

SAS® Financial Management

5.6: Process Administrator's

Guide

5.62

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2021. *SAS® Financial Management 5.6: Process Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Financial Management 5.6: Process Administrator's Guide

Copyright © 2021, SAS Institute Inc., Cary, NC, USA

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

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

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

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

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

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

September 2026

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

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

5.6-P1:fmspag

Contents

<i>About This Book</i>	<i>vii</i>
<i>What's New in SAS Financial Management</i>	<i>ix</i>

PART 1 Introduction 1

Chapter 1 • Overview of SAS Financial Management	3
SAS Financial Management Activities	3
Client Applications	3
SAS Financial Management Data	4
Additional Documentation	5

PART 2 Administering SAS Financial Management 7

Chapter 2 • Getting Started with SAS Financial Management Studio	9
Working in SAS Financial Management Studio	9
User Roles and Capabilities	12
Loading Data to the SAS Financial Management Data Mart	12
Locked Objects	13
Promoting Objects to Another System	16
Using Views	17
Using the Help System	18
Chapter 3 • The Dimensions Workspace	19
Introduction to Dimensions, Hierarchies, and Members	20
Working with Dimension Types	20
Working with Dimensions, Hierarchies, and Members	22
Dimensions View	22
Hierarchies View	24
Members View	26
Using Custom Properties	27
Custom Properties View	28
Using Data Locales	29
Data Locales View	30
Using Calendar Templates	30
Calendar Templates View	31
Chapter 4 • Member Properties	33
Account Details Page	33
Custom Properties Page	34
Formulas Page	35
Identification Page	35
Security Page	36
Time Details Page	38
Users Page	38

Chapter 5 • Required Dimension Types	39
Introduction to Required Dimension Types	40
Introduction to the Account Dimension Type	40
Introduction to Account Types	41
Asset Account Type	41
Cumulative Translation Adjustment Account Type	42
Equity Account Type	42
Expense Account Type	42
Liability Account Type	42
Retained Earnings Account Type	43
Revenue Account Type	43
Statistical Account Type	43
Statistical Balance Account Type	43
Statistical Flow Account Type	44
Balance Type Property of Accounts	44
Exchange Rate Type Property of Accounts	44
Intercompany Property of Accounts	44
Introduction to the Organization Dimension Type	45
Organization Details	45
Introduction to the Analysis Dimension Type	46
Introduction to the Currency Dimension Type	46
Introduction to the Frequency Dimension Type	46
Introduction to the Source Dimension Type	47
Introduction to the Time Dimension Type	48
Introduction to the Trader Dimension Type	48
Chapter 6 • Working with CTA Accounts	51
Creating a CTA Account: Best Practices	51
CTA Calculation	52
Other Considerations for CTA Accounts	53
Chapter 7 • Working with Retained Earnings Accounts	57
Retained Earnings: A Hybrid Account	57
Creating a Retained Earnings Account	58
Retained Earnings Accounts and Currency Conversion	61
Loading Data to a Retained Earnings Account	62
Sample Configurations	62
Other Considerations	64
Rolling Forward Balance Sheet Accounts	65
Chapter 8 • Cycles and Cycle Periods	69
Introduction to Cycles	69
Cycle Manager Window	70
Create a New Cycle	71
Copy a Cycle	71
Open a Cycle	72
Data Records in a Cycle	72
Managing Data in a Cycle	73
Periods View	74
Chapter 9 • Cycle Properties	77
Adjustments Page	77
Dimensions Page	78
Identification Page	78
Intercompany Trading Page	78
Security Page	79

Time Hierarchy Page	79
Chapter 10 • Working with Exchange Rate and Driver Rate Sets	81
About Exchange Rates	81
About Driver Rates	82
Best Practices for Exchange Rate and Driver Rate Sets	82
Base Currency for Exchange Rate Sets	83
Precision of Exchange Rates	84
Displaying and Copying Exchange Rates and Driver Rates	85
Querying Exchange Rates and Driver Rates in a Table	85
Querying Exchange Rates and Driver Rates outside a Table	86
Chapter 11 • Overview of Models	87
Introduction to Models	87
Required Features of a Model	87
Optional Features of a Model	89
Models View	90
Chapter 12 • Model Properties	93
Adjustments Page	93
Dimensions Page	94
Formats Page	96
Formulas Page	96
Identification Page	97
Periods Tab	97
Rates Page	97
Security Page	98
Override Security Settings	98
Chapter 13 • Models: Rules and Adjustments	101
Overview of Adjustments and Adjustment Rules	101
Working with Adjustment Rules	102
Working with Manual Adjustments	102
What Happens When You Post Adjustments	103
Being on Hold	104
Deleting Adjustments	104
Threshold for Adjustment Rules	105
Adjustment Rules View	107
Manual Adjustments View	109
Posting Status of Manual Adjustments and Adjustment Rules	111
Warnings	111
Chapter 14 • Models: Adjustment Rule Types	113
Intercompany Balancing Rules	113
Net Intercompany Balancing Rules	114
Reporting-Entity Balancing Rules	114
Allocation Rules	114
Reclassification Rules	115
Ownership Rules: Introduction	115
Ownership Rules: Organization Hierarchy Requirements	115
Ownership Rules: Holding Types	116
Ownership Rules: Consolidation Methods	117
Chapter 15 • Composite Models	121
Introduction to Composite Models	121
Required Features of a Composite Model	122

Optional Features of a Composite Model	122
Composite Models View	122
Chapter 16 • Formulas	125
Overview of Formula Types and Formulas	125
Order of Execution	126
Server-Side Calculated Members	126
Writing the Formula Expression: Explicit Members and Fixed Members	128
Using a Function in a Bracketed Member Reference	129
Formula Scope	130
Defining Multiple Formulas on One Member	130
Formulas That Are Ignored, Invalid, or Not Rendered	130
Working with Driver Formulas	131
Excel-Based Calculated Members	133
Dictionary of Functions	134
Chapter 17 • Forms and the Forms Workspace	153
Forms and Form Sets	153
Workflow Types	154
Form Set Summary	154
Introduction to the Forms Workspace	155
Assigning Authors to Forms	156
Assigning Reviewers to Forms	156
Status Values of Form Sets	157
Status Values of Forms	157
Form Sets View	158
Forms View	163
Chapter 18 • Task Management	167
Create a Task in SAS Financial Management Studio	167
Chapter 19 • Security in SAS Financial Management	171
About Security in SAS Financial Management	171
Object Security	172
Data Security	174
Index	177

About This Book

Audience

This book is intended for SAS Financial Management administrators and power users who perform all SAS Financial Management tasks, including:

- creating and managing form sets

For more information about SAS Financial Management Excel Add-In for Microsoft Excel, see the online help.

- creating and publishing reports

To administer SAS Financial Management, you must be familiar with the operating system on which it is installed. For example, you must know how to create folders, run scripts, and update environment variables. If using Microsoft Windows, you must also be an administrator of the machine.

What's New in SAS Financial Management

What's New in SAS Financial Management 5.62

Reports in SAS Financial Management

SAS Financial Management reports can be viewed and run only through the SAS Stored Process web application. The ETL Job Status report has been removed. The reference to the report in SAS Stored Process web application will be removed in future release.

Toolbar Options in SAS Financial Management

Beginning with SAS Financial Management 5.62, a few options are not visible on the toolbar. To view all the available options, navigate to the workspace that you want and click the view menu icon on the top right.

Post-deployment Tasks in SAS Financial Management

Beginning with SAS Financial Management 5.62, SAS administrators must perform a few steps after the deployment of the solution is complete on the machine. For more information, see [“Post-deployment Tasks” in SAS Financial Management: Installation and Configuration Guide](#).

Enable Single Sign-On in SAS Financial Management

Beginning with SAS Financial Management 5.62, SAS administrators must perform a few steps to enable the single sign-on authentication. For more information, see [“Post-installation: Enable Single Sign-On” in SAS Financial Management: Installation and Configuration Guide](#).

Notifications in SAS Financial Management

Beginning with SAS Financial Management 5.62, users will not see any notifications. Users can receive emails. For more information, see [“Receiving an Alert Notification” in SAS Financial Management: User's Guide](#).

Publish Form Sets in SAS Financial Management

Beginning with SAS Financial Management 5.62, SAS administrators should perform certain steps to publish the form sets without an exception. For more information [“Publish” on page 162](#).

What's New in SAS Financial Management 5.61

Support for Microsoft Office 2019 in SAS Financial Management

SAS Financial Management 5.61 now supports Microsoft Office 2019. You can use Microsoft Excel 2019 for SAS Financial Management Add-In.

What's New in SAS Financial Management 5.6

Some Roles Are Removed

The Planning Data Entry User, Report Consumer, and Report Provider roles are no longer available in SAS Financial Management 5.5 or later.

The Form and Report Management User Interface Has Been Rewritten in HTML5

Previously, many SAS applications and SAS solutions used the Adobe Flash Player to provide interactive user interfaces. Adobe announced that it intends to end support for Flash technology and will cease to update and distribute the Flash Player at the end of 2020. Browser vendors will disable Flash by default in 2019. For more information about Adobe Flash end-of-life, see [SAS Software and Its Use of the Adobe Flash Player](#).

The SAS Financial Management 5.6 web application has been rewritten in HTML5.

Data Entry in Excel Add-In

You can now edit the form data only through SAS Financial Management Add-In for Microsoft Excel. You cannot edit the forms in the Forms workspace. For more information about the Excel Add-In, see the online Help.

Administration and Processes Workspaces Are Removed

In the HTML5 user interface, the Administration and Processes workspaces are not provided. You can complete the tasks related to administration and business processes through SAS Financial Management Studio.

Help Documentation

All the help documentation for SAS Financial Management is available through the Forms Management Web interface. Click the User name in the banner of the application, and select **Help Center**.

Part 1

Introduction

Chapter 1

Overview of SAS Financial Management 3

Chapter 1

Overview of SAS Financial Management

SAS Financial Management Activities	3
Client Applications	3
SAS Financial Management Data	4
Cycles and Dimension Types	4
Dimension Members and Crossings	4
Hierarchies	5
Models	5
Additional Documentation	5

SAS Financial Management Activities

SAS Financial Management is an advanced system for planning. It supports the following financial management activities:

- data collection and retrieval
 - currency translation
 - management of dynamic hierarchical structures
 - intercompany eliminations
 - allocations and balancing entries
 - ownership eliminations
-

Client Applications

Users interact with SAS Financial Management by using one of the following client applications:

- SAS Financial Management Studio

In SAS Financial Management Studio, administrators build and manage the infrastructure that is required by the other two SAS Financial Management clients.

- SAS Financial Management Add-In for Microsoft Excel

SAS Financial Management Add-In for Microsoft Excel connects a desktop copy of Microsoft Excel to the SAS Financial Management database. Administrators also use the SAS Financial Management Add-In for Microsoft Excel client to create form templates for financial planning.

- SAS Financial Management Web Portal

From a web browser, administrators and end users can log on to a SAS Financial Management web application. The SAS Financial Management web application includes a workspace for managing forms for financial planning. The flow of data is controlled by a workflow that is defined in SAS Financial Management Studio.

Every SAS Financial Management deployment also includes SAS Data Integration Studio, which administrators use to load data and metadata for SAS Financial Management.

SAS Financial Management Data

Cycles and Dimension Types

In SAS Financial Management, a cycle is a structured pool of stored data. The data is stored as a dimension type. The following seven dimension types are required in a cycle:

- Account
- Analysis
- Currency
- Organization
- Time
- Source
- Trader (a mirror of Organization)

Note: Administrators can add additional dimension types to meet the needs of a site.

Dimension Members and Crossings

Each numeric value belongs to a crossing, which consists of the set of all dimension members that are associated with that value. There is one member from each relevant dimension type.

Each data record in the SAS Financial Management database consists of one crossing and one associated numeric value. The dimension members that are associated with a SAS Financial Management data record tell you what the numeric value in that record represents. For example, one record's dimension members might tell you that the numeric value represents the actual revenue for an Italian subsidiary in October 2011, expressed in euros. Another record's dimension members might tell you that the numeric value represents the planned salary expense for a Japanese subsidiary for fiscal year 2012, expressed in yen.

Hierarchies

The dimension members that are used by SAS Financial Management belong to hierarchies. The hierarchical relationships between members help define the dimension structure.

For certain dimension types (Currency, Frequency, and Analysis), there is no hierarchical relationship between the dimension members. These are known as flat hierarchies.

Models

A model is a structure for viewing and interacting with the data in a cycle. A model is the basis for both forms and reports. The structure of a model includes a set of hierarchies (from required and optional dimension types) as well as rates and formulas. A model can also have its own set of adjustments and rules.

Additional Documentation

For additional information about SAS Financial Management, see the documentation at <http://support.sas.com/documentation/onlinedoc/fm/index.html>.

- *SAS Financial Management: User's Guide*
This book is designed primarily for end users and is also available from the **Help Center** menu to users who are logged on to SAS Financial Management in a web browser.
- *SAS Financial Management: Process Administrator's Guide*
- *SAS Financial Management: System Administrator's Guide*
- *SAS Financial Management: Migration Guide*
- *SAS Financial Management: Customization Guide*
- *SAS Financial Management: Data Administrator's Guide*
- *SAS Financial Management: Data Model Reference*
- *SAS Financial Management: Formula Guide*
- *SAS Financial Management: Performance Guide*

Note: The site is password-restricted. You can find the user name and password in the preinstallation checklist or by contacting SAS Technical Support at <http://support.sas.com/techsup/contact>.

For information about the SAS Intelligence Platform, see <http://support.sas.com/94administration>.

Part 2

Administering SAS Financial Management

<i>Chapter 2</i>	
Getting Started with SAS Financial Management Studio	9
<i>Chapter 3</i>	
The Dimensions Workspace	19
<i>Chapter 4</i>	
Member Properties	33
<i>Chapter 5</i>	
Required Dimension Types	39
<i>Chapter 6</i>	
Working with CTA Accounts	51
<i>Chapter 7</i>	
Working with Retained Earnings Accounts	57
<i>Chapter 8</i>	
Cycles and Cycle Periods	69
<i>Chapter 9</i>	
Cycle Properties	77
<i>Chapter 10</i>	
Working with Exchange Rate and Driver Rate Sets	81
<i>Chapter 11</i>	
Overview of Models	87
<i>Chapter 12</i>	
Model Properties	93
<i>Chapter 13</i>	
Models: Rules and Adjustments	101

<i>Chapter 14</i>	
Models: Adjustment Rule Types	<i>113</i>
<i>Chapter 15</i>	
Composite Models	<i>121</i>
<i>Chapter 16</i>	
Formulas	<i>125</i>
<i>Chapter 17</i>	
Forms and the Forms Workspace	<i>153</i>
<i>Chapter 18</i>	
Task Management	<i>167</i>
<i>Chapter 19</i>	
Security in SAS Financial Management	<i>171</i>

Chapter 2

Getting Started with SAS Financial Management Studio

Working in SAS Financial Management Studio	9
Overview	9
Dimensions Workspace	10
Periods Workspace	10
Rates Workspace	10
Models Workspace	11
Forms Workspace	11
User Roles and Capabilities	12
Capabilities for SAS Financial Management Studio	12
Roles for Editing and Approving Forms	12
Loading Data to the SAS Financial Management Data Mart	12
Locked Objects	13
Overview of Locked Objects	13
Locked Cycles	14
Locked Cycle Periods	14
Locked Analysis Members of Cycle Periods	15
Locked Models	15
Locked Form Sets	15
Locked Forms	15
Promoting Objects to Another System	16
Using Views	17
Using the Help System	18

Working in SAS Financial Management Studio

Overview

SAS Financial Management Studio has these workspaces:

- **Dimensions**—used to define dimensions, hierarchies, members, and custom properties
- **Periods**—used to manage the time periods and the data in an open cycle
- **Rates**—used to manage exchange rates and driver rates

- **Models**—used to define and manage models, composite models, manual adjustments, adjustment rules, cell protection rules, and visibility rules
- **Forms**— used to create and manage form sets

Note: The objects in the Dimensions workspace apply to all cycles. The objects in the other four workspaces apply to the cycle that is currently open. To select and open a cycle, select **Tools** ⇒ **Cycle Manager**, or use the cycle options on the **File** menu.

Dimensions Workspace

Use the Dimensions workspace to view and manage dimension types, dimensions, hierarchies, members, data locales, custom properties, and calendar templates. This information is typically loaded from the SAS Financial Management staging area, but you can use the Dimensions workspace to view and edit it.

Access to dimensions, hierarchies, and members is available from the menu on the left of the Dimensions workspace. Access to the remaining objects is available from the **Tools** menu.

Periods Workspace

Use the Periods workspace to manage the time periods and the data in the open cycle.

In the Periods workspace, you can perform the following tasks:

- create new periods
- load new data from SAS Data Integration Studio
- load data from another model
- delete data
- delete cell comments
- lock time periods
- show properties for a period

Rates Workspace

Use the Rates workspace to view and manage exchange rates and driver rates. Exchange rates are used in the currency conversion process. In SAS Financial Management Studio, driver rates can be retrieved by the DRATE function. In the SAS Financial Management Add-In for Microsoft Excel, rates can be retrieved by the CDARate and fmRate functions.

In the Rates workspace, you can perform the following tasks:

- create new exchange rate sets and driver rate sets
- load exchange rates and driver rates for a rate set
- export exchange rates and driver rates for a rate set
- display exchange rates and driver rates for a rate set
- copy and delete exchange rate sets and driver rate sets
- display the properties of exchange rate sets and driver rate sets

Use the **Tools** menu to view and manage Exchange Rate and Driver rate types.

Models Workspace

A model provides a view of the selected cycle. Use the Models workspace to create and manage models.

In the Models workspace, you can perform the following tasks:

- create new models
- import and export models
- post adjustments for a model
- export data for a model
- lock a model
- copy and delete models
- display and manage the following types of rules for a model:
 - adjustment rules
 - manual adjustments rules
 - cell protection rules
 - cell visibility rules
 - data validation rules
- display posted adjustment data for a model
- display attachments for a model
- display properties of a model

Composite models consist of two or more models that share the same dimension types. Any adjustments and rules that are associated with the referenced models are included in these composite models.

To manage composite models, open the **Models** workspace and select **Tools** ⇒ **Composite Models**.

Forms Workspace

Use the Forms workspace to create and manage form sets.

Form distribution is based on a target hierarchy. When administrators create a form set, they select a dimension, hierarchy, and specific hierarchy members to be associated with the published forms.

Form sets have a bottom-up workflow or a top-down workflow that controls the flow of data. In a bottom-up workflow, data is entered at the bottom of the target hierarchy and then aggregated. In a top-down workflow, data is entered at the top of the target hierarchy and then allocated.

User Roles and Capabilities

Capabilities for SAS Financial Management Studio

Each workspace has a corresponding capability that grants full access to that workspace and the ability to browse content in the other workspaces. To log on to SAS Financial Management Studio, users must have at least one of the workspace capabilities.

SAS Financial Management includes a default set of roles that include these capabilities. For details, see “Assigning Groups and Roles” in the *SAS Financial Management: System Administrator’s Guide*.

Roles for Editing and Approving Forms

Users must have appropriate capabilities in order to be assigned as authors or reviewers of a form set. For details, see “Assigning Groups and Roles” in the *SAS Financial Management: System Administrator’s Guide*.

Loading Data to the SAS Financial Management Data Mart

SAS Financial Management Studio provides the following options for loading different types of data to the SAS Financial Management data mart:

- **To load members and hierarchies into a dimension**, use the **Load Dimension** option in the Dimensions view.

This option launches the Load Dimension wizard. You can use this option to load members and hierarchies into a SAS Financial Management database for the first time.

You can also use it to load members and hierarchies that have been exported from a parallel SAS Financial Management system as part of an object promotion routine.

Corresponding task in SAS Data Integration Studio: Load Dimension.

- **To load data into a cycle**, use the **Load New Data** option in the Periods view.

This option launches the Load New Data wizard. Loading data into a cycle from the SAS Financial Management staging area is part of the job of managing the data in a cycle. There are other ways in which data can enter a cycle:

- entering data in a form
- loading data from one cycle to another with the **Load Model Data** option

Corresponding task in SAS Data Integration Studio: Load Base Data.

Note: These wizards offer deletion options for existing data. There is also a **Delete Data** option that deletes specified data from a cycle without loading any data. Typically, you need to perform several of these activities in a coordinated manner.

- **To load exchange rates**, use the **Load Exchange Rates** option in the Exchange Rate Sets view.

This option launches the Load Exchange Rates wizard.

Corresponding task in SAS Data Integration Studio: Load Exchange Rates.

- **To load driver rates**, use the **Load Driver Rates** option in the Driver Rate Sets view.

This option launches the Load Driver Rates wizard.

Corresponding task in SAS Data Integration Studio: Load Driver Rates.

For details, see the online Help for each wizard.

The History page of the object's Properties window shows the history of these load operations.

For details about the SAS Data Integration Studio tasks, see the *SAS Financial Management: Data Administrator's Guide*.

Locked Objects

Overview of Locked Objects

SAS Financial management supports the locking of the following objects:

- cycles
- cycle periods
- analysis members of cycle periods
- models
- form sets
- forms

An object in a locked state can be viewed but cannot be modified. The locked state is indicated with a padlock in the object's icon .

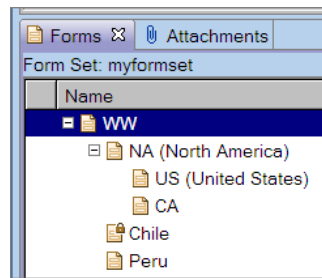
To unlock an object, reverse the process. If you locked a form set with the **Lock** button, click the **Unlock** button to unlock it.

Because objects of different types are interrelated, many objects can become locked or unlocked in more than one way. For example, you can lock a single form or a group of forms. But when you lock a form set, that action implicitly locks all the forms in the form set.

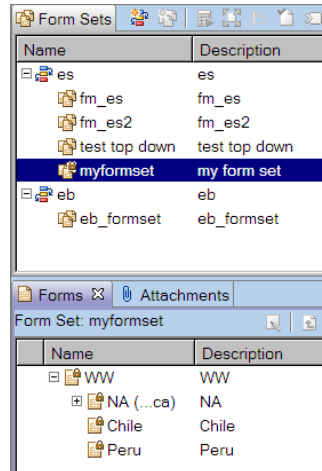
In some cases, an object is affected by more than one lock. For example, there might be a lock on a form, the related form set, and the related cycle. To undo those locks, begin by unlocking the most comprehensive lock. The subordinate locks still apply until you unlock them.

Here is an example:

- In form set myformset, you lock the form for Chile. The form is explicitly locked and its icon includes a padlock.



- Then you lock myformset. It is explicitly locked, and all its forms are implicitly locked. Notice that the icons for all the forms now include a padlock.



You cannot unlock the form for Chile until you unlock myformset. When you unlock myformset, all the forms except the form for Chile are implicitly unlocked. The form for Chile is still locked until you explicitly unlock it.

Locked Cycles

If a cycle is locked, all the cycle periods, models, form sets, and forms that belong to it are implicitly locked. You cannot create new cycle periods, models, form sets, or forms for the cycle. In addition, you cannot change the locked cycle's exchange rates or driver rates.

To lock the open cycle, select **File** ⇒ **Lock Cycle**.

To lock any unlocked cycle, use the **Lock** button in the Cycle Manager window.

Locked Cycle Periods

If a cycle period is locked, you cannot load data into it, post adjustments to it, delete data from it, or enter data into it through a form. In addition, you cannot lock or unlock analysis members for the period.

Locking does not apply in the following situations:

- the management of exchange rate and driver rate sets.
- deletion of a manual adjustment or an adjustment rule immediately deletes all postings that exist from that rule.
- deletion of a form set. If you choose the option to delete form data, the data is immediately deleted, regardless of locks.

To lock a cycle period, complete one of the following tasks:

- Use the **Lock** option in the Periods view.
- Lock the cycle to which the period belongs. This action implicitly locks all associated periods in the cycle.

Locked Analysis Members of Cycle Periods

If an analysis member of a cycle period is locked, then you cannot load data into that analysis member, post adjustments to it, delete data from it, or enter data into it through a form. However, these operations are still available for other analysis members of the same period.

To lock an analysis member of a cycle period, complete one of the following tasks:

- Open the properties for a period and select the **Analysis Member Locks** tab. Select the analysis members to lock and click **OK**.
- Select multiple periods, right-click, and select **Properties**. In the Analysis Member Locks window, select the analysis members that you want to lock and click **OK**.

A locked analysis member is indicated by a lock icon  at the right of the entry for that member.

Locked Models

If a model is locked, you cannot modify its properties, manual adjustments, or adjustment rules. Posting adjustments or rules is disabled, and form data entry is not allowed. However, you can still create and publish form sets that use the model.

To lock a model, complete one of the following tasks:

- Use the **Lock** option in the Models view.
- Lock the cycle to which the model belongs. This action implicitly locks the model.

Locked Form Sets

If a form set is locked, you cannot modify its properties, modify its template, or change its status. A locked form set implicitly locks the associated forms so that no data entry can be performed.

To lock a form set, complete one of the following tasks:

- Use the **Lock** option in Forms view.
- Lock the cycle to which the form set belongs. This action implicitly locks the form set.

Locked Forms

If a form is locked, data entry is not permitted. Locking a form can be manual or automatic:

- **Manual locking.** To lock a form manually, complete one of the following tasks:
 - Lock the form via the **Lock** option in Forms view.

- Lock the form set that the form belongs to. This action implicitly locks all the forms in the form set.
- Lock the cycle that the form set belongs to. This action implicitly locks the form set and all its forms.
- **Automatic locking.** When you create a form set or modify its properties, you can also specify automatic locking. If you enable the **Autolock** option, forms are locked when the deadline arrives.

Note: You cannot unlock a form that has been automatically locked unless you first disable automatic locking or change the form set deadline.

Promoting Objects to Another System

SAS Financial Management Studio provides the means to promote objects of certain types from one system to another. For example, you can build objects on a development system and promote them to a test system for testing. Then you can promote them to a production system when validation testing is complete.

You can promote objects by using the following methods:

- **Dimensions and hierarchies**—**Export Dimension** and **Load Dimension** options in the Dimensions workspace.

The **Export Dimension** option launches the Export Dimension wizard, which exports members and hierarchies to a set of SAS tables in a SAS library.

The **Load Dimension** option launches the Load Dimension wizard, which imports members, hierarchies, security settings, and users from a corresponding set of SAS tables in the SAS Financial Management staging area.

- **Driver Rates**—**Export Driver Rates** and **Load Driver Rates** options in the Rates workspace.

The **Export Driver Rates** option launches the Export Driver Rates wizard, which exports driver rates from a driver rate set into a target set in a SAS library.

The **Load Driver Rates** option launches the Load Driver Rates wizard, which imports driver rates from a corresponding driver rate set in the SAS Financial Management staging area.

- **Exchange Rates**—**Export Exchange Rates** and **Load Exchange Rates** options in the Rates workspace.

The **Export Exchange Rates** option launches the Export Exchange Rates wizard, which exports exchange rates from an exchange rate set into a target set in a SAS library.

The **Load Exchange Rates** option launches the Load Exchange Rates wizard, which imports exchange rates from a corresponding exchange rate set in the SAS Financial Management staging area.

- **Models**—**Export Models** and **Import Models** options in the Models workspace.

The **Export Models** option launches the Export Models wizard, which exports models to a SAS package file.

The **Import Models** option launches the Import Models wizard, which imports models from a SAS package file.

- **Allocation rules and ownership rules—Export Adjustment Rules and Import Adjustment Rules** options in the Adjustment Rules view of the Models workspace.

The **Export Adjustment Rules** option launches the Export Adjustment Rules wizard, which creates a SAS package file that contains the exported adjustment rules.

The **Import Adjustment Rules** option launches the Import Adjustment Rules wizard. This wizard imports adjustment rules into a selected model from a SAS package file that was created by the Export Adjustment Rules wizard.

- **Manual Adjustment Rules—Export Manual Adjustments and Import Manual Adjustments** options in the Manual Adjustment Rules view of the Models workspace.

The **Export Manual Adjustments** option launches the Export Manual Adjustments wizard, which creates a SAS package file that contains the exported manual adjustments.

The **Import Manual Adjustments** option launches the Import Manual Adjustments wizard. This wizard imports manual adjustments from a SAS package file into a selected model from a SAS package file that was created by the Export Manual Adjustments wizard.

- **Export and import form sets options—Export Form Sets and Import Form Sets** options in the Form Sets view of the Forms workspace.

The **Export Form Sets** option launches the Export Form Sets wizard, which creates a SAS package file that contains the exported form sets.

The **Import Form Sets** option launches the Import Form Sets wizard, which imports form sets from a SAS package file that was created by the Export Form Sets wizard. The imported form sets must use the same cycle and model as the exported form sets.

- **Cycles—Export and Import** buttons in the Cycle Manager window.

The **Export** button launches the Export Cycles wizard, which creates a SAS package file that contains the exported cycles.

The **Import** button launches the Import Cycles wizard, which imports cycles from a SAS package file that was created by the Export Cycles wizard.

For details, see the online Help for each wizard.

To promote an object of any other type, create an object in the target system that is identical to the object in the source system.

Using Views

At the top of each open view is a tab that contains the name of the view. If you right-click a view's tab, a pop-up menu appears that contains options for manipulating the appearance of the view. The **Maximize** and **Minimize** options are also available as buttons at the right end of the view's toolbar. The **Close** option is also available as an **X** on the view's tab.

To return all views to their default appearance, select **View ⇒ Restore Default Layout**.

Many views display a sortable list of items. For example, the Dimensions view displays a sortable list of dimensions. To sort the items in a sortable view, click a column heading. To reverse the sort order, click the column heading again.

In many views, options are listed in the following four places:

- the pop-up menu that you can display by right-clicking while the cursor is in the view's display area
- the toolbar at the top of the view
- the toolbar menu, which appears when you click the **View Menu** button  in the toolbar
- the task list on the task pane to the left of the view

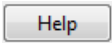
In a view where you can select items, the set of available options depends on which items are currently selected. In general, an option is available if it can be applied to the set of selected items or if it does not require a selected item.

Note: The name for an option might be slightly different depending on where it occurs. For example, the task list typically displays a phrase, such as **Show properties for this hierarchy**. However, the pop-up menu and toolbar menu typically display a single word, such as **Properties**.

Using the Help System

The Help system includes a table of contents, an index, a full-text search facility, and a bookmark facility.

Help is context sensitive:

- Press the F1 key to display the Help for the active view.
- From a window, wizard page, or property page, press the F1 key or click the **Help** button ( or ).

A list of related topics appears.

- **See also**—contains the topic that is most applicable to the selected object or view.
- **More results**—lists additional related topics.

Chapter 3

The Dimensions Workspace

Introduction to Dimensions, Hierarchies, and Members	20
Working with Dimension Types	20
Working with Dimensions, Hierarchies, and Members	22
Dimensions View	22
Hierarchies View	24
Overview of the Hierarchies View	24
Hierarchy Options	24
Member Options	25
Members View	26
Overview of the Members View	26
Available Options	26
Using Custom Properties	27
Reasons for Using Custom Properties	27
Ways to Define Custom Properties	27
Review or Define Custom Properties	27
Add Custom Property Values to Members	28
Custom Properties View	28
Using Data Locales	29
How Data Locales Work	29
Creating New Data Locales	29
Data Locales View	30
Using Calendar Templates	30
How to Use Calendar Templates	30
Create a Calendar Template	31
Add Time Periods to a Hierarchy with a Calendar Template	31
Add Time Periods to a Dimension with a Calendar Template	31
Calendar Templates View	31

Introduction to Dimensions, Hierarchies, and Members

Dimension types, dimensions, hierarchies, and members define the structures that is used for planning and reporting.

- *Dimension types* represent the categories that define your data. A dimension type defines what type of thing the members of the dimension are. Some basic dimension types include Organization, Account, Time, Analysis, and Currency.
- A *dimension* is a data element that categorizes values in a data set into non-overlapping categories that can be used to group, filter, and label the data in meaningful ways. Multiple dimensions can share the same dimension type.
- A *member* is an element of a dimension. For example, for a dimension that contains time periods, each time period is a member of the dimension.

A member can reside in multiple dimensions, and its properties can have different values in these different dimensions. This enables you to create variants of a member that can be accessed by different sets of users or that have different behavior. For example, an account could change its account type or its exchange rate type from one account dimension to another.

- A dimension can have one or more hierarchies.

A *hierarchy* is an arrangement of related objects into levels that are based on parent-child relationships. Members of a hierarchy are arranged from more general to more specific. Hierarchies represent the reporting structure for the data. Hierarchies within a dimension typically represent different groupings of information that pertains to a single concept. For example, a Time dimension might consist of two hierarchies: (1) Year, Month, and Date, and (2) Year, Week, and Day.

Note: The Analysis, Currency, and Frequency dimension types are exceptions. They support only flat member lists.

Working with Dimension Types

To open the Dimension Types view, select **Tools** ⇒ **Dimension Types**.

The following dimension types are predefined:

Dimension Type by Code	Description
ACCOUNT	Account (required)
ANALYSIS	Analysis (required)
COSTCTR	Cost center
EXTORG	External organization

Dimension Type by Code	Description
FREQUENCY	Frequency (required)
INTORG	Internal Organization (required)
ITEMCAT	Item category
PERIODS	Periods
PROFITCTR	Profit center
SOURCE	Source (required)
TIME	Time (required)
TRADER	Trader (required)

You can define additional dimension types in SAS Financial Management Studio or in SAS Data Integration Studio.

In the Dimension view, the following options are available:

New Dimension Type

launches the New Dimension Type wizard, which enables you to create a dimension type.

For details, see the online Help for the individual wizard pages.

To create a new dimension type that can be used to load data from the SAS Financial Management staging area, you must also complete certain tasks in SAS Data Integration Studio. You can complete all of the tasks required to create a new dimension type in SAS Data Integration Studio. If you choose to create a new dimension type in its entirety in SAS Data Integration Studio, do not run the New Dimension Type wizard.

For information about creating dimension types, see the *SAS Financial Management: Data Administrator's Guide*.

Delete

deletes the selected dimension type.

A confirmation window reminds you that when you delete a dimension type, you also delete all dimensions that belong to that dimension type.

Note: You can delete a dimension type only if the dimension type is not referenced by another object. In addition, many dimension types are predefined and cannot be deleted. When you select one of the predefined dimension types, the **Delete** option is not available.

Properties

displays the properties of the selected dimension type.

Working with Dimensions, Hierarchies, and Members

To work with dimensions, use the Dimensions view.

The **Load Dimension** option in the Dimensions view enables you to load hierarchies and members, including custom properties, into a selected dimension from the SAS Financial Management staging area.

To work with hierarchies:

1. In the Dimensions view, select the dimension that contains the hierarchies that you want to work with.
2. Select **Hierarchies**.

The Hierarchies view for the selected dimension opens.

3. In the Hierarchies view, select a hierarchy or a hierarchy member.

Note: You can view but not edit the Source dimension. You can view the Frequency hierarchy and reorder its members. However, you cannot add or delete members.

A dimension can contain multiple hierarchies. A hierarchy can contain all the members in a dimension or a subset of those members. Members are not required to belong to a dimension.

To work with a member outside a hierarchy context:

1. In the Dimensions view, select the dimension that contains the member.
2. Select **Members**.
3. In the Members view, select the member.

Each hierarchy has a default member. The default member is used when a query does not explicitly include that dimension. The default member is also referred to as the default read member.

Dimensions View

The Dimensionsworkspace includes these views:

- Dimensions
- Hierarchies
- Members
- Dimension Types
- Custom Properties
- Data Locales
- Calendar Templates

The Dimensions view lists all the dimensions for each dimension type.

The following options are available:

New Dimension

launches the New Dimension wizard, which enables you to create a dimension.

For details, see the online Help for the individual wizard pages.

Load Dimension

launches the Load Dimension wizard, which enables you to load members and hierarchies from the SAS Financial Management staging area into the selected dimension.

For details, see the online Help for the individual wizard pages.

Export Dimension

launches the Export Dimension wizard, which enables you to export the members and hierarchies of the selected dimension to a SAS library.

For details, see the online Help for the individual wizard pages.

Copy Here

creates a copy of the selected dimension, including all the members and hierarchies that it contains. The copy is identical to the original except for the new code, name, and description that you give it.

Delete

deletes the selected dimension.

Note: You can delete only dimensions that are not referenced by another object.

A confirmation window reminds you that when you delete a dimension, you also delete all of the hierarchies and members that belong to that dimension.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. The list narrows to display only the dimension that match the string that you are entering. Partial matches are accepted.

To further restrict the search, click the **Show options** button . In the window that appears, select one or more dimension types.

Refresh

refreshes the view.

Hierarchies

opens the Hierarchies view for the selected dimension.

Members

opens the Members view for the selected dimension.

Properties

displays the Properties window, which enables you to view (and in some cases edit) the properties of the selected dimension.

Hierarchies View

Overview of the Hierarchies View

The Hierarchies view displays the hierarchies in a dimension.

The name on the tab identifies the dimension to which the hierarchies belong (for example, **My_Account Hierarchies**).

Hierarchy Options

When you select a hierarchy, the following options are available. Many of these options are also available when you select a hierarchy member.

New Hierarchy

launches the New Hierarchy wizard, which enables you to create an entry for a new hierarchy.

For details, see the online Help for the individual wizard pages.

Add Member

launches the Add Member wizard, which enables you to add a member as a top-level member of the selected hierarchy.

You can create the member that you are adding, or copy a member that already exists in this dimension or in another dimension that belongs to the same dimension type.

For details, see the online Help for the individual wizard pages.

TIP You can also drag-and-drop a member from another hierarchy in the same dimension, or from the Members view of the same dimension.

Copy

creates a copy of the selected hierarchy and adds the copy to the bottom of the display. A wizard page enables you to specify a code, name, and description for the copy.

Delete

deletes the selected hierarchy.

Expand All

expands the entire hierarchy.

Collapse All

collapses all hierarchies completely.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Members

opens the Members view for the selected dimension, which lists all the members in the dimension.

Properties

displays the Properties window, which enables you to view (and in some cases edit) the properties of the selected hierarchy.

Find

displays or hides the Find window. To find a member, enter a character string and click **Find Next**. Partial matches are accepted.

To further restrict the search, click the **More Options** and make one or more selections.

Note: In the case of very large hierarchies, the **Find in all columns** option is not available because the search time would be excessively long.

Refresh

refreshes the view.

Member Options

The **New Hierarchy**, **Add Member**, and **Members** options also apply when you select a member in Hierarchy view. In addition, the following additional options are available:

Move Up

moves the selected member and all its subordinate members up one position in the hierarchy display.

Note: This option is active only if the selected member has at least one sibling that is currently displayed before it.

Move Down

moves the selected member and all its subordinate members down one position in the hierarchy display.

Note: This option is active only if the selected member has at least one sibling that is currently displayed after it.

Move

moves the selected member and all its subordinate members to another location in the same hierarchy.

When you select this option, the Move Member window appears, in which you specify the target location.

TIP You can also move a member and all its subordinate members to any location using drag-and-drop.

Remove

removes the member from the selected hierarchy (but not from the dimension). Members subordinate to the selected member are also removed, if there are any.

To delete a member from the dimension, remove the member from all hierarchies. Then use the **Delete** option from the Members view.

Set As Default Member

makes the selected member the default member for the hierarchy within which you select it. In the display, the default member is marked by an icon that has a thick black border.

Expand All

expands the portion of the hierarchy that is subordinate to the selected member.

Collapse All

collapses all hierarchies completely.

Refresh

updates the display to reflect any changes that occurred after you opened the Hierarchies view.

Properties

If one member is selected, then this option displays the Properties window, enabling you to view all the properties of the selected member and make changes.

If two or more members are selected, then this option displays a properties window. In this window, you can make changes to the custom properties and security settings of all the selected members.

Members View

Overview of the Members View

The Members view displays a list of the members in a dimension.

The name on the tab identifies the dimension the members belong to (for example, **My_Account Members**).

Available Options

The following options are available:

New Member

launches the New Member wizard, which enables you to add a member to the selected dimension. You can create a new member or copy a member from another dimension that belongs to the same dimension type.

For details, see the online Help for the individual wizard pages.

Copy Here

launches the Copy Members wizard, which enables you to make copies of the selected member or members. You give the copies their own codes, names, and descriptions, but they have the same properties as the selected members that you are making copies of.

For details, see the online Help for the individual wizard pages.

Delete

deletes the selected member or members from the selected dimension.

You cannot delete a member from a dimension if the member is part of a hierarchy. First, delete the member from each hierarchy that includes it, and then delete it from the dimension.

Hierarchies

opens the Hierarchies view for the selected dimension.

Properties

If one member is selected, then this option displays the Properties window, enabling you to view all the properties of the selected member and make changes.

If two or more members are selected, then this option displays a properties window. In this window, you can make changes to the custom properties and security settings of the selected members.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. Partial matches are accepted.

To further restrict the search, click the **Show options** button . In the window that appears, select one or more options.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Using Custom Properties

Reasons for Using Custom Properties

In the SAS Financial Management Add-In for Microsoft Excel, the custom properties of members have the same potential uses as standard properties. The values of custom properties can appear in tables as row, column, or slicer headings that serve to label, group, or filter the members in the table. In addition, the values of custom properties can be retrieved by the CDAProperty function or the calculated-member function fmProperty, and used by any formula that includes one of these functions.

Custom properties can also be used as custom measures in a supplemental schedule. The custom property must be associated with any member of the supplemental schedule's measure dimension. For details, see “Creating a Supplemental Schedule” in the online Help for the SAS Financial Management Add-In for Microsoft Excel.

Note: If users are denied Read access to a custom property, it does not appear in a data-entry form or a report.

Ways to Define Custom Properties

Custom properties can be defined in the Custom Properties view. They can also be defined in the SAS Financial Management staging area and loaded from there as part of the operation of loading members into a dimension.

We recommend that you register any custom properties in the SAS Financial Management staging area first. Then, when you export a dimension from SAS Financial Management Studio by using the Export Dimension wizard, the custom properties are included in the export operation. Otherwise, the changes that you make in SAS Financial Management Studio might be lost.

For information about registering custom properties, see the *SAS Financial Management: Data Administrator's Guide*.

Review or Define Custom Properties

To review existing custom properties or to define new ones:

1. Select **Tools** ⇒ **Custom Properties**.

2. Use the Custom Properties view.

Add Custom Property Values to Members

To add values for custom properties to one or more members, do one of the following in the Hierarchies view:

- Select one or more members, and then select **Tools** ⇒ **Custom Properties** and use the Custom Properties view.

This method enables you to add custom property values to many members at once.

- Select a member, and then select **Properties** and use the Custom Properties page of the Properties window.

This method limits you to working with one member at a time.

Custom Properties View

The Custom Properties view lists the existing custom properties.

The following options are available:

New Custom Property

launches the New Custom Property wizard, which enables you to create a custom property.

Delete

deletes the selected custom properties.

Properties

displays the properties of the selected custom property.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. Partial matches are accepted.

To further restrict the search, click the **Show options** button . In the window that appears, select one or more property types.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Using Data Locales

How Data Locales Work

Data locales enable SAS Financial Management Studio users to choose a language in which to display the names and descriptions of the following objects:

- dimension types
- dimensions
- hierarchies
- dimension members
- exchange rate types
- driver rate types
- models

A data locale is specified by one or both of the following two components:

- language
- country or region

The country or region component is usually, but not always, a country. For example, it might represent a geographic region within a country.

If you define two or more data locales and associate a set of names and descriptions with each one, then each user sees the names and descriptions for the data locale that best matches the regional settings of the user's computer. The data locales are ranked according to how well they match the user's regional settings, as follows:

1. The language and the country or region both match.
2. The language matches but the country or region does not match.
3. The country or region matches but the language does not match.

The current data locale is the one that best matches the user's regional settings. It is identified in the bottom margin of the main window of SAS Financial Management Studio.

For each name or description, a user sees the text from the highest-ranking data locale that has text for that name or description.

The default data locale is used if no data locale matches your computer's regional settings, or if no matching data locale has text for a particular name or description. You can set the default data locale in the Data Locales view.

Creating New Data Locales

You can create new data locales by using one of the following methods:

- Populate the CODE_LANGUAGE table in the SAS Financial Management staging area, and then run the Import Locales job to load data locale information from the SAS Financial Management staging area.
- Select **Tools** ⇒ **Data Locales** and use the Data Locales view.

A data locale must be defined in the SAS Financial Management staging area to enable you to load member and hierarchy names and descriptions for that locale from the SAS Financial Management staging area. Therefore, in general, it is best to use the CODE_LANGUAGE table and the Import Locales job to create a new data locale.

For details about loading data through the SAS Financial Management staging area, see *SAS Financial Management: Data Administrator's Guide*.

Data Locales View

The Data Locales view lists the available data locales.

The current data locale is identified in the bottom margin of the main window.

The default data locale is marked by a symbol with a black ring .

The following options are available:

New Data Locale

launches the New Data Locale wizard. This wizard enables you to create a data locale and add it to the list.

Set As Default

makes the selected data locale the default data locale.

To make this change take effect, close and reopen SAS Financial Management Studio.

Delete

deletes the selected data locale.

Properties

displays the properties of the selected data locale.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. Partial matches are accepted.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Using Calendar Templates

How to Use Calendar Templates

You can use a calendar template to add a full year of time periods to a time hierarchy in a single operation. If you load all your time periods from the SAS Financial Management staging area, then you do not need calendar templates.

Create a Calendar Template

To create a calendar template:

1. Select **Tools** ⇒ **Calendar Templates**.
2. Use the Calendar Templates view.

Add Time Periods to a Hierarchy with a Calendar Template

To use a calendar template in order to generate a set of time members and add them to a hierarchy:

1. In the Hierarchies view for the appropriate time dimension, select the hierarchy that you want to add the new time members to.
2. Select **Add Member** to launch the Add Member wizard.
3. On the Method page, select **Create members using a calendar template**.
4. Continue through the wizard.

For details, see the online Help for the individual wizard pages.

Add Time Periods to a Dimension with a Calendar Template

To use a calendar template in order to generate a set of time members outside the context of a hierarchy:

1. In the Members view for the appropriate time dimension, select any member.
2. Select **New Member** to launch the New Member wizard.
3. On the Creation Method page, select **Create members using a calendar template**.
4. Continue through the wizard.

For details, see the online Help for the individual wizard pages.

Calendar Templates View

The Calendar Templates view lists calendar templates. The following options are available:

New Calendar Template

launches the New Calendar Template wizard to create a calendar template without using any default specifications.

For details, see the online Help for the individual wizard pages.

Copy Here

launches the New Calendar Template wizard to create a calendar template using default specifications from the selected calendar template.

For details, see the online Help for the individual wizard pages.

Delete

deletes the selected calendar templates.

Properties

displays the properties of the selected calendar template.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. Partial matches are accepted.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Chapter 4

Member Properties

Account Details Page	33
Custom Properties Page	34
Overview	34
Properties for a Single Member	34
Properties for Multiple Members	34
Formulas Page	35
Identification Page	35
Overview of the Identification Page	35
Member Identification	35
Roll-Up Check Box	35
Security Page	36
Overview of the Security Page	36
Significance of Read Access	36
Significance of Write Access	36
Setting Read and Write Permissions	37
Setting Permissions for Multiple Members	37
Overriding Member-Level Security	37
Time Details Page	38
Users Page	38
Overview of the Users Page	38
Adding Users	38
Removing Users	38
How Authors and Reviewers Are Assigned to Forms	38

Account Details Page

Use the Account Details page to view and modify the properties of the selected account.

The set of properties varies with the account type and includes some of the following:

- To change the account type, use the drop-down list for the **Account type** field.
- To change the exchange rate type, use the drop-down list for the **Exchange rate type** field.

Accounts that belong to the Cumulative Translation Adjustment account type have a fixed Period Close exchange rate type. Accounts that belong to the Retained Earnings account type do not have an exchange rate type.

- To change the balance type, click the appropriate **Balance type** radio button.
- To change the value of the Intercompany property, click the **Intercompany** check box.
- To change the roll-forward method for an account that belongs to the Retained Earnings account type, use the drop-down list for the **Roll-forward method** field.
- To change the basis data selections for an account that belongs to the Retained Earnings account type, click check boxes as appropriate in the **Basis data** region.
- To add or remove source accounts for an account that belongs to the Retained Earnings account type or the Cumulative Translation Adjustment account type, use the **Add** and **Remove** buttons in the **Source accounts** region.

Custom Properties Page

Overview

Use the Custom Properties page to view or modify the custom properties for one or more selected members.

Properties for a Single Member

The Custom Properties page lists all the custom properties of the selected member.

- To add a custom property, click **Add**. When you add a custom property, you must select a value.
- To remove a custom property, select the property and click **Remove**.
- To change a property value, select from the **Value** drop-down list.
- To modify the columns that are displayed on the page, click **Customize Columns**.

Properties for Multiple Members

If you view properties for more than one member, the Custom Properties page lists all custom properties that belong to any of the selected members. If all the selected members share the same property value, it is displayed in the **Value** column. Otherwise, the **Value** column is blank.

- To modify a custom property for all selected members:
 1. Select the check box for that property. (Otherwise, the property is not changed.)
 2. From the drop-down box in the **Value** column, select a value.

CAUTION:

Each property must have a value. If you select the check box for a property and leave its **Value** column blank, the custom property is removed from these members.

- To add a custom property to all selected members, click **Add**. You must select a value for the property, and that value applies to all selected members.
- To remove a custom property from all selected members:
 1. Select (highlight) the custom property. You do not need to select the **Update** check box.
 2. Click **Remove**.

Formulas Page

Use the Formulas page of the Member Properties window to manage formulas that are associated with the selected member.

The **Type** field at the top of the page displays the formula type that applies to all of this member's formulas. To change the formula type, click **Edit**. If you change the formula type, make sure that the expression and the scope of each formula are appropriate for the new formula type.

The rest of this page works in the same way as the Formulas page in the New Member wizard.

Identification Page

Overview of the Identification Page

The Identification page includes identification information and a roll-up check box.

Member Identification

The identification information includes the following:

- the member's code
- the member's name and description for each available data locale

To change the name or the description for any listed data locale, click the appropriate cell and enter the new text in it.

Roll-Up Check Box

The check box labeled **This member rolls up into its parent** is selected by default. If you leave this check box selected, then data values for crossings that contain this member participate in the hierarchical roll-up process in any hierarchy that this member belongs to. If you deselect this check box, then data values for crossings that contain this member do not participate in the hierarchical roll-up process in any hierarchy that this member belongs to.

Here is an example. Consider a set of three crossings that are identical in every dimension type except dimension type D. In dimension type D, the three crossings

contain members A, B, and P, respectively. P is the hierarchical parent of A, B, and the virtual child of P. P is not the hierarchical parent of any other member. The A crossing contains the value 120. The B crossing contains the value 330. The virtual-child-of-P crossing contains the value zero. If this check box is selected for both A and B, then the value at the P crossing is 450. If this check box is selected for A but deselected for B, then the value at the P crossing is 120. If this check box is selected for B but deselected for A, then the value at the P crossing is 330. If this check box is deselected for both A and B, then the value at the P crossing is zero.

Note: This check box has no effect on virtual child members. In the preceding example, if A and B have virtual children, then the virtual child of A rolls up to A and the virtual child of B rolls up to B, whether this check box is selected for A and B.

Accounts that belong to the Statistical account type do not have this check box because their values never roll up.

Security Page

Overview of the Security Page

Use the Security page of the member Properties window to manage Read and Write access to the selected member.

By default, every user has both Read and Write access to every member.

Significance of Read Access

If a user is denied Read access to a given member, the following crossings are displayed as red cells without numeric values in all SAS Financial Management tables:

- all crossings that contain the member.
- all crossings that contain any hierarchical descendant of the member, unless you override this setting at a lower hierarchical level.

Note: A virtual child (VC) member always inherits its parent's security settings.

Significance of Write Access

If a user is denied Write access to a given member, the following crossings are displayed as protected cells in data-entry tables:

- all crossings that contain the member
- all crossings that contain any hierarchical descendant of the member, unless you override this setting at a lower hierarchical level.

Note: A virtual child (VC) member always inherits its parent's security settings.

The crossings are protected against direct data entry, including such actions as entering a value, Spread, or Adjust Value. However, they are not protected against indirect changes such as roll-ups or allocations.

As long as Read access is permitted, users can still view the cell values and contributing data records.

Setting Read and Write Permissions

In the Properties window for a member, you can set access permissions for an individual user or a group that the user belongs to. Follow these steps:

1. Click **Add** to add one or more users and groups.
2. On the Security page, select one or more identities.

To grant Read permission, select the **Read** check box. To deny Read permission, clear the check box.

To grant Write permission for data entry, select both the **Read data** and **Write data in a form** check boxes. To deny Write permission, clear the **Write data in a form** check box.

To restore default Read and Write access to a user or a group, click the name of the user or group in the display region, and then click **Remove**.

Note: The **Write data in a form** permission is honored only for data entry in a form, not for posting adjustments, ETL jobs, or other operations.

Setting Permissions for Multiple Members

To set permissions for more than one member:

1. From the list of dimension or hierarchy members, select the members whose security properties you want to set.
2. Right-click and select **Properties**.
3. On the Security page, set the Read and Write properties as explained above.

If a Read or Write property does not have the same value for all selected members, its check box is grayed out or highlighted. If you modify the value, it applies to all selected members. If you do not change this check box value, the members keep their original values.

Overriding Member-Level Security

You can override member-level security as follows:

- **Override group permissions.** This override applies to permissions that are directly set on a member.

If the permission is assigned to a group, you can explicitly grant or deny Read or Write permission to the user. Permission that is assigned to a user overrides permission that is assigned to a group.

If the user belongs to multiple groups, the permissions for the closest group prevail. If the groups are equally close to the user, a grant takes precedence.

- **Override inherited permissions.** If there are no directly assigned permissions, the software examines permissions that are inherited from a parent member.

To override inherited permissions, set permissions at a level that is closer to the member.

Moving down a branch of a hierarchy, you can alternate as many times as you want between denying and granting access to a user.

Time Details Page

The Time Details page shows the period type, start date, and end date of the selected time period.

Users Page

Overview of the Users Page

Use the Users page to manage the set of users who can automatically become authors or reviewers when this member is selected for a form set's workflow.

Adding Users

To add one or more users to the list, click **Add**.

The Add Users window appears.

Removing Users

To remove a user from the list, click the user to select it, and then click **Remove**.

How Authors and Reviewers Are Assigned to Forms

When a form set is created, each form is associated with a member of the target hierarchy. The default authors for a form are users who are associated with the member and who have one of the required roles. The default reviewers for a bottom-up form are the users who are associated with the member for the parent form in the workflow, if those users have one of the required roles.

You can change these automatic author and reviewer assignments by working on the form set or the individual forms in the Forms workspace.

Chapter 5

Required Dimension Types

Introduction to Required Dimension Types	40
Introduction to the Account Dimension Type	40
Introduction to Account Types	41
Asset Account Type	41
Cumulative Translation Adjustment Account Type	42
Equity Account Type	42
Expense Account Type	42
Liability Account Type	42
Retained Earnings Account Type	43
Revenue Account Type	43
Statistical Account Type	43
Statistical Balance Account Type	43
Statistical Flow Account Type	44
Balance Type Property of Accounts	44
Exchange Rate Type Property of Accounts	44
Intercompany Property of Accounts	44
Introduction to the Organization Dimension Type	45
Organization Details	45
Introduction to the Analysis Dimension Type	46
Introduction to the Currency Dimension Type	46
Introduction to the Frequency Dimension Type	46
Introduction to the Source Dimension Type	47
Introduction to the Time Dimension Type	48
Introduction to the Trader Dimension Type	48

Introduction to Required Dimension Types

Each data value that is stored in a cycle must belong to a crossing that includes a member from each of the following required dimension types:

- Account
- Analysis
- Currency
- Organization
- Source
- Time
- Trader

In addition to the seven required dimension types for a cycle, crossings that identify stored data values can also include members from other dimension types. However, all crossings in a given cycle must use exactly the same set of dimension types. For example, you can define one cycle that uses only the seven required dimension types, and another cycle that uses eight dimension types. However, you cannot mix seven-dimensional data records and eight-dimensional data records in the same cycle.

Note: The Frequency dimension type is required. You do not explicitly define the Frequency dimension when you create a cycle. However, you might include it in a table (for example, to display Year To Date values).

Introduction to the Account Dimension Type

The Account dimension type can contain multiple dimensions. Each dimension can contain multiple account hierarchies.

In general, accounts have the following properties that are specific to members of the account dimension type:

- account type
- balance type
- exchange rate type
- intercompany property

However, the following are some exceptions:

- Accounts that belong to the Retained Earnings account type inherit the exchange rate types of the accounts they reference.
- Accounts that belong to the Retained Earnings account type or the Cumulative Translation Adjustment account type have additional properties that are specific to those account types.

Typically, the members of an account hierarchy represent general ledger accounts and categories of general ledger accounts. A typical account hierarchy has several top-level

members that represent the major categories of accounts, such as Assets, Liabilities, Revenue, and Expenses. Leaf members of an account hierarchy represent the accounts for which data is loaded from the SAS Financial Management staging area or entered through forms.

An account hierarchy can also include leaf accounts and account categories for nonmonetary data, such as number of employees or unit sales of various products. These accounts are treated differently based on their account type, which should be set to either Statistical Balance or Statistical Flow.

Another special account type is Statistical, which is typically used for formulas whose input values are the results of the hierarchical roll-up process. Gross Profit is an example of a calculated member that uses the Statistical account type.

Introduction to Account Types

One of the properties of an account is the account type that it belongs to. The account type of a given crossing's account can affect how that crossing's value is computed and how that crossing's value is used in other computations. Account types fall into three categories: balance, flow, and special.

The following are the balance account types:

- Asset
- Liability
- Equity
- Statistical Balance

The following are the flow account types:

- Revenue
- Expense
- Statistical Flow

The following are the special account types:

- Cumulative Translation Adjustment
- Retained Earnings
- Statistical

Asset Account Type

The Asset account type is typically assigned to accounts that represent assets on the balance sheet. The Asset account type has a default debit balance type and a default exchange rate type of Period Close. For currency conversion purposes, the asset account type aggregates facts over a period of time and then converts them based on the exchange rate for the given periods.

For an Asset account, users are typically interested in the value for a To Date frequency member, and not the Period Activity value that is stored.

The default exchange rate type for an Asset account is Period Close.

Cumulative Translation Adjustment Account Type

The Cumulative Translation Adjustment (CTA) account type is assigned to accounts that represent the cumulative translation adjustment on the balance sheet. The CTA account type has default credit balance type and an exchange rate type of Period Close.

The CTA account type automatically calculates exchange rate differences of the accounts referenced.

Accounts of this type can reference roll-ups as well as driver and modeling formulas. For details, see “Working with CTA Accounts” in the *SAS Financial Management: Process Administrator’s Guide*.

Equity Account Type

The Equity account type is typically assigned to accounts representing equity on the balance sheet. The Equity account type has a default credit balance type and a default exchange rate type of Period Close. For currency conversion purposes, the Equity account type aggregates facts over a period of time and then converts them based on the exchange rate for the given period(s).

Expense Account Type

The Expense account type is typically assigned to accounts representing expenses on the income statement. The Expense account type has a default debit balance account type and a default exchange rate type of Period Average. For currency conversion purposes, the Expense account type currency converts facts per period. Then, it aggregates for a given period of time, limiting aggregation to a year (with the exception of the LTD frequency).

Liability Account Type

The Liability account type is typically assigned to accounts representing liabilities on the balance sheet. The Liability account type has a default credit balance and a default exchange rate type of Period Close. For currency conversion purposes, the Liability account type aggregates facts over a period of time and then converts them based on the exchange rate for the given periods.

Retained Earnings Account Type

The Retained Earnings account type is typically assigned to accounts representing retained earnings on the balance sheet. The Retained Earnings account type has a default credit balance type and uses the exchange rate types of the source accounts it references.

This account type has the following properties:

Source Accounts

specifies the members that the Retained Earnings account type should reference. Retained Earnings account types can reference roll-ups, leaf members, and driver and modeling formula types.

Roll-Forward Method

specifies the period of time for which to calculate the retained earnings activity.

Basis Data

specifies the Source dimension members that should be included in the retained earnings calculation.

Revenue Account Type

The Revenue account type is typically assigned to accounts representing revenue on the income statement. The Revenue account type has a default credit balance type and a default exchange rate type of Period Average. For currency conversion purposes, the Revenue account type currency converts facts per period. Then, it aggregates for a given period of time, limiting aggregation to a year (with the exception of the LTD frequency).

Statistical Account Type

The Statistical Account type is designed to represent values such as price and ratios. It does not participate in hierarchical roll-ups, time aggregation, or frequency aggregation. This account type has a default debit balance type and a default exchange rate type of None.

A statistical account cannot be a child or parent to other account type members. They must appear at the top level of an Account hierarchy.

Statistical Balance Account Type

The Statistical Balance account type is typically assigned to non-monetary accounts such as headcount and number of units. The Statistical Balance account type has a default

debit balance type and a default exchange rate type of None. By default, the Statistical Balance account types does not participate in currency conversion.

Statistical Flow Account Type

The Statistical Flow account type is typically assigned to non-monetary accounts such as headcount and number of units. The Statistical Flow account type has a default debit balance type and a default exchange rate type of None. By default, the Statistical Flow account types does not participate in currency conversion.

Balance Type Property of Accounts

The balance type of an account is one of the following:

Debit

indicates that the account normally has a debit balance.

Credit

indicates that the account normally has a credit balance.

For each account type, there is a default balance type. For example, Asset accounts normally have a debit balance while Liability and Equity accounts normally have a credit balance.

Exchange Rate Type Property of Accounts

The exchange rate type of an account helps determine which exchange rates are used for currency conversions of the account balance.

When you create a model, you must associate each analysis member in the model with an exchange rate set. The available exchange rate sets are listed in the Exchange Rate Sets view of the Rates workspace. Within an exchange rate set, there are exchange rates for each exchange rate type. The exchange rates that belong to a given exchange rate type are used for all the accounts that have that exchange rate type.

Intercompany Property of Accounts

The intercompany property of an account determines whether the account should be examined by intercompany balancing rules and participate in the elimination process. The elimination process depends on the Source and Trader hierarchies, which are required in all cycles.

For each account, the intercompany property has one of the following values:

Yes

indicates that the account concerns a financial relationship between the organization that maintains it and another organization in the organization hierarchy. Therefore, it should be examined by intercompany balancing rules.

No

indicates that the account does not concern a financial relationship between the organization that maintains it and another organization in the organization hierarchy. Therefore, it should not be examined by intercompany balancing rules.

The intercompany property plays a key role in the following administrative reports that you can run in the Web-based Document Manager:

- Eliminations
- ICAccounts
- Intercompany
- Non Intercompany

These reports help you verify that all the accounts that are marked with the Intercompany attribute receive the appropriate processing.

Introduction to the Organization Dimension Type

The Organization dimension type (code INTORG) can contain any number of dimensions. Each dimension can contain any number of organization hierarchies.

An organization hierarchy typically represents the organizational reporting structure of an enterprise.

Organization Details

The following properties of an organization are listed on the Organization Details page of the organization member's Properties window:

Reporting Entity

determines whether an organization member can be selected for use in manual adjustment or ownership rule.

In organization hierarchy displays, the icon for a reporting entity includes two black horizontal bars . The icon for an organization that is not a reporting entity does not have these bars .

Functional Currency

the currency that is assigned to the organization member. The functional currency is used as follows:

- Facts that are entered into a form or form set template are stored in the functional currency that is assigned to the organization member for the crossing.

- Facts that are entered for manual adjustments and rules, except for allocation rules, are stored in the functional currency that is assigned to the organization member for the crossing.

All facts that are loaded via ETL or from another model are stored in the assigned currency, which might differ from the functional currency.

Introduction to the Analysis Dimension Type

The Analysis dimension type can contain multiple dimensions. Each dimension can contain multiple flat hierarchies of analysis members.

The members of the Analysis dimension type typically identify the different sets of data that are used for comparison purposes. Examples include Actual, Forecast, and Budget.

Introduction to the Currency Dimension Type

The Currency dimension type can contain any number of dimensions. Each dimension can contain any number of flat hierarchies of currency members.

In addition,

- If you use more than one currency member, then you need to maintain exchange rates by using the Rates workspace.
- Currency members cannot carry formulas.
- In general, currency members are predefined and have three-character codes, such as EUR, JPY, and USD. The only exception is the NONE currency, which has a four-character code and is not predefined. To use NONE, you must add it to a currency hierarchy in the Dimensions workspace.
- The NONE currency has special behavior: no currency conversion takes place between a predefined, three-character currency and NONE.

Introduction to the Frequency Dimension Type

The Frequency dimension type contains a single Frequency dimension with a single predefined, flat Frequency hierarchy.

Each member of the Frequency dimension is characterized by the following two attributes:

- a length of time, such as Month, Quarter, or Year, or the generic Period
- a mode, which can be either To Date or Activity

In addition, the following usage notes apply to the Frequency dimension type:

- There is a Frequency member for each combination of a length of time and a mode. For example, some commonly used Frequency members are Month Activity, Month To Date, Period Activity, and Period To Date.
- To see the complete list of Frequency members, select the Frequency dimension in the Dimensions view, and then select **Hierarchies**. Although the Frequency hierarchy is flat, you can reorder its members.
- All facts are stored as Periodic Activity, regardless of the frequency that is used when they are entered or loaded.
- In a form, facts can be entered using the PA, PTD, or YTD frequency.
- Facts can be loaded using the PA or YTD frequency.
- For all rules, facts are queried using the PTD frequency.

Introduction to the Source Dimension Type

The Source dimension type contains a single Source dimension. The Source dimension contains a single predefined Source hierarchy. You cannot add other dimensions or hierarchies within the Source dimension type, and you cannot change the structure of the Source hierarchy.

In addition, you cannot assign calculated members to members of the Source dimension and each member of the Source dimension represents the source from which the data was derived.

The Source dimension members are as follows:

- **Total**—sum of all Source members
- **CTA**—computed value of an account that belongs to the Cumulative Translation Adjustment (CTA) account type
- **RE**—computed value of an account that belongs to the Retained Earnings (RE) account type. This value is the sum of the subordinate REElim, REAdj, and REData values
 - **REElim**—portion of the value of a Retained Earnings account that comes from values that are entered in the Elim Source dimension member
 - **REAdj**—portion of the value of a Retained Earnings account that comes from rules and manual adjustments
 - **REData**—sum of the subordinate REForms and REImport values
 - **REForm**—portion of the value of a Retained Earnings account that comes from values that are entered in the BaseForm Source member

Note: The numeric values for BaseForm crossings depend on choices that you make when you load data into the cycle.

- **REImport**—the portion of the value of a Retained Earnings account that comes from values entered in the Base or BaseJourn Source dimension member

Note: The numeric values at Base and BaseJourn crossings depend on choices that you make when you load data into the cycle.

- **Elim**—elimination values that are generated by the process of eliminating intercompany balances
- **TotalBeforeElim**—subtotal that combines all base data, all manual adjustments, and rules
 - **Adj**—subtotal of all manual adjustments and rules
 - **CPO**—adjustments that are generated by ownership rules
 - **Alloc**—adjustments that are generated by allocation rules
 - **Reclass**—adjustments that are generated by reclassification rules
 - **Bal**—adjustments that are generated by balancing rules of all types
 - **Manual**—manual adjustments
- **TotalBeforeAdj**—subtotal of BaseForm and TotalAfterImport values
 - **BaseForm**—values entered via Data Entry in Microsoft Excel and loaded from models and the SAS Financial Management staging area
 - **TotalAfterImport**—subtotal of the subordinate BaseJourn and Base values
 - **BaseJourn**—represents the values loaded from the GL_JRNL_DETAILS table in the SAS Financial Management staging area
 - **Base**—values loaded from the GL_TRANSACTION_SUM table in the SAS Financial Management staging area

Introduction to the Time Dimension Type

The Time dimension type can contain any number of dimensions. Each dimension can contain any number of hierarchies.

A *time hierarchy* defines the time periods for which the data is collected. Data can be loaded or entered only for leaf time periods. There are no virtual children in the Time dimension.

Calculated members are not supported in the Time dimension.

Introduction to the Trader Dimension Type

In a cycle, the Trader dimension type mirrors the Organization dimension type. Each trader hierarchy is an automatically generated copy of a corresponding organization hierarchy.

Each Trader dimension hierarchy supports the following two additional members in the hierarchy:

- **EXT**—typically used for external trade values

Note: This is the default write member assigned in the Trader dimension.
- **ALL**—sum of all Trader dimension members, including EXT

In a two-organization record, the trader member always identifies the second organization.

Note: The Trader dimension is not displayed in the Dimension workspace like the other dimensions. Information about the Trader dimension is located in the properties for the Organization dimension.

Chapter 6

Working with CTA Accounts

Creating a CTA Account: Best Practices	51
Overview	51
Source Accounts	51
CTA Calculation	52
Other Considerations for CTA Accounts	53
Balanced Balance Sheet	53
Writing Data to CTA Accounts	53
Multiple CTA Accounts on Balance Sheet	53
Queries and the Source Dimension	55

Creating a CTA Account: Best Practices

Overview

A Cumulative Translation Adjustment (CTA) account references values from source accounts and renders the translation differences between the source accounts' exchange rates and the CTA account's Period Close exchange rate.

SAS Financial Management assumes that the CTA account is translated at a Period Close rate. The source accounts should include any account types on the balance sheet that are not assigned a Period Close exchange rate type.

When you create a CTA account, SAS recommends the following selections:

- **Balance Type:** Credit
- **Intercompany:** No

Source Accounts

When you create a CTA account, you select the source accounts for translation. The list of accounts that are available for selection is automatically filtered to display only Asset, Liability, Equity, and Retained Earnings account types. Only these account types are displayed because CTA is intended to balance the balance sheet accounts.

To create a CTA account, you must select at least one source account. Therefore, before you create a CTA account, you should create other accounts in the balance sheet. If

necessary, you can modify the selection of source accounts later in the Properties window for the account.

To ensure that no accounts are omitted by mistake, SAS recommends that you select high-level roll-up members in the balance sheet. This example selects three accounts: Total Assets, Total Liabilities, and Total Shareholder's Equity:

CTA Calculation

The following is an example of how a simple CTA calculation works in SAS Financial Management.

Figure 6.1 Simple CTA Calculation

1	2	3	4	5	6	7 = (5-6) × 2
	US Dollars	Euros	Exchange Rate Type	Exchange Rate	CTA Peg Rate	CTA
Cash & Marketable Securities	1,500,000	1,143,503	Period Close	0.7623	0.7623	-
Total Assets	1,500,000	1,143,503	Period Close	0.7623		
Accounts Payable	500,000	381,168	Period Close	0.7623	0.7623	-
Total Liabilities	500,000	381,168	Period Close	0.7623		
Common Stock	400,000	302,000	Historic	0.7550	0.7623	2,934
Retained Earnings - Prior Years	-	-	Period Close	0.7623	0.7623	-
Current Year's Net Income	600,000	455,040	None	(from P&L accounts)		
Retained Earnings - Total	1,000,000	757,040	Period Close	0.7623		
Cumulative Translation Adjustment	-	5,295	Period Close	0.7623		
Total Shareholder's Equity	1,000,000	762,335	Period Close	0.7623		
Total Liabilities & Shareholder's Equity	1,500,000	1,143,503				
Revenue	1,000,000	758,400	Period Avera	0.7584	0.7623	3,935
Total Operating Expenses	400,000	303,360	Period Avera	0.7584	0.7623	(1,574)
Net Income	600,000	455,040	Period Avera	0.7584		
TOTAL CTA						5,295

Note: In the display, the signs of credit accounts were changed from negative to positive to provide a consistent display. For that reason, the CTA item related to Total General & Admin Expenses has its sign reversed in this example. (In SAS Financial Management, you do not need to be concerned with signs. This note is provided in case you attempt to recalculate the values in this example.)

The CTA calculation is automatic and it is not visible to the user. However, the example above shows the data elements used in the CTA calculation to provide an understanding of how the calculation works. The following table explains the columns in the report:

Column	Contents
Column 1	Account descriptions.
Column 2	The balances for an entity with a functional currency of US Dollars. The balance sheet balances in the functional currency and there is no CTA.
Column 3	The US Dollars balances of column 2, translated to euros (EUR). Note that there is a CTA amount and the balance sheet balances in euros.

Column	Contents
Column 4	A CDAProperty query of each account's exchange rate type. The exchange rate type is used in column 5.
Column 5	Using the exchange rate type of each account (from column 4), a CDAXRate query is used to look up the exchange rate for each account.
Column 6	The exchange rate that the CTA account uses as the peg rate. The peg rate is considered to be the Period Close rate that is assigned for the period.
Column 7	Recalculates CTA using the elements surfaced on the report, to prove the value calculated by SAS Financial Management. It uses columns 2, 5, and 6 in a formula: $(5 - 6) \times 2 = \text{CTA Value}$

Other Considerations for CTA Accounts

Balanced Balance Sheet

The balance sheet of a reporting entity should be in balance in its functional currency. If the balance sheet is out of balance, then the inputs to the CTA calculation are not correct and the resulting CTA calculation value is incorrect as well. In general, we recommend that each reporting entity's trial balance be validated to ensure that it is in balance before it is loaded to SAS Financial Management.

Writing Data to CTA Accounts

Any data that is loaded to a CTA account is ignored at query time. The CTA account returns the value as determined by the source accounts instead.

Multiple CTA Accounts on Balance Sheet

SAS recommends that the balance sheet contain a single CTA account. If the balance sheet contains more than one CTA account, based on the source account selection, SAS Financial Management logic ensures that the CTA calculation is not duplicated. Therefore, the balance sheet remains in balance. This logic is demonstrated in the following examples.

Example 1

In the first example, there is only one CTA account designed to balance the balance sheet. Assume that only the following three facts exist:

- Sales : -10.00 USD
- Cash and Cash Equivalents: 5.00 USD
- Investment in Subs: 5.00 USD

The exchange rates for Jan 2011 are as follows:

- Period Average EUR / USD: 0.7584
- Period Close EUR / USD: 0.7623
- Historical Rate, Investment in Subs EUR / USD: 0.7600

The source account for the CTA account is as follows:

Figure 6.2 CTA Account Details

Account Details

Account type: Cumulative Translation Adju

Exchange rate type: Period Close

Balance type: ☐ Debit ☒ Credit

☐ Intercompany

Source accounts:

Code /	Description	Formula Type	Account Type
Balance Sheet	Balance Sheet		Asset

When converted to EUR, the balance sheet remains in balance due to the CTA amount of -0.03 euros as shown below:

Figure 6.3 Single CTA Account

	Frequency		PTD	
	Time		Jan 2011	
	USD		EUR	
Sales	(10.00)		(7.58)	
Balance Sheet	0.00		0.00	
Assets	10.00		7.61	
Cash & Cash Equivalents	5.00		3.81	
Investment in Subs	5.00		3.80	
Liabilities	0.00		0.00	
Stockholder's Equity	(10.00)		(7.61)	
Retained Earnings	(10.00)		(7.58)	
Cumulative Translation Adj	0.00		(0.03)	

Example 2

In the second example, an additional CTA account is added to capture the CTA specifically related to the Investment in Subs account. As in the first example, assume the same facts and exchange rates. For the CTA – Investment in Subs account, the source account is assigned as follows:

Figure 6.4 CTA Account Details

Account Details			
Account type: Cumulative Translation Adju			
Exchange rate type: Period Close			
Balance type: ☐ Debit ☒ Credit			
☐ Intercompany			
Source accounts:			
Code /	Description	Formula Type	Account Ty...
Investme...	Investment in Subs		Asset

When converted to EUR, the balance sheet remains in balance because of the two calculated CTA amounts.

Note: The CTA account that is designed to balance the balance sheet is now -0.04 euros and the CTA- Investment in Subs account is 0.01 euro.

Figure 6.5 Multiple CTA Accounts

Frequency	PTD	
Time	Jan 2011	
	USD	EUR
Sales	(10.00)	(7.58)
Balance Sheet	0.00	0.00
Assets	10.00	7.62
Cash & Cash Equivalents	5.00	3.81
Investment in Subs	5.00	3.80
CTA - Inv in Sub	0.00	0.01
Liabilities	0.00	0.00
Stockholder's Equity	(10.00)	(7.62)
Retained Earnings	(10.00)	(7.58)
Cumulative Translation Adj	0.00	(0.04)

Queries and the Source Dimension

When the CTA account type generates a CTA value, that value is written to a member of the Source dimension called CTA.

▲ Total	Grand total
CTA	Cumulative Translation Adjustments
▲ RE	Retained earnings resulting from base data and manual and rule-based adjustments
REElim	Retained earnings resulting from elimination adjustments
REAdj	Retained earnings resulting from manual and rule-based adjustments
▲ REData	Retained earnings resulting from base data
REForm	Retained earnings resulting from form-entered data
REImport	Retained earnings resulting from imported data
Elim	Elimination adjustments
▲ TotalBeforeElim	Subtotal of base data and manual and rule-based adjustments
Adj	Subtotal of all manual and rule-based adjustments
CPO	Ownership rule adjustments
Alloc	Allocation rule adjustments
Reclass	Reclassification rule adjustments
Bal	Balancing rule adjustments
Manual	Manual adjustments
▲ TotalBeforeAdj	Subtotal of imported and form-entered data
BaseForm	Data entered via forms
▲ TotalAfterImport	Subtotal of imported data
BaseJourn	Data based on imported journal entries

If you import source system data to the Source member Base and then create a balance sheet that displays only Base data, the values that are generated by the CTA accounts are not included. The reason that the values are not included is because they appear at a higher point in the Source dimension hierarchy.

Chapter 7

Working with Retained Earnings Accounts

Retained Earnings: A Hybrid Account	57
Creating a Retained Earnings Account	58
Overview	58
Source Accounts	58
Roll-Forward Method	58
Basis Data	61
Retained Earnings Accounts and Currency Conversion	61
Loading Data to a Retained Earnings Account	62
Sample Configurations	62
Single Retained Earnings Account	62
Current and Prior Years - Retained Earnings Accounts	63
Varying Basis Data	63
Other Considerations	64
Querying the Source Dimension	64
Calculated Members as Source Accounts	65
Rolling Forward Balance Sheet Accounts	65
Overview and Objectives	65
Sample Balance Sheet Account Roll-Forward	66

Retained Earnings: A Hybrid Account

Most accounts in SAS Financial Management are either balance accounts or flow accounts.

Balance accounts (such as Cash) represent a point-in-time balance. The periodic values in these accounts are aggregated across time and are then converted to another currency by using a single exchange rate (usually the period close rate). The balance account data is stored as period activity. From the very first period of time to the current reporting period, each period must be aggregated to provide the proper balance account value for any “to date” frequency member.

Flow accounts (such as Revenue) represent a period of time. They are currency converted on a per-period basis and then aggregated over time. Unlike a balance account, a flow account aggregates values based on the period of time (time and frequency) that is being queried.

The *Retained Earnings account* type has characteristics of both balance accounts and flow accounts. A retained earnings account represents the retained earnings balance at a point in time, like a balance account. However, a retained earnings account derives its values from flow accounts (such as Net Income) that are based on periods of time and varying exchange rates. The retained earnings account type is designed to incorporate this dual nature into a single account type.

The advantages of this design are as follows:

- It eliminates the need to track complex weighted average exchange rates from period to period.
- It automatically rolls balances to the proper account, from period to period and across fiscal years.
- It automatically determines the time periods to reference to retrieve the correct values.
- Accounts that are referenced do not need to exist in the same branch of the hierarchy.

Creating a Retained Earnings Account

Overview

A retained earnings account references the values from source accounts (typically, income statement accounts) and renders that result on the balance sheet. When you create a retained earnings account, you provide information about source accounts, the roll-forward method, and basis data.

The following selections are recommended:

- **Balance Type:** Credit

This setting does not affect retained earnings accounts.

- **Intercompany:** No

Note: You are not prompted to enter an exchange rate type. The retained earnings account inherits the exchange rate types that are assigned to its source accounts. In the Dimensions workspace of SAS Financial Management Studio, the exchange rate type of a retained earnings account is displayed as “None.” The same value is returned by a CDAProperty query.

Source Accounts

Select the accounts that contribute to the retained earnings value. We recommend that you select a single parent member such as Net Income to include all accounts that affect retained earnings.

Note: For multiple hierarchies with varying members, ensure that you select the proper accounts.

Roll-Forward Method

In the roll-forward method selection, you choose the time span over which the retained earnings account accumulates data. The roll-forward method selection enables you to

segment the results of the retained earnings into current and prior year values. Each retained earnings account can have only one roll-forward method, although you can create multiple retained earnings accounts with different roll-forward methods.

The following roll-forward methods are available:

- You can choose a method that conforms to how retained earnings from the source system are rolled forward.
- You can group retained earnings by time (for example, into Current and Previous Years).
- You can tell SAS Financial Management to automatically calculate the value of retained earnings.

Here is a numeric example of the six roll-forward methods and their resulting values. In the following example, Net Income is the source account for all six retained earnings accounts. The period activity facts for the Net Income account are shown below:

Frequency	Current Period to Date			
	Dec 2011	Jan 2012	Feb 2012	Mar 2012
Net Income	100.00	75.00	55.00	60.00

The resulting values for each of the six roll-forward methods, using a period-to-date frequency are as follows:

Frequency	Current Period to Date			
	Dec 2011	Jan 2012	Feb 2012	Mar 2012
Net Income	100.00	75.00	55.00	60.00
RE 1 BOY - ECP	100.00	75.00	130.00	190.00
RE 2 BOY - BCP	0.00	0.00	75.00	130.00
RE 3 BCP - ECP	100.00	75.00	55.00	60.00
RE 4 BOT - ECP	100.00	175.00	230.00	290.00
RE 5 BOT - BCP	0.00	100.00	175.00	230.00
RE 6 BOT - BOY	0.00	100.00	100.00	100.00

In the example, December 2011 is the first period in the model and therefore it is the beginning of time. The fiscal year in the example is a calendar year (January through December). Below is a brief description of each method, with references to the rows of the example above.

Example Row	Method	Description
RE 1 BOY – ECP	From beginning of year through end of current period	This method aggregates the source account values from the beginning of the year through the end of the current query period. In the example, for Feb 2012, it aggregates the values from Net Income for Jan 2012 and Feb 2012, resulting in a value of $75.00 + 55.00 = 130.00$. This method is recommended for rolling forward retained earnings within the current year.

Example Row	Method	Description
RE 2 BOY – BCP	From beginning of year to beginning of current period	This method aggregates the source account values from the beginning of the year to the beginning of the current query period. For Feb 2012, it aggregates the value from Net Income for Jan 2012, resulting in a value of 75.00. This method is recommended for the Current Year's Retained Earnings.
RE 3 BCP – ECP	From beginning of current period through end of current period	This method aggregates the source account values from the beginning of the current period through the end of the current query period. Essentially it is the value for the query period. For Feb 2012, it aggregates the value from Net Income for Feb 2012, resulting in a value of 55.00. This method is recommended for rolling forward retained earnings by period.
RE 4 BOT – ECP	From beginning of time through end of current period	This method aggregates the source account values from the beginning of time through the end of the current query period. For Feb 2012, it aggregates the values from Net Income for Dec2011, Jan2012, and Feb 2012, resulting in a value of $100.00 + 75.00 + 55.00 = 230.00$. This method is recommended for rolling forward retained earnings for all time periods in a single retained earnings account.
RE 5 BOT – ECP	From beginning of time to beginning of current period	This method aggregates the source account values from the beginning of time through the beginning of the current query period. For Feb 2012, it aggregates the values from Net Income for Dec 2011 and Jan 2012, resulting in a value of $100.00 + 75.00 = 175.00$. As with RE 4, there is no significant time segregation.
RE 6 BOT – BOY	From beginning of time to beginning of year	This method aggregates the source account values from the beginning of time to the beginning of the year being queried. For Feb 2012, it aggregates the value from Net Income for Dec2011, resulting in a total of 100.00. This method is recommended for aggregating all retained earnings balances prior to the current fiscal year.

The various roll-forward methods give you the flexibility to account for all time periods by means of four options. The following combinations provide the proper retained earnings amount in total:

1. Roll-forward method 4: **From beginning of time through end of current period**
2. Roll-forward methods 6 and 1: **From beginning of time to beginning of year and From beginning of year through end of current period**

3. Roll-forward methods 5 and 3: **From beginning of time to beginning of current period** and **From beginning of current period through end of current period**
4. Roll-forward methods 6, 2 and 3: **From beginning of time to beginning of year**, **From beginning of year to beginning of current period**, and **From beginning of current period through end of current period**

Basis Data

The last selection to be made when you are creating a retained earnings account is the basis data. Like the roll-forward method, which segregates data by Time, **Basis Data** segregates data by Source dimension member. There are four possible selections for basis data. Each selection corresponds to a specific Source dimension member.

- **TotalAfterImport**—This data corresponds to the Source members Base and BaseJourn. Generally, this data is loaded from a source system like a general ledger or an ERP system. Because Base and BaseJourn together represent Imported Data, the manner in which the external system handles retained earnings should be the same for these two Source members.
- **BaseForm**—This data corresponds to the Source member BaseForm. Generally, this data is entered via Forms, but it might also be loaded.
- **Adj**—This data corresponds to the Source member Adj, which is the parent member for all manual and rule-based adjustments. Typically, this data is created solely within SAS Financial Management Studio.
- **Elim**—This data corresponds to the Source member Elimination Adjustments. This data is created within SAS Financial Management based on inter-company balances.

You can view these four selections for **Basis Data** in the hierarchy below:

▲ Total - CTA ▲ RE - REElim - REAdj ▲ REData - REForm - REImport - Elim ▲ TotalBeforeElim - Adj - CPO - Alloc - Reclass - Bal - Manual - TotalBeforeAdj - BaseForm ▲ TotalAfterImport - BaseJourn - Base	- Grand total - Cumulative Translation Adjustments - Retained earnings resulting from base data and manual and rule-based adjustments - Retained earnings resulting from elimination adjustments - Retained earnings resulting from manual and rule-based adjustments - Retained earnings resulting from base data - Retained earnings resulting from form-entered data - Retained earnings resulting from imported data - Elimination adjustments - Subtotal of base data and manual and rule-based adjustments - Subtotal of all manual and rule-based adjustments - Ownership rule adjustments - Allocation rule adjustments - Reclassification rule adjustments - Balancing rule adjustments - Manual adjustments - Subtotal of imported and form-entered data - Data entered via forms - Subtotal of imported data - Data based on imported journal entries - Data based on imported ledger entries
--	--

These four selections account for all Source dimension members except Retained Earnings and CTA.

Retained Earnings Accounts and Currency Conversion

In multi-currency environments, the use of retained earnings accounts greatly simplifies the currency conversion process. The retained earnings account inherits the exchange

rate type assigned to the source accounts. Consequently, the retained earnings account automatically retrieves the proper value and exchange rate for each period.

Loading Data to a Retained Earnings Account

Any data that is loaded to a retained earnings account is ignored at query time. The retained earnings account instead returns the value as determined by the source accounts, roll-forward method, and the basis data.

Sample Configurations

Single Retained Earnings Account

The next example shows a section of the balance sheet accounts for a simple retained earnings roll-forward configuration. It uses one retained earnings account with the assumption that there is no reporting need to segregate the retained earnings balance between current periods and year and prior periods and years.

Figure 7.1 Single Retained Earnings Account

Code	Name	Description	Account Type	Balance Type	Exchange Rate Type
3000	RE	Retained Earnings	Equity	Credit	Period Close
3005	Initial RE	Initial Retained Earnings Balance	Equity	Credit	Derived
3010	Initial RE	Retained Earnings - Initial Balance	Retained Earnings	Credit	None
3015	Adj_RE	Adjustments to Retained Earnings	Equity	Credit	Historic

The purpose of each account is as follows:

- Account 3000**—This account is the roll-up member in the hierarchy for presentation on a balance sheet.
 It has an account type of Equity.
 It uses a period close rate for any entries made to the virtual child.
- Account 3005**—This account stores the opening balance of retained earnings. It has an account type of Equity.
 It uses a derived exchange rate.
- Account 3010**—This account is the single retained earnings account.
 It uses Net Income as the source account.
 It uses the roll-forward method that covers all of time: **From beginning of time to end of current period.**
- Account 3015**—This account captures any adjustments to retained earnings.
 It has an account type of Equity.
 It uses a historic exchange rate.

Current and Prior Years - Retained Earnings Accounts

The following example shows a retained earnings roll-forward configuration that uses two retained earnings accounts to segregate retained earnings between current and prior years. In this example, SAS Financial Management rolls forward all retained earnings balances related to net income, with the exception of the initial balance.

Figure 7.2 Two Retained Earnings Accounts

Code	Name	Description	Account Type	Balance Type	Exchange Rate Type
RETOT	RETOT	Retained Earnings - Total	Equity	Credit	Period Close
3200	RE - Pr Yr	Retained Earnings - Prior Years	Equity	Credit	Period Close
3210	Initial RE	Retained Earnings - Initial Balance	Equity	Credit	Derived
3220	RE - Pr Yr Net Income	Prior Year's Net Income	Retained Earnings	Credit	None
3300	RE - Cur Yr Net Income	Current Year's Net Income	Retained Earnings	Credit	None
3015	Adj_RE	Adjustments to Retained Earnings	Equity	Credit	Historic

In this example, there is a single source account (Net Income). The basis data does not vary by retained earnings account. All basis data selections are checked for both retained earnings accounts.

The purpose of each account is as follows:

- Account 3200**—This account is a roll-up member that allows reporting on Prior Years' Retained Earnings from a single account.
 It aggregates the initial balance of retained earnings and the retained earnings from Net Income that is related to prior years
- Account 3210**—This account stores the opening balance of retained earnings.
 It has an account type of Equity.
 It uses a derived exchange rate.
- Account 3220**—This account captures the retained earnings from Net Income related to prior years.
 It uses the roll-forward method **From beginning of time to beginning of year**.
- Account 3300**—This account captures the retained earnings from Net Income related to the current year, using the roll-forward method **From beginning of year through end of current period**.
- Account 3015**—This account captures any adjustments to retained earnings. It has an account type of Equity.
 It uses a historic exchange rate.

Varying Basis Data

Building upon the previous example, in which retained earnings from Net Income is differentiated between current and prior years, the following example further distinguishes retained earnings amounts by basis data selection.

In this example, two separate accounts are used for prior year, based on the assumption that imported data from the source system rolls forward annually, whereas data that is generated within SAS Financial Management does not roll forward.

Figure 7.3 Different Accounts for Prior Years and Current Years

Code	Name	Description	Account Type	Balance Type	Exchange Rate Type
RETOT	RETOT	Retained Earnings - Total	Equity	Credit	Period Close
3500	CY Net Income	Current Year Net Income	Equity	Credit	Period Close
3510	CY Net Income - Imported	Current Year Net Income - Imported Data	Retained Earnings	Credit	None
3520	CY Net Inco...SAS FM Data	Current Year Net Income - SAS Fm Data	Retained Earnings	Credit	None
3600	PY Net Income	Prior Year Net Income	Equity	Credit	Period Close
3610	PY Net Income - Imported	Prior Year Net Income - Imported Data	Equity	Credit	Derived
3620	PY Net Inco...SAS FM Data	Prior Year Net Income - SAS Fm Data	Retained Earnings	Credit	None

The purpose of each prior year account is as follows:

- **Account 3600**—This account is the roll-up member that can be used for simplified reporting.
- **Account 3610**—This account is an equity account with a derived exchange rate.

The balance that is rolled forward annually is captured here.

It uses a derived exchange rate calculated externally and entered into SAS Financial Management Studio.

- **Account 3620**—This account is the retained earnings account type for prior years.

The roll-forward method is **From beginning of time to beginning of year**.

The basis data includes everything except imported data, which is included in account 3610.

This account captures all SAS Financial Management data from forms, adjustments, and eliminations. This data is not in the source system. Therefore, it is not included in the roll-forward balance supplied by the source system.

The purpose of each current year account is as follows:

- **Account 3500**—This account is the roll-up member that can be used for simplified reporting.
- **Accounts 3510 and 3520**—These accounts are configured almost identically. The only difference is that account 3510 uses the basis data selection for Imported Data, and account 3520 includes the remaining three basis data selections.

Because basis data is the only distinction, accounts 3510 and 3520 could be combined into a single account. They are displayed separately in order to mirror the configuration for prior year retained earnings.

Other Considerations

Querying the Source Dimension

When the retained earnings account type generates a retained earnings value, that value is displayed in one of four child members of the Source dimension, depending on the basis data selection.

RE	Retained earnings resulting from base data and manual and rule-based adjustments
REElim	Retained earnings resulting from elimination adjustments
REAdj	Retained earnings resulting from manual and rule-based adjustments
REData	Retained earnings resulting from base data
REForm	Retained earnings resulting from form-entered data
REImport	Retained earnings resulting from imported data

Calculated Members as Source Accounts

SAS Financial Management allows retained earnings accounts to reference calculated members with a formula type of either Driver or Modeling. Reporting formulas can be referenced as Source accounts. However, these accounts are executed after retained earnings accounts. As a result, the results of reporting formulas are not included in the retained earnings values.

Rolling Forward Balance Sheet Accounts

Overview and Objectives

Despite its name, you can use the retained earnings account type to roll forward balance sheet accounts, because source accounts can refer to both flow account types (for example, Revenue and Expense) and balance account types (for example, Asset, Liability, and Equity).

One objective in providing a balance sheet roll-forward is to display the changes in an account from one reporting period to the next. For example, a single currency environment would contain the following:

```
Opening Balance
+ Additions
- Disposals
+/- Adjustments
= Ending Balance
```

In a multi-currency environment, cumulative translation adjustment (CTA) can be included as well:

```
Opening Balance
+ Additions
- Disposals
+/- Adjustments
+/- CTA
= Ending Balance
```

In this scenario, creating a roll-forward for balance accounts is achieved by means of the following:

- Use of a combination of account types and exchange rate types:
 - Account types used: Retained Earnings, CTA, and Balance (such as Asset).
 - Exchange rate types used: Period Open, Period Average, Period Close.
- Source data loaded to the proper accounts, at the proper level of detail.
 - The balance loaded to Additions should be the activity for the desired roll-forward period, rather than all additions over time, because some of those values already are accounted for in the period's opening balance.
 - The Additions account serves as a source account for the retained earnings account. Because it is only a source account, it should not roll up the hierarchy.

The net result at the roll-up member of all these account types and exchange rate types is a balance sheet account value that is expressed at the period close rate. Additional detail is available below the roll-up member for additional reporting needs.

Sample Balance Sheet Account Roll-Forward

The following example shows a sample roll-forward configuration for a balance sheet account related to a building.

Code	Name	Description	Rolls Up	Account Type	Exchange Rate Type
RETOT	RETOT	Retained Earnings - Total	Yes	Equity	Period Close
Building	Build - End	Building - Ending Balance	Yes	Asset	Period Close
172105	Build - Opening	Building - Opening Balance	Yes	Asset	Period Open
172110	Build - Additions	Building Additions	Yes	Retained Earnings	None
172115	Build - Disposals	Building Disposals	Yes	Retained Earnings	None
172120	Build - Transfers	Building Transfers	Yes	Retained Earnings	None
172130	Build - CTA	Building - Translation Adjustment	Yes	Cumulative Translation Adjustment	Period Close
ETL - Total	ETL - Total	Source Account Data - Total	No	Asset	Period Average
172110_ETL	172110_ETL	172110 Additions - Source	Yes	Asset	Period Average
172115_ETL	172115_ETL	172115 Disposals - Source	Yes	Asset	Period Average
172120_ETL	172120_ETL	172120 Transfers - Source	Yes	Asset	Period Average

The purpose of each account is as follows:

- **Building – Ending Balance**—This account is the roll-up member of the hierarchy.

It reports the ending balance for the account at the period close rate.

Typically, there is no data entered directly to this account. However, an exchange rate type of period close is assigned to this account should any adjustments be made directly to the virtual child.

- **Building – Opening Balance – 172105**—This account is used to load the initial balance.

An exchange rate type of period open is assigned to this account, which also contains the balance at the end of the previous year. In order for the opening balance to translate correctly, the period open exchange rate type should use the same exchange rates as the period close at the end of the previous year.

Because the opening balance is valid for an entire fiscal year (the account is rolled forward annually), the period open exchange rate should be populated with the same values for all periods in the current fiscal year. That is, the period close rate from the last period of the previous fiscal year is loaded to each period of the current fiscal year, but to the period open exchange rate type.

- **Building Additions, Disposals, and Transfers – 172110, 172115, 172120**—These are the retained earnings accounts.

Roll-forward method: Each account uses the roll-forward method **From beginning of year through end of current period**.

Basis data: All four basis data selections are included.

Source accounts: Each account references a single companion source account. For example, the retained earnings account Building Additions has a companion asset account in the hierarchy called 172110 Additions – Source.

- **Source accounts for Additions, Disposals, and Transfers – 172110_ETL, 172115_ETL, 172120_ETL**—These are asset accounts that are loaded with data from the source system.

They are identified with a suffix of **_ETL** to distinguish them from their companion retained earnings accounts.

The data loaded to these accounts is the activity for each account since the last account balance roll-forward. For example, if the account balance for Building were

rolled forward at December 2010, the values loaded to the Additions account in the 2011 periods would include all activity in 2011.

The exchange rate type used for these source accounts is period average. Although the period average rate generally is used for items of income and expense, since additions, disposals, and transfers occur throughout the year each source account's activity is converted at each period's period average rate. This translation provides a more accurate value for the cash flows related to additions, disposals, and transfers.

- **Source Account Data – Total – ETL_TOTAL**—This asset account exists simply as a roll-up member for the source accounts discussed above.

This account has the same account type and exchange rate type as the source accounts (Asset, Period Average).

This account resides within the same hierarchy as the other accounts for convenience; users can refer to the values in the source accounts.

The most important setting for this account is the **Rolls up** property, which is set to **No**. With this setting, all the source account values roll up to this account, but the values do not roll up the hierarchy any further. Setting **Rolls up** to **No** permits the source accounts to reside next to their related accounts without double-counting the values.

- **Building – Translation Adj. -172130**—This account is used to calculate cumulative translation adjustment (CTA) on a balance sheet.

It is used to segregate the exchange rate effects of the roll-forward from the cash flows and to ensure that all the components, when aggregated, result in a balanced balance sheet.

In the following image, the accounts within the shaded box comprise the value at the roll-up member (account code Building). That is:

- 172105 at a period open rate.
- 172110 through 172120 at a period average rate (the rate assigned to their source accounts).
- 172130 calculates CTA on accounts 172105, 172110, 172115 and 172120.

The net result is that when the CTA value in account 172130 is added, the value is the same as if all the balances were translated at a period close rate.

Code	Name	Description	Account Type
Building	Build - End	Building - Ending Balance	Asset
172105	Build - Opening	Building - Opening Balance	Asset
172110	Build - Additions	Building Additions	Retained Earnings
172115	Build - Disposals	Building Disposals	Retained Earnings
172120	Build - Transfers	Building Transfers	Retained Earnings
172130	Build - CTA	Building - Translation Adjustment	Cumulative Translation Adjustment
ETL - Total	ETL - Total	Source Account Data - Total	Asset
172110_ETL	172110_ETL	172110 Additions - Source	Asset
172115_ETL	172115_ETL	172115 Disposals - Source	Asset
172120_ETL	172120_ETL	172120 Transfers - Source	Asset

Chapter 8

Cycles and Cycle Periods

Introduction to Cycles	69
Overview	69
Data in a Cycle	69
Objects That Belong to Cycles	70
Cycle Manager Window	70
Create a New Cycle	71
Copy a Cycle	71
Open a Cycle	72
Data Records in a Cycle	72
Managing Data in a Cycle	73
Overview of Managing Data in a Cycle	73
Loading Data from an External Database	73
Entering Data through Forms	73
Loading Data into a Cycle from a Model	73
Deleting Data from a Cycle	74
Locking Data in a Cycle	74
Deleting Cell Comments from a Cycle	74

Introduction to Cycles

Overview

A *cycle* is a structured pool of stored data with associated objects such as exchange rate sets, driver rate sets, models, and form sets.

There is no limit to the number of cycles that you can create.

Data in a Cycle

Each data record in the SAS Financial Management database belongs to a cycle. When you load data into the SAS Financial Management database, you must specify the cycle into which you are loading it. When a user enters data through a form, the data is stored in the cycle to which the form belongs.

Note: You can open only one cycle at a time. Operations such as loading or deleting data apply to the open cycle

Objects That Belong to Cycles

The following types of objects belong to a cycle:

- exchange rate sets
- driver rate sets
- models
- form sets

Some objects, such as dimensions and hierarchies, exist outside of cycles.

Cycle Manager Window

To display the Cycle Manager window, select **Tools** ⇒ **Cycle Manager** from the menu bar of the main SAS Financial Management Studio window.

The buttons on the right side of the window identify the tasks that you can perform:

New

launches the New Cycle wizard, which enables you to create a new cycle.

For details, see the online Help for the individual wizard pages.

Open

makes the selected cycle the active cycle.

Import

displays the Import Cycles window from which you can browse and select the SAS package file whose cycles you want to import into the target environment.

Export

displays the Export Cycle window from which you create a new SAS package file or browse and select an existing SAS package file to which you want to export the selected cycles.

This button is active only if at least one cycle is selected.

Copy

launches the Copy Cycle wizard.

For details, see the online Help for the individual wizard pages.

This button is active only if a single cycle is selected.

Delete

deletes the selected cycle or cycles.

You cannot delete a cycle that is currently in use or that has objects with which it is associated, such as models.

Lock

locks the selected cycle or cycles that are not currently locked.

The icon for a locked cycle includes a padlock. Locking a cycle also locks all the time periods in the cycle and all the models, form sets, and forms that belong to the cycle.

You cannot add or delete data from a locked cycle.

This button is active if at least one unlocked cycle is selected.

Unlock

unlocks the selected cycle or cycles that are currently locked.

This button is active if at least one locked cycle is selected.

Attachments

displays the Attachments window for the selected cycle, enabling you to work with the cycle's attachments.

This button is active only if a single cycle is selected.

Properties

displays the Properties window for the selected cycle, enabling you to view and modify the cycle's properties.

This button is active only if a single cycle is selected.

Close

closes the Cycle Manager window.

Create a New Cycle

To create a new cycle, use the New Cycle wizard. You can launch the New Cycle wizard in any of the following ways:

- Select **File** ⇒ **New Cycle**.
- Select **Tools** ⇒ **Cycle Manager**. In the Cycle Manager window, click **New**.
- If no cycle is open, select the **Create a new cycle and open it** radio button in any workspace that requires an open cycle, and then click **OK**.

For details, see the online Help for the individual wizard pages.

Copy a Cycle

To copy a cycle:

1. Select **Tools** ⇒ **Cycle Manager**. The Cycle Manager window is displayed.
2. Select a cycle and click **Copy**.

For details, see the online Help for the wizard.

The following items are included when you copy a cycle:

- data
- periods and period locks
- form sets

The following items are not included when you copy a cycle:

- models (including adjustment rules)
- exchange rate sets and driver rate sets
- cell comments
- locks on analysis members within a period

Because models are not copied, form sets in the copied cycle have no writable analysis members. If you want the forms to be writable, then you need to create a model and edit the form set and select at least one writable analysis member.

Open a Cycle

To open a cycle:

1. Select **File** ⇒ **Open Cycle**.

The Open Cycle window is displayed, listing all of the existing cycles to which you have access.

2. In the Open Cycle window, click a cycle to select it, and then click **OK**.

Note: You can open only one cycle at a time.

If there is no cycle open when you select the Periods, Rates, Models, or Forms workspace, then the default view in the workspace displays an error message. In addition, the following two radio buttons give you two additional ways to open a cycle:

- To create a new cycle and open it, select the **Create a new cycle and open it** radio button, and then click **OK**.

This launches the New Cycle wizard. For details, see the online Help for the individual wizard pages.

- To open an existing cycle, select the **Select a cycle to open** radio button, select a cycle from the displayed list, and click **OK**.

When a cycle is open, its name appears in the blue banner below the menu bar. If this blue banner is blank, then no cycle is open.

Data Records in a Cycle

Each data record in a cycle has a value and is associated with a single member of each required dimension in the cycle. In addition, a data record can be associated with a single member of one or more custom dimensions in the cycle.

Managing Data in a Cycle

Overview of Managing Data in a Cycle

You can use one of the following methods to load and manage data in SAS Financial Management:

- loading data from an external database, through the SAS Financial Management staging area
- entering data through forms
- loading model data

Loading Data from an External Database

To load data from an external database into a cycle:

1. Load the data into the SAS Financial Management staging area.

For details, see the *SAS Financial Management: Data Administrator's Guide*.

2. With the target cycle open, select one or more target periods in the Periods workspace.
3. Select **Load New Data** to launch the Load New Data wizard.

For details, see the online Help for the individual wizard pages.

Entering Data through Forms

To enable users to enter data into a cycle:

1. With the target cycle open, use the New Form Set wizard in the Forms workspace to create a suitable form set.

For details, see the online Help for the individual wizard pages.

2. Use the **Publish** option to publish the completed form set.

The data that users enter through the forms of the form set is saved in the target cycle. All data entered through forms is associated with the BaseForm member of the Source hierarchy.

Loading Data into a Cycle from a Model

To load data into a cycle from a model:

1. With the target cycle open, select one or more target periods in the Periods workspace.
2. Select **Load Model Data** to launch the Load Model Data wizard.

For details, see the online Help for the individual wizard pages.

Deleting Data from a Cycle

When you use any wizard to load data into a cycle, the wizard gives you choices about which data to delete from the crossings that receiving new data.

To delete data from a cycle without loading new data:

1. With the target cycle open, select one or more target periods in the Periods workspace.
2. Select **Delete Data** to launch the Delete Data wizard.

For details, see the online Help for the individual wizard pages.

Locking Data in a Cycle

There are three ways that you can lock a cycle or part of a cycle:

- lock an entire cycle.
- lock one or more periods within a cycle.
- lock one or more analysis members within a period, and leaving other analysis members unlocked

When you lock a cycle or part of a cycle, no one can add data to it or remove data from it.

Deleting Cell Comments from a Cycle

When you delete cell comments from a cycle, you can specify both the time periods and the analysis members from which to delete the comments.

1. With the target cycle open, select one or more target periods in the Periods workspace.
2. Select **Delete Cell Comments** to launch the Delete Cell Comments wizard.

For details, see the online Help for the individual wizard pages.

Periods View

The Periods view lists all the time periods in the open cycle. Its main use is loading data into the open cycle.

The following options are available:

New Periods

launches the New Periods wizard, which enables you to add leaf time periods to the open cycle. You can extend the time span that is covered by the cycle's time periods into the future, but not into the past.

Load New Data

launches the Load New Data wizard, which loads data into the open cycle from the SAS Financial Management staging area.

This option is active if at least one unlocked time period is selected.

The Load New Data wizard is equivalent to the Load Base Data job in SAS Data Integration Studio.

Load Model Data

launches the Load Model Data wizard, which loads data into the open cycle, based on a model that is associated with the same cycle or a different cycle.

This option is active if at least one unlocked time period is selected.

Delete Data

launches the Delete Data wizard, which enables you to delete data from the open cycle.

This option is active if at least one unlocked time period is selected.

Lock

locks the selected time periods that are not currently locked.

A locked time period is marked with a padlock . Data cannot be loaded into or deleted from a locked time period. Data cannot be entered into a locked time period through a form.

This option is active if at least one unlocked time period is selected.

Unlock

unlocks the selected time periods that are currently locked.

This option is active only if at least one locked time period is selected.

Properties

displays the Properties window for the selected cycle time period.

If multiple cycle periods are selected, the Properties window displays only the Analysis Member Locks page for the selected periods.

Note: If at least one analysis member of a period is locked, a lock icon  is displayed for that period.

Delete

Deletes the selected periods.

This menu item is available only if you select the last periods in the cycle (with no breaks in the selection), the cycle and the selected periods are not locked, and there is no data in any of the selected periods.

Delete Cell Comments

Deletes selected cell comments.

This menu item is available if you select one or more periods and the cycle and the selected periods are not locked.

TIP If you select target periods before you invoke the wizard, those periods are preselected in the wizard. (You can make changes if necessary.)

For details about these options, see the online Help for the individual wizard pages. For details about the SAS Data Integration Studio jobs, see *SAS Financial Management: Data Administrator's Guide*.

Chapter 9

Cycle Properties

Adjustments Page	77
Dimensions Page	78
Identification Page	78
Intercompany Trading Page	78
Security Page	79
Overview of the Security Page	79
Deny Access to a User	79
Effects of Denying Access to a User	79
Time Hierarchy Page	79

Adjustments Page

The Adjustments page shows the specified behavior for adjustments within all models that are based on the selected cycle.

The **Manual Adjustments** region has the following properties:

- The **Availability** drop-down list specifies the permissible timing of manual adjustments relative to adjustment rules. You can choose to permit the addition of manual adjustments both before and after adjustment rules run, or only before adjustment rules run, or only after adjustment rules run, or not at all.
- The **Allow unbalanced manual adjustments** check box is active only if the **Availability** field permits manual adjustments.

If you select this check box, then it is possible to create unbalanced manual adjustments in models that belong to this cycle. If you leave the check box empty, then all manual adjustments in models that belong to this cycle must be balanced.

The **Adjustment Rules** region specifies the types of adjustment rules that can be used in models that belong to this cycle.

Dimensions Page

The Dimensions page shows the dimension types and dimensions that the selected cycle uses.

You can click the column headings to change the sort order of the list.

Identification Page

The Identification Page displays the name, description, and type of the selected cycle.

To change the name or the description, enter new text in the appropriate field.

Intercompany Trading Page

The Intercompany Trading page shows the role of the Trader dimension type in the selected cycle.

The **Loaded Data** region contains the following two check boxes:

- **Intercompany accounts must be associated with an intercompany trading partner**

If this check box is selected, then every record that contains an intercompany account must also contain an organization for its Trader member. Records that do not satisfy this condition cannot be loaded into this cycle.

- If the **Non-intercompany accounts must be associated with the external trading member** check box is selected, then every record that contains a non-intercompany account must also contain EXT for its Trader member. Records that do not satisfy this condition cannot be loaded into this cycle.

In the **Intercompany Balancing Rules** region, the selected radio button specifies which organization member to use in any adjustment generated by an intercompany balancing rule that scans the data in this cycle:

- **Write values to the lowest common parent organization**

Use the hierarchically lowest organization to which both organizations in the unbalanced pair are subordinate.

- **Write values to the originating organization**

Use one of the organizations in the unbalanced pair. If the balancing account designated in the rule is a debit account, then use the organization that contributed debit accounts to the comparison. If the balancing account designated in the rule is a credit account, then use the organization that contributed credit accounts to the comparison.

Note: The **Intercompany Balancing Rules** region applies only to intercompany balancing rules, and not to net intercompany balancing rules.

Security Page

Overview of the Security Page

Use the **Security** tab of the cycle Properties window to deny certain users or groups access to the selected cycle.

Note: All users in the Administrator group retain access to all cycles, regardless of the settings on this tab.

Deny Access to a User

You can deny a user access to the selected cycle by denying access to the user individually or by denying access to a group that the user belongs to. Follow these steps:

1. Click **Add**.
2. In the Add Users and Groups window, select the user or a group the user belongs to, and click **OK**.
3. Select the user or group that you just added. Make sure that the **Can access this cycle** check box is not selected.

To restore default access to a user or a group, click the name of the user or group in the display region, and then click **Remove**.

It is possible to override security permissions that are assigned to a group.

Effects of Denying Access to a User

Denying a user access to a cycle has the following implications for that user:

- The cycle is not available to that user in SAS Financial Management Studio. This affects access to cycle properties as well as exchange rates, driver rates, models, and forms that are based on the cycle.
- Forms and read-only tables that are based on the cycle are not available to that user. The user cannot view data from the cycle or store data in the cycle.

Time Hierarchy Page

The Time Hierarchy page looks and works like the Time Hierarchy page of the New Cycle wizard.

There are two small differences:

- Because you are no longer in the wizard, there is no way to select a different time dimension.

- Your ability to select a different as-of date and/or a different time hierarchy is limited by the set of time periods that are currently part of the cycle. If you try to select a time hierarchy/as-of date combination that is incompatible with the current set of time periods, a warning message appears.

Chapter 10

Working with Exchange Rate and Driver Rate Sets

About Exchange Rates	81
About Driver Rates	82
Best Practices for Exchange Rate and Driver Rate Sets	82
Base Currency for Exchange Rate Sets	83
Precision of Exchange Rates	84
Displaying and Copying Exchange Rates and Driver Rates	85
Querying Exchange Rates and Driver Rates in a Table	85
Querying Exchange Rates and Driver Rates outside a Table	86

About Exchange Rates

Exchange rates refer to the values that are entered for the purpose of converting data from one currency to another. Exchange rates are stored in *exchange rate sets*. Exchange rate sets are required in order to create a model in the Models workspace. They are assigned to specific Analysis members within a model.

In SAS Financial Management, exchange rates can be simple or complex.

The values for *simple exchange rates* are defined by the exchange rate type and the members of the Time and Currency dimensions. For example, a simple exchange rate might be defined for the following members:

- Period Close exchange rate type
- the JAN 2011 time period
- EUR currency

The values of *complex exchange rates* can vary by the same components as simple exchange rates. A complex exchange rate definition can include a member from each of the following dimensions: Account, Organization, or any user-defined dimensions. For example, a complex exchange rate might be defined for the following members:

- Historic exchange rate type
- the JAN 2005 time period
- EUR currency

- Dividend account
- Italy organization
- a user-defined dimension such as Product

Note: Simple and complex exchange rates do not vary by Source or Trader dimensions.

To work with exchange rate sets, use the Exchange Rate Sets view in the Rates workspace.

About Driver Rates

Driver rates, formerly known as PRATEs, refer to constants that are entered for general use by SAS Financial Management. Examples of driver rates include tax rates, commission percentages, and days in the accounting cycle.

A *driver rate definition* must include at least one of the following dimensions: Time, Organization, Account, or user-defined dimensions. Driver rates are similar to complex exchange rates. However, an exchange rate definition is not required to include the Time dimension. As with simple and complex rates, driver rates do not vary by the Source or Trader dimension.

Driver rates are often useful as inputs in formula expressions. They are centrally located and their values can be managed by the Process Administrator or Rate Administrator. Other factors to consider when working with driver rates include the following:

- The assigned value cannot be converted to other currencies.
- The assigned value is not affected by Currency and Frequency member selections.

Like exchange rates, driver rates are stored in sets. To work with driver rates sets and rates:

1. Select the Rates workspace.
2. Select the **Driver Rates Set** tab to open the Driver Rate sets view, which lists the driver rate sets for the open cycle.
3. Select a driver rate set.

Driver rate types are user-defined. To work with driver rate types:

1. Select the Rates workspace.
2. Select **Tools** ⇒ **Driver Rate Types**.

Best Practices for Exchange Rate and Driver Rate Sets

When you create a rate set, SAS recommends that you assign a code, name, and description that help identify the contents of the set. For example, if an Analysis member called Actual uses actual rates, then name the related exchange rate set Actual as well.

When analyzing data with different sets, users might want to view Actual results translated at Budget rates. Users might want to view Actual results across several years using the rates of one year for all of the years, instead of using the actual rates for each

individual year. If you create additional sets with the desired values, users can apply the alternative sets to the underlying data. There are several possible ways to achieve this type of analysis, based on the user's preference and reporting needs:

- **Option 1—Copy the model.** Make a copy of the model. Then change the rate set selections for the relevant analysis member. Re-post any rules and adjustments, if applicable.

Although a copy provides a separate environment within which to report and operate, changes in the original model do not affect the copied model.

Note: This option does not allow you to compare the same analysis member with two different sets of rates from the same model. However, multiple tables or CDA reports can be used to present such a comparison.

- **Option 2—Create additional analysis members** Within the existing model, create additional analysis members and assign the desired exchange rate set to the new members. For example, a new analysis member called Actual at Budget contains actual data but is assigned the budget exchange rate set. When using this method, you must copy the Actual data, as well as any adjustments and rules on the Actual analysis member, to the new analysis member.
- **Option 3—Assign a different rate set.** Simply change the set that is assigned to a particular analysis member and re-post adjustments and rules. This approach is expedient, especially if it is done infrequently for a single reporting objective.

When using this method, note that changing the rate set affects the data for all users and reports. When any previously created dynamic reports are opened, they are refreshed with values that are based on the new rate set.

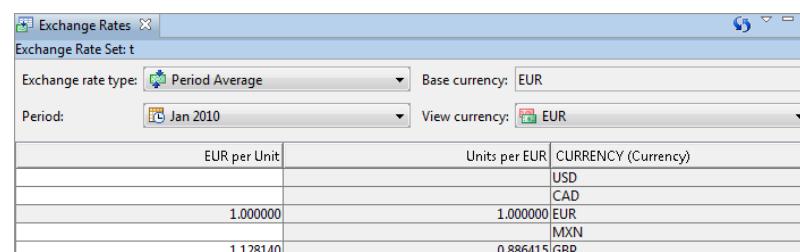
Base Currency for Exchange Rate Sets

When you create a new exchange rate set, you must select a base currency from among the available currency members in the Currency dimension.

The choice of base currency determines how exchange rates must be entered in SAS Financial Management and affects the currency and reciprocal in which the exchange rates are expressed.

The Base Currency of an exchange rate set is the TO currency for the purpose of loading exchange rates. The TO currency is the numerator of the exchange rate reciprocal. For example, if EUR is chosen as the base currency, all exchange rates must be expressed in EUR per units of the other currency. If a user enters the rate for GBP, it is expressed in terms of EUR per GBP (for example, 1.12814 EUR /GBP). When the exchange rate is entered, the exchange rate editor calculates and displays the reciprocal.

Figure 10.1 Exchange Rate and Reciprocal



EUR per Unit	Units per EUR	CURRENCY (Currency)
		USD
		CAD
1.000000	1.000000	EUR
		MXN
1.128140	0.886415	GBP

When you load exchange rates by using a SAS Data Integration Studio job, it is important to express the exchange rates in the TO and FROM reciprocal. The TO currency is the base currency of the exchange rate set, and the FROM currencies are the other currencies in the currency dimension being used by the cycle.

From a historic perspective on Euro Adoption, Council Regulation (EC) No 1103/97 from The Council of the European Union requires the exchange rate to be one euro expressed in terms of each of the national currencies. SAS Financial Management requires entry of the inverse of that rate.

All cross rates are triangulated via the base currency. For example, if EUR is the base currency, the rate for GBP / USD is determined from the EUR / GBP and EUR / USD exchange rates that are entered in the table. Assume that the following rates are entered:

- 1.128140 EUR / GBP
- 0.692488 EUR / USD

The GBP / USD rate is calculated as the following:

$$(0.692488 \text{ EUR/USD}) \times (1 / 1.128140 \text{ EUR/GBP}) = 0.613832 \text{ GBP/USD}$$

In the example above, where EUR is the base currency, historic Euro Adoption requirements for triangulation are accommodated. The rule requires monetary amounts that are to be converted from one national currency unit into another national currency unit first to be converted into a monetary amount expressed in the euro unit.

Precision of Exchange Rates

The rates and exchanges rates entered into SAS Financial Management are stored to 15 significant digits of precision. Any significant digits beyond 15 are used to round the 15th digit, and then are dropped. (Generally, if the next digit is less than or equal to 5, values are rounded down. If the next digit is greater than 5, values are rounded up.) This level of precision is sufficient for financial reporting and enables values to be converted to another currency, and then converted back to the original currency with no loss of display accuracy. Where very large values are involved (such as 100,000,000,000,000), if the full value were displayed instead of reporting in thousands or millions, a difference would be noted. In this example, it would be a difference of about 1,000 or 0.000000000001%.

SAS Financial Management always multiplies when performing conversion. It must calculate a reciprocal from the exchange rate that was entered and stored. In certain cases, the reciprocal calculation might introduce a minimal degree of inaccuracy. For example, the inverse of the value 3 is 0.33333333333333 (15 significant digits). If the value 0.33333333333333 is entered into the exchange rate editor, the related inverse is calculated as 3.00000000000003. The difference is approximately 0.000000000001%.

SAS Financial Management requires exchange rate values to be entered in Base Currency per unit. Consequently, users should enter the number of significant digits sufficient to calculate the reciprocal value to the desired level of precision. Here is an example using the Turkish Lira (TRL) value prior to January 2005:

- The base currency of the table is USD.
- The exchange rate is 1,489,400 TRL per USD.
- Because the base is USD, the TRL must be expressed in USD per TRL.
- The rate is 0.000000671411306566403, expressed to 15 significant digits (21 digits to the right of the decimal point in this case.) Note that the SAS Financial

Management rate editor allows up to 25 places to the right of the decimal, but only 15 significant digits.

If that rate is entered into the exchange rate editor, the reciprocal is calculated as 1,489,400.000000. The editor always displays six decimal places to the right of the decimal point. The precise value, displayed by holding your mouse pointer over the reciprocal in the exchange rate editor, is 1,489,399.999999999 (stored to 16 significant digits.) If the user enters only 15 digits to the right of the decimal point instead of 15 significant digits (for example, 0.000000671411307), the reciprocal displays as 1,489,399.999038.

SAS Financial Management always stores the value entered by the user in order to retain precision as entered, subject to the previously described constraints of 15 significant digits and 25 decimal places.

Displaying and Copying Exchange Rates and Driver Rates

By default, the exchange rate editor and driver rate editor display six decimal places to the right of the decimal for every value entered. This setting cannot be changed.

To view the exact value, click in the cell where the value was entered.

To view the precise value of the reciprocal, hold your mouse pointer over the specific rate with the mouse pointer.

To remove a previously entered rate, click in the desired cell and remove characters using the delete or backspace key.

It is not possible to enter a value of zero.

You can copy the value in a single cell and paste it to another cell in the same table or another table. An example includes for a different time period or exchange rate type. However, you cannot copy and paste multiple rates or exchange rates.

Querying Exchange Rates and Driver Rates in a Table

With the SAS Financial Management Add-In for Microsoft Excel, you can use calculated members to query exchange rates and driver rates in a table. The following functions are supported:

- fmRate: retrieves driver rates
- fmXRate: retrieves simple exchange rates
- fmCXRate: retrieves complex exchange rates

For more information about calculated members and these functions, see the *SAS Financial Management: Formula Guide*.

Querying Exchange Rates and Driver Rates outside a Table

To query exchange rates and driver rates outside a table, use cell data access (CDA) functions. The functions specific to rates and exchange rates are:

- CDARate—retrieves driver rates
- CDAXRate—retrieves simple exchange rates
- CDACXRate—retrieves complex exchange rates

For more information about these functions, see the *SAS Financial Management: Formula Guide*.

Chapter 11

Overview of Models

Introduction to Models	87
Required Features of a Model	87
Optional Features of a Model	89
Models View	90
Overview of the Models View	90
Options for All Models	90

Introduction to Models

A *model* provides a view of a selected cycle. Each model belongs to a single cycle.

When working with models, note the following:

- Models support *form sets*. A form set is a group of forms that are typically used for data entry in SAS Financial Management Add-In for Microsoft Excel.
 - Models are also used to generate reports in the SAS Financial Management Add-In for Microsoft Excel.
 - Models can include manual adjustments and adjustment rules.
 - Two or more models can be combined in a composite model.
-

Required Features of a Model

You specify most of the required features of a model when you create the model with the New Model wizard. A few required features are defined automatically by the software.

The following are the required features of a model:

- an association with a cycle

The underlying cycle stores the data from data-entry tables that are based on the model. The underlying cycle also provides the data for any read-only tables or CDA tables that are based on the model.

A model belongs to the cycle that is open when you create the model.

- a code, name, and description that identifies the model

After a model is created, you can modify its name and description, but not the model code.

- a hierarchy for each dimension type that is used in the cycle associated with the model

In a table that is based on the model, each data cell is associated with a crossing that consists of one member from each hierarchy that is specified for the model.

When you select a hierarchy, you select its As-of date for the model. The default As-of date (for dimensions other than Time) is **Always use the current date and time**. For each hierarchy that you select, the model uses the version that is current at the as-of-date.

Note: Data security (which is set on the **Security** tab of dimension members) is based on current specifications. As-of dates do not apply.

- a time span

By default, the New Model wizard selects a range that includes the first and last periods of the cycle.

The model gives users access to crossings whose time member falls within the model's time span.

- the default read member for each hierarchy

The default read member for each dimension is assigned in the Dimensions workspace. By default, the model inherits this default read member.

By default, the default write member is the same as the default read member. In the model properties, you can change the write member to any descendant of the read member.

If a dimension is not included on a table, the table displays the default read member for that dimension.

If a dimension is not included on a data-entry table and the default write member is a leaf member, then the crossing is writable (assuming that all its other members are leaf members). If you enter data for this crossing, then the difference in value is written to the default write member.

- an association of an exchange rate set with each analysis member in the model's analysis hierarchy

For any crossing whose value is calculated by means of a currency conversion, the exchange rate that applies is found in the exchange rate set that is associated with the crossing's analysis member. This enables you to use different sets of exchange rates for different analysis members.

- an association of a driver rate set with each analysis member in the model's analysis hierarchy

For any crossing whose value is calculated by means of a formula that contains the DRATE function, the DRATE function performs its lookup in the driver rate set that is associated with the crossing's analysis member. Among other things, this enables you to use different sets of tax rates, interest rates, or prices for Actual crossings and Budget crossings.

You set these associations on the Driver Rates page of the New Model wizard. You can change these associations on the Rates page of the model's Properties window.

- formatting specifications for debit and credit accounts

You can specify whether to display the values of debit accounts as positive numbers or as negative numbers, and likewise for credit accounts.

You make these specifications on the Debit and Credit Formats page of the New Model wizard. You can change these specifications on the Formats page of the model's Properties window.

After you create a model, you can modify most of its properties (other than the model code). Select the model and open its **Properties** window. In the model properties, you can also modify the precedence of hierarchies in determining a formula's scope.

Optional Features of a Model

A model can have the following optional features:

- attachments

To add or remove attachments, select the model, select **Attachments**, and use the **Attachments** view.

- access restrictions for users or groups

To restrict access to a model, select the model, and then select **Properties** ⇒ **Security**.

- locking

To lock a model, select the model, and then select **Lock**. To unlock a model, select **Unlock**. A locked model cannot be modified. The icon for a locked model includes a padlock.

- formulas that are associated with the model
- cell protection rules
- cell visibility rules
- data validation rules

A model can have the following optional features:

- manual adjustments

To work with manual adjustments, select the model, select **Manual Adjustments**, and use the Manual Adjustments view.

- adjustment rules

To work with adjustment rules, select the model, select **Adjustment Rules**, and use the Adjustment Rules view.

- prior period adjustments

In the New Model wizard or the model properties, you can link to another model that shares the same cycle to include adjustments from that model.

Models View

Overview of the Models View

The Models view lists the models that belong to the open cycle. To open this view, select the Models workspace.

Options for All Models

The following options are available for models:

New Model

launches the New Model wizard, which enables you to create a new model.

Import Models

launches the Import Models wizard, which enables you to import selected models from a SAS package file.

Export Models

launches the Export Models window, which enables you to export selected models to a SAS package file.

Post Adjustments

posts adjustments for the selected model. The posting includes both manual adjustments and adjustments that are generated by adjustment rules. The posting is confined to those time periods and analysis members that you specify in the Post Adjustments window, which is displayed when you select this option.

Export Data Records

launches the Export Data Records wizard, which exports data from the selected model to a target SAS library. Data is exported for every combination of the time periods and analysis members that you specify in the wizard.

Clear Warnings

removes all warnings from the selected models.

This option is active if at least one selected model has warnings.

To view a model's warnings, select the model, and then select **Properties**. The warnings are listed on the Adjustments page of the Properties window.

Lock

locks the selected models that are not currently locked.

A locked model cannot be modified. The icon for a locked model includes a padlock.

When a model is locked, every hierarchy in the model whose as-of date is **Always use the current date and time** has its as-of date automatically changed to the specific time at which the model is locked. For example, if a model is locked at precisely February 15, 2010 11:58:55 AM EST, then any as-of date value that was **Always use the current date and time** changes automatically to “Feb 15, 2010 11:58:55 AM EST.”

Unlock

unlocks the selected models that are currently locked.

Unlocking a model does not affect the as-of dates for any of its hierarchies. To change the as-of date for a hierarchy, you must use the **Dimensions** page of the model's Properties window.

Copy Here

launches the Copy Model wizard, which enables you to make a copy of the selected model that belongs to the same cycle.

When you copy a model, you have the option of including any existing visibility rules for the model. Cell protection rules, data validation rules, adjustment rules, and attachments are automatically copied.

Delete

deletes the selected models. The models cannot be referenced by another object such as form sets or composite models.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Adjustment Rules

displays the Adjustment Rules view, which enables you to work with the adjustment rules that belong to the selected model.

Manual Adjustments

displays the Manual Adjustments view, which enables you to work with manual adjustments that belong to the selected model.

Posted Adjustment Data

displays a Web page that lists all the currently posted adjustment values for the selected model. The list includes values from manual adjustments and adjustment rules.

For a list of currently posted adjustment values from a selected set of adjustment rules, use the **Posted Adjustment Data** option in the Adjustment Rules view.

Cell Protection Rules

opens a workbook in Microsoft Excel, in which you can view, modify, or add cell protection rules for the selected model. These rules are applied to all data-entry tables that reference the model.

Note: You can also export or load cell protection rules via SAS Data Integration Studio jobs. See the *SAS Financial Management: Data Administrator's Guide*.

Cell Visibility Rules

opens a workbook in Microsoft Excel, in which you can view, modify, or add visibility rules for the selected model. These rules are applied to all data-entry tables and read-only tables that reference the model, if visibility rules are enabled in the table properties in a form template or report.

Note: You can also export or load visibility rules via SAS Data Integration Studio jobs. See the *SAS Financial Management: Data Administrator's Guide*.

Data Validation Rules

displays a window in which you can define constraints for the values in data-entry forms.

Rules that are defined for a model apply to all data-entry tables that use the model (in bottom-up form sets only).

You can also export or load data validation rules via SAS Data Integration Studio jobs. See the *SAS Financial Management: Data Administrator's Guide*.

Attachments

displays the **Attachments** view, which enables you to work with attachments that belong to the selected model.

Properties

displays the Properties window, which enables you to view the properties of the selected model and to change some of them.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. The list of models displays only those models that match the string that you are entering.

Refresh

refreshes the view.

Attachments

displays the Attachments window for the selected model, enabling you to work with the model's attachments.

This button is active only if a single cycle is selected.

For details about the wizards, see the online Help for the individual wizard pages.

Chapter 12

Model Properties

Adjustments Page	93
Dimensions Page	94
Overview of the Dimensions Page	94
Select a Hierarchy	94
Specify an As-Of Date	95
Select a Default Write Member	95
Specify Time Periods	95
Manual Adjustments Warning	95
Formats Page	96
Formulas Page	96
Overview of the Formulas Page	96
Dimension Rank Order for Resolving Formula Conflicts	96
Formula Details	97
Identification Page	97
Periods Tab	97
Rates Page	97
Security Page	98
Overview of the Model Security Page	98
Set Model Access Permissions	98
Effects of Denying Access to a Model	98
Override Security Settings	98

Adjustments Page

The Adjustments page displays information about the model's manual adjustments and adjustment rules.

The **Prior Period Adjustments** region shows whether this model is tied to a prior model in a chain. Covering the time span of a cycle with a chain of models instead of a single model can be useful if you need to apply different adjustment rules to different time spans within the cycle.

To tie this model to a model that covers a preceding time span:

1. Select the **Carry over prior period adjustments from another model** check box.

Note: If there are no eligible models, then this check box is not active. A model is eligible if it belongs to the open cycle and its last time period is not the last time period of the cycle.

2. In the **Source model** field, use the drop-down list to select an eligible model.

The time period immediately following the last time period of the model that you select here automatically becomes the first time period of the model that you are editing. You can see this automatic selection in the **Start period** field when you select the Time dimension type on the Dimensions page.

To end the connection between this model and a prior model, deselect the **Carry over prior period adjustments from another model** check box.

The **Counts** region contains information about the adjustment process that the model defines.

The **Warnings** region recommends actions and gives the reasons for the recommendations.

Dimensions Page

Overview of the Dimensions Page

Use the Dimensions page to make changes that concern the hierarchies of the selected model.

Select a dimension type in the region on the left.

For any selected dimension type, you can change some of the following things:

- the hierarchy in that dimension type that the model uses
- the as-of-date value for the selected hierarchy
- the default write member for the selected hierarchy

For the Time dimension type, you can also change the following:

- the default read member for the selected hierarchy
- the set of time periods to which the model provides access to

Select a Hierarchy

To select a different hierarchy for the selected dimension type:

1. If necessary, use the **Dimension** field drop-down list to select the dimension that contains the hierarchy that you want to use.
2. Use the **Hierarchy** field drop-down list to select the hierarchy that you want to use.

To view the structure of the selected hierarchy, click **Preview**.

Note: To modify the columns that are displayed in the Hierarchy Preview window, right-click the column headings. In the pop-up window, select columns to display. If you also want to reorder the display, click **More**.

Specify an As-Of Date

The **As-of date** field shows either a specific time or the phrase “Always use the current date and time.” If this field shows a specific time, then the model uses the selected hierarchy in the form that it had at the specified time. If this field shows the phrase “Always use the current date and time,” then the model uses the selected hierarchy in the form that it has at the moment of use.

When you lock a model, the as-of date of every hierarchy in the model is automatically set to the specific time at which you locked the model. When you unlock a model, all the as-of dates remain set to the specific time at which you locked the model.

To change the content of the **As-of date** field:

1. Click the button that is next to the **As-of date** field. The As-Of Date window appears.
2. Make changes in the As-Of Date window, and then click **OK**.

You can set the same as-of-date for several hierarchies by selecting several dimension types at once and then selecting a value for the **As-of date** field. To select several dimension types at once, click each one while holding down the CTRL key. You can also select a block of dimension types by dragging the cursor across them or by using the SHIFT key.

Select a Default Write Member

To select a different default write member for the selected hierarchy:

1. Click the button that is next to the **Default write member** field. The Select Default Write Member window appears.
2. Use the Select Default Write Member window to make the selection, and then click **OK**.

Specify Time Periods

When the Time dimension type is selected, you can specify the entire set of time periods that is in the cycle that the model belongs to, or any continuous subset of it. Use the drop-down list for the **Start period** field to select the first time period to include in the model. Use the drop-down list for the **End period** field to select the last time period to include in the model. All time periods between the first period and the last period are included.

Manual Adjustments Warning

A warning message is displayed if the model has any active manual adjustments that refer to Time or Analysis dimension members that are no longer valid for the model. This situation might have occurred because of hierarchy changes or because of a change in the starting or ending time period. You should exit the model properties and place those manual adjustments on hold or delete them.

Formats Page

Use the Formats page to specify how the values of debit accounts and credit accounts are displayed in tables.

For debit accounts, select one of the following radio buttons:

Positive

Display positive debit balances as positive numbers, except when a particular table has an overriding setting.

Negative

Display positive debit balances as negative numbers, except when a particular table has an overriding setting.

For credit accounts, select one of the following radio buttons:

Positive

Display negative credit balances as positive numbers, except when a particular table has an overriding setting.

Negative

Display negative credit balances as negative numbers, except when a particular table has an overriding setting.

Formulas Page

Overview of the Formulas Page

Use the Formulas page of the Model Properties window to do the following:

- Set the rank order that will be used to resolve any multi-member formula conflicts that exist in the selected model.
- Check details about any formula in the model, including warnings for formulas that cannot be evaluated in the model.

Dimension Rank Order for Resolving Formula Conflicts

The display ranks all the dimensions in the selected model whose members can carry formulas. For each listed dimension, the **Members with Formulas** column tells you how many members in the selected model carry formulas. If two or more dimensions have members with formulas, then multi-member formula conflicts are possible at the crossings that contain two or more members with formulas. At any crossing where there is a multi-member formula conflict, the formula that is carried by the member that belongs to the most highly ranked dimension is evaluated. Any conflicting formulas are ignored.

To change the rank of a dimension in the selected model:

1. Select the dimension.

2. Use the **Move Up** button  or the **Move Down** button  to change the dimension's relative position.

Formula Details

To see detailed information about all the formulas in the model, click **Show Formulas**.
The Formulas window appears.

Identification Page

The Identification page contains the following information about the model:

- The model's code
- The model's name and description for each available data locale

To change the name or the description for any listed data locale, click the appropriate cell and enter the new text in it.

Periods Tab

Use the **Periods** tab to make certain changes that concern the time hierarchy of the selected model.

Rates Page

The Rates page shows which exchange rate set and which driver rate set is associated with each analysis member in the model.

To change the selections in any row:

1. Click the row to select it.
2. Click **Edit**.
3. Use the Rate Sets window to make the change.

Security Page

Overview of the Model Security Page

Use the Security page of the Model Properties window to deny specific users or groups access to the selected model.

Set Model Access Permissions

By default, users and groups with the appropriate capabilities have access permission for all models. You can grant or deny a user access to the selected model by setting user or group permissions. Follow these steps:

1. Add the user, or a group that contains the user, to the list in the large display region at the top. To do this, click **Add**. The Add Users and Groups window appears, enabling you to select a user or a group to add.
2. Select the user or group that you just added. Select or clear the **Can access this model** check box.

To restore default access to a user or a group, select the user or group name and click **Remove**.

Note: All users in the Administrators group retain access to all models, regardless of the settings on this tab.

Effects of Denying Access to a Model

If a user is denied access to a model, the model is not available for view or selection in SAS Financial Management Studio or in a form. In a read-only table in the SAS Financial Management Add-In for Microsoft Excel, the user can select a compatible model, if one exists. For details, see “Security in SAS Financial Management” in the *SAS Financial Management Process Administrator’s Guide*.

Note: The user is also denied access to a composite model that includes that model. The other effects of denying access to a composite model are the same as denying access to a model.

Override Security Settings

To override object-level security, use one of these approaches:

- **Add user permissions**—If the permission is assigned to a group, you can explicitly grant or deny access to the user. Permission that is assigned to a user overrides permission that is assigned to a group.
- **Add another group**—If the user belongs to multiple groups, the permissions for the closest group prevail. If the groups are equally close to the user, a grant takes precedence.

Note: Such overrides apply to security for cycles, dimension types, dimensions, hierarchies, models, and custom properties.

Chapter 13

Models: Rules and Adjustments

Overview of Adjustments and Adjustment Rules	101
Working with Adjustment Rules	102
Working with Manual Adjustments	102
What Happens When You Post Adjustments	103
Overview	103
If a Rule Fails	103
Reposting Adjustments	104
Being on Hold	104
Deleting Adjustments	104
Threshold for Adjustment Rules	105
Overview	105
Operations That Are Not Affected by the Materiality Threshold	106
Setting the Materiality Threshold	106
Materiality Threshold in Data-Entry Forms	106
Adjustment Rules View	107
Manual Adjustments View	109
Posting Status of Manual Adjustments and Adjustment Rules	111
Warnings	111

Overview of Adjustments and Adjustment Rules

The *adjustment process* can be a powerful feature of a model. You use manual adjustments and adjustment rules to enter and manage a variety of accounting entries, adjustments, and allocations that are not performed in an ERP system or entered through data-entry forms.

In the Models workspace, an administrator can create both manual adjustments and adjustment rules that are associated with a model. During a query that references that model, posted adjustment values are added to the values from the cycle. (The adjustments do not change the values that are stored in the cycle.)

In a form or report, users can select **Contributing data** for a crossing to view the contributions from adjustments.

manualadjustment	MAdj 22	sasdemo	5/8/2013 12:36 PM	3.00
manualadjustment	MAdj 23	sasdemo	5/8/2013 12:36 PM	5.00
rule	Reclass 101	sasdemo	5/8/2013 12:36 PM	(23.00)

The Source dimension uses the following members to capture data that is associated with manual adjustments and adjustment rules:

- Manual
- Bal
- Reclass
- Alloc
- CPO

Working with Adjustment Rules

SAS Financial Management supports six different types of adjustment rules. Depending on the types of adjustments allowed by the cycle's properties, the following types of rules are available:

- Intercompany balancing rules
- Net intercompany balancing rules
- Reporting-entity balancing rules
- Reclassification rules
- Allocation rules
- Ownership rules

To create, edit, delete, or copy the adjustment rules that belong to a model:

1. In the Models view, select the model.
2. Select **Adjustment Rules**.

In the Adjustment Rules view, all the adjustment rules for the selected model are displayed in a single list, in the order in which the rules are executed.

3. Use the Adjustment Rules view as explained in the online Help for that view.

Working with Manual Adjustments

A *manual adjustment* is an entry that is made at a model level for specified crossings. Values for a model are assumed to have a frequency of Period Activity.

SAS Financial Management supports both balanced and unbalanced manual adjustments. A cycle's properties determine whether manual adjustments are available, the adjustment rules that are permitted, and whether unbalanced manual adjustments are permitted in models that are associated with the cycle.

Balanced manual adjustments require an Organization member that has been designated as a reporting entity.

For each manual adjustment, you must designate a currency. All the adjustment values that are part of the adjustment are expressed in the designated currency.

To create, edit, delete, or copy the manual adjustments that belong to a particular model, do the following:

1. In the **Models** view, select the model.
2. Select **Manual Adjustments**.

The Manual Adjustments view for the selected model is displayed.

3. In the Manual Adjustments view, ensure that the selections are correct in the **Analysis** and **Time** fields and that the correct radio button is selected: **Before rules** or **After rules**.

This view maintains a separate set of manual adjustments for each combination of analysis member, time period, and position in the posting sequence.

4. Use the Manual Adjustments view as explained in the online Help for that view.

When you copy a manual adjustment to another time period, you have the option of making the copy a reversing entry instead of an ordinary copy.

What Happens When You Post Adjustments

Overview

The administrator can select whether to post a manual adjustment before or after adjustment rules. If a manual adjustment is posted first, then its output can be referenced in adjustment rules.

For a specified set of time periods and analysis members, the sequence of events in the posting process is as follows:

1. All the currently posted adjustment values are deleted.
2. All the **Before rules** manual adjustment values are posted.
3. The adjustment rules run in the order that is specified in the Adjustment Rules view. For each rule, adjustment values are generated and posted. The input to a given rule can include values from any **Before rules** manual adjustment and any adjustment that was generated by a rule that has already run.

By default, the adjustments from certain adjustment rules are subject to a threshold.

4. All the **After rules** manual adjustment values are posted.

If a Rule Fails

If a rule fails, then the failure is displayed in an error dialog box. In addition, the error dialog box provides any available information, such as the time and analysis members for which the rule was posted. The posting stops. Adjustments that were posted in this session remain.

To recover from the error:

1. Fix the rule that failed.

Alternatively, you can place the rule on hold, so that it is not included in postings. Later, you can fix the rule and take it off hold.

2. Repost the adjustments.

Reposting Adjustments

If you make changes to manual adjustments or adjustment rules, you must repost the adjustments, as well as any rules that rely on reposted values.

Being on Hold

When you place a manual adjustment or adjustment rule on hold, it is temporarily excluded from the posting process. Its previously posted adjustments are deleted from the selected time periods and analysis members during the next posting.

To place a manual adjustment or an adjustment rule on hold, select the item and select **Put On Hold**. The item is marked with the On Hold icon.

To remove the hold, select **Take Off Hold** instead.

These selections are also available in the properties for the manual adjustment or adjustment rule.

Deleting Adjustments

There are several approaches to deleting adjustments, depending on what you want to accomplish:

- **Delete an adjustment rule**—The rule and all its posted adjustments are immediately deleted, even if there are locks on some time or analysis members.
- **Delete a manual adjustment**—If the manual adjustment has no posted adjustments, it is immediately deleted.

If the manual adjustment has posted adjustments and its time and analysis members are not locked, both the item and its posted adjustments are immediately deleted.

If the manual adjustment has posted adjustments and its time and analysis members are locked, the deletion fails.

- **Place an adjustment rule or manual adjustment on hold**—This approach can be used to delete adjustments without deleting the rule or manual adjustment.

When you repost adjustments, all currently posted values are deleted for the selected time and analysis members. No new adjustments are posted for items that are on hold. This can affect any rules that previously relied on those adjustments.

Post Adjustments cannot be run for locked time or analysis members. Consequently, you cannot use the **On hold** mechanism to delete adjustments from time-analysis member combinations that are locked.

Threshold for Adjustment Rules

Overview

Adjustments that are generated by certain adjustment rules are subject to a materiality threshold that ignores (does not store) very small values that are considered immaterial. This threshold affects the following:

- allocation rules
- reclassification rules
- intercompany balancing rules
- net intercompany balancing rules
- self-elimination rules
- reporting entity rules

Absolute values that are lower than the threshold value are dropped (not posted). For example, assume the following adjustments:

Account	Value
A	5.0 (target)
B	0.0008 (target)
C	–5.0 (offset)
D	–0.0008 (offset)

With a threshold value of 0.001, any entry whose absolute value is less than 0.001 is not stored in the database. In the previous example, there would be two line items posted rather than four:

Account	Value
A	5.0 (target)
C	–5.0 (offset)

If the entry to the target crossing is below the threshold, the corresponding offset entry is dropped so that the entry continues to have a net value of zero. In this example, three line items are generated for posting:

Account	Value
D	6.0 (target)

Account	Value
E	0.0008 (target)
F	–6.0008 (offset)

With a threshold value of zero, all three of the line items are posted. With a threshold value of 0.001, only two line items are posted:

Account	Value
D	6.0 (target)
F	–6.00 (offset)

The threshold operates on the output of the rule immediately before storing the posted adjustments, using the rule's Currency and Period Activity frequency.

The same concept holds true for the balancing accounts and offsets that are generated by intercompany balancing rules and net intercompany balancing rules.

Note: The number of decimal places in the threshold value does not affect the precision of the output. For example, with a threshold value of 0.1, adjustment values might be 3.1416, 5.2, or 0.123. Those values are not rounded or truncated to match the threshold.

Operations That Are Not Affected by the Materiality Threshold

The materiality threshold does not affect the following:

- ownership rules
- facts that are entered directly by the user (not formula-generated)
- ETL facts

Setting the Materiality Threshold

The default materiality threshold is 0.001. You can modify the threshold, or you can effectively eliminate it by setting its value to 0. See “Post-Configuration Steps” in the *SAS Financial Management: System Administrator's Guide*.

After modifying the threshold value (and restarting the managed servers), you can repost adjustment rules to use the new threshold.

Materiality Threshold in Data-Entry Forms

In data-entry forms, the materiality threshold also affects the following:

- automatic allocation in bottom-up forms and form templates
- allocation in top-down forms and form templates
- redistribution of values because of hold rules

- driver formulas, whether they are initiated from writeback or from selecting **Run driver formulas** in SAS Financial Management Studio

Note: The materiality threshold does not affect queries. For example, it plays no part in sorting, ranking, data filtering, or data validation.

Adjustment Rules View

The Adjustment Rules view lists the adjustment rules for the selected model. The selected model is identified immediately below the view's tab.

The adjustment rules are listed in the order in which they run when you post adjustments for the selected model. Because this order is important, you cannot sort the list. To change the order of the list, select **Reorder Adjustment Rules**.

The options that are available in this view can be restricted by properties of the open cycle. If the cycle imposes no restrictions, then the following options are available:

New Intercompany Balancing Rule

launches the New Intercompany Balancing Rule wizard. This wizard enables you to create an intercompany balancing rule.

New Reporting-Entity Balancing Rule

launches the New Reporting-Entity Balancing Rule wizard, which enables you to create a reporting-entity balancing rule.

New Net Intercompany Balancing Rule

launches the New Net Intercompany Balancing Rule wizard, which enables you to create a net intercompany balancing rule.

New Allocation Rule

launches the New Allocation Rule wizard, which enables you to create an allocation rule.

New Reclassification Rule

launches the New Reclassification Rule wizard, which enables you to create a reclassification rule.

New Ownership Rule

launches the New Ownership Rule wizard, which enables you to create an ownership rule.

You can create only one ownership rule per model. If the selected model already has an ownership rule, then this option is not available.

Import Adjustment Rules

launches the Import Adjustment Rules wizard, which enables you to import an ownership rule or allocation rules from a SAS package file.

Export Adjustment Rules

launches the Export Adjustment Rules wizard, which enables you to export the selected adjustment rules to a SAS package file.

Note: This option is active if any adjustment rules are selected.

Put On Hold

excludes the selected adjustment rules from posting.

This option is active if at least one selected adjustment rule is not on hold.

Take Off Hold

takes off hold all the selected adjustment rules that are currently on hold.

Note: This option is active if at least one selected adjustment rule is on hold.

Delete

deletes the selected adjustment rules. When you delete an adjustment rule, the rule and all posted values that were produced by that rule are immediately deleted, regardless of any locks that might exist.

Copy Here

launches the Copy Rule wizard. This wizard enables you to make a copy of the selected adjustment rule for the same model.

Note: This option is active only if a single adjustment rule is selected.

Copy To

launches the Copy Rule To wizard. This wizard enables you to make a copy of the selected adjustment rule for another model.

You can copy an adjustment rule from one model to another model. The result might not be an exact copy if the source and target models do not use the same set of dimension types or the same dimensions within each dimension type.

If a dimension type is in the source model but not in the target model, then all references to it are discarded.

If a dimension type is in the target model but not in the source model, then its default write member is added to the rule wherever appropriate.

If source and target members do not match up properly within a dimension type, then a warning is displayed.

Note: This option is active only if a single adjustment rule is selected.

Reorder Adjustment Rules

displays the Reorder Adjustment Rules window. This window enables you to change the order of the adjustment rules for the selected model.

The adjustment rules run in the order in which they are listed in this view.

Posted Adjustment Data

displays a Web page that lists the currently posted values that were produced by the selected adjustment rules.

For a list of all the currently posted adjustment values for a model, use the **Posted Adjustment Data** option in the Models view.

Properties

displays the properties of the selected adjustment rule, enabling you to edit the properties that can be changed.

Note: This option is active only if exactly one adjustment rule is selected.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field and click the Search button. Partial matches are accepted.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Manual Adjustments View

The Manual Adjustments view lists the manual adjustments that belong to the selected model. The adjustments are filtered based on the following selections:

- **Before rules**—manual adjustments that are posted before any adjustment rules are posted
- **After rules**—manual adjustments that are posted after all adjustment rules are posted
- **Analysis**—manual adjustments that apply to the selected analysis member
- **Time**—manual adjustments that apply to the selected time period

Depending on the types of adjustments allowed by the cycle's properties, the following options are available:

New Balanced Manual Adjustment

launches the New Balanced Manual Adjustment wizard, which enables you to create a balanced manual adjustment.

New Unbalanced Manual Adjustment

launches the New Unbalanced Manual Adjustment wizard, which enables you to create an unbalanced manual adjustment.

Import Manual Adjustments

launches the Import Manual Adjustments wizard, which enables you to import selected manual adjustments from a SAS package file.

Export Manual Adjustments

launches the Export Manual Adjustments wizard, which enables you to export selected manual adjustments to a SAS package file.

At the time of export, the target environment must include the related dimension types, dimensions, hierarchies, cycle, and model.

Note: This option is active when at least one manual adjustment is selected.

Put On Hold

excludes the selected adjustment rules from posting.

Note: This option is active if at least one selected manual adjustment is not on hold.

Take Off Hold

takes off hold all the selected manual adjustments that are currently on hold.

Note: This option is active if at least one selected manual adjustment is on hold.

Move After

moves the selected manual adjustments from **Before rules** to **After rules**.

This option is active only if the selected manual adjustments are **Before rules** manual adjustments.

Move Before

moves the selected manual adjustments from **After rules** to **Before rules**.

This option is active if the selected manual adjustments are **After rules** manual adjustments.

Delete

deletes the selected manual adjustments.

If the manual adjustment has no posted adjustments, then it is immediately deleted.

If there are posted adjustments but no locks on the time or analysis members, then the manual adjustment and its posted adjustments are immediately deleted.

Otherwise, the deletion fails.

Copy Here

launches the Copy Manual Adjustment wizard. This wizard enables you to make a copy of the selected manual adjustment for the same analysis member, time period, and relationship to adjustment rules.

Note: This option is active only if a single manual adjustment is selected.

Copy To

launches the Copy Manual Adjustment To wizard. This wizard enables you to copy the selected manual adjustment to any available model, analysis member, and time period. The option to copy as a reversing entry is also available.

You can copy a manual adjustment from one model to another model. The result might not be an exact copy if the source and target models do not use the same set of dimension types or the same dimensions within each dimension type.

If a dimension type is in the source model but not in the target model, then all references to that dimension type are discarded in the copy.

If a dimension type is in the target model but not in the source model, then its default write member is added to the adjustment wherever appropriate.

If source and target members do not match up properly within a dimension type, then a warning is displayed.

Note: This option is active only if a single manual adjustment is selected.

Find Errors

displays the Manual Adjustment Errors window, which displays a summary of any rules in error, as well as the analysis and period for each rule that is in error.

Find Manual Adjustments

displays the Manual Adjustments window, which lists the analysis members and time periods that have manual adjustments.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field and click the Search button. Partial matches are accepted.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Properties

displays the properties of the selected manual adjustment, enabling you to edit the properties that can be changed.

Note: This option is active only if a single manual adjustment is selected.

For details, see the online Help for the wizard or property pages.

Posting Status of Manual Adjustments and Adjustment Rules

Every manual adjustment and every adjustment rule has a posting status. There are three possible posting status values:

Not posted

The item has never been posted.

Posted

The item has been posted. It has not been modified since it was last posted.

Modified since last post

The item has been posted. However, it has been modified since it was last posted.

Therefore, new reports do not reflect the latest modifications until you post the item again.

An item's current posting status is displayed in the **Posting Status** column of the view where the item is listed. You can also see it on the **General** tab of its Properties window.

In the **Models** view, if a model includes any manual adjustment or adjustment rule that is not on hold and has a posting status of "Not posted" or "Modified since last post," then that model is marked with the **Warning** icon. In addition, the Warning icon and a warning message are on the Adjustments page of the model's Properties window. The warning lets you know that there are adjustments or rules that need to be posted.

Warnings

In the Models view, a model might be marked with the Warning icon ().

These warnings are present only to notify administrators that there have been changes that might affect data values. For information about the changes that are responsible for the warning, select **Properties** and examine the **Warnings** region on the Adjustments page of the model's Properties window. Here are some example warning messages:

- Data records have changed
- Exchange rates have changed
- Driver rates have changed
- Hierarchies have changed

Chapter 14

Models: Adjustment Rule Types

Intercompany Balancing Rules	113
Net Intercompany Balancing Rules	114
Reporting-Entity Balancing Rules	114
Allocation Rules	114
Reclassification Rules	115
Ownership Rules: Introduction	115
Ownership Rules: Organization Hierarchy Requirements	115
Ownership Rules: Holding Types	116
Ownership Rules: Consolidation Methods	117
Overview of Consolidation Methods	117
Cost Method	117
Equity Method	117
Full Consolidation Method	118
Full Consolidation with Minority Interest Method	118
Proportional Method	119
Treasury Stock Method	119
Parent Method	119
“No Consolidation” Method	119
“No Method”	119

Intercompany Balancing Rules

An *intercompany balancing rule* compares the selected intercompany debit and credit accounts for every possible Organization/Trader combination, using the PTD balances for the TotalBeforeElim source member. Any difference in the totals for the selected accounts is recorded in the adjustment account, with an offsetting entry recorded in the specified offset account.

Adjustments are recorded based on the **Intercompany Balance Rule** selection in the cycle properties. This selection is made when the cycle is created and cannot later be modified. If the **Intercompany Balance Rule** selection is **Write values to the originating organization**, then the following rules apply:

- If the balancing account is a debit account, then the balancing account is adjusted for the organization that contributed the debit accounts to the unequal comparison.
- If the balancing account is a credit account, then the balancing account is adjusted for the organization that contributed the credit accounts to the unequal comparison.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

Net Intercompany Balancing Rules

A net intercompany balance rule compares the selected intercompany accounts for every possible Organization/Trader combination, using the TotalBeforeElim source member and the Period To Date frequency. Any difference in the net totals for the selected accounts is recorded in the adjustment account, with an offsetting entry recorded in the specified offset account.

The adjustment is always made to the lowest common parent organization of the two compared organizations.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

Reporting-Entity Balancing Rules

A reporting-entity balancing rule compares the selected non-intercompany debit and credit accounts for every reporting entity, using the PTD balances for the Total source member. Any difference in the total value for a selected account results in a single-sided entry to the specified adjustment account. The typical use of a reporting-entity balancing rule is to verify the standard balance-sheet equation, $\text{Total Assets} = \text{Total Liabilities} + \text{Total Equity}$.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

Allocation Rules

An *allocation rule* allocates the PTD value of a specified source crossing to a set of target crossings. A single allocation rule can allocate values from multiple source crossings, provided that all the allocations have the same pattern. The allocation table displays a preview of the allocation weights based on the source and target crossings and how the allocation should be performed (equal, weighted, or proportional).

You have the option of selecting an offset account for a balanced entry.

The selections that you make in the allocation wizard define the allocation table. The source crossings define the rows of the allocation table, and the target crossings define the columns of the allocation table. The allocation method then determines the weights per crossing.

It is possible to chain two or more allocation tables together in a single allocation rule. The outputs from the first allocation table become the inputs to the second allocation table, and so on. Each allocation table describes a distinct step of the allocation.

For example:

- A one-step allocation rule can distribute certain sales and marketing costs among several product lines.
- A two-step allocation rule can distribute certain sales and marketing costs among several product lines, and then distribute each resulting product line amount among several sales channels.

By default, the allocation rule allocates the PTD balance of the Total member of the Source dimension. Only the last step of an allocation rule is recorded to the Alloc member of the Source hierarchy.

Reclassification Rules

A *reclassification rule* reclassifies the values at selected source crossings to selected target crossings. You have the option of assigning an offset to create a balanced entry.

By default, the reclassification rule allocates the PTD balance of the Total Source dimension member.

The values generated by the reclassification rule are stored in the Reclass Source dimension member.

Ownership Rules: Introduction

An *ownership rule* generates adjustments to take proper account of a specified set of partial ownership relations among the organizations in a model's organization hierarchy.

If an ownership rule is part of a model that includes non-required dimension types, then you can specify which members from those dimension types to use in the adjustments that are generated by the rule. You can use the **Write Members** tab of the rule's Properties window to specify default write members for the ownership rule as a whole. You can then override the rule-level write members with transaction-specific write members when you define transactions within the ownership rule. Transaction-specific write members enable you to classified holdings by line of business or geographic region, as required by some accounting standards.

Ownership Rules: Organization Hierarchy Requirements

In order to support an ownership rule, a model must use an organization hierarchy that is designed with the requirements of an ownership rule in mind.

In such an organization hierarchy, every real-world organization for which a consolidation is performed must be represented by the following two members:

- consolidation parent—contains the real-world organization’s complete consolidated results.
- stand-alone parent—contains data for the real-world organization’s separate financial statements as defined by international accounting standards.

In addition, note the following:

- Each stand-alone parent must be a child of its corresponding consolidation parent.
- The consolidation parent must be a reporting entity.
- The stand-alone parent must be a reporting entity and must have no descendants that are reporting entities.

Note: In addition, it is a good idea to adopt a naming convention for the members of the organization hierarchy that makes plain the relationship between each consolidation parent and its corresponding stand-alone parent. The ownership rule uses the Direct Parent holding type to mark this relationship.

Ownership Rules: Holding Types

After you have completely defined an ownership rule, many pairs of organizations in the underlying organization hierarchy are connected by *holding relationships*. Each holding relationship belongs to one of the following four types:

Direct Parent

the special relationship that exists between a consolidation parent and its companion stand-alone parent. Both the consolidation parent and its companion stand-alone parent represent the same real-world organization. The consolidation parent contains the real-world organization’s complete consolidated results. The stand-alone parent contains the data for the real-world organization’s separate financial statements as defined by international accounting standards.

Direct Transacted

a relationship that is the result of past transactions between the holding organization and the held organization. Such transactions can include acquisitions, disposals, cash infusions, dividends, and others. The holding organization is always a stand-alone parent.

Direct Non-Transacted

a relationship that mirrors an underlying Direct Transacted relationship. For each Direct Transacted relationship, a corresponding Direct Non-Transacted relationship is formed by replacing each stand-alone parent with its companion consolidation parent. The direct ownership percentage of the Direct Non-Transacted relationship is identical to that of the underlying Direct Transacted relationship. For example, if the stand-alone parent A_s acquires a 30% share of B, this transaction gives rise to these two direct relationships:

- a Direct Transacted relationship with a 30% ownership share between A_s and B
- a Direct Non-Transacted relationship with a 30% ownership share between the consolidation parent A_c and B

Indirect

a relationship that results from a chain of direct relationships, which can be either Direct Transacted or Direct Non-Transacted. For example, if A has a direct ownership share of B and B has a direct ownership share of C, then A has an indirect ownership share of C.

Ownership Rules: Consolidation Methods

Overview of Consolidation Methods

The *consolidation method* that you associate with a holding relationship determines how the data for the held organization affects the computed financial results of the holding organization.

For a holding organization that is a consolidation parent, all the consolidation methods are available.

For a holding organization that is a stand-alone parent, only the following consolidation methods are available:

- Cost method
- Equity method
- Proportional method (only if the proportional method has been selected for the corresponding Direct Non-Transacted relation in which the companion consolidation parent is the holding organization)

Cost Method

In the *cost method*, the holding organization values its investment in the held organization at cost. Any dividends that the held organization pays to the holding organization are recognized by the holding organization as income, but they do not change the value of the investment.

There is no consolidation of the held organization's accounts into the holding organization's accounts. If the held organization is a hierarchical descendant of the holding organization, then the ownership rule generates adjustments to block the normal hierarchical aggregation process between the held organization and the holding organization. From the point of view of the holding organization, the held organization is a black box that might or might not pay a dividend.

As a rule, the cost method is appropriate when the holding organization does not exercise control over the held organization and its ownership percentage is low (20% or less).

Equity Method

In the *equity method*, the holding organization values its investment in the held organization initially at cost. Over time, it adjusts the investment value up or down to reflect its cumulative share of the net income of the held organization.

To achieve the consolidation of the appropriate share of the held organization's net income, you must specify the following three accounts:

- Source—contains the held organization's net income.
- Target—one of the holding organization's income statement accounts, typically Equity in Earnings of Subsidiary. Its value is computed as the value of the Source account times the ownership percentage.
- Offset—one of the holding organization's investment accounts, typically Investment in Subsidiary. Its value is computed by reversing the sign of the Target account. On the holding organization's balance sheet, changes to this investment account balance the changes in retained earnings that flow from Equity in Earnings of Subsidiary.

Except for this consolidation of a share of the held organization's net income, there is no consolidation of the held organization's accounts into the holding organization's accounts. If the held organization is a hierarchical descendant of the holding organization, then the ownership rule generates adjustments to block the normal hierarchical aggregation process between the held organization and the holding organization. For the holding organization, the held organization is a black box that generates a profit or loss in which the holding organization has an ownership share.

Typically, the equity method is appropriate when the holding organization does not exercise control over the held organization but owns a substantial percentage of it (20% to 50%).

Full Consolidation Method

In the *full consolidation method*, all the accounts of the held organization are consolidated fully into the corresponding accounts of the holding organization.

The full consolidation method is appropriate when the holding organization owns 100% of the held organization.

Full Consolidation with Minority Interest Method

In the *full consolidation with minority interest method*, the accounts of the held organization are fully consolidated into the corresponding accounts of the holding organization. In addition, the holding organization maintains a Minority Interest in Equity of Subsidiary account and a Minority Interest in Income of Subsidiary account.

The Minority Interest in Equity of Subsidiary account contains the amount of the held organization's equity that the holding organization does not own. This account typically appears in the holding organization's consolidated balance sheet either as a line in the equity section or as a line between the liabilities section and the equity section. To enable the software to compute the value of this account, you must specify a set of three accounts:

- The Source account contains the held organization's equity. (If there are components of the held organization's equity in several accounts, then you need to specify each of them as a Source account in a different account set. The Target and Offset accounts are the same in each account set.)
- The Target account is an equity account that is eliminated in the consolidation. Its value is computed as the value of the Source account times the percentage that is not owned times (-1).
- The Offset account is the Minority Interest in Equity of Subsidiary account. Its value is computed by reversing the sign of the Target account.

The Minority Interest in Income of Subsidiary account contains the amount of the held organization's net income that the holding organization does not own. To enable the software to compute the value of this account, you must specify a set of three accounts:

- The Source account contains the held organization's net income.
- The Target account is an income statement account that is eliminated in the consolidation. Its value is computed as the value of the Source account times the percentage that is not owned times (-1) .
- The Offset account is the Minority Interest in Income of Subsidiary account. Its value is computed by reversing the sign of the Target account.

The full consolidation with minority interest method is appropriate when the holding organization has full control of the held organization, but owns less than 100% of it.

Proportional Method

In the *proportional method*, the values of the held organization's accounts are multiplied by the ownership percentage and the results are consolidated into the corresponding accounts of the holding organization. If the ownership percentage is 100%, then the proportional method and the full consolidation method produce the same result.

The proportional method is often used for joint ventures.

Treasury Stock Method

The *treasury stock method* resolves circular or reciprocal holding relationships. If A owns part of B, which in turn owns part of A, then A can effectively reacquire B's share of A as treasury stock. This converts the circle into a net holding of B by A.

Parent Method

The SAS Financial Management software applies the *Parent method* automatically to the relationship between a consolidation parent and the associated stand-alone parent. These two members of an organization hierarchy represent the same real-world organization; the consolidation parent contains its consolidated financial results and the stand-alone parent contains its separate financial results as defined by international accounting standards. The Parent method performs a full consolidation of the stand-alone parent into the consolidation parent and applies adjustments to prevent the double counting of data from held organizations that is reflected in the stand-alone parent.

"No Consolidation" Method

The "*No consolidation*" method means that the data for the held organization does not affect the holding organization in any way. If the held organization is a hierarchical descendant of the holding organization, then the ownership rule generates adjustments to block the normal hierarchical aggregation process between the held organization and the holding organization.

You can use the "no consolidation" method to investigate the impact of excluding all data from the held organization.

"No Method"

The "*No method*" means that the relationship between the held organization's data and the holding organization's data is not affected by the ownership rule. If the held organization is a hierarchical descendant of the holding organization, then the normal hierarchical aggregation process takes place between the held organization and the holding organization. If the held organization is not a hierarchical descendant of the

holding organization, then the data for the held organization does not affect the holding organization in any way.

Chapter 15

Composite Models

Introduction to Composite Models	121
Required Features of a Composite Model	122
Optional Features of a Composite Model	122
Composite Models View	122
Overview of the Composite Models View	122
Options for Composite Models	123

Introduction to Composite Models

A *composite model* combines two or more models. The models can belong to the same cycle or different cycles, as long as the cycles use the same set of dimension types.

When working with composite models, note the following:

- You can use composite models to combine models that span different time periods or Analysis members.
- In the SAS Financial Management Add-In for Microsoft Excel, composite models can be used in read-only tables and in CDA tables.
- A composite model might reference more than one cycle with values for the same crossing. In that case, the value that is displayed for the crossing is the sum of those values.
- A composite model uses all the posted adjustments of all its referenced models. You cannot create additional adjustments for the composite model itself.

To work with composite models:

1. Open a cycle.
2. Select the Models workspace.
3. Select **Tools** ⇒ **Composite Models**. The **Composite Models** view is displayed.

Required Features of a Composite Model

You specify most of the required features of a composite model when you create the composite model with the New Composite Model wizard. A few required features are defined automatically by the software.

The following are the required features of a composite model:

- a code, name, and description that identifies the composite model
- two or more models
- a hierarchy for each dimension type that is used in the models

Although the selected models must use the same set of dimension types, they might not use the same set of hierarchies. Therefore, you need to select a hierarchy and as-of date for each dimension in the composite model.

- the default read member for each hierarchy

For a composite model, the default read members are always the default members of the hierarchies.

- an exchange rate set is associated with each analysis member in the composite model's analysis hierarchy
- a driver rate set is associated with each analysis member in the composite model's analysis hierarchy
- formatting specifications for debit and credit accounts

Optional Features of a Composite Model

The following are the optional features of a composite model:

- attachments

To add or remove attachments, select the composite model, select **Attachments**, and use the **Attachments** view.

- access restrictions for users or groups

To restrict access to a composite model, select the composite model and select **Properties** ⇌ **Security**.

Composite Models View

Overview of the Composite Models View

The Composite Models view lists all composite models, regardless of the cycles to which the models belong.

Options for Composite Models

The following options are available:

New Composite Model

launches the New Composite Model wizard, which enables you to create a new composite model.

Copy Here

makes a copy of the selected composite model.

Delete

deletes the selected composite model.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Attachments

displays the **Attachments** view, which enables you to add, open, or delete attachments for the selected composite model.

Properties

displays the Properties window, which enables you to view the properties of the selected composite model and to change some of them.

Filter

displays or hides the **Search** field at the top of the view. To filter the display, enter a character string in the **Search** field. The list of composite models narrows to display only those sets that match the string that you are entering. Partial matches are accepted.

Refresh

refreshes the view.

Chapter 16

Formulas

Overview of Formula Types and Formulas	125
Order of Execution	126
Server-Side Calculated Members	126
Account Types for Calculated Members	126
Resolving Conflicts between Dimensions	127
Virtual Child Members	127
Viewing Formula Information in SAS Financial Management Studio	127
Viewing Formula Information in Microsoft Excel	128
Writing the Formula Expression: Explicit Members and Fixed Members	128
Using a Function in a Bracketed Member Reference	129
Formula Scope	130
Defining Multiple Formulas on One Member	130
Formulas That Are Ignored, Invalid, or Not Rendered	130
Working with Driver Formulas	131
Overview	131
What Triggers Execution of a Driver Formula	131
Driver Formula Execution at Data Entry	132
The Run Driver Formulas Option	132
Form Design and the Run Driver Formulas Option	132
Excel-Based Calculated Members	133
Dictionary of Functions	134

Overview of Formula Types and Formulas

A *formula* is a mathematical expression that returns a value when it runs. A formula can be created on a calculated member in the Dimensions workspace in SAS Financial Management Studio.

SAS Financial Management supports the following four types of formulas:

- reporting formulas
- modeling formulas
- driver formulas

- Excel-based calculated member formulas

The formula type names reflect when the formulas run and where they are most commonly used in reporting and planning.

Reporting formulas and Excel-based calculated member formulas run post-query, meaning that the formulas are typically based on data inputs from a query. The most common examples are ratios such as Gross Profit Percentage and Earnings per Share.

Modeling formulas and driver formulas can be thought of as data-creation formulas. They generate data that can be consumed by reporting formulas and Excel-based calculated member formulas. These formulas also generate accounting logic such as retained earnings and cumulative translation adjustment (CTA) accounts. For this reason, modeling formulas and driver formulas are often termed pre-query formulas. Typical examples of these formulas include Sales based on Price and Units, Estimated Benefits Expense based on Salaries, and Training Expense based on Headcount.

Order of Execution

The order in which formulas are processed is as follows:

1. facts
2. intercompany eliminations
3. driver formulas
4. modeling formulas
5. retained earnings and cumulative translation adjustment accounts
6. reporting formulas

This process order enables both modeling formulas and driver formulas to be indirectly or directly referenced as source accounts of retained earnings accounts and cumulative translation adjustment accounts.

Reporting formulas are calculated after retained earnings accounts and cumulative translation adjustment accounts. Therefore, reporting formulas do not contribute to the results of retained earnings accounts and cumulative translation adjustment accounts. Each level can consume its own level as well as the preceding levels.

Note: Currency conversion occurs at each level of execution.

Server-Side Calculated Members

Account Types for Calculated Members

The behavior of calculated members in the Account dimension is influenced by the member's account type. The following are the available account types, grouped by category:

Balance account types

- Asset
- Liability

- Equity
- Statistical Balance

Flow Account Types

- Revenue
- Expense
- Statistical Flow

Other Account Types

- Statistical

Note: Because the Retained Earnings and Cumulative Translation Adjustment account types cannot be associated with formulas, they are excluded from the preceding list.

Formula results are calculated by a distinct method for each account type category.

- Balance account types—aggregate and then convert results
- Flow account types—convert and then aggregate results

Statistical accounts do not participate in aggregation or conversion.

The calculation and currency conversion methods depend on the account type of the calculated member.

Resolving Conflicts between Dimensions

Formula conflicts are limited to reporting formulas. Due to the order in which formulas are processed, driver formulas and modeling formulas always run before reporting formulas and are available only in the Account dimension. Because only the Account dimension permits driver formulas and modeling formulas, there are no conflicts with these formula types.

Virtual Child Members

In SAS Financial Management, a *virtual child* is automatically available to a member that is designated as a roll-up. A virtual child enables you to enter values at a roll-up point where less detail is required or where spreading or allocations are involved.

Virtual children are available for the following dimension types:

- IntOrg
- Account
- Custom

You can run a formula for a virtual child member just like you can run a formula for an ordinary leaf member. However, you cannot assign a formula to a virtual child like you can to an ordinary leaf member.

A formula on a roll-up member is always ignored, and the result is the sum of leaf values.

Viewing Formula Information in SAS Financial Management Studio

SAS Financial Management Studio provides formula information in the Dimensions and Models workspaces.

In the **Dimensions** workspace, formula information is available on the **Members** and **Hierarchies** tabs for all dimension types that support calculated members.

The Account dimension type offers optional informational headers and columns for both **Formula Type** and **Formula Count**.

Viewing Formula Information in Microsoft Excel

Formula information is available on a crossing-by-crossing basis in the SAS Financial Management Add-In for Microsoft Excel.

For either a data-entry table or a read-only table, select **Tools** ⇒ **Cell Information** from the **SAS Financial Management** menu to view formula information. The following formula information is displayed:

- formula type
- dimension of calculated member
- formula expression
- formula name
- fixed members, if any

Writing the Formula Expression: Explicit Members and Fixed Members

In SAS Financial Management, a formula expression reads from the same crossings where it runs. However, there are two exceptions: explicit members in the formula expression text and fixed member references.

- **an explicit member**—If a member is explicitly named in the expression, then that member is used. For example:

```
[ "ACCOUNT"="A100" ] [ "PRODUCT"="P1100" ] + [ "ACCOUNT"="A200" ] [ "PRODUCT"="P1100" ] )
```

The bracketed expression uses this syntax:

```
[dimension-type=member-code]
```

- **a fixed member**—At the next level of precedence is a *fixed member*. Fixed members are useful for complex formulas. Rather than specifying the same member multiple times, you can define a fixed member.

The fixed member is applied to all formula inputs that do not explicitly include a member of that dimension. For example, you select a fixed member of the Product dimension, “P1100”. Using the example above, but adding a fixed member of “P1100”, the formula can now be written as follows:

```
[ "ACCOUNT"="A100" ] + [ "ACCOUNT"="A200" ]
```

As long as the expression does not explicitly include a member of that dimension, the fixed member applies. In the following formula expression, a fixed member of “P1100” would apply only to the first operand:

```
[ "ACCOUNT"="A100" ] + [ "ACCOUNT"="A200" ] [ "PRODUCT"="P1200" ]
```

- **the same member as in the target crossing for the formula**—If there are no fixed-member or explicit-member references, then the formula expression runs at the same

crossings that it reads from (after considering formula scope). If a dimension is off the table, then the formula reads from the default read member and executes at the default write member for that dimension.

Using a Function in a Bracketed Member Reference

With some exceptions, you can use a function within a bracketed member reference. Here is an example:

```
SUM(["ACCOUNT"="Accounts & Notes Receivable"]
["TIME"=OPENINGPERIOD(ANCESTOR("TIME","Year"))]:
["ACCOUNT"="Accounts & Notes Receivable"]
["TIME"=CURRENT("TIME"))] / (PROPERTY("TIME","Month_number"))
```

In this example, the OPENINGPERIOD function works with the SUM function and the PROPERTY function to calculate the year-to-date average balance of the Accounts and Notes Receivable account. The PROPERTY function retrieves the values of a custom property (Month_number) of the Time dimension.

The following string functions are supported using this syntax:

- ANCESTOR
- CLOSINGPERIOD
- COMPRESS
- CURRENT
- FIRSTCHILD
- FIRSTSIBLING
- LASTCHILD
- LEFT
- LOWCASE
- NEXT
- OPENINGPERIOD
- PARENT
- PREVIOUS
- PROPERTY
- REPEAT
- REVERSE
- RIGHT
- SUBSTR
- SCAN
- TRIM
- UPGRADE
- VIRTUALCHILD

Note: If a function is not supported using this syntax, you can still use it within an IF or NESTIF function.

Formula Scope

Formula scope is an optional means of restricting where a formula runs. It can be applied on an expression-by-expression basis and is available for all server-side formulas (reporting, driver, and modeling formulas). From a performance standpoint, formula scope is most effective when you use it with modeling formulas to limit the number of crossings where a formula runs.

Defining Multiple Formulas on One Member

You can define an unlimited number of formulas on a single calculated member. To improve formula performance, maintenance, and manageability, define multiple formulas on one member as an alternative to the following functions:

- SUBSTR
- IF
- NESTIF

Use fixed members and formula scope to customize the formula and where it is applied.

When you create a formula, it is assigned a rank. The initial rank order for a calculated member is based on the order in which the formulas were created. The rank of a formula determines which formula is executed in the event of formula scope overlap.

Formulas That Are Ignored, Invalid, or Not Rendered

This section identifies formula expressions that are ignored, invalid, or not rendered in SAS Financial Management. A warning message is provided on the Formulas page of the model properties.

- **Ignored**—Calculated members in this category are not executed due to the priority of other computations such as roll-up logic. Because the formula expression is ignored, there is no cell information. Examples of this category include the following:
 - formulas on a roll-up member
 - formulas that run out of bounds
 - referencing a dimension or member that is not in the model
- **Invalid**—Calculated members in this category are not run at query time because they are incorrect in the context of the model. In most cases, these expressions pass validation in the formula editor. However, the expressions have warning messages as described above.

In a table, an invalid driver formula is displayed as a non-writable cell with a value of zero. Modeling and reporting formulas render as red cells. Select **Tools** ⇒ **Cell information** to view warnings for modeling and driver formulas. Examples of this category include the following:

- circular references
- divide by zero
- **Not rendered**—This type of formula restriction applies to the use of constants in driver and modeling formulas. For both formula types, the result is a non-writable cell with a value of zero. No warning message is available in **Cell Information**. The following is an example of this category:
 - referencing only constant values

Working with Driver Formulas

Overview

Driver formulas provide a method to create data using the context of a data-entry table. The following items define the range of processing for a driver formula:

- formula scope (defined on the member)
- writable analyses (defined in the form set)
- writable crossings on the form set template (after excluding crossings with hide visibility rules and crossings removed by using filter member combinations)
- the driver formulas included on the data entry table in the form set

When working with driver formulas, note the following:

- Driver formulas are not executed at TIME members that are locked in the cycle.
- With SAS Financial Management Studio 5.3 and later, only drivers formulas included on the data-entry table are executed. Prior releases of SAS Financial Management Studio executed all driver formulas in a hierarchy, regardless of whether they were included on the table or not.
- With SAS Financial Management Studio 5.4 and later, drivers that are removed from the data-entry table by using the filter member combinations are no longer executed. A crossing must be navigable on the table for the driver formula to execute.

What Triggers Execution of a Driver Formula

Based on their design and limited range of execution, driver formulas provide an efficient way to calculate values. There are two ways to run driver formulas:

- entering data into a form.

Driver formula execution is triggered by the change in a value on a data-entry form. When the value is entered, the formula result is calculated and stored in the database. No additional action is required.

- selecting the **Run driver formulas for this form set** option in the Forms workspace of SAS Financial Management Studio.

This option triggers the execution of driver formulas for the entire form set.

Note: Selecting **Refresh** for the form set does not trigger the execution of driver formulas.

Driver Formula Execution at Data Entry

The following guidelines apply to driver formula processing at the time of data entry (in Microsoft Excel):

- Driver formulas read from and write to the BaseForm member of the Source dimension. A driver formula can read from other members of the Source dimension if you explicitly state the members in the formula expression or implicitly state them as default members. A driver formula can write only to BaseForm; this cannot be modified.
- Driver formula inputs that are referenced in the formula expression must be included on a data-entry table to trigger driver formula execution at writeback.
- Driver formula expressions with relative time references such as `[ "TIME"=CURRENT ( "TIME" ) -1 ]` can read from Time members that are not included in the form. However, the formula expressions write only to those Time members that are included in the form. Driver formulas do not execute at Time members that are locked in the cycle.

The Run Driver Formulas Option

The **Run driver formulas for this form set** option is required for the following scenarios:

- changes to global values such as exchange rates, PRATE or DRATE rates, and formula expression inputs that are not in a given form
- changes to a driver formula expression after data input
- creation or deletion of a driver formula after data input
- loading data records that affect driver formulas
- formula inputs that are not included on the table

Form Design and the Run Driver Formulas Option

Form design defines the scope for execution of driver formulas when you select the **Run driver formulas for this form set** option. This option examines member selection for all slicers, rows, and columns. Limiting the number of slicers and members in a form limits the number of drivers that need to be executed. This improves performance.

To optimize the performance of the **Run driver formulas for this form set** option, we recommend the following guidelines:

- Limit the dimension members in rows, columns, and slicers to the members that are required for data entry (in Microsoft Excel). Be sure to consider the Source and Trader dimensions.
- Use a separate read-only table for data that is needed in a form for information purposes only. For example, if you need to enter data for a budget, maintain the Actual data in a separate read-only table in the same form.

- Use Visibility Rules (hide) or Filter Member Combinations to restrict the number of crossings that are required to run driver formulas.

With SAS Financial Management 5.4, only drivers included on the data-entry table are executed. The **Run Driver Formulas** option makes the following accommodations for members that are not on the data-entry table of a form template:

- If a dimension type is not on the table, run drivers first examines the default read member in the model hierarchy for that dimension type. (If the default read member is overridden in the form template, the value from the form template is considered instead.)

If the default read member is excluded by visibility rules or by a filter member combination, then the run drivers does not proceed.

- Otherwise, run drivers uses the default write member for the model hierarchy for the missing dimension type.

If the default write member is not a leaf member, then run drivers cannot complete the operation. There is an exception: if either the **Allow data to be entered for Parent members other than time** option or the **Allocate from Parent members other than Time using predefined weights** option is selected, then the run drivers uses the VC member of the default write member.

Note: A visibility rule or filter member combination that includes the Frequency, Currency, or Source dimension has no effect on run drivers.

Excel-Based Calculated Members

With the SAS Financial Management Add-In for Microsoft Excel, you can create Excel-based calculated members on a table-by-table basis. Excel-based calculated members look the same as members that are created in SAS Financial Management Studio. They are most similar in behavior to reporting formulas, sharing similar calculation methods and currency conversion methods. They differ primarily in reuse because formulas are created and saved locally, not on the server. Excel-based calculated members support any combination of the following in a formula expression:

- reference to any member in the same dimension
- absolute reference to any crossing in a table in the same workbook
- absolute reference to any cell in the same workbook
- any Excel function or valid Excel expression
- any of the SAS Financial Management calculated-member functions:
 - **fmValue**
 - **fmCode**
 - **fmProperty**
 - **fmRate**
 - **fmXRate**
 - **fmCXRate**

For a description of each of these functions, see the “Dictionary of Calculated-Member Functions” in the online Help for the SAS Financial Management Add-In for Microsoft Excel.

Dictionary of Functions

The following functions can be used in SAS Financial Management Studio formulas. They are available for selection on the **Functions** tab of the Formula Editor window.

CAUTION:

Every complete formula must return a numeric value. Therefore, only functions that return numeric values can be used as complete formulas. Any function that returns a string value or a Boolean value must be used in an argument of another function.

ABS

returns the absolute value of its only argument. The argument must have a numeric value.

ACOS

returns the arc cosine, in radians, of its only argument. The argument must have a numeric value between -1 and +1, inclusive.

ANCESTOR

The ANCESTOR function has two signatures:

- **ANCESTOR(dimtypecode, number_of_levels)** returns the member code of the ancestor a specified number of levels above the current member for a dimension.
- **ANCESTOR(dimtypecode, level_name)** returns the member code of the ancestor at a named level (such as "Year") above the current member for a dimension.

Use this function to navigate a hierarchy, typically the Time hierarchy. It enables a formula to use a member that is one or more levels above the current member, or to use a specified period type, such as Year.

The ANCESTOR function takes two arguments and supports an optional third argument:

- The first argument can be either a dimension type code or a function that returns a member code:
 - If the first argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
 - If the first argument is a function that returns a member code, then the returned code explicitly specifies a member.
- The second argument is an ancestor designator. The ancestor designator can take two forms:
 - *number_of_levels*: An integer indicates a number of hierarchical levels above the member that is specified by the first argument. You can use an integer as the second argument no matter what the first argument is.
 - *level_name*: If the first argument is the Time dimension type code or a function that returns a time period code, then you can use a quoted string that contains one of the following period type values to indicate a particular level of the time hierarchy: **AllYears**, **Year**, **HalfYear**, **QuarterYear**, **Month**, or **Day**.

- (Optional) Third argument: The optional third argument specifies a number of time periods in the past or future. This argument is typically used to get a prior year value.

This formula might be applied to the Beginning Balance - Equipment account to return the prior year value of the Ending Balance, Equipment account:

```
[ "ACCOUNT"="Ending Balance, Equipment" ] [ "TIME"=ANCESTOR ( "TIME", "Year", -1 ) ]
```

ASIN

returns the arc sine, in radians, of its only argument. The argument must have a numeric value between -1 and $+1$, inclusive.

ATAN

returns the arc tangent, in radians, of its only argument. The argument must have a numeric value.

BETAINV

returns the p th quantile from the beta distribution with shape parameters a and b . The probability that an observation from a beta distribution is less than or equal to the returned quantile is p .

Syntax:

```
BETAINV (  $p$ ,  $a$ ,  $b$  )
```

p is a numeric probability in the range $0 \leq p \leq 1$.

a is a numeric shape parameter, in the range $a > 0$.

b is a numeric shape parameter, in the range $b > 0$.

For example, **BETAINV**(0.001,2,4) returns a value of 0.0101017879.

Note: BETAINV is the inverse of the PROBBETA function.

CEIL

returns the smallest integer that is greater than or equal to its only argument. For example, **CEIL**(5.3) returns 6 and **CEIL**(5.0) returns 5. The argument must have a numeric value.

CLOSINGPERIOD

returns the code of the member that is the last-listed leaf member that is hierarchically subordinate to the member that is specified either explicitly or implicitly by the argument of the function. If the member specified by the argument is itself a leaf member, then the CLOSINGPERIOD function returns the code of that same leaf member.

The CLOSINGPERIOD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

The CLOSINGPERIOD function can be used with any hierarchy, but it is primarily intended for use with time hierarchies.

Consider a year-quarter-month time hierarchy and the following reference:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=CLOSINGPERIOD ( "TIME" ) ]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the last month of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the last month of that quarter. When evaluated at a month crossing, this reference retrieves the value of the Headcount account for the month in that crossing.

Instead of a dimension type code, the argument of CLOSINGPERIOD can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=CLOSINGPERIOD ( FIRSTCHILD ( ( "TIME" ) ) ) ]
```

COMPRESS

returns a character string that is formed by removing certain characters from an input string. You can use this function with either one argument or two arguments:

- If you use COMPRESS with one argument, then the argument is the input string and the function removes all the spaces from it. For example, **COMPRESS ("ABCD")** returns the string **ABCD**.
- If you use COMPRESS with two arguments, then the first argument is the input string and the function removes from it all the characters that are specified in the second argument. For example, **COMPRESS ("A+B-C=D", "+-= ' ")** returns the string **ABCD**.

COS

returns the cosine of its only argument. The argument must have a numeric value, which represents an angle measure in radians.

CURRENT

returns the code of the member that satisfies these two conditions:

- It belongs to the dimension type that is specified in the function.
- It is in the crossing where the function is evaluated.

The CURRENT function takes one argument: a dimension type code. It is most often used as part of a relative reference within a time hierarchy. For example, the following reference, when evaluated for a certain month, retrieves the value of the Headcount account for the same month of the previous year:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=CURRENT ( "TIME" ) -12 ]
```

CURRENTNAME

As of SAS Financial Management 5.4, the CURRENTNAME function is no longer available.

DATE

returns the SAS integer representation of the date on which it is evaluated. This function does not take an argument. For example:

- On January 1, 1960, **DATE ()** returned **1**.
- On January 2, 1960, **DATE ()** returned **2**.
- On February 1, 1960, **DATE ()** returned **32**.

DATE () and **TODAY ()** are equivalent.

DATETIME

returns the number of seconds that have elapsed since the beginning of January 1, 1960. This function does not take an argument. The syntax is:

```
DATETIME ( )
```

DAYOFMONTH

returns the sequence number of the day on which it is evaluated within the month in which it is evaluated. For example, on the twelfth day of any month **DAYOFMONTH ()** returns **12**.

DAYOFWEEK

returns the sequence number of the day on which it is evaluated within the week in which it is evaluated, starting with Sunday as day 1. For example, on any Thursday **DAYOFWEEK ()** returns **5**.

DAYOFYEAR

returns the sequence number of the day on which it is evaluated within the year in which it is evaluated, starting with January 1 as day 1. For example, on February 10 of any year **DAYOFYEAR ()** returns **41**.

DRATE

retrieves numeric values from driver rate sets.

A driver rate set consists of tables that are specific to driver rate types. Each driver-rate-type-specific table contains a column of numeric values and several columns that represent dimension types. Each table row associates the numeric value that it contains with the combination of dimension members that it contains.

The syntax of the DRATE function is:

```
DRATE(rate_type)
```

rate_type is a string that represents the name of a driver rate type.

The driver rate set that the DRATE function searches in is determined by these two things:

- the analysis member for which the function is evaluated
- the driver rate set that is associated with that analysis member in the governing model

Within that target driver rate set, the DRATE function uses the table for the driver rate type that is specified by its argument. From that target table, the DRATE function retrieves the numeric value that is associated with the same combination of dimension members that the function is being evaluated for. If that target table contains no numeric value that is associated with the same combination of dimension members that the function is being evaluated for, then the DRATE function returns a missing value.

In this example formula, the DRATE function works on the Income Taxes account member. The desired outcome is to multiply Income Before Taxes by a predefined rate that varies by Organization to return an estimated Income Tax value. If Income Before Taxes is less than zero, then Income Taxes is zero.

```
=IF(["ACCOUNT"="Income Before Taxes"]<0,  
["ACCOUNT"="Income Before Taxes"]* DRATE("TaxRate2012")*-1,0)
```

The DRATE for TaxRate2012 can be displayed in an Excel report using an Excel calculated member with the following syntax: **=fmRate("TaxRate2012")**.

EXP

returns the result of raising the constant *e* to the power that is specified in its only argument. The argument must have a numeric value. *e* is the base of the natural logarithms, which is approximately 2.718.

EXP is the inverse of LOG . For example, **EXP(LOG(2.65))** returns **2.65**.

FIRSTCHILD

returns the code of the member that is the first-listed hierarchical child of the member that is specified either explicitly or implicitly by the argument of the function.

The FIRSTCHILD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=FIRSTCHILD ( "TIME" ) ]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the first quarter of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the first month of that quarter.

Instead of a dimension type code, the argument of FIRSTCHILD can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=FIRSTCHILD ( NEXT ( ( "TIME" ) ) ) ]
```

If the FIRSTCHILD function is applied to a member that has one child, then it returns the code of that one child.

If the FIRSTCHILD function is applied to a leaf member, then the operand that the function is part of receives a value of 0.

FIRSTSIBLING

returns the code of the member that is the first-listed hierarchical sibling of the member that is specified either explicitly or implicitly by the argument of the function.

The FIRSTSIBLING function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=FIRSTSIBLING ( "TIME" ) ]
```

When evaluated at a January, February, or March crossing, this reference retrieves the value of the Headcount account for January. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the first quarter of the same year.

Instead of a dimension type code, the argument of FIRSTSIBLING can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=FIRSTSIBLING ( PARENT ( "TIME" ) ) ]
```

If the FIRSTSIBLING function is applied to an only child, then it returns the code of that one child.

FLOOR

returns the largest integer that is less than or equal to its only argument. For example, **FLOOR(5.3)** returns 5 and **FLOOR(5.0)** returns 5. The argument must have a numeric value.

IF

returns a value that depends on the truth value of a Boolean expression. The IF function takes three arguments, as illustrated in the following example:

```
IF(["ACCOUNT"="Net Sales"]=0,0,["ACCOUNT"="Net Income"]/["ACCOUNT"="Net Sales"])
```

The first argument is the Boolean expression. The second and third arguments can be expressions of any kind. If the Boolean expression is true, then the IF function returns the value of the second argument. If the Boolean expression is false, then the IF function returns the value of the third argument.

The Boolean expression can compare two character values or two numeric values. Within the Boolean expression, you can use any of the Boolean operators and comparison operators that are available on the symbol toolbar of the Formula Editor window. You can also use the ISCURRENT, ISLEAF, and ISLEAFCROSSING functions, which return Boolean values.

The second and third arguments must be of the same data type. They must both yield numeric values, or they must both yield character-string values, or they must both yield Boolean values. (Recall that a character-string or Boolean value can be returned only as an argument of another function.)

INDEX

returns an integer that indicates the starting position of one string within another string. The INDEX function takes two arguments:

- The first argument is the string to search in.
- The second argument is the string to search for.

If the second string occurs more than once in the first string, then the INDEX function returns the starting position of its first occurrence. If the second string does not occur at all in the first string, then the INDEX function returns 0. For example:

- **INDEX("herewego", "we")** returns 5
- **INDEX("nono", "no")** returns 1
- **INDEX("yesyes", "no")** returns 0

INDEXC

returns an integer that indicates the first position in an input string that contains any character in a specified set of characters. The INDEXC function takes two arguments:

- The first argument is the input string.
- The second argument is a string that represents the set of characters to search for.

If none of the specified characters is in the input string, then the INDEXC function returns 0. For example:

- **INDEXC("education", "aeiou")** returns 1
- **INDEXC("school", "aeiou")** returns 4
- **INDEXC("jklmn", "aeiou")** returns 0

ISCURRENT

returns a Boolean value that indicates whether a specified member is in the crossing where the function is evaluated.

The ISCURRENT function takes two arguments: a dimension type code and the code of a member that belongs to the dimension type. The ISCURRENT function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF (ISCURRENT ("TIME", "JAN2007"), 100, 0)
```

ISLEAF

returns a Boolean value that indicates whether the member of a specified dimension type that is in the crossing where the function is evaluated is a leaf member.

The ISLEAF function takes one argument: a dimension type code. The ISLEAF function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF (ISLEAF ("TIME"), 100, 0)
```

Note: A virtual child member counts as a leaf member.

ISLEAFCROSSING

returns a Boolean value that indicates whether the crossing where the function is evaluated consists entirely of leaf members.

The ISLEAFCROSSING function does not take an argument. The ISLEAFCROSSING function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF (ISLEAFCROSSING(), 100, 0)
```

Note: A virtual child member counts as a leaf member.

ISVIRTUALCHILD

returns a Boolean value that indicates whether the member of a specified dimension type that is in the crossing where the function is evaluated is a virtual child member.

The ISVIRTUALCHILD function takes one argument: a dimension type code. The ISVIRTUALCHILD function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF (ISVIRTUALCHILD ("INTORG"), 100, 0)
```

LASTCHILD

returns the code of the member that is the last-listed hierarchical child of the member that is specified either explicitly or implicitly by the argument of the function.

The LASTCHILD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"] ["TIME"=LASTCHILD ("TIME")]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the last quarter of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the last month of that quarter.

Instead of a dimension type code, the argument of LASTCHILD can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=LASTCHILD (NEXT ( ( "TIME" ) ) ) ]
```

If the LASTCHILD function is applied to a member that has one child, then it returns the code of that one child.

If the LASTCHILD function is applied to a leaf member, then the operand that the function is part of receives a value of 0.

LASTSIBLING

returns the code of the member that is the last-listed hierarchical sibling of the member that is specified either explicitly or implicitly by the argument of the function.

The LASTSIBLING function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=LASTSIBLING ( "TIME" ) ]
```

When evaluated at a January, February, or March crossing, this reference retrieves the value of the Headcount account for March. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the last quarter of the same year.

Instead of a dimension type code, the argument of LASTSIBLING can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=LASTSIBLING (PARENT ( ( "TIME" ) ) ) ]
```

If the LASTSIBLING function is applied to an only child, then it returns the code of that one child.

LEFT

returns a string that is formed by removing all the leading spaces from an input string. It takes one argument, which must have a character value. For example, **LEFT (" abc")** returns the character string **abc**.

LENGTH

returns the length of an input string. It takes one argument, which must have a character value. For example, **LENGTH ("January")** returns 7.

LOG

computes the natural logarithm of a number. It takes one argument, which must have a numeric value greater than zero.

LOG is the inverse of EXP. For example, **LOG (EXP (2.65))** returns 2.65.

LOWCASE

returns a character string that is formed from an input character string by converting each uppercase letter to the corresponding lowercase letter. This function takes one argument, which must have a character value. For example, **LOWCASE ("HIGH5")** returns **high5**.

MAX

returns the value of the argument that has the largest value. The MAX function can have any number of arguments. Each argument must be numeric. For example, **MAX (-5, -2.1, 0.3, 1.1, 1.3)** returns 1.3.

MEAN

returns the mean of the values of all its arguments. The MEAN function can have any number of arguments. Each argument must be numeric. The mean is the sum of all the values divided by the number of values. For example, **MEAN(1, 2, 3, 3, 16)** returns 5.

MIN

returns the value of the argument that has the smallest value. The MIN function can have multiple arguments. Each argument must be numeric. For example, **MIN(-5, -2.1, 0.3, 1.1, 1.3)** returns -5.

MOD

returns the remainder when one number is divided by another number. The MOD function takes two arguments. The first argument is the dividend. The second argument is the divisor. Both arguments must have numeric values. The returned remainder always has the same sign as the dividend. For example:

- **MOD(20.3, 6.2)** returns 1.7
- **MOD(-20.3, 6.2)** returns -1.7
- **MOD(20.3, -6.2)** returns 1.7
- **MOD(-20.3, -6.2)** returns -1.7

NESTIF

returns a value that depends on the truth values of one or more Boolean expressions. The NESTIF function takes an even number of arguments, which are arranged in pairs. The second member of each pair is an expression whose value might be returned. These are the even-numbered arguments of the function. The first member of each pair is a Boolean expression that is associated with the second member of the pair. These are the odd-numbered arguments of the function.

A Boolean expression can compare two character values or two numeric values. Within the Boolean expressions, you can use any of the Boolean operators and comparison operators that are available on the symbol toolbar of the Formula Editor window. You can also use the ISCURRENT, ISLEAF, and ISLEAFCROSSING functions, which return Boolean values.

The function returns the value of the first even-numbered argument that is associated with a true Boolean expression. For example:

- **NESTIF(1=2, "first", 1=1, "second")** returns **second**
- **NESTIF(1=1, "first", 1=1, "second")** returns **first**

There is no limit to the number of arguments that the NESTIF function can take. The number must be even, however.

All the even-numbered arguments must be of the same data type. They must all yield numeric values, or they must all yield character-string values, or they must all yield Boolean values.

If all the Boolean expressions in the odd-numbered arguments are false, then the NESTIF function returns a default value that depends, in the following way, on the data type of the even-numbered arguments:

- Numeric data type: SAS missing value
- Character-string data type: zero-length empty string
- Boolean data type: Boolean **FALSE**

NEXT

returns the code of the member that is at the same hierarchical level, and immediately after, the member that is specified either explicitly or implicitly by the argument of the function.

The NEXT function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

The NEXT function is most often used as part of a relative reference within a time hierarchy. For example, the following reference, when evaluated for a certain month, retrieves the value of the Headcount account for the next month:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=NEXT ( "TIME" ) ]
```

Instead of a dimension type code, the argument of NEXT can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=NEXT ( PARENT ( "TIME" ) ) ]
```

If the NEXT function is applied to a member for which there is no next member, then the operand that the function is part of receives a value of 0.

OPENINGPERIOD

returns the code of the member that is the first-listed leaf member that is hierarchically subordinate to the member that is specified either explicitly or implicitly by the argument of the function. If the member specified by the argument is itself a leaf member, then the OPENINGPERIOD function returns the code of that member.

The OPENINGPERIOD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

The OPENINGPERIOD function can be used with any hierarchy, but it is primarily intended for use with time hierarchies.

Consider a year-quarter-month time hierarchy and the following reference:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=OPENINGPERIOD ( "TIME" ) ]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the first month of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the first month of that quarter. When evaluated at a month crossing, this reference retrieves the value of the Headcount account for the month in that crossing.

Instead of a dimension type code, the argument of OPENINGPERIOD can be a function that returns a member code, as in the following example:

In this case, when the formula is evaluated at a year, quarter, or month, the value of the Headcount account is for the first month of that year.

```
["ACCOUNT"="Headcount"]
["TIME"=OPENINGPERIOD(ANCESTOR("TIME","Year"))]
```

PARENT

returns the code of the member that is the hierarchical parent of the member that is specified either explicitly or implicitly by the argument of the function.

The PARENT function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"] ["TIME"=PARENT("TIME")]
```

When evaluated at a month crossing, this reference retrieves the value of the Headcount account for the quarter that contains that month. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the year that contains that quarter.

Instead of a dimension type code, the argument of PARENT can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="Headcount"] ["TIME"=PARENT(NEXT(("TIME"))]
```

If the PARENT function is applied to a member that has no parent, then the operand that the function is part of receives a value of 0.

PI

returns the value of pi (3.14159...) to as many decimal places as your computer supports. This function does not take an argument. For example, **COS(PI()/3)** returns 0.5.

POW

returns the result of raising its first argument to the power given by its second argument. It takes two arguments. Both arguments must have numeric values. For example:

- **POW(2,4)** returns 16.
- **POW(9,0.5)** returns 3.
- **POW(3,-2)** returns 1/9 or 0.11111...

PREVIOUS

returns the code of the member that is at the same hierarchical level, and immediately before, the member that is specified either explicitly or implicitly by the argument of the function.

The PREVIOUS function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

The PREVIOUS function is most often used as part of a relative reference within a time hierarchy. For example, the following reference, when evaluated for a certain month, retrieves the value of the Headcount account for the previous month:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=PREVIOUS ( "TIME" ) ]
```

Instead of a dimension type code, the argument of PREVIOUS can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="Headcount" ] [ "TIME"=PREVIOUS ( FIRSTSIBLING ( "TIME" ) ) ]
```

If the PREVIOUS function is applied to a member for which there is no previous member, then the operand that the function is part of receives a value of 0.

PROBBETA

returns the probability that an observation from a beta distribution, with shape parameters a and b , is less than or equal to x .

```
PROBBETA (  $x$ ,  $a$ ,  $b$  )
```

x is a numeric random variable in the range $0 \leq x \leq 1$.

a is a numeric shape parameter in the range $a > 0$.

b is a numeric shape parameter in the range $b > 0$.

For example, **PROBBETA** (. 2 , 3 , 4) returns a value of 0 . 09888 .

PROBIT

returns the p th quantile from the standard normal distribution. The probability that an observation from the standard normal distribution is less than or equal to the returned quantile is p . The syntax is :

```
PROBIT (  $p$  )
```

p is a numeric probability in the range $0 < p < 1$.

For example, **PROBIT** (. 025) returns a value of -1 . 959963985 .

Note: The result could be truncated to lie between -8.222 and 7.941.

Note: PROBIT is the inverse of the PROBNORM function.

PROBNORM

returns the probability that an observation from the standard normal distribution is less than or equal to x . The syntax is:

```
PROBNORM (  $x$  )
```

x is a numeric random variable.

For example, **PROBNORM** (1 . 96) returns a value of 0 . 9750021049 .

Note: PROBNORM is the inverse of the PROBIT function.

PROPERTY

returns the value of a specified property of the member that satisfies these two conditions:

- It belongs to the dimension type that is specified in the function.
- It is in the crossing where the function is evaluated.

The PROPERTY function takes two arguments: a dimension type code and a property code. The property code is case-sensitive.

For example, the following use of the PROPERTY function returns the account type of the account that is in the crossing where the function is evaluated:

```
PROPERTY ("ACCOUNT", "AccountType")
```

Here are some property codes that you can use in this function and the values that the function can return for each of them:

- **AccountBehavior**: the category to which the account type belongs. This property is valid only if the member is an account. The following values can be returned: **Balance**, **CTA**, **Flow**, **Hybrid**, and **NonFrequency**. (**NonFrequency** is another name for the Statistical account type. **Hybrid** is another name for the RetainedEarnings account type.)
- **AccountType**: the account type of an account. This property is valid only if the member is an account. The following values can be returned: **Asset**, **Liability**, **Equity**, **Revenue**, **Expense**, **RetainedEarnings**, **CTA**, **StatisticalBalance**, **StatisticalFlow**, **NonFrequency** (another name for the Statistical account type).
- **BalanceType**: the balance type of an account. This property is valid only if the member is an account. The following values can be returned: **Credit**, **Debit**
- **ExchangeRateType**: the exchange rate type of an account. This property is valid only if the member is an account. The following values can be returned: **PeriodAverage**, **PeriodClose**, **PeriodOpen**, **Custom1**, **Custom2**, **Derived**, **Historic**.
- **FunctionalCurrency**: the functional currency of an organization. This property is valid only if the member is an organization. Any currency code can be returned.
- **Intercompany**: whether an account is an intercompany account. This property is valid only if the member is an account. The following values can be returned: **True**, **False**.
- **Level**: the period type of a time period. This property is valid only if the member is a time period. The following values can be returned: **AllYears**, **Year**, **HalfYear**, **QuarterYear**, **Month**, **Week**, **Day**.
- **ReportingEntity**: whether an organization is a reporting entity. This property is valid only if the member is an organization. The following values can be returned: **True**, **False**.

REPEAT

returns a string that is formed from an input string by appending the input string to itself a specified number of times. The REPEAT function takes two arguments. The first argument is the input string. The second argument is the number of repetitions, beyond the first occurrence. For example:

- **REPEAT ("no", 1)** returns **nono**
- **REPEAT ("ha", 2)** returns **hahaha**

REVERSE

returns a string that is formed by reversing the character sequence of an input string. It takes one argument, which must have a character value. For example, **REVERSE ("nuts")** returns **stun**.

RIGHT

returns a string that is formed by removing all the trailing spaces from an input string. It takes one argument, which must have a character value. For example, **RIGHT ("abc ")** returns the character string **abc**.

ROUND

returns the result of rounding its first argument to the number of decimal places given by its second argument. The first argument must have a numeric value. The value of the second argument must be a nonnegative integer. The second argument is optional. Omitting the second argument is equivalent to giving the second argument the value 0: the result is rounded to the nearest integer.

If the most significant digit that is dropped is 5, then the least significant digit that is not dropped is increased by 1. In other words, the absolute value increases.

For example:

- **ROUND (2.425, 2)** returns **2.43**
- **ROUND (2.425, 1)** returns **2.4**
- **ROUND (2.425, 0)** returns **2**
- **ROUND (2.425)** returns **2**
- **ROUND (-2.425, 2)** returns **-2.43**
- **ROUND (-2.425, 1)** returns **-2.4**
- **ROUND (-2.425, 0)** returns **-2**
- **ROUND (-2.425)** returns **-2**

SCAN

returns a requested segment of a segmented input string. The following characters are treated as delimiters that divide the input string into countable segments:

`< (+&!$*); ^-/, %|`

The SCAN function takes two arguments:

- The first argument is the input string.
- The second argument is the numeric position of the requested segment. Positions are counted from the beginning of the input string if the second argument is positive, and from the end of the input string if the second argument is negative.

Here are two examples:

- **SCAN ("12+34-56+78", 4)** returns **78**
- **SCAN ("The%quick%brown%fox%jumped", -2)** returns **fox**

SIN

returns the sine of its only argument. The argument must have a numeric value, which represents an angle measure in radians.

SQRT

returns the square root of its only argument. The argument must have a nonnegative numeric value.

STD

returns the standard deviation of a set of arguments. The syntax is:

`STD (argument-1, argument-2 [, ... argument-n])`

Each argument specifies a numeric constant, variable, or expression. At least two arguments that do not contain invalid data are required.

For example, **STD (2, 6)** returns a value of **2.8284271247**.

SUBSTR

returns a substring of an input string. The SUBSTR function can take either two or three arguments:

- The first argument is the input string.
- The second argument is the numeric position in the input string of the first character of the requested substring.
- If there is a third argument, then it specifies the length of the requested substring. If there is no third argument, then the returned substring runs to the end of the input string.

Here are two examples:

- **SUBSTR("reiterate",3,2)** returns **it**
- **SUBSTR("reiterate",3)** returns **iterate**

SUM

returns the sum of its arguments. The SUM function can take any number of arguments. All the arguments must have numeric values. For example, **SUM(1, 2, 3, 3, 16)** returns **25**.

You can also use a colon (:) to specify a range of crossings, as in the following examples:

```
SUM ( ["TIME"="JAN2005"] : ["TIME"="JUN2005"] )
```

```
SUM ( ["ACCOUNT"="Interest"] ["TIME"="JAN2005"] : ["ACCOUNT"="Interest"]  
["TIME"="JUN2005"] )
```

In such a range specification, the first and last crossings must be at the same level in the hierarchy of variation. The specified range includes crossings at only that one level, so as to avoid double counting. Each example above specifies a six-location range at the month level of a time hierarchy.

TABLEC

returns a character value that it retrieves from a specified table. Ask your SAS consultant for details.

TABLEN

returns a numeric value that it retrieves from a specified table. Ask your SAS consultant for details.

TAN

returns the tangent of its only argument. The argument must have a numeric value, which represents an angle measure in radians.

TIME

returns the current clock time in military format, to the nearest second. For example, if TIME is executed at exactly noon, it returns **12:00:00**. The syntax is **TIME()**.

TODAY

returns the SAS integer representation of the current date. For example:

- January 1, 1960 is **1**
- January 2, 1960 is **2**
- February 1, 1960 is **32**

This function does not take an argument. **DATE()** and **TODAY()** are equivalent.

TODEGREES

converts an angle measure in radians to degrees. The argument must have a numeric value. This value represents an angle measure in radians. Because all the trigonometric functions return a result in radians, you must use this function to find the angle in degrees whose tangent, sine, or cosine has a given value. For example, **TODEGREES (ASIN (0.5))** returns 30.

TORADIANS

converts an angle measure in degrees to radians. The argument must have a numeric value. This value represents an angle measure in degrees. Because all the trigonometric functions assume that the argument is in radians, you must use this function in order to apply a trigonometric function to an angle measure in degrees. For example, **SIN (TORADIANS (30))** returns 0.5.

TRANSLATE

returns a character string that is formed by replacing certain characters in an input string with designated substitution characters. The TRANSLATE function normally takes three arguments:

- The first argument is the input string that undergoes translation.
- The second argument lists one or more substitution characters.
- The third argument lists the characters to replace, in a sequence that corresponds to the sequence of substitution characters in the second argument.

For example, **TRANSLATE ("bone", "iw", "ob")** returns **wine**.

In a variant syntax, you can spread the translation instructions over additional pairs of arguments. For example, **TRANSLATE ("bone", "i", "o", "w", "b")** also returns **wine**.

TRIM

returns a character string that is formed by removing all the leading spaces and all the trailing spaces from an input character string. The TRIM function takes one argument, which must have a character value. For example, **TRIM (" abc ")** returns the character string **abc**.

UPCASE

returns a character string that is formed from an input character string by converting each lowercase letter to the corresponding uppercase letter. This function takes one argument, which must have a character value. For example, **UPCASE ("they8it")** returns **THEY8IT**.

VAR

returns the variance of a set of arguments

```
VAR (argument-1, argument-2 [, ... argument-n])
```

Each argument contains a numeric constant, variable, or expression. At least two arguments that do not contain invalid data are required.

For example, **var (4, 2, 3.5, 6)** returns a value of **2.7291666667**.

VERIFY

checks whether one or more characters are present in one or more strings. The first argument is a list of characters to check for. Each subsequent argument is a string to check. If every character in the first argument is found in at least one subsequent argument, then VERIFY returns the numeric value 0. Otherwise, VERIFY returns the numeric position of the first character that is not found in any subsequent string. For example:

- **VERIFY("aeiou", "state", "union")** returns 0 because each vowel is in either “state” or “union.”
- **VERIFY("aeiou", "state", "onion")** returns 5 because “u” is not in “state” or “onion.”

VIRTUALCHILD

returns a reference to the virtual child of a specified non-leaf member. If a leaf member is specified as input, then VIRTUALCHILD returns a reference to that same leaf member.

The VIRTUALCHILD function takes one argument, which is the member code of the member whose virtual child you want to refer to. The specified member code must not belong to any of the following dimension types, in which there are no virtual children:

- Analysis
- Currency
- Frequency
- Time
- Source

The VIRTUALCHILD function is equivalent to the .vc suffix. For example, each of the following refers to the virtual child of the CURRLIAB member of a model’s account hierarchy:

```
[ "ACCOUNT"=VIRTUALCHILD ( "CURRLIAB" ) ]
```

```
[ "ACCOUNT"="CURRLIAB.vc" ]
```

The .vc suffix must be lowercase.

Instead of a literal member code, the argument of VIRTUALCHILD can be a function that returns a member code, as in the following example:

```
[ "ACCOUNT"="UnitsSold" ] * -1 * [ "ACCOUNT"="Price" ]  
[ "PRODUCT"=VIRTUALCHILD ( PARENT ( CURRENT ( "PRODUCT" ) ) ) ]
```

XRATE

returns a specified exchange rate.

The XRATE function takes two arguments: an exchange rate type code and the code of the target currency. It gets the other information that it needs from the crossing where it is evaluated.

For any exchange rate type, the XRATE function uses the following members of the crossing where it is evaluated:

- The analysis member determines which exchange rate set to use.
- The time member determines the relevant time period.
- The currency member determines the source currency.

For a complex exchange rate that depends on the members of certain other dimension types, the XRATE function also uses the members of those other dimension types in the crossing where the function is evaluated.

The first argument must be a quoted string that contains one of the following exchange rate type codes: **PeriodAverage**, **PeriodClose**, **PeriodOpen**, **Custom1**, **Custom2**, **Derived**, or **Historic**.

For example, to retrieve the Period Average exchange rate for converting the value at the crossing where the function is evaluated to a value in euros, use the following:

```
XRATE("PeriodAverage", "EUR")
```


Chapter 17

Forms and the Forms Workspace

Forms and Form Sets	153
Workflow Types	154
Overview	154
Bottom-Up Workflows	154
Top-Down Workflows	154
Form Set Summary	154
Introduction to the Forms Workspace	155
Assigning Authors to Forms	156
Assigning Reviewers to Forms	156
Status Values of Form Sets	157
Status Values of Forms	157
Form Sets View	158
Overview of the Form Sets View	158
Phase Options	158
Form Set Options	159
Forms View	163
Overview of the Forms View	163
Form Options in SAS Financial Management Studio	163

Forms and Form Sets

A *form set* is a group of forms that are typically used for data entry via SAS Financial Management Add-In for Microsoft Excel. Each form set includes a workflow process for submitting, approving, rejecting, and recalling forms.

When you create a form set, you select a target hierarchy that determines the members to be included in the workflow.

Workflow Types

Overview

SAS Financial Management supports the following two types of workflows:

- bottom-up workflows
- top-down workflows

Each workflow type has a predefined set of states through which a form advances. You can customize a workflow by writing a *stored process*. A stored process is a SAS program that executes before or after the form is advanced to the next state in the workflow. For details, see “Customizing a Workflow” in the *SAS Financial Management: Customization Guide*.

Bottom-Up Workflows

In a bottom-up workflow, users enter data for leaf members of the target hierarchy and then submit it to higher levels for review. Each review can result in either approval or rejection.

Forms that have been submitted but not yet approved can also be recalled by the author.

Note: Data that has not been approved is not visible in parent forms.

Top-Down Workflows

In a top-down workflow, an initial form author enters data at the highest level of the form set’s target hierarchy, using the virtual child (VC) member. The form author allocates these amounts to subordinate members of the target hierarchy and pushes the form to the next level in the workflow. Subsequent authors determine the amount to allocate to their children and descendants, if applicable.

Form Set Summary

The following table summarizes the capabilities of a form set.

Table 17.1 Form Set Capabilities

Feature	Description
Form template	Can include data-entry tables, read-only tables, CDA tables, and supplemental schedules.
Use of models	Can contain multiple models that share the same cycle.

Feature	Description
As-of date	The as-of date and time are specified in the form set properties for the target hierarchy. This date can be the current date or an earlier date.
Target hierarchy	Can be an organization hierarchy, an account hierarchy, or a hierarchy from any non-required dimension type in the underlying cycle. By default, there is one form for each selected member of the target hierarchy in the workflow. However, a form can include writable descendants.
Non-target hierarchies	Every form has access to all crossings as defined by the tables in the template, with consideration for security and the target hierarchy.
Form authors and reviewers	A bottom-up form can have multiple authors and reviewers. A top-down form can have multiple authors.
Formula types	Driver, modeling, reporting, and Excel-based calculated members are supported.
Form data entry	Users can edit forms in Microsoft Excel depending on the form set properties.

Introduction to the Forms Workspace

In the Forms workspace, you can create and manage form sets that belong to the open cycle.

A *form set* is a set of similarly structured data-entry forms that enable users to enter data into the cycle associated with the form set. The structure of the data-entry forms is defined by a template that you create in Microsoft Excel.

The Forms workspace contains the following three views:

- **Form Sets view**—opens when you select the Forms workspace. The Form Sets view lists all of the form sets in the open cycle and the phases that contain the form sets. *Phases* are grouping devices, like folders, in which you can group form sets.
- **Forms view**—opens when you select the **Forms** option for a selected form set. The Forms view displays all of the forms that belong to the selected form set.
- **Attachments view**—opens when you select the **Attachments** option for a selected form set. The Attachments view displays all of the attachments of the selected form set.

To create a phase in the Form Sets view, select **New Phase**.

To create a form set, select a phase and then select **New Form Set** to launch the New Form Set wizard.

Assigning Authors to Forms

Each form in a form set can have one or more authors. *Authors* are assigned to forms automatically if any users with the required capabilities are associated with the form's target member. You can also assign authors in the Form Sets view and in the Forms view.

To assign an author in the Form Sets view:

1. Select a form set.
2. Select **Authors**.

The Authors window for the selected form set is displayed.

3. In the Authors window, view and assign authors for any form in the selected form set.

To assign an author in the Forms view:

1. Select a form.
2. Select **Properties**.

The Properties window for the selected form is displayed.

3. In the **Authors** page of the Properties window, view or assign authors.

Assigning Reviewers to Forms

Each form in a bottom-up form set can have one or more reviewers who can approve or reject the data that a form author has entered into that form. You can assign reviewers in the Form Sets view and in the Forms view.

Note: Each assigned reviewer must have the required capabilities.

By default, any assigned reviewer can approve a form. You have the option of defining a review sequence that consists of multiple levels, with one or more reviewers for each level.

To assign reviewers in the Form Sets view:

1. Select a form set.
2. Select **Reviewers**.

The Reviewers window for the selected form set is displayed.

3. In the Reviewers window, view or modify the review process for any form in the selected form set.

To assign reviewers in the Forms view:

1. Select a form.
2. Select **Properties**.

The Properties window for the selected form is displayed.

3. In the **Reviewers** page of the Properties window, define the review process for the selected form.

Status Values of Form Sets

A form set has a life cycle that takes it through a series of status values. The following life cycle status values are possible:

No template

The form set has been created, but no data-entry template has been created for it yet.

Note: “No template” is displayed for a form set with a brand new template. The status value changes to “Draft” when you select **Refresh** or when you exit from SAS Financial Management Studio and then log on again.

Draft

The form set has been created and a data-entry template has been created for it. The form set can be published at any time.

Published

The forms in the form set are available in the SAS Financial Management web application.

Completed

The forms in the form set are no longer available in the SAS Financial Management web application.

In addition, every form set has a current lock state—either locked or unlocked. The status value and lock state of a form set help determine which options are available for it in the Form Sets view.

Status Values of Forms

For forms that are not available to be worked on, the following status values are possible:

Approved

The form belongs to a bottom-up form set whose status is “Completed”.

Completed

The form belongs to a top-down form set whose status is “Completed”.

Draft

The form belongs to a form set whose status is “Draft”.

Holding

The form belongs to a top-down form set whose status is “Published”. However, the form is not yet available for editing because data has not been pushed down to it from a higher level of the target hierarchy.

No template

The form belongs to a form set whose status is “No template”.

In addition, every form has a current lock state—either locked or unlocked. The status value and lock state of a form help determine which options are available for it in the Forms view.

Form Sets View

Overview of the Form Sets View

The **Form Sets** view lists form sets inside phases. *Phases* are grouping devices, like folders, in which you can group form sets.

Use the plus and minus signs next to a phase to expand and collapse the phases in the display.

To search for a particular phase or form set, click the **Find** button .

Note: One set of options is available when a phase is selected. Another set of options is available when a form set is selected.

Phase Options

When a phase is selected, the following options are available:

New Phase

launches the New Phase wizard, which enables you to create a new phase.

For details, see the online Help for the individual wizard pages.

New Form Set

launches the New Form Set wizard, which enables you to create a new form set in the selected phase.

For details, see the online Help for the individual wizard pages.

You can also create a new form set by selecting an existing form set and then selecting **Copy**. This method can be convenient if the new form set resembles the existing form set.

Import Form Sets

launches the Import Form Sets wizard, which enables you to import form sets into the selected phase from a designated SAS package file.

The **Export Form Sets** and **Import Form Sets** options are useful for promoting form sets from one system to another (for example, from a test system to a production system). The imported form sets must use the same cycle and model as the exported form sets.

For details, see the online Help for the individual wizard pages.

Delete

deletes the selected phase.

Note: This option is available only if the phase contains no form sets.

Reorder Phases

displays the Reorder window, which enables you to change the order of the phases in the Form Sets view.

Reorder Form Sets

displays the Reorder window, which enables you to change the order of the form sets in the selected phase.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Properties

displays the Properties window for the selected phase, enabling you to view and modify its properties.

Refresh

refreshes the view.

Form Set Options

When a form set is selected, the following options are available:

New Phase

launches the New Phase wizard, which enables you to create a new phase.

For details, see the online Help for the individual wizard pages.

New Form Set

launches the New Form Set wizard, which enables you to create a new form set in the same phase as the selected form set.

For details, see the online Help for the individual wizard pages.

Note: To refresh the form set status, click the **Refresh** button .

You can also create a new form set by selecting an existing form set and then selecting the **Copy** option. This method can be convenient if the new form set resembles the existing form set.

Import Form Sets

launches the Import Form Sets wizard, which enables you to import form sets into the same phase as the selected form set from a designated SAS package file.

The **Export Form Sets** and **Import Form Sets** options are useful for promoting form sets from one system to another (for example, from a test system to a production system). The imported form sets must use the same cycle and model as the exported form sets.

For details, see the online Help for the individual wizard pages.

Export Form Sets

launches the Export Form Sets wizard, which enables you to export the selected form sets to a SAS package file at a designated location.

For details, see the online Help for the individual wizard pages.

Send Notification

displays the Send Notification window, which enables you to send a message to authors and reviewers of the forms in the selected form set.

Note: This option is also available when you select a single form or any set of forms. When forms are selected, it enables you to send a message to authors and reviewers of those selected forms.

Lock

locks the selected form sets that are not currently locked. When a form set is locked, none of the forms in the form set can be used, and certain options do not apply to it.

Note: This option is available only if at least one unlocked form set is selected.

Unlock

unlocks the selected form sets that are currently locked.

Note: This option is available only if at least one unlocked form set is selected.

Delete

deletes the selected form sets.

When you select this option, a confirmation window appears that contains these radio buttons:

Delete existing form data and comments

Deletes the selected form sets and any data that was entered through the forms in the form set (regardless of any locks on time and analysis members). Any cell comments that were entered in the forms are also deleted.

Preserve existing form data but delete comments

Deletes the selected form sets, but retains any data that was entered through the forms in the form set. Any cell comments that were entered in the forms are deleted.

Preserve existing form data and comments visible outside this form set

Deletes the selected form sets, but retains any data that was entered through the forms in the form sets. Any cell comments that were visible only in the form set are deleted. Cell comments that were visible outside the form set are retained.

Select the appropriate radio button, and then click **Yes**.

Copy

launches the Copy Form Set wizard, which enables you to create a form set as a copy of the selected form set.

For details, see the online Help for the individual wizard pages.

This is a flexible copy operation. The wizard enables you to make the new form set differ from the original form set in a variety of ways.

You can copy the form set to the same cycle or to a different cycle.

Note: Data and comments are not included in the copy. If the new form set is in the same cycle as the original, both form sets are associated with the same data and comments. If the new form set is in a different cycle, the form set is associated with that cycle's data and comments, if any. (Comments are available only if the form set properties permit them.)

If the form set that you are copying has a template, then a copy of the template is created in Draft status. The new template opens automatically when you complete the wizard, enabling you to edit it. The tables in the new template might or might not be based on the models that you want. You should always check all table-model associations in a template that is created by copying a form set.

Move

displays the Move window, which enables you to move the selected form set to a different phase in the same cycle.

Note: This option is available only if the Form Sets view contains two or more phases.

Reorder Phases

displays the Reorder window, which enables you to change the order of the phases in the Form Sets view.

Note: This option is available only if the Form Sets view contains two or more phases.

Reorder Form Sets

displays the Reorder window, which enables you to change the order of the form sets in the phase that contains the selected form set.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Template

enables you to create a template for a selected form set that has no template, or opens the template of a selected form set that has a template. Your ability to edit the template, or only view it, depends on the status of the form set.

Note: The status value of a form set with a brand new template is **No template**. The status value changes to **Draft** when you select **Refresh** or when you exit from SAS Financial Management Studio and then log on again.

Forms

displays the Forms view for the selected form set.

The Forms view shows the form set's hierarchy of forms and enables you to work with the individual forms.

Authors

displays an Authors window, which enables you to assign different authors to any form in the form set.

Reviewers

displays a Reviewers window, which enables you to assign different reviewers to any form in the form set.

Note: This option is available only for form sets that have a bottom-up workflow.

Data Validation Rules

displays a window in which you can define constraints for the values in data-entry forms for this form set only.

Note: This option is available only for form sets that have a bottom-up workflow.

You can also export or load data validation rules by using SAS Data Integration Studio jobs. For detailed information, see the *SAS Financial Management: Data Administrator's Guide*.

Attachments

opens the Attachments view for the selected form set, which enables you to attach documents to the form set and review its current attachments.

Documents that are attached to a form set in SAS Financial Management Studio become attachments of each form in the form set in the SAS Financial Management web application. These attachments are typically used to communicate instructions to the users who must enter data into the forms.

Properties

displays the Properties window for the selected form set, which enables you to view and modify properties.

Which properties you can modify depends on the status of the form set. A form set with the status of Draft is more open to modification than a form set with the status of Published.

Run driver formulas

computes and stores driver formula output values for all crossings that are included in the template for the selected form set. Formula validation is performed first, and a warning message lists any errors that are encountered. After reviewing any errors, you can then decide whether to continue with Run driver formulas.

It is important to select this option after any event (other than Excel data entry) that affects the computed values of a form set's driver formulas. Examples of such events include changes to loaded cycle data, changes to driver rates, changes to adjustments, and changes to the driver formulas themselves.

Note: This option is not needed when users enter new values in forms, because that event triggers the recomputation of all affected driver formula output values in the form view.

Note: This option is not available for locked form sets or form sets that have no template. It is also not available when the target hierarchy for the form set is from the Account dimension.

Publish

makes the forms in the selected form set available for use in the SAS Financial Management web application. Ensure that you complete work on a form set's form template before you publish the form set.

Every published form must have at least one assigned author. In a form set with a bottom-up workflow, assigned reviewers are also required. If you attempt to publish a form set that does not meet the requirements for authors or reviewers, a warning message is displayed.

Beginning with SAS Financial Management 5.62, SAS administrators should perform the following steps to publish the form sets without an exception:

- Navigate to <https://host:port/SASAdmin>.
- Navigate to the **Authorization** workspace.
- Under **Assign Roles**, perform these steps:
 - From the step 1 **Select an item**, select **Users**.
 - From the step 2 **You must select an identity**, select a user name.
 - From the step 3 **Role**, select **ROLE_ADMIN**.

Note: The **Publish** option is available only for form sets whose status is Draft; it changes the status from Draft to Published. If you reset and then publish a form set, any history and form comments that were previously associated with the forms are no longer available in the SAS Financial Management web application. Form users can view history and form comments that occurred after the republish, but not before the republish. However, the full history and form comments are still available in SAS Financial Management Studio. (This note does not apply to cell comments. Cell comments are not affected by a reset.)

Reset

returns a published and unlocked form set to Draft status.

Complete

removes all forms in the form set from the SAS Financial Management web application.

Note: The **Complete** option is available only for unlocked form sets whose status is Published; it changes the status from Published to Completed. Use this option only after the completion of all Microsoft Excel data entry, review, and approval for the entire form set.

Reactivate

makes the forms in the selected form set available for use again, by changing the form set status back to Published.

Note: The **Reactivate** option is available only for unlocked form sets whose status is Completed.

Find

displays or hides the Find window. Enter a character string and click **Find Next**. Partial matches are accepted.

To further restrict the search, click the **More Options** and make one or more selections.

Refresh

refreshes the view.

Forms View

Overview of the Forms View

The Forms view displays all the forms in a selected form set, select a form set and select **Forms**.

When working in the Forms view, note the following:

- Click the **List View** button  to toggle between a hierarchical display of forms and a simple list of forms.
- In the list display, sort the list by clicking a column heading.
- In the hierarchical display, use the plus and minus signs to expand and collapse portions of the hierarchy. Alternatively, you can use the **Expand all** and **Collapse all** options to expand or collapse the entire hierarchy.
- To search for a particular form, click the **Find** button .

Note: A pencil icon  to the left of a form name indicates that a user modified the form by using the SAS Financial Management Add-in for Microsoft Excel and saved it by using the **Save Form Design** option. If you republish the form set, then all forms are overwritten and these changes are lost.

Form Options in SAS Financial Management Studio

End users open forms from the SAS Financial Management web application. If you are a form administrator or a process administrator, you can use many of the same options in SAS Financial Management Studio. You can also use some additional options that are not available in the SAS Financial Management web application.

The following form options are available in SAS Financial Management Studio:

Approve (bottom-up workflows only)

moves a form's data up into its parent form. A form might require one approval by one reviewer or a sequence of approvals by two or more reviewers. (The review order is defined in the form properties.)

If only one approval is required, then approving a form changes its status from Submitted to Approved.

If a sequence of approvals is required, then the form's data moves up into its parent form after the last required approval. The approvals affect the form's status in the following way:

- The first approval changes the form's status from Submitted to Partially Approved.
- An intermediate approval in a long sequence of approvals leaves the status of Partially Approved unchanged.
- The last approval changes the form's status from Partially Approved to Approved.

Lock

locks the selected forms that are not currently locked. The icon for a locked form includes a padlock. When a form is locked, it cannot be used.

Open in Excel

opens a form in Microsoft Excel. The status of the form determines whether the form is writable or read-only.

The first time someone opens a form, the form's status changes from Unedited to Edited. Any form that has the status of Edited can be edited again without a change of status.

Bottom-up workflows: The first time someone opens a newly rejected form, the form's status changes from Rejected to Edited.

Top-down workflows: Below the top level, you cannot change the total monetary value that is allocated to the form that you are editing.

Properties

displays the Properties window for the selected form, enabling you to view and modify properties. The properties of a form include the authors and reviewers who are assigned to it.

Push (top-down workflows only)

copies the amounts that you have allocated to the form's children into the child forms and makes those child forms available for editing.

When you push a form, its status changes to Pushed and its child forms take on a status of Unedited.

Push to All (top-down workflows only)

copies the amounts that you have allocated to all the form's descendants into the descendant forms.

When you push a form to all its descendants, the status of the pushed form changes to Pushed, and the status of all the descendant forms changes to Completed.

Recall

Bottom-up workflows: reverses the effect of submitting a form. Only the user who submitted a form can recall it. Recalling a form changes its status from Submitted to Edited and makes it available for further editing.

Top-down workflows: reverses the effect of a previous Push or Push to All. Only the user who pushed a form can recall it. Recalling a form changes its status from Pushed to Edited and makes it available for further editing. It also removes from the forms list all the subordinate forms that had received data as a result of the previous Push or Push to All.

Reject (bottom-up workflows only)

reverses the effect of submitting a form. Only a reviewer of a form can reject it.

Rejecting a form changes its status from Submitted to Rejected and makes it available for further editing.

Submit (bottom-up workflows only)

makes a form available for review.

Submitting a form changes its status to Submitted.

Submitting a form that has unapproved children also changes the status of each unapproved child from Submitted to Approved.

Send Notification

displays the Send Notification window, in which you can send a message to authors and reviewers of the selected form or forms.

Note: This option is also available when you select a form set. It enables you to send a message to authors and reviewers of all the forms in the selected form set.

Undo Check-out

cancels a form's Checked Out status.

A user who is editing a form in Microsoft Excel can check out the form and edit it offline. A copy of the form is saved on the user's desktop. The status of the form becomes Checked Out, and no one else can edit the form until the user checks it back in.

If necessary, you can cancel the check-out from SAS Financial Management Studio. The form's status reverts to Edited. Any changes since the form was checked out are discarded.

Find

displays or hides the Find window. Enter a character string and click **Find Next**. Partial matches are accepted.

To further restrict the search, click the **More Options** and make one or more selections.

Customize Columns

displays the Customize Columns window. In this window, you can specify the columns to include in the view and the order in which to display them.

Refresh

refreshes the view.

Chapter 18

Task Management

Create a Task in SAS Financial Management Studio	167
Overview	167
SAS Financial Management Studio Task Types	168
Override Task Properties	169

Create a Task in SAS Financial Management Studio

Overview

Certain tasks must be created in SAS Financial Management Studio. You select an option, such as Load Model Data, and make all the selections that the wizard requires. However, instead of running the task at that moment, save it as a task that is associated with a process definition.

To create a task in SAS Financial Management Studio, complete the following steps:

1. Make sure the process definition is not being edited in the Processes workspace on the web. Otherwise, you cannot save a task for that process definition in SAS Financial Management Studio.
2. Select an option. (See the list below.)
3. On the Execution page, select **Save as a task in a process definition**.
4. Complete the steps in the wizard.

After you create the task, it is available in the selected process definition, at the end of the list of tasks. It has the same due date offset as the preceding task. You are listed as the task owner. In the process definition, you can modify those properties and assign a task schedule and notifications.

Note: Member selection rules are applied when the task is run, based on the hierarchy's as-of date in the model.

For some tasks, you can select different time and analysis members. See [“Override Task Properties” on page 169](#).

SAS Financial Management Studio Task Types

The following tasks can be created by wizards in SAS Financial Management Studio. Each requires an additional, task-related capability, such as Form Administration.

When you assign a task owner, only users with the required capabilities are available for selection. In an active process, another check is made when a user attempts to run the task.

The table lists the available task types, the corresponding option names in SAS Financial Management Studio, and the additional capabilities that are required.

Table 18.1 Task Types That Can Be Created in SAS Financial Management Studio

Task Type	Option	Description	Task-Related Capability
Delete cycle data	Delete Data	Delete data from the specified cycle.	Cycle Administration
Export model data	Export Data Records	Export data from the selected model to a SAS library.	Model Administration
Load dimension	Load Dimension	Load members and hierarchies from the SAS Financial Management staging area into the selected dimension.	Dimension Administration
Load driver rates	Load Driver Rates	Load driver rates from the SAS Financial Management staging area.	Rate Administration
Load exchange rates	Load Exchange Rates	Load both simple and complex exchange rates from the SAS Financial Management staging area.	Rate Administration
Load data to cycle	Load New Data	Load data into the specified cycle from the SAS Financial Management staging area.	Cycle Administration
Load model data to cycle	Load Model Data	Load data to a cycle, based on a model that is associated with the same cycle or a different cycle.	Cycle Administration
Post adjustments	Post Adjustments	Post adjustments for the selected model, time members, and analysis members.	Model Administration

Override Task Properties

When you create a task in SAS Financial Management Studio, you usually are asked to select time and analysis members as well as other dimension members. However, in an active process, you might not always want to use the same time and analysis members.

For example, you might create a **Load data to cycle** task in SAS Financial Management Studio, using the Load New Data wizard. Each time you run the process that contains this task, you want to load data to a different period of the same cycle.

Note: You cannot override the cycle that was selected in the wizard.

To override analysis members, you would select **Use process analysis members instead of task analysis members** in the task properties and select the appropriate analysis members in the process properties.

The following task types permit overrides:

- Delete cycle data
- Export model data
- Load data to cycle
- Load exchange rates (time members only)
- Post adjustments

Note: Depending on the task type, the Type page of the task definition might also display member selection rules for other dimensions. Those rules cannot be overridden. (If the **all members** option was selected for a dimension, the dimension is not listed.)

Chapter 19

Security in SAS Financial Management

About Security in SAS Financial Management	171
Object Security	172
Why Use Object Security?	172
Access Permissions for Cycles, Models, and Composite Models	172
Read, Update, and Delete Permissions	172
Denying Access to an Object	173
Superuser Access to Cycles and Models	173
Precedence Rules for Object Security	173
Data Security	174
Why Use Data Security?	174
Effects of Data Security	175
Precedence Rules for Data Security	175

About Security in SAS Financial Management

The security features in SAS Financial Management include the following:

- Metadata permissions and operating system permissions
 You can use metadata permissions and operating system permissions to define permission settings for folders and files.
 For information about metadata permissions, see “Authorization” in the *SAS Intelligence Platform: Security Administration Guide*. For information about operating system permissions, see “What to Do Next: Administration Tasks” in the *SAS Intelligence Platform: System Administration Guide*.
- Role-based security
 Role-based security determines a user's ability to perform various tasks, such as creating a dimension or editing a form. Roles are mapped to capabilities, which determine the operations that a user can perform.
 For information about groups and roles for SAS Financial Management, see “Assigning Groups and Roles” in the *SAS Financial Management: System Administrator's Guide*.
- Object security

Object security consists of permissions that you apply to cycles, dimension types, dimensions, hierarchies, models, composite models, and custom properties, to restrict users' access to those objects.

- Data security

Data security consists of permissions that you apply to members of one or more dimensions, to restrict users' access to data.

The remainder of this chapter discusses the object security and data security features of SAS Financial Management.

Object Security

Why Use Object Security?

By default, SAS Financial Management grants access to all objects, as long as a user has the necessary capabilities.

For cycles, models, and composite models, security can be used to restrict access to the object. For dimension types, dimensions, hierarchies, and custom properties, there are separate Read, Update, and Delete permissions.

Access Permissions for Cycles, Models, and Composite Models

For cycles, models, and composite models, permissions are defined simply in terms of access. If a user is denied access to any of these objects, they are also denied access to any objects that depend on it. For example, if a user is denied access to a specific model, the user is also denied access to any composite models, form sets, reports, and CDA formulas that reference that model.

Read, Update, and Delete Permissions

Dimension types, dimensions, hierarchies, and custom properties have separate Read, Update, and Delete permissions.

- Read permission

If Read permission is denied, it affects access to any objects that depend on it. For example, if a user is denied Read permission for a dimension type, the user cannot access any dimensions, hierarchies, cycles, models, or form sets that rely on that dimension type.

If a user is denied Read permission for a custom property, the user cannot access that property in SAS Financial Management Studio. In a form or report, the custom property is omitted (without affecting other access to the form or report).

- Update permission

If Update permission is denied, users cannot modify the associated object.

Update permission requires Read permission.

- Delete permission

If Delete permission is denied, users cannot delete the associated object.

Delete permission requires both Read and Update permission.

A dimension inherits Read and Update permissions from its dimension type. A hierarchy inherits Read and Update permissions from its dimension. For example, if you have Read-Only access to a dimension type, you have Read-Only access to its dimensions and to the hierarchies that belong to those dimensions.

A dimension member also inherits Read and Update permissions from its dimension. These permissions affect the ability to view or modify dimension member properties in SAS Financial Management Studio. They do not affect data security, which is set in the member properties.

Delete permissions are object specific and do not affect any objects that depend on that object. For example, a user might be denied Delete permission for a dimension but be granted Delete permission for a hierarchy within that dimension.

Note: Permissions that are applied to dimension types, dimensions, and hierarchies do not affect permissions that are assigned to custom properties.

Denying Access to an Object

To deny access to a cycle, dimension type, dimension, hierarchy, model, composite model, or custom property, complete the following steps:

1. Log on to SAS Financial Management Studio. You must have the capability for the appropriate workspace and the Security Administration capability.
2. Open the object's properties and click the **Security** tab.

You can modify security for multiple objects of the same type: select the objects, right-click, and select **Properties**. The Properties window is displayed with the security properties.

Note: You must modify security properties for each cycle and model separately.

3. Set the permissions for one or more identities.

For details, see the online Help.

Superuser Access to Cycles and Models

There is an exception to object security: members of the Administrators group can access all objects, regardless of the security settings.

Note: Members of the Administrators group still require roles with specific capabilities in order to manipulate these objects.

This super user status does not apply to data security. You can apply member permissions that deny the Administrators group access to data.

Precedence Rules for Object Security

The precedence rules for object security are as follows:

- Permissions that are assigned to a user take precedence over permissions that are assigned to a group.
- If two groups with the same relationship to the user (identity precedence) apply conflicting permissions to the object, then the grant permission wins.
- When there is no specific permission, the default is to grant access.

In the following examples, assume that user Joe belongs to Group1 and Group2. Group2 is a member of Group3.

Scenario	Results
In SAS Financial Management Studio, you deny Group1 access to ModelA. You grant Joe access to ModelA.	Joe is granted access to ModelA. The permission that you assigned directly to the user takes precedence.
You grant Group1 access to ModelA. You deny Joe access to ModelA.	Joe is denied access to ModelA. Again, the permission that you assigned directly to the user takes precedence.
You deny Group1 access to ModelA. (You assign no permissions to Joe.)	Joe is denied access to ModelA.
You deny Group1 access to ModelA. You grant Group2 access to ModelA. (You assign no permissions to Joe or Group3.)	Joe is granted access to ModelA. Because both Group1 and Group2 are equally close to Joe, the grant permission takes precedence.
You deny Group1 access to ModelA. You grant Group3 access to ModelA. (You assign no permissions to Joe or Group2.)	Joe is denied access to ModelA. Because Group1 is closer to Joe than Group3 is, Group1's permission takes precedence.

Data Security

Why Use Data Security?

Data security secures the data that is displayed in a report, in the output from a SAS program such as a stored process report, or in a planning form. The security settings can be broad or very specific.

By default, all users and groups have Read and Write permission for all members of each dimension. Regardless of the as-of date for a hierarchy, the permissions that are currently set on the member apply.

There are two ways to set permissions for data security:

- In SAS Financial Management Studio, you can set the permission on the **Security** tab of a selected member or members. This method directly populates the tables in the SAS Financial Management Data Mart.
- You can also create an external permissions file and use SAS Data Integration Studio jobs to populate the tables in the SAS Financial Management Data Mart.

TIP If you delete a member, it is archived along with its properties, including security settings. If you re-create the member, the security settings are also re-created.

Effects of Data Security

Denying Read or Write access to a dimension member has the following effects:

Task or Area	Effects
Editing a data-entry form or opening an Excel report	If a user is denied Read access to a member, all crossings that include that member are displayed as red cells. Their contents are not visible, and the cells are not writable. If a user is denied Write access (but not Read access), all crossings that include that member are protected. Their contents are visible but the cells are not writable. In Microsoft Excel, if the user right-clicks one of these cells and selects Tools ⇒ Cell information, a pop-up message explains the restriction.
Opening an Excel report	When a user opens a report that contains a read-only table, the slicer opens to the top-level member to which the user has access.
SAS programs	In a SAS program (such as a stored process), values are returned as NaN if data security prevents Read access.

Data security is inherited by subordinate members in a hierarchy. If there are no specific security permissions on a member, the permissions of its parent are applied. If a member does have security permissions, those permissions take precedence.

Precedence Rules for Data Security

By default, all users and groups have Read and Write permission for all members of each dimension. Authorization for member permissions takes the following path:

1. If permissions are directly assigned to the member and include the user or a group that the user belongs to:
 - Permissions that are assigned to a user take precedence over permissions that are assigned to a group. The closer the relationship between the user and the group, the higher the precedence.
 - If two groups with the same relationship to the user apply conflicting permissions to the member, the grant permission wins.
2. Otherwise, the authorization process checks to see whether any permissions are directly assigned to the parent member in the hierarchy. If not, the process works its way up the hierarchy.

Index

A

ABS function [134](#)
 access restrictions
 composite models [122](#)
 cycles [69](#)
 members [36, 89](#)
 models [89](#)
 roles of users [12](#)
 Account dimension type [40](#)
 account types
 Asset [41](#)
 Cumulative Translation Adjustment [42](#)
 Equity [42](#)
 Expense [42](#)
 Liability [42](#)
 overview [41](#)
 Retained Earnings [43](#)
 Revenue [43](#)
 Statistical [43](#)
 Statistical Balance [43](#)
 Statistical Flow [44](#)
 ACOS function [134](#)
 adjustment rules
 managing [102](#)
 overview [102](#)
 promoting [16](#)
 threshold [105](#)
 Adjustment Rules view [107](#)
 adjustments
 balanced manual [102](#)
 manual [102](#)
 overview [101](#)
 posting [103, 111](#)
 posting errors [103](#)
 reposting [104](#)
 unbalanced manual [102](#)
 ALL member of a trader hierarchy [48](#)
 allocation rules
 managing [102](#)
 overview [114](#)
 promoting [16](#)
 Analysis dimension type [46](#)
 analysis members
 locking [15](#)

ANCESTOR function [134](#)
 approving
 forms [163](#)
 as-of date
 hierarchy in a composite model [122](#)
 hierarchy in a model [87](#)
 target hierarchy of a form set [153](#)
 ASIN function [135](#)
 Asset account type [41](#)
 ATAN function [135](#)
 attachments
 composite models [122](#)
 cycles [69](#)
 form sets [158](#)
 models [89](#)
 automatic allocation
 materiality threshold [106](#)

B

balance account types [41](#)
 balance type property of accounts [44](#)
 balanced manual adjustments
 managing [102](#)
 overview [102](#)
 base currency
 for exchange rate sets [83](#)
 BETAINV function [135](#)
 bottom-up workflow [154](#)
 creating form sets [153](#)

C

calendar templates [30](#)
 Calendar Templates view [31](#)
 CEIL function [135](#)
 cell comments [74](#)
 CLOSINGPERIOD function [135](#)
 columns of a view
 customizing [17](#)
 Completed status of form sets [157](#)
 Completed status of forms [157](#)
 completing form sets [158](#)
 composite models

- attachments 122
 - copying 122
 - creating 122
 - deleting 122
 - optional features 122
 - overview 121
 - properties 122
 - required features 122
 - security 122, 172
 - status 122
 - Composite Models view 122
 - COMPRESS function 136
 - consolidation methods in ownership rules 117
 - consolidation parents in ownership rules 115
 - context-sensitive help 18
 - copying
 - adjustment rules 102
 - composite models 122
 - cycles 70
 - dimensions 22
 - form sets 158
 - hierarchies 22
 - manual adjustments 102
 - members 22
 - models 90
 - COS function 136
 - cost method of consolidation 117
 - creating
 - adjustment rules 102
 - composite models 122
 - custom properties 27
 - cycles 71
 - dimension types 20
 - dimensions 22
 - form sets 158
 - hierarchies 22
 - manual adjustments 102
 - members 22
 - models 90
 - Credit balance type of accounts 44
 - CTA accounts
 - and balanced balance sheets 53
 - and Source dimension 55
 - best practices 51
 - calculation 52
 - loading data 53
 - multiple 53
 - source accounts 51
 - Cumulative Translation Adjustment
 - account type 42
 - Cumulative Translation Adjustment accounts
 - See* CTA accounts
 - Currency dimension type 46
 - current data locale 29
 - CURRENT function 136
 - custom properties
 - creating 27
 - overview 27
 - Custom Properties view 28
 - customizing columns of a view 17
 - cycle periods
 - locking analysis members 15
 - cycles
 - attachments 69
 - copying 70, 71
 - creating 71
 - data management for 73
 - data record structure 4, 72
 - deleting 70
 - dimension types required 40
 - loading data into 12, 73
 - locking 13, 70
 - opening 72
 - overview 69
 - properties 70
 - security 69, 172
 - unlocking 13, 70
- D**
- data locales 29
 - Data Locales view 30
 - data management for cycles 73
 - data record structure 4, 72
 - data security 174
 - DATE function 136
 - DATETIME function 136
 - DAYOFMONTH function 137
 - DAYOFWEEK function 137
 - DAYOFYEAR function 137
 - deadlines
 - for forms 15
 - Debit balance type of accounts 44
 - default data locale 29
 - default read members in models 87
 - default write members in models 87
 - deleting
 - adjustment rules 102
 - composite models 122
 - cycles 70
 - data from cycles 73
 - dimension types 20
 - dimensions 22
 - form sets 158
 - hierarchies 22
 - manual adjustments 102
 - members 22
 - models 90
 - dimension types

- Account 40
- Analysis 46
- creating 20
- Currency 46
- deleting 20
- Frequency 46
- Organization 45
- overview 20
- properties 20
- required in a cycle 40
- Source 47
- Time 48
- Trader 48
- dimensions
 - copying 22
 - creating 22
 - deleting 22
 - exporting 22
 - loading members into 12
 - members 22
 - overview 20
 - promoting 16
 - properties 22
- Dimensions view 22
- Dimensions workspace 9
- Direct Non-Transacted holding type 116
- Direct Parent holding type 116
- Direct Transacted holding type 116
- Draft status of form sets 157
- Draft status of forms 157
- DRATE function 137
- driver formulas
 - materiality threshold 106
- driver rates 82
 - and calculated member functions 85
 - and CDA functions 86
 - best practices 82
 - copying 85
 - displaying 85
 - loading 12

E

- editing
 - adjustment rules 102
 - form templates 158
 - forms 163
 - manual adjustments 102
- Equity account type 42
- equity method of consolidation 117
- exchange rate sets
 - base currency 83
- exchange rate type property of accounts 44
- exchange rates 81
 - See also* exchange rate sets

- and calculated member functions 85
- and CDA functions 86
- best practices 82
- copying 85
- displaying 85
- loading 12
- precision 84
- EXP function 137
- Expense account type 42
- exporting
 - data records to staging tables 90
 - dimensions 22
- EXT member of a trader hierarchy 48

F

- Financial Management: Form Approver
 - role 12
- Financial Management: Form Submitter
 - role 12
- Financial Management: Planning Data
 - Entry User role 12
- Financial Management: Process
 - Administrator role 12
- FIRSTCHILD function 138
- FIRSTSIBLING function 138
- FLOOR function 139
- flow account types 41
- Form Approver role 154, 156
- form authors 156
- form reviewers 156
- form sets
 - attachments 158
 - authors 156
 - completing 158
 - copying 158
 - deadline 13
 - deleting 158
 - in a copied cycle 71
 - locking 13, 158
 - overview 154, 155
 - promoting 16
 - properties 158
 - publishing 158
 - reactivating 158
 - resetting 158
 - reviewers 156
 - running driver formulas 158
 - status 157
 - unlocking 13, 158
- Form Sets view 158
- Form Submitter role 154
- form templates
 - designing 158
- forms 154
 - approving 163

- editing 163
- locking 13, 163
- overview 3, 155
- properties 163
- pushing 163
- rejecting 163
- status 157
- submitting 163
- unlocking 13, 163
- Forms view 163
- Forms workspace 9, 155
- formulas
 - functions in 134
 - returning numeric values 134
- Frequency dimension type 46
- full consolidation 117
- full consolidation with minority interest 117
- functional currencies 45
- functions 134
 - in a bracketed member reference 129

H

- Help system 18
- hierarchies
 - as-of date in a composite model 122
 - as-of date in a model 87
 - overview 20, 22
 - roll-up arithmetic 4
 - target hierarchy of a form set 153
- Hierarchies view 24
- Holding status of forms 157
- holding types in ownership rules 116

I

- IF function 139
- importing
 - data into cycles 12, 73
 - data locales 29
 - driver rates 12
 - exchange rates 12
 - members into a dimension 12
- INDEX function 139
- INDEXC function 139
- Indirect holding type 116
- intercompany balancing rules
 - managing 102
 - overview 113
- intercompany property of accounts 44
- INTORG dimension type 45
- ISCURRENT function 139
- ISLEAF function 140
- ISLEAFCROSSING function 140
- ISVIRTUALCHILD function 140

L

- LASTCHILD function 140
- LASTSIBLING function 141
- LEFT function 141
- LENGTH function 141
- Liability account type 42
- loading
 - custom properties 27
 - data into cycles 12, 73
 - data locales 29
 - driver rates 12
 - exchange rates 12
 - members into a dimension 12
- locales (data) 29
- locking
 - analysis members 15
 - cycles 13, 70, 73
 - form sets 13, 158
 - forms 13, 163
 - models 13, 89, 90
 - periods in a cycle 13, 73, 74
- LOG function 141
- LOWCASE function 141

M

- manual adjustments
 - managing 102
 - overview 102
- Manual Adjustments view 109
- materiality threshold
 - and adjustment rules 105
 - and allocations 106
 - and driver formulas 106
 - setting 106
- MAX function 141
- MEAN function 142
- members of dimensions and hierarchies
 - importing 12
 - loading 12
 - overview 20
 - security 36
 - virtual child 40, 45
 - working with 22
- Members view 26
- MIN function 142
- MOD function 142
- models 87, 89, 90
 - attachments 89
 - composite 121
 - copying 90
 - creating 90
 - deleting 90
 - exporting data records to staging tables 90
 - locking 13, 89, 90

- optional features 89
- overview 87
- required features 87
- security 89, 172
- status 111
- unlocking 13
- warnings 111
- Models view 90
- Models workspace 9, 87

N

- NESTIF function 142
- net intercompany balancing rules
 - managing 102
 - overview 114
- NEXT function 143
- No template status of form sets 157
- No template status of forms 157
- NONE currency 46

O

- object security 172
- online Help 18
- opening cycles 72
- OPENINGPERIOD function 143
- Organization dimension type 45
- organizations
 - functional currencies 45
 - ownership rule requirements for 115
 - reporting entities 45
- ownership rules
 - consolidation methods 117
 - consolidation parents 115
 - holding types 116
 - holdings by geography 115
 - holdings by line of business 115
 - organization hierarchies for 115
 - overview 115
 - promoting 16
 - stand-alone parents 115
 - write members 115

P

- PARENT function 144
- parent method of consolidation 117
- periods in a cycle
 - deleting data from 74
 - loading data into 74
 - locking 13, 74
 - properties 74
 - unlocking 13, 74
- Periods view 74
- Periods workspace 9, 74

- phases in the Form Sets view 158
- PI function 144
- Planning Data Entry User role 154
- posting adjustments 90, 103, 111
- POW function 144
- precision
 - of exchange rates 84
- PREVIOUS function 144
- process definitions
 - creating a task in SAS Financial Management Studio 167
 - overriding task properties 169
 - task types 168
- promoting
 - adjustment rules 16
 - dimensions 16
 - form sets 16
- properties
 - composite models 122
 - cycles 70
 - dimension types 20
 - dimensions 22
 - form sets 158
 - forms 163
 - hierarchies 22
 - members 22
 - organizations 45
- PROPERTY function 145
- proportional method of consolidation 117
- Published status of form sets 157
- publishing
 - form sets 158
- pushing
 - forms 163

R

- rates
 - driver 82
 - exchange 81
- Rates workspace 9
- reactivating form sets 158
- reclassification rules
 - managing 102
 - overview 115
- rejecting
 - forms 163
- REPEAT function 146
- reporting entities 45
- reporting-entity balancing rules
 - managing 102
 - overview 114
- resetting form sets 158
- Retained Earnings account type 43
- Revenue account type 43
- REVERSE function 146

- reversing entries for manual adjustments 102
- RIGHT function 146
- roles of users for security 12
- roll-forward method for retained earnings updates 43
- roll-up arithmetic in hierarchies 4
- ROUND function 147
- rules that generate adjustments 102
- running driver formulas 158

S

- SAS Financial Management staging area 12, 27
- SAS Financial Management Studio
 - tasks for process management 167
- SAS Financial Management Web application 163
- SCAN function 147
- security
 - composite models 122
 - cycles 69
 - data 174
 - features in SAS Financial Management 171
 - for cycles 172
 - for models 172
 - members 36, 89
 - models 89
 - object 172
 - object security 172
 - roles of users 12
 - superuser access to cycles and models 173
- SIN function 147
- Source dimension type and hierarchy 47
- SQRT function 147
- staging tables
 - exporting data records to 90
- stand-alone parents in ownership rules 115
- Statistical account type 43
- Statistical Balance account type 43
- Statistical Flow account type 44
- status
 - composite models 122
 - form sets 157
 - forms 157
 - models 111
- submitting
 - forms 163
- SUBSTR function 148
- SUM function 148
- supplemental schedules
 - overview 154

T

- TABLEC function 148
- TABLEN function 148
- TAN function 148
- target hierarchy of a form set 153
- task pane 9
- templates for forms
 - designing 158
- threshold
 - for adjustment rules 105
- Time dimension type 48
- TIME function 148
- time periods in a cycle
 - loading data into 12
- TODAY function 148
- TODEGREES function 149
- top-down workflow 154
 - creating form sets 153
- TORADIANS function 149
- Trader dimension type 48
- TRANSLATE function 149
- treasury stock method of consolidation 117
- TRIM function 149

U

- unbalanced manual adjustments
 - managing 102
 - overview 102
- unlocking
 - cycles 13, 70, 73
 - form sets 13, 158
 - forms 13, 163
 - models 13, 89, 90
 - periods in a cycle 13, 73, 74
- UPCASE function 149
- user roles 12

V

- VERIFY function 149
- View menu 9
- views
 - Adjustment Rules 107
 - Calendar Templates 31
 - Composite Models 122
 - Custom Properties 28
 - Data Locales 30
 - Dimensions 22
 - Form Sets 158
 - Forms 163
 - Hierarchies 24
 - Manual Adjustments 109
 - Members 26
 - Models 90

- overview [17](#)
- Periods [74](#)
- virtual child members [40, 45](#)
- VIRTUALCHILD function [150](#)

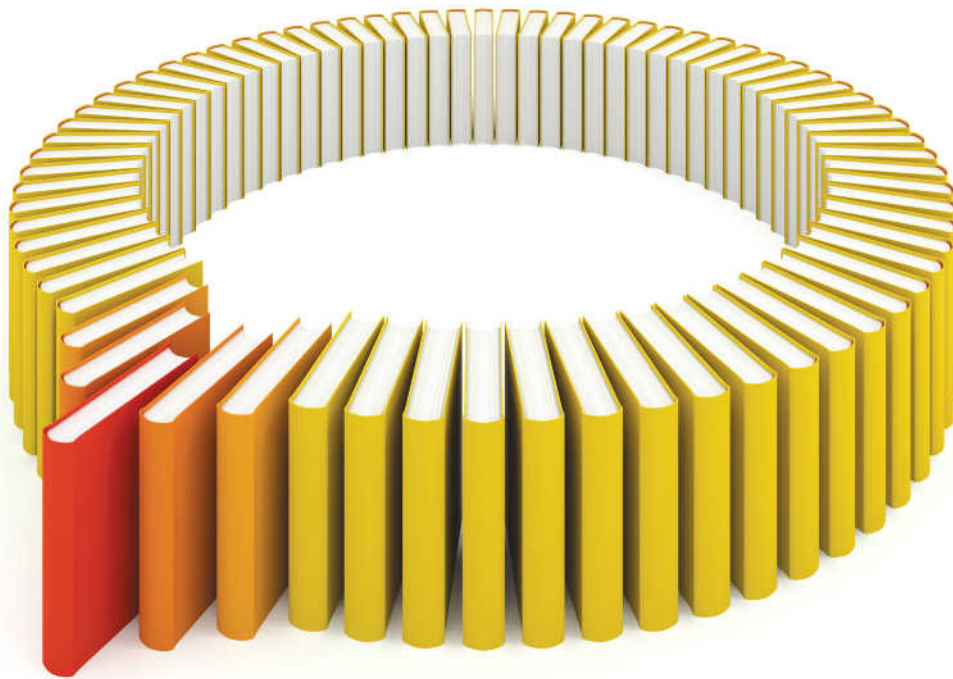
W

- warnings
 - models [111](#)
- workflow of a form set [153](#)
- workflow types [154](#)
- workspaces
 - Dimensions [9](#)

- Forms [9, 155](#)
- Models [9, 87](#)
- overview [9](#)
- Periods [9, 74](#)
- Rates [9](#)
- writable analysis members
 - in a copied cycle [71](#)
- write members in ownership rules [115](#)

X

- XRATE function [150](#)



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.



SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

