,450.00</td>
<td></td>
</tr>
<tr>
<td>southeast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jim's Sports</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Susie Sports</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midwest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Northeast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pacific Northwest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Southeast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>West Coast</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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**

A calculated attribute stores a number that is calculated from numeric properties of a model and/or from other calculated attributes, based on an equation that you define. For more information, see Chapter 12, “Using Equations with Calculated Drivers and Calculated Attributes.”

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 equation for a calculated driver, which remains the same for all periods, the equation 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:

\[
\text{Cost/"Orders Processed"}
\]

**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:

\[
\text{"Cases Loaded"/"FTE"}
\]

---

**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 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.”)

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

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.
Using Equations with Calculated Drivers and Calculated Attributes

Introduction

When you create a calculated driver or a calculated attribute, you must define the associated equation. An equation can contain the following elements:

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

For information about calculated drivers, see Chapter 9, “Drivers.” For information about calculated attributes, see Chapter 11, “Attributes.” For information about the functions you can use in equations, see the Help.

Defining Equations

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

If you use a numeric attribute in a calculated driver equation, 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

An equation 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 an equation, you see the notation Source. or Destination., followed by a property or attribute. For example, Source.Cost or Destination.UnitCost.

**Property Names in Equations**

The property names in equations 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 an equation.

Occasionally, the property name that is used in an equation contains an abbreviation, such as DimLevelName for the property Dimension Level Name, or TDQ for the property Total Driver Quantity.

**Testing Equations**

Testing an equation validates the operators and ensures that the syntax is correct. However, numeric properties and numeric attributes are not validated. Equation testing does not guarantee that the equation is valid or that the equation will yield the intended value.

During calculation, if an equation references an item that does not exist, a warning is displayed. If you use an invalid equation, SAS Activity-Based Management halts the calculation as soon as it encounters the invalid equation. 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 equation, 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"
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

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 on the OLAP page 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 on the OLAP page.
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 performs calculations that pre-aggregate numeric data to give you faster performance when you work on the OLAP page. 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.”

- the number of dimensions
  
  When you import model data, select fewer than six or eight dimensions. For information about dimensions, see Chapter 6, “Dimensions.”

- 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.”

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.”

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.”

**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 on the OLAP Page**

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 on the OLAP page 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 Page 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 page. 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 on the OLAP page.

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 page 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  87
  Model Size and Report Performance  87
  Report Header  88
  Setting up a Report  88
  Saving Report Data  88
  Exporting a Report  88
  Publishing a Report  88
  Report Configurations  89
    Saving a Report Configuration  89
    Selecting a Period/Scenario Association  89
  Report Templates  89
    Pre-Defined Report Templates  89

Reports  87

A report is created as an Adobe Acrobat PDF file.

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

As a Viewer, you can view, print, and export a report set up for you by a Designer. If you want to be able to set up reports yourself, ask your SAS Activity-Based Management administrator to assign a Designer license to you.

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>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Name</td>
<td>the model selected for the report</td>
</tr>
<tr>
<td>Module</td>
<td>one or more modules selected for the report; each module starts on a new page</td>
</tr>
<tr>
<td>Period</td>
<td>the period selected for the report</td>
</tr>
<tr>
<td>Scenario</td>
<td>the scenario selected for the report</td>
</tr>
<tr>
<td>View Perspective</td>
<td>the dimension selected for the report</td>
</tr>
<tr>
<td>Filtered</td>
<td>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</td>
</tr>
</tbody>
</table>

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.

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.”

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.
Introduction  91
Showing Multiple Period/Scenario Associations  92
Default Column Layout  92
Formatting Columns  92
   Color, Style, and Alignment  93
   Type  93
      Currency Type  93
      Number Type  93
      Percentage Type  94
      Text Type  94
      Checkbox Type  94
   Available Format Types  94
   Negative Numbers  94
   Precision  94
   Currency Formatting  95
   Displaying the Currency Symbol  95
Saving a Column Layout  95

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

For more information, see Chapter 7, “Modules.”

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.

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Display Reference</th>
<th>2003 Cost</th>
<th>2001 Cost</th>
<th>2000 Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource (Primary Pane)</td>
<td></td>
<td>$2,417,843.00</td>
<td>$2,352,343.00</td>
<td>$2,366,843.00</td>
</tr>
<tr>
<td>Personnel</td>
<td>PER</td>
<td>$123,100.00</td>
<td>$57,500.00</td>
<td>$72,100.00</td>
</tr>
<tr>
<td>Compensation</td>
<td>PER-COMP</td>
<td>$104,000.00</td>
<td>$38,500.00</td>
<td>$53,000.00</td>
</tr>
<tr>
<td>Salaries - Personnel</td>
<td>PER-SALARIES</td>
<td>$104,000.00</td>
<td>$38,500.00</td>
<td>$53,000.00</td>
</tr>
<tr>
<td>Temp Salaries</td>
<td>510-8110350</td>
<td>$0.00</td>
<td>$1,000.00</td>
<td>$15,000.00</td>
</tr>
<tr>
<td>Part Time Salaries</td>
<td>510-8110200</td>
<td>$4,000.00</td>
<td>$2,500.00</td>
<td>$5,000.00</td>
</tr>
<tr>
<td>Non-Exempt Salaries</td>
<td>510-8110100</td>
<td>$8,000.00</td>
<td>$10,000.00</td>
<td>$8,000.00</td>
</tr>
<tr>
<td>Exempt Salaries</td>
<td>510-8110050</td>
<td>$75,000.00</td>
<td>$25,000.00</td>
<td>$25,000.00</td>
</tr>
</tbody>
</table>

### 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 29, “Working with Column Layouts.”

### Formatting Columns

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 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.”

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:

<table>
<thead>
<tr>
<th>Underlying data</th>
<th>Available format types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Text</td>
</tr>
<tr>
<td>Cost or rate</td>
<td>Currency (default)</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td>Number or quantity</td>
<td>Number (default)</td>
</tr>
<tr>
<td></td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td>Percentage</td>
</tr>
<tr>
<td>Percentage</td>
<td>Percentage (default)</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td>Boolean</td>
<td>Checkbox</td>
</tr>
</tbody>
</table>

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: 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.

<table>
<thead>
<tr>
<th>Change this...</th>
<th>In this dialog box...</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive currency format</td>
<td>Customize Regional Options</td>
<td>1,1 F</td>
</tr>
<tr>
<td>Negative currency format</td>
<td>Customize Regional Options</td>
<td>-1,1 F</td>
</tr>
<tr>
<td>Decimal symbol</td>
<td>Customize Regional Options</td>
<td>,</td>
</tr>
<tr>
<td>No. of digits after decimal</td>
<td>Customize Regional Options</td>
<td>2</td>
</tr>
<tr>
<td>Digit grouping symbol</td>
<td>Customize Regional Options</td>
<td>a space</td>
</tr>
<tr>
<td>Digit grouping</td>
<td>Customize Regional Options</td>
<td>123 456 789</td>
</tr>
</tbody>
</table>

### 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
Currencies and Exchange Rates

Currencies

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 30, “Working with Currencies.”

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 30, “Working with Currencies.”

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 30, “Working with Currencies.”
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 31, “Working with Importing Data and Exporting Data.”

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

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:

- an Oros model (not addressed in this document)
- 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

The following steps describe how to import data:

1. Verify that a Designer license has been assigned to you.
2. Import the data. You can choose to import the entire data set at once, or you can import the data in groups and populate the model in the following general steps:

   **Import 1**: The first import step defines the existence of the model and includes the Dimension, DimensionOrder, Period, and Scenario tables. This step is equivalent to the finishing point of the New Model Wizard when you interactively build a model.

   **Import 2**: The second import step defines the model’s content (resources, activities, and cost objects) and includes the Account, DimensionLevel, and DimensionMember tables. This step is equivalent to the finishing point of the New Account Wizard when you interactively create accounts. The Account table includes the revenue and sold quantities.

   **Import 3**: The third import step loads costs into the model and includes the EnteredCostElement and ExternalUnit tables. This step is equivalent to the point where you create cost elements in the New Account Wizard when you interactively build a model.

   **Import 4**: The fourth import step flows costs through the model using assignments with quantities for flow calculation. This step includes the Assignment and Driver tables. 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.
3 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 31, “Working with Importing Data and Exporting Data.”

**Import Process Summary**

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

<table>
<thead>
<tr>
<th>Table name</th>
<th>Requires dimension signature</th>
<th>Requires reference numbers</th>
<th>Has multiple keys?</th>
<th>Import step order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 1</td>
</tr>
<tr>
<td>DimensionOrder</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 1</td>
</tr>
<tr>
<td>Period</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 1</td>
</tr>
<tr>
<td>Scenario</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 1</td>
</tr>
<tr>
<td>Account*</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 2</td>
</tr>
<tr>
<td>DimensionLevel</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 2</td>
</tr>
<tr>
<td>DimensionMember</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 2</td>
</tr>
<tr>
<td>EnteredCostElement</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 3</td>
</tr>
<tr>
<td>ExternalUnit</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 3</td>
</tr>
<tr>
<td>Assignment</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 4</td>
</tr>
<tr>
<td>Driver</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 4</td>
</tr>
<tr>
<td>DimensionAttributeAssociation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 5</td>
</tr>
<tr>
<td>ValueAttribute</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Import 5</td>
</tr>
<tr>
<td>ValueAttributeAssociation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Import 5</td>
</tr>
<tr>
<td>AssignmentNonUnique</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Optional</td>
</tr>
<tr>
<td>CurrencyRate</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
</tr>
<tr>
<td>PeriodLevel</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
</tr>
<tr>
<td>ScenarioLevel</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
</tr>
</tbody>
</table>

*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 the “Understanding Keys” section later in this chapter.
Updating a Model with New Structure and New Periodic Values

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

<table>
<thead>
<tr>
<th>Table name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
</tr>
<tr>
<td>Scenario</td>
</tr>
<tr>
<td>Account</td>
</tr>
<tr>
<td>DimensionLevel</td>
</tr>
<tr>
<td>DimensionMember</td>
</tr>
<tr>
<td>EnteredCostElement</td>
</tr>
<tr>
<td>ExternalUnit</td>
</tr>
<tr>
<td>Assignment</td>
</tr>
<tr>
<td>Driver</td>
</tr>
<tr>
<td>DimensionAttributeAssociation</td>
</tr>
<tr>
<td>ValueAttribute</td>
</tr>
<tr>
<td>ValueAttributeAssociation</td>
</tr>
<tr>
<td>ValueAttributePeriodicDef</td>
</tr>
</tbody>
</table>

Updating a Model with New Periodic Values

The following table lists the required tables for updating a model with new periodic values:

<table>
<thead>
<tr>
<th>Table name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
</tr>
<tr>
<td>EnteredCostElement</td>
</tr>
<tr>
<td>ExternalUnit</td>
</tr>
<tr>
<td>Assignment</td>
</tr>
<tr>
<td>ValueAttributeAssociation</td>
</tr>
<tr>
<td>ValueAttributePeriodicDef</td>
</tr>
</tbody>
</table>

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.”

### Importing Data from a Database

Before you import data from a database, ensure that the SAS Activity-Based Management server can access the database. Then, review the information about connection strings that is presented in the “Connection Strings” section later in this chapter. To import the data, you can use the wizard, or you can write a program.

You can import information from any data source that supports OLEDB. The following types of data sources are the most common:

- SAS
- Microsoft SQL Server
- Oracle
- Microsoft Access
- Microsoft Excel workbook (spreadsheet ranges)
- SAS Activity-Based Management Adapter for SAP R/3

### 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 **IntsectnName**) or the intersection reference (the column **IntsectnRef**) properties.
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>
<thead>
<tr>
<th>Property</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DimRef1</td>
<td>Region</td>
</tr>
<tr>
<td>DimMemberRef1</td>
<td>Beaverton, Eugene</td>
</tr>
<tr>
<td>DimRef2</td>
<td>General Ledger Account (GL)</td>
</tr>
<tr>
<td>DimMemberRef2</td>
<td>Wages, Operating Expenses, Equipment Expenses</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>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.</td>
</tr>
<tr>
<td>Period</td>
<td>Specifies the period names, the descriptions, and the start dates and the end dates.</td>
</tr>
<tr>
<td></td>
<td>If the required periods already exist on the SAS Activity-Based Management server, then this table is not required when importing data.</td>
</tr>
<tr>
<td>PeriodLevel</td>
<td>Specifies the level names for period hierarchies.</td>
</tr>
<tr>
<td></td>
<td>If the required period levels already exist on the SAS Activity-Based Management server, then this table is not required when importing data.</td>
</tr>
<tr>
<td>Scenario</td>
<td>Specifies the scenario names and the descriptions.</td>
</tr>
<tr>
<td></td>
<td>If the required scenarios already exist on the SAS Activity-Based Management server, then this table is not required when importing data.</td>
</tr>
<tr>
<td>ScenarioLevel</td>
<td>Specifies the level names for scenario hierarchies.</td>
</tr>
<tr>
<td></td>
<td>If the required scenario levels already exist on the SAS Activity-Based Management server, then this table is not required when importing data.</td>
</tr>
<tr>
<td>CurrencyRate</td>
<td>Specifies currency exchange rates for each period.</td>
</tr>
<tr>
<td></td>
<td>If multiple currencies are not required in the model, then this table is not required when importing data.</td>
</tr>
<tr>
<td>Dimension</td>
<td>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 Boolean attributes. This table is required when importing data.</td>
</tr>
<tr>
<td>DimensionOrder</td>
<td>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.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DimensionLevel</td>
<td>Specifies level names in a dimension hierarchy. This table is required when importing data.</td>
</tr>
<tr>
<td>DimensionMember</td>
<td>Specifies the hierarchy for each dimension. This table is required when importing data.</td>
</tr>
<tr>
<td>Account</td>
<td>Specifies the dimension intersections in the Resource module, the Activity module, and the Cost Object module. The combination of the columns DimRef(&lt;n&gt;) and DimMemberRef(&lt;n&gt;) 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.</td>
</tr>
<tr>
<td>ExternalUnit</td>
<td>Specifies the dimension intersections for the External Units module. The combination of the columns DimRef(&lt;n&gt;) and DimMemberRef(&lt;n&gt;) 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.</td>
</tr>
<tr>
<td>EnteredCostElement</td>
<td>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(&lt;n&gt;) and DimMemberRef(&lt;n&gt;). If the model does not use entered cost elements, then this table is not required when importing data.</td>
</tr>
<tr>
<td>Driver</td>
<td>Specifies the drivers. This table is required when importing data.</td>
</tr>
<tr>
<td>Assignment</td>
<td>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(&lt;n&gt;), SourceDimMemberRef(&lt;n&gt;), and DestinationDimRef(&lt;n&gt;), and DestinationDimMemberRef(&lt;n&gt;). This table is required when importing data.</td>
</tr>
<tr>
<td>AssignmentNonUnique</td>
<td>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(&lt;n&gt;) and DimMemberRef(&lt;n&gt;). This table is not required when importing data because the unique and nonunique driver quantities can be specified in the Assignment table.</td>
</tr>
<tr>
<td>ValueAttribute</td>
<td>Specifies the attribute hierarchy for numeric attributes, text attributes, and Boolean attributes. This table is not required when importing data.</td>
</tr>
<tr>
<td>ValueAttributePeriodicDef</td>
<td>Specifies the default value and formula for the numeric attributes. This table is required when importing data that has value attributes.</td>
</tr>
</tbody>
</table>
### Table Name Description

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueAttributeAssociation</td>
<td>Specifies the accounts that are associated with numeric attributes, text attributes, and Boolean attributes. The accounts can be specified with the column ItemReference or the columns DimRef&lt;n&gt; and DimMemberRef&lt;n&gt;. This table is not required when importing data.</td>
</tr>
<tr>
<td>DimensionAttributeAssociation</td>
<td>Specifies the accounts that are associated with dimension attributes. The accounts can be specified with the column ItemReference or the columns DimRef&lt;n&gt; and DimMemberRef&lt;n&gt;. This table is not required when importing data.</td>
</tr>
</tbody>
</table>

### 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 31, “Working with Importing Data and Exporting Data.”

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 ValueAttributeAssociation tables) to include all of the dimensions that you used in the model that you want to import.

<table>
<thead>
<tr>
<th>Number of dimensions in the imported model</th>
<th>Difference in the number of dimensions in the sample model</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Units module: two dimensions</td>
<td>Add one dimension</td>
</tr>
<tr>
<td>Resource module: one dimensions</td>
<td>Remove one dimension</td>
</tr>
<tr>
<td>Activity module: three dimensions</td>
<td>Add one dimension</td>
</tr>
<tr>
<td>Cost Object module: five dimensions</td>
<td>Add two dimensions</td>
</tr>
</tbody>
</table>

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 API to 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 API, see the SAS Activity-Based Management product page on your local server: http://<Your_Server_Name>/SasSolutions/ABM/.

**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
Importing Data from an XML File

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

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.

<table>
<thead>
<tr>
<th>Reason for exporting model data</th>
<th>Export destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>To export model data and to manipulate the data, and then to import the data back into the model or into another model</td>
<td>Database</td>
</tr>
<tr>
<td>To archive a model</td>
<td>Database</td>
</tr>
<tr>
<td>To export some of the items in the model, but to exclude other items</td>
<td>Database</td>
</tr>
<tr>
<td></td>
<td>XML file (requires less disk space)</td>
</tr>
</tbody>
</table>

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
Exporting Model Data

Chapter 17

- 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.”

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
- Entered values: DestinationSoldQuantity
- Calculated values: ContribCost, ContribPcnt, DestinationCost, DestinationOutputQuantity
- Attributes specific to the model design for each beginning or final account (resources, cost objects)
  - Dimensional attributes—used for grouping
  - Numerical attributes—entered values
  - Calculated attributes—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.”
<table>
<thead>
<tr>
<th>Report</th>
<th>Number of standard report templates</th>
<th>Multiple currencies supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination furthest</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dimensional attribute cost</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Dimensional attribute unit cost</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Dimensional view</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Driver - cost and rate</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Idle capacity</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Module hierarchy</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Multi-level contributions</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Multi-level contributions with cost elements</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Profit and loss – depth – horizontal</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Profit and loss – depth – vertical</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Profit and loss – resource – horizontal</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Profit and loss – resource – vertical</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Profit cliff</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Resource contributions</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Resource contributions by attribute</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Resource contributions with cost elements</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Single-stage assignments</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Single-stage contributions</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Unassigned costs</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Unit cost</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total number of export report tables</td>
<td>21</td>
<td>7</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Field type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIKE</td>
<td>Wildcard</td>
<td>Text</td>
</tr>
<tr>
<td></td>
<td>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 (<em>) to specify a single character. For example, Name LIKE account</em> 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”.</td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>Equal</td>
<td>Text or numeric</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal</td>
<td>Text or numeric</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td>Numeric</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>Numeric</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
<td>Numeric</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

## Connection Strings

When you import data from a database or when you export data to a database, you must set up a connection string to the database. If possible, you should name your source tables and fields so that they match the target tables and fields. Matching the names enables the Import Wizard to automatically map the source tables and fields to the target tables and fields.

OLEDB is Microsoft’s system-level data access interface to various database management system services. OLEDB can access both relational data sources and non-relational data sources and provides access to a particular data source by using a data provider. The data provider is often referred to as an OLEDB provider.

*Note: There are many data providers. This section discusses only the most common data providers. The data provider that you must use depends on the database to which you want to connect.*
For SAS Activity-Based Management, the most common OLEDB providers include:

- Microsoft OLEDB Provider for SQL Server, which provides access to databases that were built with Microsoft SQL Server versions 6.5 and 7.0.
- Oracle Provider for OLEDB, which provides access to Oracle databases. (An Oracle client must be installed on the same computer.)
- Microsoft Jet 4.0 OLEDB Provider, which provides access to data in Microsoft Access databases, as well as access to tabular data that is stored in Microsoft Excel workbooks.

**Connecting to a Microsoft Access Database or to a Microsoft Excel Workbook**

The SAS Activity-Based Management architecture is client/server; therefore, it is important to understand how the process of importing data or exporting data relates to file sharing and security settings in Microsoft Windows.

To connect to a Microsoft Access database or to a Microsoft Excel workbook, the Import Wizard or Export Wizard on a SAS Activity-Based Management client must build a standard OLEDB connection string to the data source. Although the OLEDB connection string is created on a SAS Activity-Based Management client, the connection string is passed to the SAS Activity-Based Management server to be processed. It is the server, not the client that actually connects to the data source. Therefore, both the client and the server must be able to access the data source through the same path. The requirement that both the client and the server use the same path means that the path cannot be a relative path. The easiest path to use is a Universal Naming Convention (UNC) path. For example, `\ServerName\Stagetables`.

Further, when the SAS Activity-Based Management server connects to a Microsoft Access database or to a Microsoft Excel workbook, the server must use the ASP.NET user account. The ASP.NET user account allows connections only to files that are located on the SAS Activity-Based Management server. Therefore, the Microsoft Access database or the Microsoft Excel workbook must be located on the SAS Activity-Based Management server.
Building a Model

Chapter 18........Working in SAS Activity-Based management 125
Chapter 19........Working with Workspaces and Workspace Manager 129
Chapter 20........Working with Models 133
Chapter 21........Working with Periods and Scenarios 147
Chapter 22........Working with Dimensions 155
Chapter 23........Working with Accounts and Cost Elements 159
Chapter 24........Working with Drivers 167
Chapter 25........Working with Assignments 175
Chapter 26........Working with Attributes 179
Chapter 27........Working with OLAP 187
Chapter 28........Working with Reports 195
Chapter 29........Working with Column Layouts 215
Chapter 30........Working with Currencies 223
Chapter 31........Working with Importing Data and Exporting Data 229
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.”

Connect to a Server

1. Start SAS Activity-Based Management. The Connect dialog box appears.

   ![Connect dialog box](image)
2 To log on using a network account other than your current network logon information (credentials), clear the Log in using my current credentials option.

3 To select a different server, select a server from the Server drop-down list, or type the server name.

4 If you cleared the Log in using my current credentials option, then do the following:
   a Type your domain and User name. For example, HQ\AlexW.
   b Type your Password.

5 If multiple languages are installed on your computer, select the Language that you want to use for the interface.

   Note: If only one language is installed, the Language drop-down list is not displayed.

---

**Refresh Information**

Click ![refresh icon](#).

---

**Resource module: Drivers**

After the information is updated, the icon returns to its usual state ![refresh icon](#).

---

**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.
Introduction

This chapter contains task-based information about workspaces and Workspace Manager. For more conceptual information, see Chapter 3, “Workspaces and Workspace Manager.”

Create a Shortcut to a Workspace Item

1. Click the Home link.
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.
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.**
Introduction

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

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.” For information about dimensions, see Chapter 6, “Dimensions.” For information about the modules, see Chapter 7, “Modules.”

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**.
From the Select the initial period list, select a period. For information about periods, see Chapter 5, “Periods and Scenarios.”

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

2 Type the **Description**.

3 Click **OK**.

4 From the **Select the initial scenario** list, select a scenario. For information about scenarios, see Chapter 5, “Periods and Scenarios.”

5 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.”

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.
6 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.”

### Start Model Mode

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

Click \[\text{Model}\] at the top of the SAS Activity-Based Management window.

### 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 Model Home page, as shown:

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.”

1. From the list of \text{Folders}, select a folder.
2. From the list of \text{Models}, click a model.

\textit{Tip:} Alternatively, you can click \[\text{Open}\].

### Open a Model with a Model Already Open

1. From the \text{Model} drop-down list, select a model.

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.”
2 From the Period/Scenario drop-down list, select a period/scenario association.

3 Click .

The model opens and the Resource module page is displayed.

---

**Calculate Costs**

1 Open a model in Model mode, and select **Model > Calculate Costs and Generate Cubes**.

The Calculate Costs, Generate Fact Tables, and Generate Cubes dialog box appears.

2 For **Perform this action**, select the **Calculate costs** option.

3 From the **Using this period/scenario association** drop-down list, select either a period/scenario association or <All>.
4 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.”

5 For **Errors and Warnings**, select an option.

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

---

**Validate a Model**

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

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.”
Introduction

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

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.

   ![Manage Scenarios dialog box]

2. From the **Scenario** column, select a scenario.

3. Click **Levels**.
   
The Scenario Level Names dialog box appears.

   ![Scenario Level Names dialog box]

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

Create a Period/Scenario Association

1 Select Model > 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.

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

Copy Period/Scenario Data

1 Select Model > Copy Period/Scenario Data.
   The Copy Period/Scenario Data dialog box appears.

   ![Copy Period/Scenario Data Dialog Box]

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.
Introduction

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

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.”

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**.
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."

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.


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:

Create accounts using: **parent dimension members only**

**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.

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.

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Northeast x Salaries_1021</td>
<td>Northeast x Salaries_1021</td>
<td></td>
</tr>
<tr>
<td>Salaries_1021-1</td>
<td>Salaries_1021_1021-1</td>
<td></td>
</tr>
</tbody>
</table>

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 element's 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**.
Introduction

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

Create a Driver

1. Open a model in Model mode, and select Model > Drivers Page.
   The Drivers page appears.

2. Select DRIVERS.

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 **Equation Editor.**

   The Equation Editor dialog box appears.
Tip: You can also modify the equation in the **Equation** box.

For information about equations, see Chapter 12, “Using Equations with Calculated Drivers and Calculated Attributes.”

8 Define an equation:

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 equation.

*Tip:* You can also type the equation.

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

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

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

f Click **Test**.

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

9 Click the **Advanced** tab.
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.

Specify driver sequencing:

- Either select or clear the **Use this sequence number** option. If you select this option, then specify the sequence number in the box provided.

- 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.

- 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.

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.
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.
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. From the **Properties** list, locate the **Driver Name** property.

4. 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.

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Display Reference</th>
<th>Cost</th>
<th>DrvName</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY (PRIMARY PA)</td>
<td>New York</td>
<td>$100.00</td>
<td></td>
</tr>
<tr>
<td><em>Take Orders</em></td>
<td><em>Take Orders</em></td>
<td>$96.15</td>
<td><em>Alloc User Entered</em></td>
</tr>
<tr>
<td>Process Orders</td>
<td>Process Orders</td>
<td>$3.85</td>
<td></td>
</tr>
</tbody>
</table>

- Basic Fixed
- Basic Fixed and Variable
- Basic Non-Unique
- Basic Unique
- Basic Variable
- Bill of Cost
- BOC Fixed
- BOC Fixed and Variable
- BOC Variable
- Evenly Assigned
- Percentage
Introduction

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

Create an Assignment

1. Decide which assignments panes to show.

2. Select Model > Assignments > Add Accounts in Left Pane (or Add Accounts in Right Pane).

   The Add Accounts for Assignments dialog box appears.
From **Show accounts from this module**, select an option.

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.

**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.

*Tip:* To quickly create assignments to many accounts, select **Model > Assignments > Assign All Left, Assign All Right, or Assign All Left and Right**.
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.
Introduction

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

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 **Equation Editor**.

      The Equation Editor dialog box appears.
Tip: You can also modify the equation in the Equation box. For information about equations, see Chapter 12, “Using Equations with Calculated Drivers and Calculated Attributes.”

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 equation.

Tip: You can also type the equation.

e Continue to insert items until you have built the entire equation.

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

g To delete the entire equation, click Clear.

h Click Test.

If no errors are displayed, the syntax of the equation is valid.
Add an Attribute to an Account

1. On a module page, select an 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRAIBUTES</td>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value Added</td>
<td>Value Added</td>
<td>Text</td>
<td>New York x Take Orders 1021</td>
</tr>
<tr>
<td>Non Value Added</td>
<td>Non Value Added</td>
<td>Text</td>
<td>New York x Process Orders 1022</td>
</tr>
</tbody>
</table>

### Remove an Attribute from an Account

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. Click **< Remove**.
   
The attribute is removed from the **Attribute hierarchy** list.
Introduction

This chapter contains task-based information about working with OLAP. For more conceptual information, see Chapter 13, “Analyzing Model Data with OLAP.”

Start OLAP Mode

Click at the top of the SAS Activity-Based Management window.

Create an OLAP View

1 Select File > New > OLAP View.

The New OLAP View Wizard appears.
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 OLAP home page.
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.

Specify the Properties for the Multi-Stage Contributions Cube

1. Open a model in Model mode, and select Model > Properties.
   The Model Properties dialog box appears.

2. Click the Cube tab.
3 To define each module as a stage, select the **Define stages by each module** option.

4 To use the Stages dimension attribute folder to define stages, do the following:
   a Select the **Define stages by a dimension attribute named Stages** option.
      For information about the Stages attributes, see Chapter 11, “Attributes.”
   b Select or clear the **Include external units** option.

5 In the **Use the cost as it flows in or out of each stage** list, select the **Cost Flow** for each module.

---

**Generate Cubes**

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

1 Select **Model > Calculate Costs and Generate Cubes**.

The Calculate Costs, Generate Fact Tables, and Generate Cubes dialog box appears.
2. For **Perform this action**, select the **Generate cubes** option.

3. To force costs to be calculated, select the **Calculate costs** option.

4. From the **Using this period/scenario association** drop-down list, select either a period/scenario association or `<All>`.

5. Select one or more options in the **Generate these cubes** section.

6. 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.”

7. For **Errors and Warnings**, select an option.
Open a Model and a Cube

1. From the Model drop-down list, select a model.
2. From the Cube drop-down list, select a predefined cube.
3. Click 🔄.

Open an OLAP View without an OLAP View Already Open

If you start OLAP mode without an OLAP view open, you must open an OLAP view from the OLAP home page, as shown:

![OLAP View Home Page](image)

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.

Tip: Alternatively, you can click 🔄.
Open an OLAP View with an OLAP View Already Open

*Note:* If an OLAP view is not already open, open an OLAP view from the OLAP home page.

1. From the **OLAP View** drop-down list, select a saved OLAP view.
2. Click 📄.

Delete an OLAP View

1. Do one of the following:
   - If no OLAP view is open, click the **OLAP** link.
   - If an OLAP view is open, click the **OLAP Home** link.

   The OLAP home page appears.

2. Click the **Delete** link that is next to the OLAP view.

*Note:* If you are a Designer, but your role permission is Read-only, you do not have the ability to delete.
Introduction

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

Start Reports Mode

Click \texttt{Reports} at the top of the SAS Activity-Based Management window.

Create a Report

1. Select \texttt{File > New > Report}.

The Report Wizard appears.
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**.
To include data from all the modules and all the accounts, select the **Use all modules** option.

To limit data to a specific module, select the **Select a module** option, and select a module.

To limit data to specific accounts, select the **Use selection** check box, then select one or more accounts from the list.

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:

Select one to ten dimensions:
- Activities
- Customers
- External Units
- General Ledger
- Organization
- Products and Services

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

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

Next, you will review the report summary and create the report.

Click **Next**.

Review the report summary.

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.

To save the report configuration so that the report can be easily run again, do the following:

- Select the **Save configuration as** option.
- Type the **Name**.
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 must open a report from the Reports home page, as shown:

<table>
<thead>
<tr>
<th>Folders</th>
<th>Report Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports</td>
<td><img src="image" alt="Fiscal Year-end Report" /></td>
</tr>
</tbody>
</table>

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.

**Tip:** Alternatively, you can click ![image](image).

---

Open a Report with a Report Already Open

1 From the Report Configuration drop-down list, select a configuration.

2 Click ![image](image).
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 are a Designer, but your role permission is Read-only, 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.

   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 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 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 ![arrow up] or ![arrow down].

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**.
There are no options for this report type.
15 Click ....

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.

![Publish a Report dialog box](image)

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 are a Designer, but your role permission is Read-only, you do not have the ability to publish.*

## Delete a Report Configuration

1. Do one of the following:
   - If no report configuration is open, click the **Report** link.
   - If a report configuration is open, click the **Report Home** link.

   The Reports home page appears.

2. To the right of a report configuration, click the **Delete** link.
Note: If you are a Designer, but your role permission is Read-only, you do not have the ability to delete.
CHAPTER 29

Working with Column Layouts

Introduction  215
Add a Column  215
Order Columns  217
Remove a Column  218
Format a Column  218
Save a Column Layout  219
Copy a Column Layout to another Module  220
Apply a Saved Column Layout  221

Introduction

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

Add a Column

1 On a module page, select Model > Column Layout.

Tip: Alternatively, you can double-click a column heading.

The Column Layout dialog box appears.
216 Add a Column  Chapter 29

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.
Order Columns

1. On a module page, select Model > Column Layout.
   The Column Layout dialog box appears.

2. Select the Module to which this column layout applies.
From the list of **Displayed columns**, select the pane from the appropriate tab (**Left**, **Primary**, or **Right**) that you want to order.

Select a row (which represents a column in the view).

To move the row up, click **Move Up**.

The column will be moved to the left.

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**.

   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**.

   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:
   
   a Select or clear the **Override default precision** option.

   b 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**.

   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 🔄.
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.”

Add a Currency


The Manage Exchange Rates dialog box appears.
2. Click the **Add/remove currency** link.

The Add/Remove Currencies dialog box appears.
Working with Currencies

Remove a Currency

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 Microsoft 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**:
   a. Click the **Add/remove currency** link.
      The Add/Remove Currencies dialog box appears.
   b. 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.
Select the Period/Scenario association that has the exchange rates table that you want to copy.
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.”

Import Model Data from a Database with the Import Data Wizard

Before attempting to import data from a database, see the “Connect to a Database” section at the end of this chapter, 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 > 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.
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.

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.

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.
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<code>number</code> and DimMemberRef<code>number</code> only if you chose to identify accounts by their dimension signatures, which is the only situation in which these values are needed.

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 <code>&lt;None&gt;</code>. Map this new column to the **Target Column Name** Period, and specify the **Default** value as MyPeriod.

Repeat steps 15 through 18 for every table column that you want to import.

Click the appropriate tab at the bottom of the grid to map the columns for another table.

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**.
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 > 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 the “Connect to a Database” section at the end of this chapter, 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**.
   
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 **...**.

   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 **Which model do you want to export from** drop-down list, select a model.

8 From the **Which period/scenario associations do 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.

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 **Which model do you want to export from** drop-down list, select a model.

8 From the **Which period/scenario associations do 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 the “Connect to a Database” section at the end of this chapter, 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**.

   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 ....

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 Which model do you want to export from drop-down list, select a model.

7 From the Which period/scenario associations do you want to export from list, select the check box next to one or more period/scenario associations.
Which period/scenario associations do you want to export from?

- [ ] 2005 / ACTUAL

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.
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.

**Click Next.**
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:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td></td>
</tr>
<tr>
<td>Filter Conditions</td>
<td></td>
</tr>
<tr>
<td>Name = 'Headquarters'</td>
<td></td>
</tr>
<tr>
<td>BaseCurrency = 'USD'</td>
<td></td>
</tr>
</tbody>
</table>

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

You must perform two major tasks when you connect to a database to import data or export data: preparing to connect and creating a connection string. These tasks can differ for each type of database. The tasks for the most common databases are described in this section.

Microsoft Access Database

Prepare to Connect

1 In Microsoft Windows Explorer, create a folder on the SAS Activity-Based Management server’s file system in which to save the Microsoft Access database.

2 Select the folder and select File > Properties.

   The Properties dialog box appears.

3 Click the Sharing tab.

4 Select the Share this folder option.

5 For the User limit, select the Maximum allowed option.
6 Click the **Security** tab.

7 From the **Group or user names** list, select **ASP.NET Machine Account**.

8 In the list of **Permissions**, select every check box in the **Allow** column.
Repeat steps 7 and 8 for the user account of the person who will be importing data and exporting data.

Save the Microsoft Access database in the folder you just created on the SAS Activity-Based Management server's file system.

Display the properties for the Microsoft Access database on the server, and ensure that the ASP.NET Machine Account has the same permissions as the new folder.

You are now ready to import data from or export data to the Microsoft Access database.

Create a Connection String

When you use a wizard to import data or to export data, you are asked to set up a database connection. Click … in the wizard.

The Data Link Properties dialog box appears.
2 On the **Provider** tab, select **Microsoft Jet 4.0 OLE DB Provider**.

3 Click **Next**.

The Connection tab appears.
4 In the **Select or enter a database name** box, type the Universal Naming Convention (UNC) path to the Microsoft Access database that resides on the SAS Activity-Based Management server. Or, click ... to browse.

5 Type the **User name** of the account that can open the Microsoft Access database.

6 If the Microsoft Access database requires a password, do the following:
   a Clear the **Blank password** option, and type the **Password**.
   b Select the **Allow saving password** option.

7 Click **Test Connection**.

   If you see a message that the test connection failed, review the information that you specified and correct any errors.

8 Click **OK**.

   You return to the SAS Activity-Based Management wizard to continue importing data or exporting data.
Microsoft Excel Workbook

Prepare to Connect

1. In Microsoft Windows Explorer, create a folder on the SAS Activity-Based Management server's file system in which to save the Microsoft Excel workbook.

2. Select the folder and select File > Properties.

   The Properties dialog box appears.

3. Click the Sharing tab.

4. Select the Share this folder option.

5. For the User limit, select the Maximum allowed option.

6. Click the Security tab.

7. From the Group or user names list, select ASP.NET Machine Account.

8. In the list of Permissions, select every check box in the Allow column.
9 Repeat steps 7 and 8 for the user account of the person who will be importing data.

10 Save the Microsoft Excel workbook in the folder you just created on the SAS Activity-Based Management server's file system.

11 Display the properties for the Microsoft Excel workbook on the server, and ensure that the ASP.NET Machine Account has the same permissions as the new folder.

You are now ready to import data from the Microsoft Excel workbook.

**Create a Connection String**

1 When you use a wizard to import data, you are asked to set up a database connection. Click … in the wizard.

   The Data Link Properties dialog box appears.
2 On the **Provider** tab, select **Microsoft Jet 4.0 OLE DB Provider**.

3 Click **Next**.

The Connection tab appears.
4 In the **Select or enter a database name** box, type the Universal Naming Convention (UNC) path to the Microsoft Excel workbook that resides on the SAS Activity-Based Management server. Or, click ... to browse.

5 If you browse to locate the Microsoft Excel workbook, ensure that you choose to display all file types in the Select Access Database dialog box.

   By default, this dialog box displays only Microsoft Access database files.

6 Leave the **User name** as Admin.

7 If the Microsoft Excel workbook requires a password, do the following:

   a Clear the **Blank password** option, and type the **Password**.

   b Select the **Allow saving password** option.

8 Click the **All** tab.

9 In the **Name** column, select **Extended Properties**.

10 Click **Edit Value**.

   The Edit Property Value dialog box appears.
11 In the **Property Value** box, type **Excel 8.0**.

12 Click **OK**.

13 Click the **Connection** tab, and click **Test Connection**.

   If you see a message that the test connection failed, review the information that you specified and correct any errors.

14 Click **OK**.

   You return to the SAS Activity-Based Management wizard to continue importing data.

---

**SAS/SHARE**

**Prepare to Connect**

1 Set up a SAS/SHARE server and then configure and test the SAS/SHARE clients.

2 Note the server name, the server location, and the library name.

   You will need this information later when you create the connection string in SAS Activity-Based Management.
3 From the desktop of the computer that will import data or export data, run the command `sasoledb.exe`.

### Create a Connection String

1 When you use a wizard to import data or export data, you are asked to set up a database connection. Click ... in the wizard.

The Data Link Properties dialog box appears.

2 On the **Provider** tab, select **SAS SHARE Data Provider 9.1**.

3 Click **Next**.

The Connection tab appears.
4 In the **Data Source** box, type the name of the SAS/SHARE server.

5 In the **Location** box, type the location of the SAS/SHARE server.

6 Click the **All** tab.

7 In the **Name** column, select **Extended Properties**.

8 Click **Edit Value**.

The Edit Property Value dialog box appears.

9 In the **Property Value** box, type the name of the SAS/SHARE library.
If you are using SAS Version 8, do the following:

a. From the All tab, select SAS Server Release, and click Edit Value.
b. In the Edit Property Value dialog box, type a Property Value of 8.
c. Click OK.

Click the Connection tab, and click Test Connection.

If you see a message that the test connection failed, review the information that you specified and correct any errors.

Click OK.

You return to the SAS Activity-Based Management wizard to continue importing data or exporting data.

Oracle Database

Prepare to Connect

1. Obtain the Oracle Service Name (SID) and the Oracle user ID and password from your Oracle database administrator.

You will need this information later when you create the connection string in SAS Activity-Based Management.

2. Ensure that the Oracle client software is installed on both the SAS Activity-Based Management server and client with the Administrator option.

Create a Connection String

1. When you use a wizard to import data or to export data, you are asked to set up a database connection. Click ... in the wizard.

The Data Link Properties dialog box appears.
2 On the **Provider** tab, select **Oracle Provider for OLE DB**.

Do not select **Microsoft OLE DB Provider for Oracle** because this provider might not be compatible with your Oracle server.

3 Click **Next**.

The **Connection** tab appears.

4 Type the SID in the **Data Source** box.

5 Select the **Use a specific user name and password** option, and type the Oracle **User name** and **Password**.

6 Select the **Allow saving password** option.

If you fail to select this option, the SAS Activity-Based Management client will not be able to pass the user credentials to the SAS Activity-Based Management server for processing.
7 Click Test Connection.

If you see a message that the test connection failed, review the information that you specified and correct any errors.

8 Click OK.

You return to the SAS Activity-Based Management wizard to continue importing data or exporting data.

**Microsoft SQL Server Database**

**Prepare to Connect**

For any Microsoft SQL Server database that will serve as the source for importing data or as the destination for exporting data, identify or define a Microsoft SQL Server user account that has appropriate permissions to that database. For exporting data, the user account **dbowner** is best. For importing data, public access should be adequate. Do not use a Microsoft Windows user account for this type of access to Microsoft SQL Server.
Create a Connection String

1 When you use a wizard to import data or to export data, you are asked to set up a database connection. Click … in the wizard.

The Data Link Properties dialog box appears.

2 On the Provider tab, select Microsoft OLE DB Provider for SQL Server.

3 Click Next.

The Connection tab appears.

4 From the Select or enter a server name list, select or type the name of the Microsoft SQL Server instance that contains the database.

5 Select the Use a specific user name and password option, and type the Microsoft SQL Server User name you have identified or defined for this purpose and the Password.

6 Select the Allow saving password option.

If you fail to select this option, the SAS Activity-Based Management client will not be able to pass the user credentials to the SAS Activity-Based Management server for processing.
Select the **Select the database on the server** option, then select or type the name of the database.

8 Click **Test Connection**.

If you see a message that the test connection failed, review the information that you specified and correct any errors.

9 Click **OK**.

You return to the SAS Activity-Based Management wizard to continue importing data or exporting data.
Index

A
access conflicts 15
Account table 115
accounts 47–49
See also cost elements
activity accounts 49
adding attributes to 184
adding stages to 74–77
associating drivers with item properties 172
corresponding dimension member 48
cost object accounts 49
creating 48, 159, 163
creating cost elements when accounts are created 164
deleting 48
destination accounts 67, 79–80, 177
going to 178
minimal information 159, 162
optional information 163
removing attributes from 185
resource accounts 49
rollup accounts 47
showing accounts with added attributes 185
source accounts 67, 79–80, 177
validating a model 50–51
Activities dimension 42
activity accounts 49
activity-based costing 3–4
activity-based management 3
Activity module 42–43
Activities dimension 42
creating structural dimensions 42–43
Organization dimension 42–43
alignment for columns 93
All dimension concept 32–33
archiving models 112
to databases 243–246
to XML files 246–250
assigned cost element 50
assignment panes 68
assignment paths 67
Assignment table 115
assignments 67–72
assignment panes 68
assignment paths 67
bills of costs 68–71
creating 72, 175–177
inner-module assignments 67
 recyclable costing 71–72
showing destination accounts for an account 177
showing source accounts and destination accounts for an account 177–178
showing source accounts for an account 177
associations
See period/scenario associations
attribute dimensions 34
attribute folders 77
creating 179–180
attribute names in equations 80
attribute values
exporting 118
attributes 73–77
adding to an account 184
attribute folders 77, 179–180
Boolean attributes 73
created attributes 76, 79–80
dimension attributes 74–75
dimension member attributes 74–75
dimension value attributes 74–75
numeric attributes 75–76
removing from accounts 185
showing accounts with added attributes 185
specifying attribute's value 184
stages attributes 77
text attributes 76
automatic saving 8
average cost of processing a customer order 76
average number of cases loaded per employee 76
base currency  97
basic driver  54
bill of costs driver  54
bills of costs  68–71
discrete-manufacturing environments  69
process-manufacturing environments  69
bills of costs quantities  69–70
fixed quantities  70
variable quantities  69–70
variable quantities and fixed quantities  70–71
Boolean attributes  73
budgeting and planning, exporting data for  113–114
business analysis
  exporting data for further analysis  112–114
business process  42
calculated attributes  76
  average cost of processing a customer order  76
  average number of cases loaded per employee  76
  equations with  79–80
  calculated drivers  54–55
    carrying costs of inventory of finished goods  55
    delivering a product to customers  54–55
    equations with  79–80
    processing customer orders  55
    shipping a product  54
    storage costs for inventory of finished goods  55
  calculated results
    exporting, filtering, and limiting  119–120
    calculating costs  See cost calculation
canceling server tasks  12, 126–127
carrying costs of inventory of finished goods  55
Channels dimension  41
Checkbox column type  94
collecting data  21
color for columns  93
column formatting
  available format types  94
  Checkbox type  94
color, style, and alignment  93
  currency formatting  95
currency type  93
displaying the currency symbol  95
format types  94
negative numbers  94
Number type  93–94
Percentage type  94
precision  94–95
Text type  94
column layouts
  adding a column  215–217
  applying a saved column layout  221
copying a saved column layout  220–221
default column layout  217
  names of  22
  ordering columns  217–218
  removing a column  218
  saving  217–218
  showing multiple period/scenario associations  92
cost calculation  23–24, 143–144
  changing model structure after  24
  errors  24
  reciprocal costs  71–72
cost elements  50
  See also accounts
  assigned cost element  50
  creating an entered cost element on a module page  164–165
  creating when accounts are created  164
  external unit cost element  50
  internal unit cost element  50
  validating a model  50–51
cost object accounts  49
Cost Object module  40–41
Channels dimension  41
creating structural dimensions 41
Customers dimension 41
Products dimension 41
costing
activity-based costing 3–4
bills of costs 68–71
product costing 113
reciprocal costing 71–72
service costing 114
shared services costing 114
credentials, current 7
cross-module assignments 67
cube fact table 116–117
cubes 82–84
availability for OLAP views 85
fact tables 82, 116–117
generating 82, 83, 190–191
multi-stage contributions cube 84, 116–117
189–190
opening a model and a cube 192
pre-defined 83–84
resource contributions cube 84, 117
single-stage contributions cube 83–84
currencies and exchange rates
adding currencies 223–225
adding euro exchange rates 226
base currency 97
copying exchange rate tables 226–227
currencies 97
Currency column type 93
currency formatting for columns 95
currency symbol in columns 95
exchange rates 97–98
removing currencies 225
selecting currencies for exchange rate tables 98
setting up exchange rate tables 226
current credentials 7
customers
average cost of processing orders 76
delivering a product to customers 54–55
 Customers dimension 41

D
data collection 21
data schema 103
database tables in 105–108
database connections 263–279
Microsoft Access 121
Microsoft Excel workbook 121, 268–272
Microsoft SQL Server 277–279
Oracle 275–277
SAS/SHARE 272–275
database tables
creating sample tables 108–110
in data schema 105–108
relationships, dependencies, and data values 110
databases
archiving models to 243–246
importing data from 103–111
importing from multiple databases 110
importing model data from 229–239
dates for periods 26
default column layout 92
default driver 58, 171
delivering a product to customers 54–55
Designer license 4
destination accounts 67
equations and 79–80
showing for an account 177
dimension attributes 74–75
dimension levels 31–32
dimension member attributes 74–75
dimension members 32–33
All and No displayed in Model mode 32–33

corresponding to accounts 48
creating 155–156
deleting 157
dimension signature key 103–104
dimension signatures 34, 103, 104
dimension value attributes 74–75
dimensions 28, 29, 31–37
All and No/None 32–33
attribute dimensions 34
creating 34–35
creating dimension members 155–156
deleting dimension members 157
dimension levels 31–32
dimension members 32–33
exporting 118
multiple dimensions for better analysis 35–37
structural dimensions 33–34
types of 33–34
discrete-manufacturing environment 69
drill-down 33
driver quantities 58–63
fixed 58
idle 61–62
unique and non-unique 60–61
variable 59
variable and fixed together 59–60
weighted 60
driver sequencing 63–64

drivers 53–65
	associating, using account's item properties 172
	associating, using module grid 173
	basic driver 54
	bill of costs driver 54

calculated driver 54, 55, 79–80

creating 65, 167–170

default driver 58, 171

driver quantities 58–63

driver sequencing 63–64

evenly assigned driver 55–56

percentage driver 56

sales volume driver 56

specifying default driver 171

types of 54–57

user-entered cost allocation 64–65

weighted driver 56–57, 170–171

E

end dates for periods 26

entered cost element 50

creating on module page 164–165

equations
	attribute names in 80

calculated drivers and calculated attributes with 79–80

defining 79–80

property names in equations 80

source accounts and destination accounts 79–80

testing 80

euro exchange rates 226

evenly assigned drivers 55–56

evenly assigned quantities 63

exchange rates

See currencies and exchange rates

Export wizard

archiving models to databases 243–246

archiving models to XML files 246–250

exporting data 112–120

exporting model data to databases 250–263

exported report tables 117–118

exporting data 99, 112–120

See also connection strings

See also database connections

archiving models 112

archiving models to databases 243–246

archiving models to XML files 246–250

attribute values 118

dimensions 118

Export wizard for 112, 120, 250–263

exporting, filtering, and limiting calculated results 119–120

exporting model data 112–120

exporting model data to databases 250–263

filtering data 120

for further business analysis 112–114

planning and budgeting 113–114

product costing 113

profitability analysis 113

properties 118

service costing 114

shared services costing 114

tables 118

tables to export for business analysis 114–118

exports reports 88, 203–213

external unit cost element 50

external units 44

External Units module 44

F

fact tables 82, 116–117

features, organization of 9–10

filtering

calculated results 119–120

data for exporting 120

finished goods inventory carrying costs of 55

storage costs for 55

fixed quantities

bills of cost quantities 70–71

driver quantities 58, 59–60

folders

attribute folders 77, 179–180

creating 131

font style for columns 93

formatting columns

See column formatting

G

General Ledger dimension 43–44

grid

See column layouts

See columns

H

headers 88

Home page 9

home pages for modes 10
idle quantities 62–63
  evenly assigned quantities 63
  user-entered quantities 62
  user-proportioned quantities 62–63
import configurations
  naming 22
  XML 111
Import Data Wizard
  importing model data from databases 229–239
  importing model data from XML files 239–243
importing data 99–111
  See also connection strings
  See also database connections
building a model by 100–111
creating sample database tables 108–110
data schema 103
database table relationships, dependencies, and data values 110
database tables in data schema 105–108
from databases 103–111
general steps 100–101
Import Data Wizard for 229–239
keys 103–105
from Microsoft Excel workbooks 110–111
model data from databases 229–239
from multiple databases 110
period/scenario associations 111
process summary for 101
source data preparation 102–103
updating models with new periodic values 102
updating models with new structure 102
from XML files 111, 239–243
XML import configurations 111
inner-module assignments 67
internal unit cost element 50
inventory of finished goods
  carrying costs of 55
  storage costs for 55
item properties
  associating drivers with 172
  workspace items 130–131
K
  keys 103–105
    dimension key 103–104
    dimension signatures and references 103–104
    reference key 104
L
  layout
    See column layouts
  licenses 4–5
    access conflicts with roles 15
  limiting calculated results 119–120
  locale 98
M
  measures 81
  Microsoft Access databases
    connection strings 265–267
    preparing to connect 263–265
  Microsoft Excel workbooks
    connection strings 269–272
    importing data from 110–111
    preparing to connect 268–269
  Microsoft SQL Server databases
    connection strings 278–279
    preparing to connect 277
  minimal information
  model data
    analyzing with OLAP 81–85
    exporting 112–120
    exporting to databases 250–263
    importing from databases 229–239
    reporting 87–89
  Model mode 9
    All and No displayed in 32–33
    starting 142
  model structure
    anticipating user needs 20–21
    changing after cost calculation 24
    collecting data 21
    creating 20–21
    paper model for 21
    structural information
      updating 102
    models 19–24
      See also periods
      See also scenarios
  archiving 112, 246, 250
  availability 8
  building by importing data 100–111
  calculating costs 23–24, 143–144
  creating 133–142
  naming 22
  naming items 22–23
  opening 142–143
  opening a model and a cube 192
  report performance and size of 87
  saving items 22
Index

sharing data with others  23
sharing models, configurations, and data  22, 23
updating with new periodic values  102
validating  50–51, 144–145
modes  9
changing  9–10
mode home pages  10
Model mode  9, 32–33, 142
OLAP mode  9, 187
Reports mode  9, 195
module rollup  47
modules  39–45
Activity module  42–43
associating drivers using grid  173
columns in  40
copying column layouts to another module  220–221
Cost Object module  40–41
creating  44–45
creating entered cost element on module page  164–165
cross-module assignments  67
External Units module  44
inner-module assignments  67
references  40
Resource module  43–44
multi-stage contributions cube  84
cube fact table with calculated values  116–117
properties for  189–190
My Shortcuts  14–15

N
names
attribute names in columns  80
column layouts  22
import configurations and models  22
model items  22, 23
OLAP views  23
period level names  150–151
property names in equations  80
report configurations  23
scenario level names  152–153
workspace items  15
negative numbers in columns  94
No/None dimension concept  32–33
non-unique driver quantities  80–81
Number column type  93–94
numeric attributes  75–76

O
OLAP
See also cubes
analyzing model data with  81–85
changing appearance of windows  189
measures  81
periods and scenarios in  28–29
OLAP mode  9
starting  187
OLAP page  84–85
OLAP views  85
performance  85
OLAP views  85
creating  187–188
cube availability  85
deleting  193
naming  23
opening  192–193
saving  85, 188
opening a model and a cube  192
optional information  163
Oracle databases
connection strings  275–277
preparing to connect  275
ordering columns  217–218
Organization dimension
Activity module  42–43
Resource module  44
organization of features  9, 10
ownership of server area items  15

P
paper model  21
Percentage column type  94
percentage drivers  56
time spent on tasks  56
performance
OLAP page  85
reports  87
period level names  150–151
period/scenario names  27
calculation status  27
copying data from one to another  27
importing  111
publishing or unpublishing  154
selecting for report configurations  89
showing multiple  92
status of associations  27
periods  25–29
See also scenarios
copying period/scenario data  154
creating 26, 147–149
deleting 26
in OLAP analysis 28–29
organizing 26
period level names 150–151
setting up 25–26
start dates and end dates 26
updating models with periodic values 102
permissions for server area items 15
planning and budgeting
exporting data for 113–114
pre-defined cubes 83–84
multi-stage contributions cube 84
resource contributions cube 84
single-stage contributions cube 83–84
pre-defined report templates 89
precision for columns 94–95
process-manufacturing environment 69
processing customer orders 55
products
costing 113
delivery to customers 54–55
shipping 54
Products dimension 41
profitability analysis 113
properties
associating drivers with item properties 172
column layouts 91
exporting 118
multi-stage contributions cube 189–190
property names in equations 80
workspace items 130–131
publishing
period/scenario associations 154
reports 88, 213
headers 88
model size and report performance 87
opening 202
publishing 88, 213
saving 88
setting up 88
Reports mode 9
starting 195
resource accounts 49
resource contributions cube 84, 117
Resource module 43–44
creating structural dimensions 43–44
General Ledger dimension 43–44
Organization dimension 44
roles 4–5
access conflicts with licenses 15
permissions for server area items 15
rollup accounts 47
S
sales volume drivers 56
sample database tables 108–110
SAS Activity-Based Management
organization of features 9–10
SAS Activity-Based Management server
See servers
SAS/SHARE databases
connection strings 273–275
preparing to connect 272–273
saved column layout 40, 221
saving
automatic 8
column layouts 95–96, 219–220
data 8–9
model items 22
OLAP views 85, 188
report configurations 89, 203
reports 88
scenario level names 152–153
scenarios 25–29
See also period/scenario associations
See also periods
copying period/scenario associations
See also periods
creating 149–150
in OLAP analysis 28–29
organizing 26
setting up 25–26
sequencing, driver 63–64
server area 14
server area items
access conflicts between roles and licenses 15
reciprocal costing 71–72
reference key 104
references 40, 103–104
refreshing information 11, 126
report configurations 89
changing 203
deleting 213–214
naming 23
saving 89, 203
selecting a period/scenario association 89
report templates 89
reports 87–89
creating 195–202
exported report tables 117–118
exporting 88, 203–213
creating and deleting 14
ownership and permissions 15
servers
canceling tasks 12, 126–127
connecting to 7, 125–126
current credentials 7
service costing 114
shared service costing 114
shipping a product 54
shortcuts 14, 15, 129–130
single-stage contributions cube 83–84
sorting columns 217–218
source accounts 67, 177
source data preparation 102–103
splitter bar 68
stages
adding to accounts 77
attributes 77
start dates for periods 26
status bar 11
storage costs for inventory of finished goods 55
structural dimensions 33–34
Activity module 42–43
Cost Object module 41
Resource module 43–44
structure
See model structure
style for columns 93
T
tables
Account table 115
Assignment table 115
cube fact table 116–117
database tables 105–110
exchange rate tables 98, 226–227
exported report tables 117–118
exporting 118
exporting for business analysis 114–118
fact tables 82, 116–117
task bars 10–11
testing equations 80
text attributes 76
Text column type 94
time spent on tasks 56
U
unique driver quantities 60–61
unpublishing period/scenario associations 154
user-entered cost allocation 64–65
user-entered quantities 62
user needs 20–21
user-proportioned quantities 62–63
V
validating models 50–51, 144–145
variable driver quantities 59–60
variable quantities 69–70
Viewer license 5
W
weighted driver quantities 60
weighted drivers 56–57
processing customer orders 57
specifying weights for 170–171
supporting products 57
work flow 12
workspace items 13
creating shortcuts to 129–130
moving 132
naming guidelines 15
opening 132
reviewing or changing properties of 130–131
Workspace Manager 9, 13–15
My Shortcuts 14–15
server area 14
workspaces 13
X
XML files
archiving models to 246–250
import configurations 111
importing model data from 239–243
Your Turn

If you have comments or suggestions about *SAS® Activity-Based Management 6.3: User's Guide*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

- SAS Publishing
- SAS Campus Drive
- Cary, NC 27513
- E-mail: yourturn@sas.com

For suggestions about the software, please return the photocopy to

- SAS Institute Inc.
- Technical Support Division
- SAS Campus Drive
- Cary, NC 27513
- E-mail: suggest@sas.com
SAS® Publishing gives you the tools to flourish in any environment with SAS!

Whether you are new to the workforce 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.

SAS® Press Series
Need to learn the basics? Struggling with a programming problem? You’ll find the expert answers that you need in example-rich books from the SAS Press Series. Written by experienced SAS professionals from around the world, these books deliver real-world insights on a broad range of topics for all skill levels.

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: 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, and hard-copy books.

SAS® Learning Edition 4.1
Get a workplace advantage, perform analytics in less time, and prepare for the SAS Base Programming exam and SAS Advanced Programming exam with SAS® Learning Edition 4.1. This inexpensive, intuitive personal learning version of SAS includes Base SAS® 9.1.3, SAS/STAT®, SAS/GRAPH®, SAS/QC®, SAS/ETS®, and SAS® Enterprise Guide® 4.1. Whether you are a professor, student, or business professional, this is a great way to learn SAS.