THE POWER TO KNOW.

SAS® Financial Management 5.2

User's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® Financial Management 5.2: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Financial Management 5.2: User's Guide

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

1st electronic book, November 2010

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

Contents

What's New in SAS Financial Management 5.2 . . . . . *xi*

PART 1 Introduction 1

Chapter 1 • Overview of SAS Financial Management **3**
SAS Financial Management Activities 3
SAS Financial Management Architecture 3
SAS Financial Management Data 4
Comparing Financial and Operational Planning 6
Additional Documentation 7

PART 2 Administering SAS Financial Management 9

Chapter 2 • Getting Started with SAS Financial Management Studio **11**
A Map of SAS Financial Management Studio 11
User Roles 14
Loading Data from the Cross Industry Detail Data Store 15
Locked Objects 16
Promoting Objects to Another System 19
Using Views 20
Using Wizards 21
Using the Help System 21

Chapter 3 • The Dimensions Workspace **23**
Important Documentation Convention 24
Introduction to Dimension Types, Dimensions, and Hierarchies 24
Working with Dimension Types 25
Dimension Types View 25
Working with Dimensions, Hierarchies, and Members 26
Dimensions View 27
Hierarchies View 28
Members View 32
Using Custom Properties 34
Custom Properties View 35
Using Data Locales 36
Data Locales View 37
Using Calendar Templates 38
Calendar Templates View 39

Chapter 4 • Survey of Required Dimension Types **41**
Introduction to Required Dimension Types 42
Introduction to the Account Dimension Type 42
Introduction to Account Types 43
Asset Account Type 44
Cumulative Translation Adjustment Account Type 44

Equity Account Type . . . 45
Expense Account Type . . . 45
Liability Account Type . . . 46
Retained Earnings Account Type . . . 46
Revenue Account Type . . . 47
Statistical Account Type . . . 47
Statistical Balance Account Type . . . 48
Statistical Flow Account Type . . . 48
Balance Type Property of Accounts . . . 48
Exchange Rate Type Property of Accounts . . . 48
Intercompany Property of Accounts . . . 49
Introduction to the Organization Dimension Type . . . 49
Organization Details . . . 50
Introduction to the Analysis Dimension Type . . . 50
Introduction to the Currency Dimension Type . . . 51
Introduction to the Frequency Dimension Type . . . 51
Introduction to the Source Dimension Type . . . 52
Introduction to the Time Dimension Type . . . 53
Introduction to the Trader Dimension Type . . . 54

Chapter 5 • Cycles and Cycle Periods . . . **55**
Introduction to Cycles . . . 55
Cycle Types . . . 57
How Many Cycles Do You Need? . . . 59
Cycle Manager Window . . . 59
Create a Cycle . . . 60
Open a Cycle . . . 60
Structure of the Data Records in a Cycle . . . 61
Managing Data in a Cycle . . . 62
Periods View . . . 64

Chapter 6 • Exchange Rates . . . **67**
Introduction to Exchange Rates . . . 67
Survey of Exchange Rate Types . . . 68
Exchange Rate Types View . . . 70
Exchange Rate Sets View . . . 70
Exchange Rates View . . . 71
Resolving Conflicts between Two Rows of a Rate Table . . . 74

Chapter 7 • Driver Rates . . . **77**
Introduction to Driver Rates . . . 77
Driver Rate Types View . . . 78
Driver Rate Sets View . . . 79
Driver Rates View . . . 80

Chapter 8 • Models: General Considerations . . . **83**
Introduction to Models . . . 83
Required Features of a Model . . . 84
Optional Features of a Model . . . 85
Models View . . . 86

Chapter 9 • Models: The Adjustment Process . . . **89**
Overview of the Adjustment Process . . . 89
Differences between Posting Manual Adjustments and Posting Adjustment Rules . . . 90
Adjustment Options in the Models View . . . 90
Posting Status of Manual Adjustments and Adjustment Rules . . . 91

Being On Hold . . . 92
What Happens When You Post Adjustments . . . 92
Warnings for Financial Models . . . 94
Working with Manual Adjustments . . . 95
Manual Adjustments View . . . 96
Working with Adjustment Rules . . . 98
Adjustment Rules View . . . 99

Chapter 10 • Models: Adjustment Rule Types . . . **103**
Intercompany Balancing Rules . . . 103
Net Intercompany Balancing Rules . . . 104
Reporting-Entity Balancing Rules . . . 105
Allocation Rules . . . 106
Reclassification Rules . . . 106
Ownership Rules: Introduction . . . 107
Ownership Rules: Organization Hierarchy Requirements . . . 108
Ownership Rules: Holding Types . . . 108
Ownership Rules: Consolidation Methods . . . 109

Chapter 11 • Composite Models . . . **113**
Introduction to Composite Models . . . 113
Required Features of a Composite Model . . . 114
Optional Features of a Composite Model . . . 115
Composite Models View . . . 116

Chapter 12 • Forms and the Forms Workspace . . . **117**
Forms and Form Sets . . . 117
Workflow Types . . . 117
The Forms Workspace . . . 118
Comparison of Financial and Operational Form Sets . . . 118

Chapter 13 • Working with Forms and Form Sets . . . **121**
Creating and Publishing a Form Set . . . 122
Form Sets and the Target Hierarchy . . . 125
Assigning Form Authors and Reviewers . . . 129
Enforcing Form Set Deadlines . . . 133
Time Periods in a Data-Entry Table . . . 135
Cell Protection for Data-Entry Tables . . . 138
Allow Data to be Entered for Parent Members (Financial Forms Only) . . . 144
Allocating Using Predefined Weights (Financial Forms Only) . . . 145
Supplemental Schedules . . . 151

Chapter 14 • Built-In Computations . . . **155**
Survey of Computations . . . 155
Generating Adjustments from Adjustment Rules . . . 156
Generating Intercompany Eliminations . . . 156
Frequency Conversion . . . 157
Currency Conversion . . . 159
Combining Frequency Conversion and Currency Conversion . . . 160
Computing Retained Earnings . . . 161
Computing Cumulative Translation Adjustments . . . 163
Hierarchical Roll-Ups . . . 164

Chapter 15 • Formulas . . . **167**
Introduction to Formulas . . . 167
Creating Formulas . . . 168

Formula Types . . . 169
Formula Scopes . . . 175
Formula Expressions . . . 176
Resolving Conflicts between Formulas . . . 179
Avoiding Overuse of the IF and NESTIF Functions . . . 180
Dictionary of Functions . . . 181

Chapter 16 • Security in SAS Financial Management . . . **199**
About Security in SAS Financial Management . . . 199
Object-Level Security . . . 200
Member-Level Security . . . 202

PART 3 SAS Financial Management Add-In for Microsoft Excel 205

Chapter 17 • Getting Started with the Excel Add-In . . . **207**
Overview of the SAS Financial Management Add-In for Microsoft Excel . . . 207
Summary of Tasks . . . 207
Ways to Access the Excel Add-In . . . 208
Combining Features of the Add-In with Features of Excel . . . 208
Logging On and Logging Off . . . 208
Accessibility Features . . . 209

Chapter 18 • Uses of the Excel Add-In . . . **211**
Designing a Financial Report . . . 211
Viewing a Financial Report . . . 213
Designing a Data-Entry Form Template . . . 214
Entering Data as Part of a Bottom-Up Workflow . . . 221
Reviewing Data as Part of a Bottom-Up Workflow . . . 222
Entering Data as Part of a Top-Down Workflow . . . 222
Entering Data Offline . . . 223
Entering Data into a Supplemental Schedule . . . 224

Chapter 19 • Creating a Table with the Table Wizard . . . **227**
Model Page . . . 227
Table Layout Page . . . 228
Members Page . . . 229
Selection Summary Page . . . 233
Selection Rules . . . 233

Chapter 20 • Working with Tables . . . **235**
Creating Tables . . . 236
Deleting Tables . . . 237
Table Layout Principles . . . 237
Using Member Properties as Filters, Groupers, or Labels . . . 239
Using the Table Pop-Up Menu . . . 240
Changing the View Given by a Table . . . 240
Pivoting a Table . . . 241
Using Calculated Members . . . 243
Using the Frequency Dimension . . . 243
Associating Comments with Table Cells . . . 244
Formatting Table Cells . . . 244
Coordinating Slicers between Tables . . . 245

Exporting Values to a Metric Table . . . 246
Converting a Read-Only Table to a CDA Table . . . 246
Converting a CDA Table to a Read-Only Table . . . 247
Protecting Cells in a Data-Entry Table . . . 247

Chapter 21 • Dictionary of Menu Options . . . **249**
About SAS Financial Management Add-In . . . 251
About SAS Solutions Add-In . . . 251
Adjust Values . . . 251
Allocate . . . 251
Allocation Weights . . . 252
Attach Modifications . . . 252
Automatically Refresh Tables . . . 252
Calculated Members . . . 253
CDA Table . . . 253
Cell Information . . . 253
Change Description . . . 254
Check In Form . . . 254
Check Out Form . . . 254
Clear Modifications . . . 255
Clear Table's Modifications . . . 255
Collapse . . . 255
Contributing Data . . . 256
Convert to CDA . . . 256
Copy as CDA . . . 256
Copy as Read-Only Table . . . 257
Copy as Table . . . 257
Data-Entry Table . . . 257
Delete Position . . . 258
Document . . . 258
Drill Down . . . 258
Drill Up . . . 258
Edit Protection Rules . . . 259
Expand . . . 259
Expand All . . . 259
Export as Measure . . . 259
Filter Member Combination . . . 260
Filters . . . 260
Format Members . . . 261
Hide Member . . . 261
Log Off . . . 261
Log On . . . 262
Member Labels . . . 262
New Position . . . 262
Pivot . . . 263
Prompt for File Format . . . 263
Properties . . . 263
Protect Cell . . . 263
Publish . . . 264
Read-Only Table . . . 264
Reallocate . . . 265
Refresh . . . 265
Refresh All . . . 265
Repeat New Position . . . 265
Repeat Spread . . . 266
SAS Financial Management Add-In Help . . . 266

SAS Solutions Add-In Help . . . 266
Save All Supplemental Data . . . 266
Save Form Design . . . 267
Save Template . . . 267
Show Members . . . 268
Spread . . . 268
Supplemental Schedule . . . 268
Table Properties . . . 269
Unassign . . . 269
Unprotect All Cells . . . 269
Unprotect Cell . . . 270
Validate Web Form . . . 270

Chapter 22 • Dictionary of CDA Functions . . . **273**
Using CDA Functions in Microsoft Excel Formulas . . . 273
CDACXRate . . . 274
CDADesc . . . 274
CDAGet . . . 275
CDAName . . . 275
CDAProperty . . . 275
CDARate . . . 277
CDAXRate . . . 278

Chapter 23 • Dictionary of Calculated-Member Functions . . . **279**
Using Calculated-Member Functions . . . 279
fmCode . . . 280
fmCXRate . . . 281
fmProperty . . . 282
fmRate . . . 283
fmValue . . . 284
fmXRate . . . 285

PART 4 SAS Financial Management on the Web 287

Chapter 24 • Data Entry: General Considerations . . . **289**
Introduction to Data Entry . . . 289
Form Set Types . . . 290
Facilities for Data Entry . . . 290
Security Roles for Data Entry . . . 291
Form Manager Page . . . 291
Form Editor Page . . . 293
Actions for Forms . . . 297
Status Values for Forms . . . 297

Chapter 25 • Data Entry: Bottom-Up Workflow . . . **299**
Description of the Bottom-Up Workflow . . . 299
Bottom-Up Actions . . . 300
Bottom-Up Status Values . . . 302

Chapter 26 • Data Entry: Top-Down Workflow . . . **305**
Description of the Top-Down Workflow . . . 305
Top-Down Actions . . . 306
Top-Down Status Values . . . 307

Chapter 27 • Reports in the Document Manager **309**
Types of Reports 309
Viewing Reports 309
List of Administrative Reports 310
Index **313**

What's New in SAS Financial Management 5.2

Overview

SAS Financial Management 5.2 has new or changed features that affect the following areas:

- creating and publishing a form set
- working with forms and form templates
- reusing a form set
- required roles
- thresholds for allocation and reclassification rules

Creating and Publishing a Form Set

- (Financial forms only) By default, there is a separate form for each member selected in the target hierarchy. As an option, the form for a target member can include its writable descendants, so that users who are responsible for updating multiple forms can open a single form.
- Member selection rules for financial form sets are saved and can be edited in the form set properties, just as they are for operational form sets. When the form set is published, the rules are applied based on the hierarchy's as-of date and time.
- During form set creation, calculated members are available for selection as writable analysis members.
- Administrators can specify a form set deadline that includes both a due date and time.
- Automatic notifications can be sent to form users (depending on the form’s status), reminding them of the approaching deadline
- The deadline can include automatic locking. If this option is selected, the forms are locked when the deadline is reached.
- When a form is locked (either automatically or manually), it is no longer available in Form Manager. If the form is open, it becomes a read-only form. (An administrator can

manually lock a form by locking the form itself, its containing form set, or its containing cycle.)

Forms and Form Templates

- Financial Form Manager is now called Form Manager.
- A floating time rule can now be used in both data-entry tables and read-only tables. This functionality improves the ability to have rolling forecasts.
- (Financial forms only) Cells (crossings) in a data-entry table can be protected against the following actions: manual data entry, spread, client side allocation, and automatic allocation. The values of these cells can still change as the result of indirect actions such as manual adjustments or changes in exchange rates.

  Administrators can define cell protection rules for a model and for a form set. In the SAS Financial Management Add-In for Microsoft Excel, users can protect additional cells within a form.
- (Financial forms in a bottom-up workflow only) In a form template, process administrators can enable the following actions:
  - entering values directly at roll-ups
  - automatically allocating these values to subordinate members, based on predefined weights
- When they edit a form in Microsoft Excel, users can select a subset of the available analysis members to appear in the table. If they save the form design, their selection is available the next time they open the form.
- Multiple sets of system filters are permitted in a form.

  Daisy chaining is not permitted. That is, the target of a filter cannot act as the source of another filter in the same filter set.

Reusing a Form Set

- Resetting and publishing a form set from SAS Financial Management Studio has the following effects:
  - Comments from the previous publish operation are retired. In Form Manager, users can view only comments that occurred since the most recent publish operation.
  - In Form Manager, users can view only history that occurred since the most recent publish operation.

  In SAS Financial Management Studio, administrators can view the full comments and history.
- Using the SAS Financial Management Java API, administrators can write batch processes for the following purposes:
  - publishing a form set, with options to notify users and retire existing form comments. (If you do not retire form comments, the comments from the previous publish are still available in Form Manager. However, history from a previous publish operation is not available in Form Manager.)

- resetting a form set, with the option to notify users.
- setting the deadline for a form set.
- (financial forms only) setting the as-of date for a form set's target hierarchy.
- locking or unlocking all the forms in a form set.

Required Roles

SAS Financial Management 5.2 includes a new set of roles. The Financial Management: Process Administrator role grants access to all features of SAS Financial Management. In addition, each workspace in SAS Financial Management Studio has a corresponding role that grants full access to that workspace and permits users to browse content in the other workspaces.

Threshold for Allocation and Reclassification Rules

By default, adjustments that are generated by an allocation rule or a reclassification rule are subject to an absolute value threshold. Adjustments with an absolute value that is less than the threshold value are not posted.

The default threshold value is 0.001. In previous releases of SAS Financial Management, the default threshold value was 0.0 (which results in no threshold being applied). You can change the threshold value in the JVM options for the managed server. See "Post-Configuration Steps" in the *SAS Solutions Services: System Administration Guide*.

Part 1

Introduction

Chapter 1
Overview of SAS Financial Management *3*

Chapter 1

Overview of SAS Financial Management

SAS Financial Management Activities 3

SAS Financial Management Architecture 3

SAS Financial Management Data 4
Resemblance to Spreadsheet Data 4
Number of Coordinates 5
Significance of the Coordinates 5
Coordinate Values Are Members of Hierarchies 5

Comparing Financial and Operational Planning 6

Additional Documentation 7

SAS Financial Management Activities

SAS Financial Management facilitates these important activities:

- financial reporting
- financial planning
- operational planning

You can use SAS Financial Management for any or all of these activities. The parts of the product that you use, and the way that you use the product, depend in part on which of these activities you are engaged in.

For financial reporting, SAS Financial Management receives data from a general ledger system or other third-party database, performs the computations that you specify, and presents results in reports that you design. For financial and operational planning, SAS Financial Management receives data from users through data-entry forms that you design and presents results in those same forms or in other reports. Financial planning involves the entry of large amounts of monetary data. Operational planning involves the entry of even larger amounts of monetary or non-monetary data.

SAS Financial Management Architecture

SAS Financial Management consists of a server and three clients. The server contains the shared database and performs most of the computations. Following are the three clients:

- The SAS Financial Management Add-In for Microsoft Excel connects your desktop copy of Microsoft Excel to the SAS Financial Management database on the server. You can use this client to build reports that are based on data in the SAS Financial Management database, and to view the reports that you or others have built. You can also use this client to build the forms that enable you to enter financial or operational planning data into the SAS Financial Management database, and to enter financial planning data through financial planning forms that are enabled for use in Excel.
- The SAS Financial Management Web application includes a Form Manager that displays a list of data-entry forms to each user who must enter or review planning data. It also includes a Form Editor for entering planning data into forms and reviewing planning data that has been entered by others.

  From the list of forms in the Form Manager, a user can launch any selected operational-planning form in the Form Editor. For a selected financial-planning form, there are three possibilities, which depend on the design of the form set that the form belongs to. Some financial-planning forms can be launched only in the Web application's Form Editor, some can be launched only in Excel, and some can be launched in either the Form Editor or Excel.

  The SAS Financial Management Web application is part of an integrated SAS Web environment that might give you access to other SAS products, such as SAS Human Capital Management or SAS Strategy Management, depending on what SAS products you have purchased and installed.
- SAS Financial Management Studio enables you to build and manage the infrastructure that the other two clients need. Here are a few examples. The two-way flow of data between Microsoft Excel or the Web data-entry interface and the SAS Financial Management database is controlled by complex structures called models, which you must build using SAS Financial Management Studio. The data-entry process of financial or operational planning uses form sets, which you must build using SAS Financial Management Studio. Models and form sets are based on cycles, which must also be defined in SAS Financial Management Studio.

Most users never see SAS Financial Management Studio. A typical user does some combination of financial reporting and financial or operational planning using the SAS Financial Management Add-In for Microsoft Excel and/or the SAS Financial Management Web application.

In addition to the three SAS Financial Management clients, every SAS Financial Management system also includes SAS Data Integration Studio, which is supplied with many jobs and tables that are tied to SAS Financial Management. These jobs and tables enable you to supply SAS Financial Management with essential data and metadata, including user definitions, dimension definitions, periodic accounting data, currency exchange rates, driver rates, and report filters. The SAS Financial Management database typically receives planning data through Web-based or Excel-based data-entry forms, and factual accounting data through jobs and tables that reside in SAS Data Integration Studio.

SAS Financial Management Data

Resemblance to Spreadsheet Data

Each record in the SAS Financial Management database associates a single numeric value with a particular set of dimension members. This record structure resembles the association between a numeric value in a spreadsheet cell and the cell's set of three location coordinates,

which identify the row, the column, and the sheet that the cell belongs to. However, there are important differences, which are explained in the following sections.

Number of Coordinates

A spreadsheet cell is always associated with a set of three coordinates. In SAS Financial Management, data records are stored in containers called cycles. The number of dimension members that each record associates with a numeric value can vary from one cycle to another. There are two types of cycle—financial and operational. In a financial cycle, every data record contains one member from each of these seven required dimension types:

- Account
- Analysis
- Currency
- Organization
- Source
- Time
- Trader (mirrors Organization)

In an operational cycle, the Source and Trader dimension types are not used, so there are only five required dimension types—Account, Analysis, Currency, Organization, and Time. A cycle of either type can include additional, optional dimension types; each data record in a cycle contains a member from each optional dimension type that the cycle uses.

The set of all the dimension members that are associated with a given numeric value, including one member from each relevant dimension type, is known as the crossing that the numeric value belongs to. Each data record in the SAS Financial Management database consists of one crossing and one associated numeric value.

Significance of the Coordinates

The row, column, and sheet coordinates of a spreadsheet cell identify the cell but tell you nothing about the significance of the numeric value that is in the cell. By contrast, the dimension members in a SAS Financial Management data record tell you exactly 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 2009 expressed in euros, while another record's dimension members might tell you that the numeric value represents the planned salary expense for a Japanese subsidiary for fiscal year 2011 expressed in yen.

Coordinate Values Are Members of Hierarchies

Spreadsheet coordinates belong to flat sets—the set of rows, the set of columns, the set of sheets. By contrast, the dimension members used by SAS Financial Management typically belong to hierarchies. For example, an organization typically belongs to a hierarchy of organizations and an account typically belongs to a hierarchy of accounts. Each member is a member of a hierarchy in addition to being a member of the dimension and the dimension type that the hierarchy belongs to.

The hierarchical relationships between members help to define the functional relationships between numeric values. In a spreadsheet, all functional relationships between numeric values are defined by user-specified formulas. Formulas are important in SAS Financial Management too. However, they work together with functional relationships of several

other sorts, including the roll-up arithmetic that is implicit in the hierarchical structures of members. For example, suppose that an organization hierarchy has a European division with three subordinate organizations—an Italian subsidiary, a French subsidiary, and a German subsidiary. In this case, the actual October 2009 revenue for the European division is computed as the sum of the actual October 2009 revenue values for the three national subsidiaries; no formula is involved.

The concept of a hierarchy that SAS Financial Management uses is more inclusive than the ordinary concept of a hierarchy. A SAS Financial Management hierarchy can be a strict hierarchy or a set of two or more strict hierarchies that has no single top member. The extreme case of such a multiple-hierarchy hierarchy is a completely flat hierarchy. This is simply a set of members that has no hierarchical structure at all. In the Currency dimension type and the Analysis dimension type, all hierarchies must be flat.

Comparing Financial and Operational Planning

In SAS Financial Management, financial planning and operational planning are broadly similar activities. For both types of planning, all of the following are true:

- You define form sets in the Forms workspace of SAS Financial Management Studio.
- Each form set has a target hierarchy. A form is assigned to selected members of the target hierarchy.
- The person who creates the form set can choose between a bottom-up workflow, in which data is entered at the bottom of the target hierarchy and then aggregated, and a top-down workflow, in which data is entered at the top of the target hierarchy and then allocated.
- You design the form template that defines the form set's forms using the SAS Financial Management Add-In for Microsoft Excel.
- Users enter data into data-entry tables in the forms. The data-entry work of a number of users is coordinated and combined by means of the bottom-up or top-down workflow.

However, there are numerous differences between the two types of planning, including the following:

- Financial planning data is stored in financial cycles. Operational planning data is stored in operational cycles. A key difference between the two cycle types is that financial cycles require the Source and Trader dimension types, whereas operational cycles do not use these dimension types. When you create a cycle to support a planning process, you must create the right type of cycle.
- You can load data from an operational cycle into a financial cycle by opening the target financial cycle and using the Load Operational Data wizard in the Periods workspace. You cannot load data from a financial cycle to an operational cycle.
- A financial model can include manual adjustments and adjustment rules, provided that these are enabled in the underlying cycle. An operational model cannot include manual adjustments and adjustment rules. Accordingly, the Export Data Records option is available for financial models but not for operational models.
- You can lock and unlock financial models using the **Lock** and **Unlock** options in the Models view. The **Lock** and **Unlock** options are not available for operational models. However, an operational model is automatically locked if any operational form set that uses it is published.

- A read-only table or a CDA table in the SAS Financial Management Add-In for Excel can be based only on a financial model, which retrieves data from a financial cycle. Operational models are not available for selection when you create a read-only table or a CDA table.
- Although financial form sets and operational form sets are broadly similar, they differ in many important ways.

See Also

Cycle Types on page 57

Additional Documentation

For additional information, see the appropriate versions of the following books. They are available at **http://support.sas.com/documentation/onlinedoc/fm/index.html**.

- *SAS Financial Management: User's Guide*
- *SAS Solutions Services: Data Administration Guide*
- *SAS Solutions Services: Data Model Reference*

Part 2

Administering SAS Financial Management

Chapter 2
Getting Started with SAS Financial Management Studio *11*

Chapter 3
The Dimensions Workspace *23*

Chapter 4
Survey of Required Dimension Types *41*

Chapter 5
Cycles and Cycle Periods *55*

Chapter 6
Exchange Rates *67*

Chapter 7
Driver Rates *77*

Chapter 8
Models: General Considerations *83*

Chapter 9
Models: The Adjustment Process *89*

Chapter 10
Models: Adjustment Rule Types *103*

Chapter 11
Composite Models *113*

Chapter 12
Forms and the Forms Workspace *117*

Chapter 13
Working with Forms and Form Sets *121*

Chapter 14

Built-In Computations *155*

Chapter 15
Formulas *167*

Chapter 16
Security in SAS Financial Management *199*

Chapter 2

Getting Started with SAS Financial Management Studio

A Map of SAS Financial Management Studio **11**
Overview 11
Dimensions Workspace 12
Periods Workspace 12
Rates Workspace 13
Models Workspace 13
Forms Workspace 14

User Roles **14**
Roles for SAS Financial Management Studio 14
Roles for Editing and Approving Forms 14

Loading Data from the Cross Industry Detail Data Store **15**

Locked Objects **16**
Overview of Locked Objects 16
Locked Cycles 17
Locked Cycle Periods 18
Locked Financial Models 18
Locked Operational Models 18
Locked Form Sets 18
Locked Forms 18

Promoting Objects to Another System **19**

Using Views **20**

Using Wizards **21**

Using the Help System **21**

A Map of SAS Financial Management Studio

Overview

In SAS Financial Management Studio, you work mainly in views. The views are grouped into five workspaces. The workspaces are listed on the **View** menu and in the task pane on the left side of the main window. You can hide the task pane or vary its size by using the **Task Pane** option on the **View** menu.

One overarching aspect of SAS Financial Management Studio, working with cycles, is outside the domain of workspaces and views. To work with cycles, select **Tools** ⇨ **Cycle Manager** or use the cycle options on the **File** menu.

Dimensions Workspace

Use the Dimensions workspace to manage the complete system of dimension types, dimensions, hierarchies, and members that is used to specify what numeric values represent. This information is typically loaded from the Cross Industry Detail Data Store, but you can use the Dimensions workspace to view and edit it.

The Dimensions workspace includes seven views, which you open as follows:

- To open the Dimensions view, select the Dimensions workspace.
- To open the Hierarchies view for any dimension, select the dimension in the Dimensions view, and then select **Hierarchies**.

  Alternatively, select a member in the Members view for the dimension, and then select **Hierarchies**.
- To open the Members view for any dimension, select the dimension in the Dimensions view, and then select **Members**.

  Alternatively, select a hierarchy or a member in the Hierarchies view for the dimension, and then select **Members**.
- To open the Dimension Types view, select the Dimensions workspace, and then select **Tools** ⇨ **Dimension Types**.
- To open the Custom Properties view, select the Dimensions workspace, and then select **Tools** ⇨ **Custom Properties**.

  Custom properties of members can be used to taper the granularity of the forms as you climb the target hierarchy of an operational form set. They can also be in the SAS Financial Management Add-In for Microsoft Excel in all the ways that standard member properties can be used.
- To open the Data Locales view, select the Dimensions workspace, and then select **Tools** ⇨ **Data Locales**.

  Data locales enable you to provide names and descriptions of certain object types in more than one language.
- To open the Calendar Templates view, select the Dimensions workspace, and then select **Tools** ⇨ **Calendar Templates**.

  Calendar templates are used to create sets of time periods within a time dimension.

Periods Workspace

Use the Periods workspace to manage the time periods and the data in the open cycle. The Periods workspace includes only one view, the Periods view, which enables you to add periods, load data, and delete data.

To open the Periods view, select the Periods workspace.

You can also load data into a cycle using SAS Data Integration Studio. For details, see *SAS Solutions Services: Data Administration Guide*.

See Also

Additional Documentation on page 7

Rates Workspace

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

The Rates Workspace includes a set of three views for exchange rates and a parallel set of three views for driver rates, which you open as follows:

- To open the Exchange Rate Sets view, select the Rates workspace.
- To open the Exchange Rates view for any exchange rate set, select the exchange rate set in the Exchange Rate Sets view, and then select **Exchange Rates**.
- To open the Exchange Rate Types view, select the Rates workspace, and then select **Tools** ⇨ **Exchange Rate Types**.
- To open the Driver Rate Sets view, select the Rates workspace, and then click the tab of the Driver Rate Sets view.
- To open the Driver Rates view for any driver rate set, select the driver rate set in the Driver Rate Sets view, and then select **Driver Rates**.
- To open the Driver Rate Types view, select the Rates workspace, and then select **Tools** ⇨ **Driver Rate Types**.

Models Workspace

Use the Models workspace to create and manage models. A model is a complex framework that connects the data that is stored in its cycle with the numeric values that are displayed in reports or entered into planning forms. Each model includes a hierarchy for each dimension that is used by the cycle. A financial model can define an adjustment process that uses a combination of manual adjustments and adjustment rules.

The Models workspace includes five views, which you open as follows:

- To open the Models view, select the Models workspace.
- To open the Manual Adjustments view for any financial model, select the model in the Models view, and then select **Manual Adjustments**.
- To open the Adjustment Rules view for any financial model, select the model in the Models view, and then select **Adjustment Rules**.
- To open the Attachments view for any model, select the model in the Models view, and then select **Attachments**.

  The Attachments view for a model and the Attachments view for a form set have the same appearance and behavior.
- To open the Composite Models view, select the Models workspace when a financial cycle is open, and then select **Tools** ⇨ **Composite Models**.

  A composite model is a framework for displaying reports that contain numeric values that are generated by two or more different financial models.

Forms Workspace

Use the Forms workspace to create and manage form sets. A form set is a set of related data-entry forms. Forms in the form set are assigned to designated members of a designated hierarchy. For example, in order to implement a budgeting process, you might assign forms to all the organizations in a certain organization hierarchy that have responsibility for preparing budgets.

The Forms workspace includes three views, which you open as follows:

- To open the Form Sets view, select the Forms workspace.
- To open the Forms view for any form set, select the form set in the Form Sets view, and then select **Forms**.
- To open the Attachments view for any form set, select the form set in the Form Sets view, and then select **Attachments**.

  The Attachments view for a model and the Attachments view for a form set have the same appearance and behavior.

User Roles

Roles for SAS Financial Management Studio

Users with the Financial Management: Finance Process Administrator role have full access to all features of SAS Financial Management Studio. In addition, each workspace has a corresponding role 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 the Financial Management: Finance Process Administrator role or one of the workspace roles.

For details, see "Assigning Groups and Roles" in the *SAS Solutions Services: System Administration Guide*.

Roles for Editing and Approving Forms

When you assign authors and reviewers to the forms of a form set, keep in mind that authors and reviewers must have appropriate roles in order to do their work:

- **Financial forms:**
  - To be assigned as author of a financial form, a user must have the Financial Management: Form Submitter role or the Financial Management: Process Administrator role.
  - To be assigned as reviewer of a financial form, a user must have the Financial Management: Form Approver role or the Financial Management: Process Administrator role.
- **Operational forms:** To be assigned as author or reviewer of an operational form, a user must have the Financial Management: Planning Data Entry User role or the Financial Management: Process Administrator role.

Note: From the Form Manager or from the Forms workspace of SAS Financial Management Studio, a user with the Financial Management: Process Administrator

role or the Financial Management: Form Administrator role can access all forms (subject to any security restrictions that are in place).

For details, see "Assigning Groups and Roles" in the *SAS Solutions Services: System Administration Guide*.

Loading Data from the Cross Industry Detail Data Store

SAS Financial Management Studio provides the following options for loading different types of data from the Cross Industry Detail Data Store into the SAS Financial Management database:

- To load members and hierarchies into a dimension, use the **Load Dimension** option in the Dimensions view. This option launches the Load Dimension wizard. It is equivalent to the Load Dimension job in SAS Data Integration Studio. For details, see the online Help for the individual wizard pages.

  You can use this option to load members and hierarchies into a SAS Financial Management database for the first time or to load members and hierarchies that have been exported from a parallel SAS Financial Management system as part of an object promotion routine. See Promoting Objects to Another System on page 19.

  The **History** page of a dimension's Properties window shows the complete history of successful and unsuccessful attempts to load members and hierarchies into the selected 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. This wizard is slightly different for financial cycles and operational cycles. The financial-cycle variant is equivalent to the Load Base Data job in SAS Data Integration Studio. The operational-cycle variant is equivalent to the Load OP Data job in SAS Data Integration Studio. For details, see the online Help for the individual wizard pages.

  The **History** page of a cycle time period's Properties window shows the complete history of successful and unsuccessful attempts to load data into the selected time period of the selected cycle.

  Loading data into a cycle from the Cross Industry Detail Data Store is part of the job of managing the data in a cycle. There are several other ways in which data can enter a cycle. For any cycle, users can enter data through forms, if you are using the product's planning facilities. For a financial cycle, you can load data from another financial cycle using the **Load Model Data** option and you can load data from an operational cycle using the **Load Operational Data** option. All the wizards for loading data offer choices regarding which related data to delete. 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. It is equivalent to the Load Exchange Rates job in SAS Data Integration Studio. For details, see the online Help for the individual wizard pages.

  The **History** page of an exchange rate set's Properties window shows the complete history of successful and unsuccessful attempts to load exchange rates into that exchange rate set.

- To load driver rates, use the **Load Driver Rates** option in the Driver Rate Sets view. This option launches the Load Driver Rates wizard. It is equivalent to the Load Driver Rates job in SAS Data Integration Studio. For details, see the online Help for the individual wizard pages.

  The **History** page of a driver rate set's Properties window shows the complete history of successful and unsuccessful attempts to load driver rates into that driver rate set.

In each case, it is important to make sure that the data reaches the Cross Industry Detail Data Store before you try to move it from there into the SAS Financial Management database.

You can find details about loading data through the Cross Industry Detail Data Store in the *SAS Solutions Services: Data Administration Guide*. (See Additional Documentation on page 7.)

See Also

Managing Data in a Cycle on page 62

Locked Objects

Overview of Locked Objects

Objects of several types can be in a locked state or an unlocked state. An object in a locked state cannot be modified. The locked state is indicated with a padlock in the object's icon  .

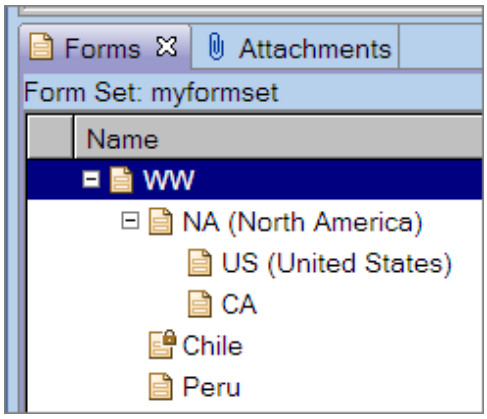
To unlock an object, reverse the process. If you locked a form set with the **Lock** button, click the **Unlock** button to unlock it.

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.

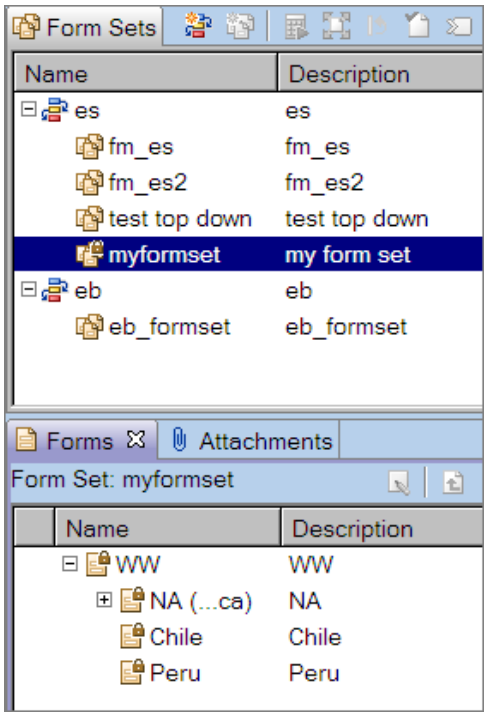
In some cases, an object is affected by more than one lock. For example, there might be a lock on a form, the related form set, and the related cycle. To undo those locks, begin by unlocking the most comprehensive lock. The subordinate locks still apply until you unlock them.

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, delete data from it, or enter data into it through a form.

To lock a cycle period, do one of the following:

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

Locked Financial Models

If a financial 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 financial model, do one of the following:

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

Locked Operational Models

If an operational model is locked, you cannot modify its properties. However, you can still create and publish form sets that use the model.

You cannot lock an operational model directly. To lock an operational model, perform one of the following actions. Each of these actions implicitly locks the model:

- Publish an operational form set that uses the model.
- Lock the cycle that the model belongs to.

If you locked an operational model by publishing an operational form set that uses the model, the model is not unlocked until each associated form set has a status of either Draft or Completed.

Locked Form Sets

If a form set is locked, you cannot modify its properties, modify its template, or change its status. Its forms are implicitly locked, and no one can use them to enter data.

To lock a form set, do one of the following:

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

Locked Forms

If a form is locked, no one can use it to enter data. Locking can be manual or automatic:

- **Manual locking.** To lock a form manually, do one of the following:
  - 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 automatic locking 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 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, promote them to a test system for testing, and then promote them again to a development system when they pass the relevant tests.

The following object promotion facilities are available:

- You can promote the members and hierarchies of a dimension using the **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 and hierarchies from an identically structured set of SAS tables in the Cross Industry Detail Data Store.
- You can promote allocation rules and ownership rules using the **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, which imports adjustment rules into a selected financial model from a SAS package file that was created by the Export Adjustment Rules wizard. For details, see the online Help for the individual wizard pages.
- You can promote financial form sets using the **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. For details, see the online Help for the individual wizard pages.

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 on 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**.

It is difficult to make statements that are true of every view, but there are some important features that are shared by many views.

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 by the information in any column, click the column heading. To reverse the sort order, click the same column heading again.

In many views, options are listed in these 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 View Menu ( ) in the toolbar
- the task list on the task pane to the left of the view

These four places contain similar, but not exactly identical, sets of options.

Each option has a short description and a long description, which are used as follows:

- The pop-up menu and the toolbar menu use the short descriptions.
- The toolbar tips are long descriptions.
- The task list uses the long descriptions when the task pane is full size and the short descriptions when the task pane is medium size. When the task pane is small size or hidden, there is no task list.

  To change the size of the task pane, select **View** ⇨ **Task Pane**.
- This online Help system uses the short descriptions.

In many views, the following generic options are available:

Filter
displays or hides the **Search** field at the top of the view. If you enter a string in the **Search** field and click the magnifying glass to the right of the field, the list is limited to those items for which the entered string is a substring of the text in some column of the view.

This option is available on the toolbar and the toolbar menu.

Customize Columns
displays the Customize Columns window, which enables you to specify which columns of information to include in the view, and what order to display the columns in.

This option is available on the pop-up menu, the toolbar, and the toolbar menu.

Refresh

refreshes the view so that it shows all the currently existing items with their current properties.

This option is available on the toolbar and the toolbar menu.

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.

Using Wizards

Many tasks are performed with wizards. The option for the task launches the wizard for that task. You work through the wizard page by page. When you come to the end of the wizard and click **Finish**, you have completed the task. Every page of every wizard has a Help topic. The page-by-page Help is the detailed documentation for the task.

In general, tasks that are not performed with wizards are very simple. They consist of one or two short steps.

Using the Help System

The Help system includes a table of contents, an index, a full-text search facility, and a bookmark facility. Buttons at the bottom of the Help display enable you to switch from each of these to any other.

To expand or collapse portions of the table of contents, click the + and – signs next to the group headings. To display any topic that is listed in the table of contents, click the topic title.

If a view is active, you can press the F1 key to display its Help topic.

If a window, wizard page, or property page is active, you can either press the F1 key or click the Help button on the active item to display its Help topic. Look for one of the following buttons:

-  in the lower left corner
- 

in the lower right corner

You can select Help text in any displayed Help topic, and then use the Help topic pop-up menu to print the selected text or copy it to a document or e-mail message.

Chapter 3

The Dimensions Workspace

Important Documentation Convention **24**

Introduction to Dimension Types, Dimensions, and Hierarchies **24**

Working with Dimension Types **25**

Dimension Types View **25**

Working with Dimensions, Hierarchies, and Members **26**

Dimensions View **27**

Hierarchies View **28**
- Overview of the Hierarchies View 28
- Navigating Hierarchies 28
- Hierarchy Options 28
- Member Options 29
- Create a Hierarchy 30
- Modify a Hierarchy 31
- Delete a Hierarchy 31
- Move a Member within a Hierarchy Using Drag-and-Drop 32

Members View **32**
- Overview of the Members View 32
- Available Options 33
- Add a Member to a Hierarchy Using Drag-and-Drop 33

Using Custom Properties **34**
- Reasons for Using Custom Properties 34
- Ways to Define Custom Properties 34
- Review or Define Custom Properties 35
- Add Custom Property Values to Members 35

Custom Properties View **35**

Using Data Locales **36**
- How Data Locales Work 36
- Defining Data Locales 36

Data Locales View **37**

Using Calendar Templates **38**
- Utility of Calendar Templates 38
- Create a Calendar Template 38
- Add Time Periods to a Hierarchy with a Calendar Template 38
- Add Time Periods to a Dimension with a Calendar Template 38

Calendar Templates View **39**

Important Documentation Convention

The features of the Dimensions workspace of SAS Financial Management Studio are also available as a separate client called the SAS Solutions Dimension Editor, which can be used to manage dimensions for other products. Therefore, with the exception of this topic, the entire discussion of these features is written so that it can be used in both contexts. This means, in particular, that the discussion never refers to the Dimensions workspace, even though there are certain places where it might be helpful to do so.

In SAS Financial Management Studio you must have the Dimensions workspace selected in order to use any of the following views:

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

See Also

A Map of SAS Financial Management Studio on page 11

Introduction to Dimension Types, Dimensions, and Hierarchies

Dimension types, dimensions, and hierarchies constitute a three-tiered organizational system for the members that are used to identify numeric values:

- The dimension types represent the information categories that are used to identify numeric values. Some basic dimension types are Organization, Account, Time, Analysis, and Currency.
- Within a dimension type, there can be any number of dimensions. In many cases, you need only one dimension within a dimension type, but there are reasons why you might want to maintain two or more dimensions within a dimension type. One reason is to use dimensions like folders in order to keep different sets of members separate from one another. Another reason is to use dimensions to maintain two or more variants of the same member. 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.
- Within a dimension, there can be any number of hierarchies. Each hierarchy embeds some or all of the members of the dimension in a hierarchy-like structure. The hierarchy-like structure can be a strict hierarchy with a single top member, or a set of

strict hierarchies, or even a simple ordered set that has no vertical structure at all. The Analysis and Currency dimension types permit only completely flat hierarchies.

Working with Dimension Types

Many dimension types are predefined. In general, if you need to create additional dimension types, you should define them in the Cross Industry Detail Data Store and load them from there. Only a dimension type that is defined in the Cross Industry Detail Data Store can be used to load numeric data.

To review the existing dimension types or to create new ones:

1. Select **Tools** ⇨ **Dimension Types**.
2. Use the Dimension Types view.

See Also

Dimension Types View on page 25

Dimension Types View

The Dimension Types view lists all the existing dimension types.

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 Cross Industry Detail Data Store, you must also do work in SAS Data Integration Studio. You can choose to do all the work of creating a new dimension type in SAS Data Integration Studio. In that case, do not run the New Dimension Type wizard. For details, see *SAS Solutions Services: Data Administration 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. If you click **Yes** in the confirmation window, then the selected dimension type is deleted.

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.

See Also

- Using Views on page 20

- Working with Dimension Types on page 25

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 Cross Industry Detail Data Store. There are two reasons why it is generally better to load members into dimensions from the Cross Industry Detail Data Store than to define the members here:

- Any member that is used to load numeric values must be defined in the Cross Industry Detail Data Store.
- Defining a lot of members here is a time-consuming task.

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, you can select any hierarchy or any member of any hierarchy.

   Some dimensions cannot be edited. For such a dimension, you can view hierarchies and members but you cannot edit them.

A member is always a member of a dimension, but it might not be a member of any hierarchy in that dimension. You can work with a member in the context of any hierarchy that it belongs to or outside the context of any hierarchy.

To work with a member in the context of a hierarchy, follow the preceding steps for working with a hierarchy.

To work with a member outside any hierarchy context:

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

   The Members view for the selected dimension opens.
3. In the Members view, you can select any member of the dimension.

For each dimension, you can also open the Members view from the Hierarchies view, using the **Members** option, and open the Hierarchies view from the Members view, using the **Hierarchies** option. The Hierarchies view and the Members view are closely related views that provide complementary perspectives on the contents of a selected dimension.

See Also

- Dimensions View on page 27
- Hierarchies View on page 28

- Members View on page 32

Dimensions View

The Dimensions view lists all the existing dimensions for all the existing dimension types. The **Dimension Type** column shows which dimension type each dimension belongs to.

The following options are available:

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

Load Dimension
launches the Load Dimension wizard, which enables you to load members and hierarchies from the Cross Industry Detail Data Store 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 package file.

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.

A confirmation window reminds you that when you delete a dimension, you also delete all the hierarchies and members that belong to that dimension. If you click **Yes** in the confirmation window, then the selected dimension is deleted.

Hierarchies
opens the Hierarchies view for the selected dimension, enabling you to work with the hierarchies that belong to that dimension and the members in their hierarchical contexts.

Some dimensions have members and hierarchies that cannot be edited. If you select such a dimension, the Hierarchies view enables you to view its members and hierarchies, but not to modify them.

Members
opens the Members view for the selected dimension, enabling you to work with the members that belong to that dimension outside their hierarchical contexts.

Some dimensions have members that cannot be edited. If you select such a dimension, the Members view enables you to view its members, but not to modify them.

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

See Also

- Using Views on page 20

• Working with Dimensions, Hierarchies, and Members on page 26

Hierarchies View

Overview of the Hierarchies View

The Hierarchies view displays the hierarchies that belong to the dimension that is identified in the view's tab. The view includes a left-justified row to identify each hierarchy and indented rows for all the members of each hierarchy.

If you select a hierarchy, then hierarchy options are available. If you select a member, then member options are available.

The **Members** option, which is available whether you select a hierarchy or a member, opens the Members view for the selected dimension. The Members view displays a list of all the members in the dimension.

Some dimensions cannot be edited. For such a dimension, the options that modify hierarchies and members are not available.

Navigating Hierarchies

To expand a hierarchy or a part of a hierarchy, do one of the following:

- Click the + sign that is next to the item that you want to expand.
- Select the item that you want to expand, and then select **Expand All**.

To collapse a hierarchy or a part of a hierarchy, click the – sign that is next to the item that you want to collapse.

To collapse all hierarchies, select any item, and then select **Collapse All**.

To search for a particular member or hierarchy, select **Find**. The **Find** option is available on the toolbar and the toolbar menu.

Hierarchy Options

When you select a hierarchy, the following options are available:

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 either 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.

Copy Here
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.

When you select this option, a confirmation window appears that contains a check box labeled "Delete its members (that are not in another hierarchy)." If you leave this check box empty and click **Yes**, then the selected hierarchy is deleted but all its members remain in the dimension. If you select this check box and click **Yes**, then the selected hierarchy is deleted and all members of the selected hierarchy that are not in any other hierarchy are deleted from the dimension.

Expand All
expands the entire hierarchy.

Collapse All
collapses all hierarchies completely.

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 the properties of the selected hierarchy and to change some of them.

Member Options

When you select a member, the following options are available:

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 child of the selected member.

You can either 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.

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

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.

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, enabling you to specify the target location.

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

Remove
removes the selected member from the selected hierarchy. Members subordinate to the selected member are also removed, if there are any.

A member that is removed from a hierarchy remains in the hierarchy's dimension. To delete it from the dimension, use the **Delete** option in the Members view, after removing the member from all hierarchies.

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.

Members
opens the Members view for the selected dimension, which lists all the members in the 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 the Custom Properties window, enabling you to make changes to the custom properties of all the selected members.

Create a Hierarchy

To create a hierarchy, use one of the following methods:

- Build the entire new hierarchy one member at a time.
- Copy and then modify an existing hierarchy.

To build an entire hierarchy:

1. With either a hierarchy or a member selected, select **New Hierarchy** to launch the New Hierarchy wizard.
2. Use the New Hierarchy wizard to create an empty hierarchy.
3. Build the hierarchy one member at a time.

To copy and then modify an existing hierarchy:

1. Select the hierarchy that you want to copy.
2. Select **Copy Here** to launch the Copy Hierarchy wizard.
3. In the Copy Hierarchy wizard, provide a code, name, and description for the copy.

   The copy is created and added to the bottom of the display.
4. Modify the copy as appropriate.

Modify a Hierarchy

To modify an existing hierarchy, use the following options:

Add Member
launches a wizard that enables you to insert a member at the selected location.

Properties
displays the Properties window, which enables you to change certain properties of the selected hierarchy or the selected member.

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, enabling you to specify the target location.

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

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

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.

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

Remove
removes the selected member from the selected hierarchy.

A member that is removed from a hierarchy remains in the hierarchy's dimension. To delete it from the dimension, use the **Delete** option in the list of members, after removing the member from all hierarchies.

Delete a Hierarchy

To delete a hierarchy:

1. Select the hierarchy.
2. Select **Delete**.
3. Click **Yes** in the deletion confirmation window.

Move a Member within a Hierarchy Using Drag-and-Drop

You can use drag-and-drop to move a member from one location in a displayed hierarchy to another location in the same hierarchy. This is an alternative to the **Move**, **Move Up**, and **Move Down** options.

To begin a drag-and-drop operation, click the code of a member and hold the mouse button down. As you drag a member, any member that marks a possible drop location actually marks three possible drop locations, which correspond to the top, middle, and bottom thirds of the member:

- When the cursor is over the top third of a member, a bold horizontal line appears immediately above the member. If you release the mouse button here, the dragged member is inserted immediately before the member that marks the drop location, and at the same hierarchical level.
- When the cursor is over the middle third of a member, the member is highlighted. If you release the mouse button here, the dragged member is inserted as a child of the member that marks the drop location.

  For analysis and currency hierarchies, which must be flat, this possibility is not available.
- When the cursor is over the bottom third of a member, a bold horizontal line appears immediately below the member. If you release the mouse button here, the dragged member is inserted immediately after the member that marks the drop location, and at the same hierarchical level.

As you drag a member, you might discover that the desired drop location is not visible. You can make any possible drop location visible using the following techniques:

- To scroll the hierarchy down, drag a member to the bottom of the hierarchy display and wait while the display scrolls down.
- To scroll the hierarchy up, drag a member to the top of the hierarchy display and wait while the display scrolls up.
- To expand a collapsed portion of the hierarchy, drag a member over the code of a collapsed member and wait while the collapsed member is expanded.

See Also

- Using Views on page 20
- Working with Dimensions, Hierarchies, and Members on page 26

Members View

Overview of the Members View

The Members view displays a list of all the members in the dimension that is identified in the view's tab.

Some dimensions cannot be edited. For such a dimension, the options that modify members are not available.

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 the Custom Properties window, enabling you to make changes to the custom properties of all the selected members.

Add a Member to a Hierarchy Using Drag-and-Drop

You can drag-and-drop a member in the Members view for a certain dimension into a hierarchy in the Hierarchies view for the same dimension. The dragged member must not be in the target hierarchy already.

Before you begin a drag-and-drop operation, make sure that the appropriate Hierarchies view is visible and that the target hierarchy is expanded. To begin a drag-and-drop operation, click the code of a member in the Members view and hold the mouse button down. As you drag the member into the Hierarchies view, any member of the target hierarchy that marks a possible drop location actually marks three possible drop locations, as follows:

- When the cursor is over the top third of a member, a bold horizontal line appears immediately above the member. If you release the mouse button here, the dragged member is inserted immediately before the member that marks the drop location, and at the same hierarchical level.
- When the cursor is over the middle third of a member, the member is highlighted. If you release the mouse button here, the dragged member is inserted as a child of the member that marks the drop location.

For analysis and currency hierarchies, which must be flat, this possibility is not available.

- When the cursor is over the bottom third of a member, a bold horizontal line appears immediately below the member. If you release the mouse button here, the dragged member is inserted immediately after the member that marks the drop location, and at the same hierarchical level.

As you drag a member, you might discover that the desired drop location is not visible. You can make any possible drop location visible using the following techniques:

- To scroll a hierarchy down, drag a member to the bottom of the hierarchy display and wait while the display scrolls down.
- To scroll a hierarchy up, drag a member to the top of the hierarchy display and wait while the display scrolls up.
- To expand a collapsed portion of a hierarchy, drag a member over the code of a collapsed member and wait while the collapsed member is expanded.

See Also

- Using Views on page 20
- Working with Dimensions, Hierarchies, and Members on page 26

Using Custom Properties

Reasons for Using Custom Properties

In the SAS Financial Management Add-In for Microsoft Excel, 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 also play a key role in operational planning. When you create a form set for operational planning, you must select a custom property whose values group the members of the form set's workflow hierarchy. You can then define a different level of detail for the forms in each group. Therefore, before you create an operational form set, at least one custom property must be defined. In addition, the members in the hierarchy that will be the form set's target hierarchy must have appropriate values for the custom property.

Ways to Define Custom Properties

Custom properties can be defined in the Custom Properties view. They can also be defined in the Cross Industry Detail Data Store and loaded from there as part of the operation of loading members into a dimension. New custom properties that are loaded from the Cross Industry Detail Data Store are added to the list in the Custom Properties view. If a custom property that is loaded from the Cross Industry Detail Data Store has the same code as a custom property that is already in the Custom Properties view, then the newly loaded custom property replaces the previously existing custom property that has the same code.

It is generally best to load custom properties from the Cross Industry Detail Data Store, because defining custom properties in the Custom Properties view has two disadvantages:

- If you subsequently load that dimension with members from the Cross Industry Detail Data Store, then the custom properties that were defined in the Custom Properties view are lost.
- If you export the dimension using the Export Dimension wizard, then the custom properties that were defined in the Custom Properties view are not exported with it.

For any dimension whose members you define in the Hierarchies view, any necessary custom property values of those members must be defined in the Custom Properties view.

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 **Custom Properties** and use the Custom Properties window.

  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.

See Also

Custom Properties View on page 35

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.

See Also

- Using Views on page 20
- Using Custom Properties on page 34

Using Data Locales

How Data Locales Work

Data locales give users a choice of languages for the names and descriptions of the following objects:

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

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

- language
- country/region

  The country/region component is usually a country, but it can be a geographical region that is not 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 desktop locale that is selected in the desktop operating system. The data locales are ranked according to how well they match the desktop locale, as follows:

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

The data locale that matches the desktop locale best is known as the current data locale. The current data locale is identified in the bottom margin of the main window.

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

If there is no defined data locale that matches the user's desktop locale at all, then the text from the designated default data locale is displayed. The default data locale is available on any desktop from which you log on to the SAS Financial Management server. In the Data Locales view, the icon for the default data locale has a black ring ( ).

Defining Data Locales

You can define data locales in two ways:

- Populate the CODE_LANGUAGE table in the Cross Industry Detail Data Store, and then run the Import Locales job to load data locale information from the Cross Industry Detail Data Store.
- Select **Tools** ⇨ **Data Locales** and use the Data Locales view.

In general, it is best to use the CODE_LANGUAGE table and the Import Locales job, because a data locale must be defined in the Cross Industry Detail Data Store to enable you to load member and hierarchy names and descriptions for that data locale from the Cross Industry Detail Data Store.

For details about loading data through the Cross Industry Detail Data Store, see *SAS Solutions Services: Data Administration Guide*. You can obtain this document at the following Web location: `http://support.sas.com/documentation/`. Select the product that you installed to access its documentation.

See Also

Data Locales View on page 37

Data Locales View

The Data Locales view lists the available data locales.

The default data locale is marked by the icon with a black ring (). The default data locale is the data locale that is used if no data locale matches your desktop locale or if no data locale that matches your desktop locale has text for a particular name or description.

The current data locale is identified in the bottom margin of the main window. The current data locale is the data locale that matches your desktop locale best and that is therefore ranked first on your desktop. If the current data locale has text for a given name or description, then the text for the current data locale is displayed.

The following options are available:

New Data Locale
launches the New Data Locale wizard, which 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, you must close and reopen SAS Financial Management Studio.

Delete
deletes the selected data locale.

Properties
displays the properties of the selected data locale.

See Also

- Using Views on page 20
- Using Data Locales on page 36

Using Calendar Templates

Utility of Calendar Templates

Using a calendar template, you can add a full year of time periods to a time hierarchy in a single operation. If you load all your time periods from the Cross Industry Detail Data Store, 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.

See Also

Calendar Templates View on page 39

Calendar Templates View

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

New Calendar Template
launches the **New Calendar Template** wizard, which enables you 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, which enables you to create a calendar template using the selected calendar template as a source of default specifications.

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.

See Also

- Using Views on page 20
- Using Calendar Templates on page 38

Chapter 4

Survey of Required Dimension Types

Introduction to Required Dimension Types 42
Introduction to the Account Dimension Type 42
Introduction to Account Types 43
Asset Account Type 44
Cumulative Translation Adjustment Account Type 44
Equity Account Type 45
Expense Account Type 45
Liability Account Type 46
Retained Earnings Account Type 46
Revenue Account Type 47
Statistical Account Type 47
Statistical Balance Account Type 48
Statistical Flow Account Type 48
Balance Type Property of Accounts 48
Exchange Rate Type Property of Accounts 48
Intercompany Property of Accounts 49
Introduction to the Organization Dimension Type 49
Organization Details 50
Introduction to the Analysis Dimension Type 50
Introduction to the Currency Dimension Type 51
Introduction to the Frequency Dimension Type 51
Introduction to the Source Dimension Type 52
Introduction to the Time Dimension Type 53
Introduction to the Trader Dimension Type 54

Introduction to Required Dimension Types

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

- Account on page 42
- Analysis on page 50
- Currency on page 51
- Organization on page 49
- Source on page 52
- Time on page 53
- Trader on page 54

Operational cycles do not use the Source and Trader dimension types. Each data value that is stored in an operational cycle must belong to a crossing that includes a member from each of the following required dimension types:

- Account
- Analysis
- Currency
- Organization
- Time

In addition to the seven required dimension types for a financial cycle or the five required dimension types for an operational 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 financial cycle that uses only the seven required dimension types, and another financial cycle that uses eight dimension types. However, you cannot mix seven-dimensional data records and eight-dimensional data records in the same cycle.

There is another required dimension type that is not used to store data in cycles, but is used to display data in the SAS Financial Management Add-In for Microsoft Excel and the Web-based Form Editor. This is the Frequency dimension type.

Introduction to the Account Dimension Type

The Account dimension type can contain any number of dimensions. Each dimension can contain any number of 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

There are some exceptions:

- Accounts that belong to the Retained Earnings account type do not have exchange rate types.
- 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.

You supply values for the relevant properties when you create an account with the New Member wizard. You can view or edit the account-specific properties of an account on the **Account Details** tab of the account's Properties window.

Generally speaking, the members of an account hierarchy represent general ledger accounts and categories of general ledger accounts. An account hierarchy can have any number of top-level members. 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 particular accounts for which data is loaded from the Cross Industry Detail Data Store or entered through forms, such as Cash and Marketable Securities, Interest Income, or Cost of Goods Sold. Intermediate-level members can be used for subcategories of accounts, such as Current Assets and Fixed Assets. Each non-leaf member of an account hierarchy has a virtual child (vc) member associated with it, which can be used to load or enter data at that point in the hierarchy.

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. A simple example is a formula that computes the profit margin by dividing aggregate profit by aggregate revenue.

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 the way in which that crossing's value is computed and the way in which that crossing's value is used in other computations. Account types fall into three categories—balance, flow, and special.

The balance account types are the following:

- Asset on page 44
- Liability on page 46
- Equity on page 45
- Statistical Balance on page 48

The flow account types are the following:

- Revenue on page 47
- Expense on page 45
- Statistical Flow on page 48

The special account types are the following:

- Cumulative Translation Adjustment on page 44

- Retained Earnings on page 46
- Statistical on page 47

Asset Account Type

An account that belongs to the Asset account type represents a balance sheet asset. Typical examples are cash, accounts receivable, and fixed assets.

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.

See Also

Frequency Conversion on page 157

Cumulative Translation Adjustment Account Type

An account that belongs to the Cumulative Translation Adjustment (CTA) account type is used to compute the cumulative translation adjustment that typically accompanies the currency conversion of a balance sheet.

If you import accurate data, then a balance sheet for a given leaf organization is in balance in that organization's functional currency. But it might not be in balance in any other currency, unless a translation adjustment is included. This is because different exchange rates are often used to convert different balance sheet accounts. The following pattern is typical:

- Asset and liability accounts are converted using the appropriate Period Close exchange rate.
- The retained earnings account reflects a history of conversions of net income using Period Average exchange rates for a series of time periods.
- The proceeds of a past common stock offering are converted using a Historic exchange rate.

Every CTA account has the Period Close exchange rate type; you cannot select any other exchange rate type when you create or edit a CTA account. The Period Close exchange rate type of a CTA account has a special purpose: it determines the so-called peg rate for the CTA computation.

If the peg rate were used to convert all the balance sheet accounts, then the converted balance sheet would be in balance. For each account that is converted using some other exchange rate, the software computes the difference between the actual converted value and the hypothetical converted value if the peg rate were used. The software then combines these differences to yield the cumulative translation adjustment, which brings the converted balance sheet into balance.

In addition to a peg rate, the CTA computation needs access to the complete set of accounts that make up the balance sheet. You provide this information by designating source accounts for the CTA account. You can designate every relevant leaf account, but a better approach is to designate non-leaf accounts that span the relevant leaf accounts. For

example, depending on the structure of your account hierarchy, you might be able to span all the balance sheet accounts by designating three source accounts—Total Assets, Total Liabilities, and Total Equity. This approach makes the job of specifying the source accounts much easier. It also insulates the specification from any changes that might be made in the set of leaf accounts.

CTA computations are performed only for leaf organizations. Balance sheets of higher-level organizations are compiled through the hierarchical roll-up process, possibly including an ownership rule.

See Also

- Computing Cumulative Translation Adjustments on page 163
- Frequency Conversion on page 157

Equity Account Type

An account that belongs to the Equity account type represents a balance sheet equity item. Typical examples are common stock and retained earnings. The Equity account type should also be used for the account that holds the grand total of all liabilities and all equity.

For an Equity 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 Equity account is Period Close.

There is an important difference between a retained earnings account that has the Equity account type and an account that has the Retained Earnings account type:

- A retained earnings account that has the Equity account type holds cumulative retained earnings for the life of the firm.
- An account that has the Retained Earnings account type holds an increment to retained earnings for a specified time span, which is computed from revenue, expense, and other relevant accounts for that time span.

See Also

Frequency Conversion on page 157

Expense Account Type

An account that belongs to the Expense account type represents an income statement expense item. Typical examples are salary expense and interest expense.

For an Expense account, users are often interested in the values for both Period Activity and To Date frequency members.

The default exchange rate type for an Expense account is Period Average.

See Also

Frequency Conversion on page 157

Liability Account Type

An account that belongs to the Liability account type represents a balance sheet liability. Typical examples are accounts payable and notes payable.

For a Liability 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 a Liability account is Period Close.

See Also

Frequency Conversion on page 157

Retained Earnings Account Type

An account that belongs to the Retained Earnings account type is used to compute a change in retained earnings.

The computation combines the values of a set of input crossings that you designate by specifying time periods, members of the Source hierarchy, and contributing accounts such as Net Income and Dividends. These specifications correspond to the following special account properties:

Roll-Forward Method
: specifies the time span that the computed change in retained earnings covers.

Basis Data
: specifies the members of the Source hierarchy that the computed change in retained earnings includes.

Source Accounts
: specifies the complete set of accounts whose values contribute to the computed change in retained earnings. You can list every relevant leaf account, but a better approach is to list a set of non-leaf accounts that spans all the relevant leaf accounts. For example, depending on how your account hierarchy is structured, you might be able to list only Net Income. This makes the task of specifying the source accounts much easier. It also insulates the specification from any changes that might be made in the set of underlying leaf accounts.

An account that belongs to the Retained Earnings account type does not have an exchange rate type. This is because the value of such an account is computed separately for each currency, from the currency-converted source accounts.

The main use of an account that belongs to the Retained Earnings account type is to update the balance sheet retained earnings account, which typically belongs to the Equity account type. It is important to understand the difference between these two types of accounts and the way they work together in retained earnings computations.

See Also

- Introduction to the Source Dimension Type on page 52
- Computing Retained Earnings on page 161
- Frequency Conversion on page 157

Revenue Account Type

An account that belongs to the Revenue account type represents an income statement revenue item, such as sales of a particular product or all products in a certain category. The Revenue account type should also be used for the Net Income account.

For a Revenue account, users are often interested in the values for both Period Activity and To Date frequency members.

The default exchange rate type for a Revenue account is Period Average.

See Also

Frequency Conversion on page 157

Statistical Account Type

An account that belongs to the Statistical account type can serve one of these purposes related to formulas:

- Holding the output of a reporting formula. Typical examples are standard analytic ratios such as profit margin, debt-to-equity ratio, and revenue per employee.
- Holding a constant value that is an input to a driver formula or modeling formula. An example is a unit price that is multiplied by a number of units to yield the revenue for a product. Statistical accounts can be used in this way because they do not participate in the hierarchical roll-up process.

  This use of a Statistical account is an alternative to using the DRATE function to access constant values that are stored in driver rate sets. Compare these two methods of providing constant inputs to formulas before selecting one.

A crossing that contains a Statistical account displays the same value no matter what frequency member it contains.

The default exchange rate type for a Statistical account is None.

See Also

- Formula Types on page 169
- Frequency Conversion on page 157

Statistical Balance Account Type

An account that belongs to the Statistical Balance account type is normally used for a nonmonetary quantity that pertains to a point in time. Typical examples are headcount and the number of units of a product that are in inventory.

For a Statistical Balance 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 a Statistical Balance account is None.

See Also

Frequency Conversion on page 157

Statistical Flow Account Type

An account that belongs to the Statistical Flow account type is normally used for a nonmonetary quantity that pertains to a period of time. Typical examples are the number of new hires and the number of units of a product that are shipped.

For a Statistical Flow account, users are often interested in the values for both Period Activity and To Date frequency members.

The default exchange rate type for a Statistical Flow account is None.

See Also

Frequency Conversion on page 157

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 to 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 financial cycles but not used by operational cycles. Therefore, the intercompany property has no significance in operational cycles, models, and form sets.

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 command and reporting structure of an enterprise. It can have any number of top-level members.

Each organization either is or is not a reporting entity; this is an important distinction with a variety of implications.

There are special constraints when you design an organization hierarchy for use with an ownership rule.

Typically, data is loaded or entered for leaf organizations, while organizations at higher points in the hierarchy represent pure aggregation points. However, each non-leaf member

of an organization hierarchy has a virtual child (vc) member associated with it, which can be used to load or enter data at that point in the hierarchy.

See Also

- Organization Details on page 50
- Ownership Rules: Organization Hierarchy Requirements on page 108

Organization Details

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

Reporting Entity
determines whether an organization is available for selection as a reporting entity in those contexts that require a reporting entity. These contexts include the following:

- In a balanced manual adjustment, you must specify the reporting entity within which the components of the adjustment must balance.
- In an ownership rule, only reporting entities can participate in the ownership relationships that you define.

An organization is a reporting entity if and only if the **Reporting Entity** check box is selected.

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
is the currency in which the organization conducts its business. Uses of the functional currency include the following:

- Typically, every monetary value that is loaded into a cycle uses the functional currency of the relevant organization.
- In a form set that is based on an organization hierarchy, the data entered through a form is stored in the functional currency of the organization that the form is assigned to.
- An organization's functional currency is used by default for manual adjustments that are posted to that organization.
- All rule-generated adjustments that are posted to an organization use the organization's functional currency, with the exception of adjustments that are generated by allocation rules.

Introduction to the Analysis Dimension Type

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

Analysis members are typically used to distinguish the different relationships that numeric values can have to the real world, such as describing the past, predicting the future, specifying a goal, projecting the likely consequences of a contemplated action, and so on. Some standard examples are ACTUAL, FORECAST, and BUDGET. For an iterative budget process, you might use BUDGET1, BUDGET2, and BUDGET3.

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.

If you use more than one currency member, then you need to maintain exchange rates between pairs of currencies. Exchange rates are maintained in 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. You can use the NONE currency to display together in a convenient manner the numeric values for many organizations that have different functional currencies. For example, you might use such a display to verify that the base data for a variety of leaf organizations has been correctly loaded from the Cross Industry Detail Data Store. Note, however, that the NONE currency yields meaningless values for any organization that aggregates values from two or more organizations that have different functional currencies. For example, in a display that uses the NONE currency, 3 dollars in one organization and 4 euros in another organization would be aggregated to a meaningless 7 in their common parent.

See Also

Organization Details on page 50

Introduction to the Frequency Dimension Type

The Frequency dimension type contains a single Frequency dimension that contains a single predefined, flat Frequency hierarchy. Each member of the Frequency dimension is characterized by 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

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**.

Frequency members are not used in stored data records. All numeric values that are stored in cycles are implicitly associated with the Period Activity member. In tables in the SAS Financial Management Add-In for Microsoft Excel or the Web-based Form Editor,

Frequency members are sometimes needed to differentiate two numeric values whose crossings are identical in every other dimension type. A standard example is the distinction between revenue for a given month and revenue for a given year through a given month. The relevant month is the same for both of these numeric values, so they are not differentiated in the Time dimension type. In the Frequency dimension type, however, revenue for a given month is associated with Month To Date or Month Activity, whereas revenue for a given year through a given month is associated with Year to Date or Year Activity.

Frequency conversion from the Period Activity values that are stored in cycles is an important step in the computation of displayed values for various members of the Frequency dimension

See Also

Frequency Conversion on page 157

Introduction to the Source Dimension Type

The Source dimension type contains a single Source dimension, which 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. Operational cycles do not use the Source dimension type.

Members of the Source hierarchy cannot carry formulas.

The name of the Source hierarchy reflects the fact that its leaf members represent the different ways in which numeric values can enter the system. The structure of the Source hierarchy enables numeric values that come from these various sources to be combined.

The top member of the Source hierarchy, which is named Total, marks the grand total of all the numeric values that are associated with the subordinate members. Below the Total member, the Source hierarchy has the following members structured in the following way:

- CTA—the computed value of an account that belongs to the Cumulative Translation Adjustment (CTA) account type.

  Crossings that combine the CTA member with any account that is not a CTA account have no meaning and never hold numeric values.

- RE—the 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.

  Crossings that combine the RE member or one of its descendants with any account that is not an RE account have no meaning and never hold numeric values.

  - REElim—the portion of the value of a Retained Earnings account that comes from Elim input crossings.
  - REAdj—the portion of the value of a Retained Earnings account that comes from CPO, Alloc, Reclass, Bal, or Manual input crossings.
  - REData—the sum of the subordinate REForms and REImport values.
    - REForm—the portion of the value of a Retained Earnings account that comes from BaseForm input crossings.

Note that 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 Base or BaseJourn input crossings.

  Note that 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 trades.
- TotalBeforeElim—the subtotal that combines all base data, all manual adjustments, and all adjustments that are generated by adjustment rules.
  - Adj—the subtotal of all manual adjustments and all adjustments that are generated by adjustment 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—the subtotal of the subordinate BaseForm and TotalAfterImport values.
    - BaseForm—The numeric values at BaseForm crossings depend on choices that you make when you load data into the cycle. If you always load data to Base and BaseJourn, then BaseForm values are the values that are entered through forms. If you always load data to BaseForm, then BaseForm values are the sum of values that are entered through forms and values that are loaded.
    - TotalAfterImport—the subtotal of the subordinate BaseJourn and Base values.

      BaseJourn—The numeric values at BaseJourn crossings depend on choices that you make when you load data into the cycle. If you always load data to Base and BaseJourn, then BaseJourn values are the values that are loaded from the GL_JRNL_DETAILS table in the Cross Industry Detail Data Store. If you always load data to BaseForm, then BaseJourn crossings do not have values.

      Base—The numeric values at Base crossings depend on choices that you make when you load data into the cycle. If you always load data to Base and BaseJourn, then Base crossings have values. If you always load data to BaseForm, then Base crossings do not have values.

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 that numeric data values are associated with. Data can be loaded or entered only for leaf time periods; higher-level time periods do not have virtual children.

For financial reporting, time hierarchies typically have a three-level structure of fiscal years, quarters, and months. Under such a structure, financial data is loaded or entered for each month, and rolled up to yield quarterly and yearly figures.

Time members cannot carry formulas.

See Also

Using Calendar Templates on page 38

Introduction to the Trader Dimension Type

In a financial cycle, the Trader dimension type mirrors the Organization dimension type. Each trader hierarchy is an automatically generated copy of a corresponding organization hierarchy. Operational cycles do not use the Trader dimension type.

Two special members, which are kept hidden in all organization hierarchies, are available for display in all trader hierarchies. The two special members are a unique top-level or root member whose code is ALL and a leaf member directly underneath ALL whose code is EXT.

Any record that describes a financial relationship between two organizations in an organization hierarchy must be eliminated when those organizations are consolidated into their common parent organization. For example, sales made by a European subsidiary to a North American subsidiary do not represent sales of the global corporation that owns both subsidiaries, and so they must be eliminated at the global level. The Trader dimension type is used to identify the other participating organization in such two-organization records. It is also used to distinguish these two-organization records from the single-organization records that do not get eliminated in the consolidation process.

In a single-organization record, the trader member is the special member EXT. In a two-organization record, the trader member always identifies the second organization. For example, a record that represents sales made by a European subsidiary to a North American subsidiary would identify the European subsidiary in the organization dimension and the North American subsidiary in the trader dimension.

For any organization dimension, information about the special members of the trader hierarchies in the corresponding trader dimension is displayed on the **Trader** page of the organization dimension's Properties window.

See Also

Introduction to the Organization Dimension Type on page 49

Chapter 5

Cycles and Cycle Periods

Introduction to Cycles **55**
Overview 55
Features of a Cycle 56
Data in a Cycle 56
Objects that Belong to Cycles 56
Being Open 57

Cycle Types **57**
Overview 57
Reporting 57
Planning 58

How Many Cycles Do You Need? **59**

Cycle Manager Window **59**

Create a Cycle **60**

Open a Cycle **60**

Structure of the Data Records in a Cycle **61**

Managing Data in a Cycle **62**
Overview of Managing Data in a Cycle 62
Loading Data from an External Database 62
Entering Data through Forms 62
Loading Data into a Financial Cycle from a Model 62
Loading Data into a Financial Cycle from an Operational Cycle 63
Deleting Data from a Cycle 63
Locking Data in a Cycle 63

Periods View **64**

Introduction to Cycles

Overview

You can think of a cycle as a large container. A cycle is a container for stored data and objects of several types, including exchange rate sets, driver rate sets, models, and form sets.

Features of a Cycle

The following features of a cycle are specified when you create the cycle using the New Cycle wizard:

- a name and description
- the cycle type—financial or operational
- the dimensions that are represented in the cycle's data records

  You must select a dimension for each required dimension type. If other dimension types have been defined at your site, then you can choose to include dimensions from them as well.
- a time hierarchy and a set of leaf time periods from it

  All data records that are stored in the cycle must use one of these time periods.
- certain details concerning the accounting logic that is used to process the data that is stored in the cycle (financial cycles only)

The following features of a cycle can be specified only after the cycle is created:

- attachments

  In the Cycle Manager window, select a cycle and click **Attachments**.
- security restrictions

  In the Cycle Manager window, select a cycle and click **Properties**. In the Properties window, select the **Security** page.

  You can deny access to the cycle to designated users and groups. A user who is denied access to a cycle has no access to the cycle, the data that is stored in the cycle, or any of the objects that belong to the cycle.

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 that the form belongs to, although this fact is not apparent to the user.

See also:

- Structure of the Data Records in a Cycle on page 61
- Managing Data in a Cycle on page 62

Objects that Belong to Cycles

Each object of any of the following types belongs to a cycle:

- Exchange rate sets contain currency exchange rates that are used to convert numeric values from one currency to another.

  To work with exchange rate sets, use the **Exchange Rates** view in the Rates workspace.
- Driver rate sets contain driver rates. Driver rates are numeric values that are associated with specific combinations of members and that can be used in formulas.

  To work with driver rate sets, use the **Driver Rates** view in the Rates workspace.

- Models are complex objects that mediate between the numeric values that users can see and the more limited sets of numeric values that are stored in data records. A model is at work whenever a report is generated from stored data records and whenever new data records are created from the numeric values that users enter into data-entry forms.

  To work with models, use the Models workspace.

- Form sets provide the framework for data-entry projects. Each form set includes a target hierarchy and a form template. When you define a form set, you select members of the target hierarchy to participate in a data-entry project and the nature of each person's participation. The form template defines the data-entry forms that the participants use.

  To work with form sets, use the Forms workspace.

Being Open

To load data into a cycle or to delete data from a cycle, you must have the cycle open. To create an object that belongs to a cycle or to work with any existing object that belongs to a cycle, you must have the cycle open. Only one cycle can be open at a time. The currently open cycle is identified in the blue banner immediately below the menu bar. If this banner is blank, then no cycle is open.

There are some things that you can do to a cycle whether the cycle is open or not. You can perform these tasks in the Cycle Manager window. To display the Cycle Manager window, select **Tools** ⇨ **Cycles**. You can use the **File** menu to perform most of these tasks for the currently open cycle.

For details about the tasks that do not require a cycle to be open, see Cycle Manager Window on page 59.

Cycle Types

Overview

There are two types of cycles:

- financial reporting and planning cycles (financial cycles)
- operational planning cycles (operational cycles)

When you create a cycle, you must specify which type of cycle you are creating. To decide whether to create a financial cycle or an operational cycle, consider the following comparisons.

Reporting

Most reporting capabilities are available only for financial cycles. Specifically, financial cycles and operational cycles have the following differences that concern reporting capabilities:

- When you create read-only tables or CDA tables in the SAS Financial Management Add-In for Microsoft Excel, you can select only models that retrieve data from financial cycles.
- The accounting logic that is important for most financial reports—including manual adjustments, adjustment rules, retained earnings computations, cumulative translation

adjustment computations, and intercompany eliminations—is available only in financial cycles.

- The data records in operational cycles do not have a field for the Trader dimension type.
- Models that belong to operational cycles cannot include manual adjustments or adjustment rules. In the Models workspace, all the options for working with manual adjustments and adjustment rules are absent if the open cycle is an operational cycle.

It is possible to include a very simple report in an operational form set by designing a data-entry table that has only protected cells that display stored data.

Planning

Both financial cycles and operational cycles give you the ability to create form sets that enable users to enter data into the cycle.

Operational form sets provide a potential performance advantage that can be important for forms that contain a very large number of crossings. The main reason for creating an operational cycle is to obtain the performance advantage of operational form sets.

You might prefer to use a financial form set for one or more of the following reasons:

- To enter data into an operational form, a user must use the Web-based Form Editor. Financial form sets can be designed to permit data entry using the Web-based Form Editor, or the Excel Add-In, or both.
- Supplemental schedules, including supplemental schedules that retrieve data from SAS Human Capital Management, are available only in the Excel Add-In. Therefore, an operational form set cannot include a supplemental schedule.
- An operational form can have only one author or reviewer associated with it. A financial form can have many authors or reviewers associated with it.
- A form in an operational form set ignores all reporting formulas and all modeling formulas. It also ignores all driver formulas that include any reference to a member of any dimension other than the dimension of the member that carries the formula. Only driver formulas that have no cross-dimensional references are executed in an operational form. In a financial form, formulas of all types can be executed.
- In order to produce read-only tables or CDA tables that display data that was entered by means of an operational form set, you must use the Load Operational Data wizard to load the data into a financial cycle. With a financial form set, you can produce reports from the same financial cycle that the data is entered into.

To decide whether to use a financial form set (and therefore a financial cycle) or an operational form set (and therefore an operational cycle), you should weigh all these considerations as they apply to a given case.

See Also

Comparing Financial and Operational Planning on page 6

How Many Cycles Do You Need?

There is no limit to the number of cycles that you can create, but you might find that you need very few. You might even find that you can conveniently maintain all your data in a single cycle.

Here are some reasons why you might need to create additional cycles:

- If you need to do reporting and you also need to create form sets with very large numbers of crossings, then you need at least one cycle of each type—a financial cycle for reporting and an operational cycle to improve the performance of the form sets.
- If you need to load or enter data for different time period types (both months and quarters, for example), then you need different cycles that use different time hierarchies.
- If you need to load corrected data while keeping the uncorrected data in the system, you can accomplish this by loading uncorrected data into one cycle and corrected data into another cycle.
- You might want to create a cycle that is dedicated to a special project.

Cycle Manager Window

Use the Cycle Manager window to perform cycle tasks that do not require a cycle to be open.

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

The tasks are represented by the buttons on the right side of the window:

New
launches the New Cycle wizard, which enables you to create a new cycle by providing a full set of specifications for it.

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

Copy
launches the Copy Cycle wizard, which enables you to create a new cycle that has the same set of dimensions as the selected cycle, but can differ from the selected cycle in other ways.

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

This button is active only if exactly one cycle is selected.

Delete
deletes the selected cycle or cycles.

A deletion confirmation window appears. You must click **Yes** in the deletion confirmation window to complete the deletion.

You cannot delete a cycle that has a model. You cannot delete a cycle that another user has open.

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. Data cannot be added to or deleted from a locked cycle.

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

See Locked Objects on page 16.

Unlock
unlocks the selected cycle or cycles that are currently locked.

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

See Locked Objects on page 16.

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

This button is active only if exactly one 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 exactly one cycle is selected.

To select a single cycle, click it. To select a solid block of cycles, click the cycle at one end of the block, and then click the cycle at the other end of the block while holding down the SHIFT key. To select any set of cycles, click each cycle in the set while holding down the CTRL key.

Create a Cycle

To create a 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 about the New Cycle wizard, see the online Help for the individual wizard pages.

Open a Cycle

To open a cycle:

1. Select **File** ⇨ **Open Cycle**.

   The Open Cycle window appears, displaying a list of all existing cycles that you have access to.

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

   The selected cycle is opened. If a different cycle was open, it is closed.

If no cycle is open when you select the Periods, Rates, Models, or Forms workspace, then the default view in the workspace displays the message "This view requires a cycle." Below this message, there are two radio buttons that 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 about the wizard, see the online Help for the individual wizard pages.

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

  The selected cycle is opened.

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.

Structure of the Data Records in a Cycle

Each data record in a cycle has several dimension fields and one field that contains a numeric value. Each dimension field contains a member of the dimension that defines the field. The collection of all the members in a given record constitutes a crossing that determines what the numeric value in that record measures or represents.

Every data record in every cycle contains a dimension field for each of the following dimension types:

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

  In an operational cycle, the Source dimension field of every record contains the BaseForm member and none of the other Source dimension members are used in computations. In effect, this field is functionally absent even though it is physically present.

Every data record in a financial cycle also contains a dimension field for the Trader dimension type.

In addition to these required dimension types, every data record in a particular cycle contains a dimension field for each optional dimension type that you specified when you created that cycle.

Managing Data in a Cycle

Overview of Managing Data in a Cycle

You can load data into any cycle from an external database, through the Cross Industry Detail Data Store. You can load data into a financial cycle from a financial cycle or from an operational cycle. Users can enter data into any cycle through forms. You can delete data from any cycle. Managing the data in a cycle typically involves coordinating several of these activities with each other and with other business processes outside SAS Financial Management. This might mean setting up a monthly schedule.

You can lock an entire cycle or any subset of its periods to prevent the loading, entering, or deleting of data. It is a good practice to lock the data in a financial cycle before you create tables that present the data to users. It is a good practice to lock the data in an operational cycle before you load that data into a financial cycle.

Loading Data from an External Database

To load data from any external database into any financial cycle or any operational cycle:

1. Load the data into the Cross Industry Detail Data Store.

   For details, see *SAS Solutions Services: Data Administration Guide* (see Additional Documentation on page 7).

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 about the Load New Data wizard, see the online Help for the individual wizard pages.

Entering Data through Forms

To enable users to enter data into any financial cycle or any operational 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 about the New Form Set wizard, 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. It replaces any data that was previously associated with the same crossings. All data entered through forms is associated with the BaseForm member of the Source hierarchy.

Loading Data into a Financial Cycle from a Model

To load data into a financial cycle from a model that belongs to a financial cycle:

1. With the target financial 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 about the Load New Data wizard, see the online Help for the individual wizard pages.

Loading Data into a Financial Cycle from an Operational Cycle

To load data into a financial cycle from an operational cycle:

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

   For details about the Load Operational Data wizard, 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 concerning which data to delete from the crossings that will receive 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 about the Delete Data wizard, see the online Help for the individual wizard pages.

Locking Data in a Cycle

Using a **Lock** option, you can lock an entire cycle or any set of periods within a cycle. If a cycle or a period of a cycle is locked, then no one can add data to it or remove data from it. You should always lock data before you generate final reports from that data.

The data in a financial cycle is available to queries that use a model that belongs to the cycle. For example, when you create a read-only table using the SAS Financial Management Add-In for Microsoft Excel, the numbers in the resulting report come from the financial cycle that underlies the model on which the read-only table is based. Some numbers in the report might come directly from the underlying cycle, while others are produced by computations whose inputs come from the underlying cycle.

See Also

- Loading Data from the Cross Industry Detail Data Store on page 15
- Locked Objects on page 16

Periods View

The Periods view lists all the time periods in the open cycle. Its main use is loading data into the open cycle. You can load data into an open financial cycle from the Cross Industry Detail Data Store, a financial cycle, or an operational cycle. You can load data into an open operational cycle from the Cross Industry Detail Data Store.

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.

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

Load New Data
launches the Load New Data wizard, which enables you to load data into the open cycle from the Cross Industry Detail Data Store.

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

This option is active only if at least one unlocked time period is selected. To make your work in the wizard simpler, select the exact set of time periods to which you want to load data before you launch the wizard.

The Load New Data wizard is slightly different for financial cycles and operational cycles. The financial-cycle variant is equivalent to the Load Base Data job in SAS Data Integration Studio. The operational-cycle variant is equivalent to the Load OP Data job in SAS Data Integration Studio.

For details about the SAS Data Integration Studio jobs, see *SAS Solutions Services: Data Administration Guide*.

Load Model Data
launches the Load Model Data wizard, which enables you to load data into the open financial cycle from any financial cycle, using a model that belongs to the financial cycle that you are loading data from.

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

This option is active only if the open cycle is a financial cycle and at least one unlocked time period is selected. To make your work in the wizard simpler, select the exact set of time periods to which you want to load data before you launch the wizard.

Load Operational Data
launches the Load Operational Data wizard, which enables you to load data into the open financial cycle from any operational cycle.

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

This option is active only if the open cycle is a financial cycle and at least one unlocked time period is selected. To make your work in the wizard simpler, select the exact set of time periods to which you want to load data before you launch the wizard.

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

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

This option is active only if at least one unlocked time period is selected. To make your work in the wizard simpler, select the exact set of time periods from which you want to delete data before you launch the wizard.

Lock
locks the selected time periods that are not currently locked.

The icon for a locked time period includes 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 only if at least one unlocked time period is selected.

See Locked Objects on page 16.

Unlock
unlocks the selected time periods that are currently locked.

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

See Locked Objects on page 16.

Properties
displays the Properties window for the selected cycle time period, enabling you to view and modify its properties.

This option is active only if exactly one time period is selected.

See Also

- Using Views on page 20
- Loading Data from the Cross Industry Detail Data Store on page 15
- Managing Data in a Cycle on page 62

Chapter 6

Exchange Rates

Introduction to Exchange Rates **67**

Survey of Exchange Rate Types **68**
Overview of Exchange Rate Types 68
Simple Exchange Rate Types 69
Complex Exchange Rate Types 69

Exchange Rate Types View **70**

Exchange Rate Sets View **70**

Exchange Rates View **71**
Overview of the Exchange Rates View 71
Working with Simple Exchange Rate Types 72
Working with Complex Exchange Rate Types 72

Resolving Conflicts between Two Rows of a Rate Table **74**
Overview of Rate Conflicts 74
Superset Rule 74
Precedence Rule 74

Introduction to Exchange Rates

Exchange rates are stored numeric values that are used to convert monetary values from one currency to another. Each stored exchange rate is associated with an exchange rate type and a time period. Exchange rates that are associated with one of the complex exchange rate types (Derived or Historic) are also associated with members from other dimension types, such as organizations or accounts.

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

Ordinarily, you load exchange rates into exchange rate sets by using the **Load Exchange Rates** option in the Exchange Rate Sets view, or the corresponding SAS Data Integration Studio job. However, it is possible to enter exchange rates manually.

To view or edit the exchange rates in an exchange rate set:

1. In the Exchange Rate Sets view, select the exchange rate set.
2. Select **Exchange Rates**.

   The Exchange Rates view opens for the selected exchange rate set, enabling you to work with the exchange rates.

To work with exchange rate types, use the Exchange Rate Types view. To open the Exchange Rate Types view:

1. Select the Rates workspace.
2. Select **Tools** ⇨ **Exchange Rate Types**.

In summary, managing exchange rates involves three views that have similar names:

- The Exchange Rate Sets view lists the exchange rate sets for the open cycle. Exchange rate sets are the containers for exchange rates.
- The Exchange Rates view gives you access to the exchange rates that are in a selected exchange rate set.
- The Exchange Rate Types view lists the exchange rate types that are used in the exchange rate sets.

When you create a model for a given cycle, all the exchange rate sets that belong to that cycle are available for use in the model. In creating the model, you associate each analysis member in the model with a particular exchange rate set. Therefore, when you create an exchange rate set, you should have in mind the analysis member or members that you intend to associate it with. For example, for a given cycle you might maintain a set of actual past exchange rates, a set of likely future exchange rates, a set of exchange rates that would become likely if a certain event were to occur, and so on.

One of the properties of each account is the applicable exchange rate type. In all the computations that are governed by the model, the exchange rate that is used to convert a given data value from one currency to another is retrieved from the exchange rate set that is determined by the relevant analysis member, and from the exchange rate type that is determined by the relevant account.

The SAS Financial Management Add-In for Microsoft Excel has several functions that retrieve exchange rates for use in special computations:

- You can retrieve exchange rates for the simple exchange rate types with the CDA function CDAXRate and the calculated-member function fmXRate.
- You can retrieve exchange rates for the complex exchange rate types with the CDA function CDACXRate and the calculated-member function fmCXRate.

See Also

- Survey of Exchange Rate Types on page 68
- Loading Data from the Cross Industry Detail Data Store on page 15

Survey of Exchange Rate Types

Overview of Exchange Rate Types

Each exchange rate set contains exchange rates for several exchange rate types. The exchange rate type of each account determines which exchange rates to use for currency conversion of that account.

For each account type, there is a default exchange rate type, which is generally the appropriate exchange rate type for all accounts of that type. The default exchange rate types are as follows:

- The default exchange rate type for the monetary balance account types (Asset, Liability, and Equity) is Period Close.
- The default exchange rate type for the monetary flow account types (Revenue and Expense) is Period Average.
- The default exchange rate type for the Statistical Flow, Statistical Balance, and Statistical accounts is None. The nonmonetary values of such accounts are transferred across all currencies without change, because the associated currencies are irrelevant to them.

  You can think of None as a special exchange rate type or as the absence of an exchange rate type. Exchange rate sets do not contain any exchange rates for it.
- Cumulative Translation Adjustment (CTA) accounts always use the Period Close exchange rate type. This value cannot be changed.

  The Period Close exchange rate type of a CTA account has a special purpose: it determines the so-called peg rate for the CTA computation. It is not used for currency conversion of the value of the account.
- There is no default exchange rate type for Retained Earnings accounts. This is because the value of a Retained Earnings account is computed separately within each currency. The operation of transferring a value from one crossing to another crossing that differs only in the currency member does not apply.

Simple Exchange Rate Types

A simple exchange rate type is an exchange rate type whose exchange rates vary by time period and analysis member, but not by the members of any other dimension type. The most important simple exchange rate types are the following:

Period Average
: the average over a time period of the continuously varying market exchange rate. Exchange rates of this type are typically used for Revenue and Expense accounts, which reflect transactions that take place at various times within a time period.

  In principle, one could convert each transaction using the market exchange rate at the moment of the transaction, but that level of detail is considered immaterial. The use of a Period Average exchange rate for all transactions within a period yields an acceptable approximation, as long as the period is not too long.

Period Close
: the market exchange rate that is in effect at the end of a time period. Exchange rates of this type are typically used for Asset, Liability, and Equity accounts, which contain the monetary values of various entities at the end of each time period.

Period Open
: the market exchange rate that is in effect at the beginning of a time period.

Complex Exchange Rate Types

A complex exchange rate type is an exchange rate type whose exchange rates vary by time period and analysis member, and also by the members of at least one other dimension type. The other dimension types that are most likely to be relevant are account and organization.

Complex exchange rates have the same behavior as driver rates. There are two complex exchange rate types:

Historic
: the market exchange rate that was in effect at the time of a particular significant transaction, to which the exchange rate must be applied. Exchange rates of this type are typically used for dividend payouts, primary sales of stock, and other major financial events.

  Note that the software associates each historic exchange rate with a time period, not with the particular moment in that period when the exchange rate was in effect. You cannot define two different historic exchange rates for two different moments in the same time period.

Derived
: an exchange rate that you compute by dividing the value of an account expressed in one currency by the value of the same account expressed in another currency. Exchange rates of this type are typically used for certain accounts in the first time period for which you use SAS Financial Management. For example, in the first time period a retained earnings account must be currency-converted using a derived exchange rate, because the software has no record of prior earnings from which to compute retained earnings in each currency. In this case, the derived exchange rate is in effect a blend of all the market exchange rates that have been applied to periodic earnings throughout the history of the enterprise prior to the use of SAS Financial Management.

See Also

Introduction to Exchange Rates on page 67

Exchange Rate Types View

The Exchange Rate Types view lists the available exchange rate types.

See Also

- Using Views on page 20
- Introduction to Exchange Rate Types on page 68

Exchange Rate Sets View

The Exchange Rate Sets view lists all the exchange rate sets that belong to the open cycle.

The following options are available:

New Exchange Rate Set
: launches the New Exchange Rate Set wizard, which enables you to create a new exchange rate set.

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

Load Exchange Rates
: launches the Load Exchange Rates wizard, which enables you to load numeric exchange rates from a specified Cross Industry Detail Data Store exchange rate set into

the selected exchange rate set. In the wizard, you specify which time periods and exchange rate types to include in the load operation.

All exchange rates in the selected exchange rate set for the specified time periods and exchange rate types are deleted before new exchange rates are loaded.

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

This option is active only if exactly one exchange rate set is selected.

Exchange Rates
opens the Exchange Rates view for the selected exchange rate set, enabling you to view, edit, and delete the exchange rates that it contains.

This option is active only if exactly one exchange rate set is selected.

Copy Here
launches the Copy Exchange Rate Set wizard, which enables you to create a copy of the selected exchange rate set that belongs to the open cycle. The copy is identical to the original except for the new name and description that you give it.

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

This option is active only if exactly one exchange rate set is selected.

After you create the copy, you can modify it by using the **Properties** option and the **Exchange Rates** option.

Copy To
launches the Copy Exchange Rate Set To wizard, which enables you to create a copy of the selected exchange rate set and give it to another cycle. The copy is identical to the original except for the new name and description that you give it.

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

This option is active only if exactly one exchange rate set is selected.

After you create the copy, you can modify it by opening the target cycle and using the **Properties** option and the **Exchange Rates** option in the Exchange Rate Sets view.

Delete
deletes the selected exchange rate sets.

Properties
displays the properties of the selected exchange rate set.

This option is active only if exactly one exchange rate set is selected.

See Also

- Using Views on page 20
- Introduction to Exchange Rates on page 67

Exchange Rates View

Overview of the Exchange Rates View

Use the Exchange Rates view to work with exchange rates in the selected exchange rate set, which is identified below the view tab.

Use the **Exchange rate type** and **Period** fields to select the exchange rate type and time period that you want to work with. The appearance and behavior of the window depend on whether you select a simple exchange rate type or a complex exchange rate type.

Working with Simple Exchange Rate Types

All exchange rate types except Historic and Derived are simple exchange rate types. A simple exchange rate type is simple in the sense that each numeric exchange rate depends only on a time period. Within that time period, the numeric exchange rate applies to all accounts that have the appropriate exchange rate type, all organizations, and all members of any non-required dimension types that are part of the cycle that the exchange rate set belongs to.

To view the simple exchange rates for any currency, select that currency in the **View currency** field. For a selected combination of currency, exchange rate type, and time period, there is a table that has the following three columns, from right to left:

- **Currency**
- The exchange rates by which you must multiply values that are expressed in the selected **View currency** currency in order to yield equivalent values that are expressed in the **Currency** column currencies.
- The exchange rates by which you must multiply values that are expressed in the **Currency** column currencies in order to yield equivalent values that are expressed in the selected **View currency** currency.

In other words, the first two cells of each row of the table show the pair of reciprocal exchange rates that relate the currency that is specified in the third cell of that row to the currency that is selected in the **View currency** field.

You cannot edit simple exchange rates for a locked period. You cannot edit simple exchange rates for any currency other than the base currency.

If the selected period is not locked, then you can edit simple exchange rates for the base currency. To do this, you must select the same currency in the **View currency** field that is displayed in the **Base currency** field. For example, if the base currency is EUR, then you must select EUR in the **View currency** field. The base currency is specified when you create the exchange rate set. It cannot be changed.

If the base currency is selected in the **View currency** field, then you can enter exchange rates in the first column of the table. To place an exchange rate in a cell, click the cell to select it, and then enter the value. If you have a policy of loading all simple exchange rates from the Cross Industry Detail Data Store, then you should not edit them in this window.

If you change the value of any exchange rate, then all other exchange rates that depend on the value that you change are updated automatically. This includes the reciprocal exchange rate in the second cell of the row. It also includes all exchange rates between the currency in the third cell of the row and all other currencies.

Working with Complex Exchange Rate Types

Overview of Complex Exchange Rate Types

The complex exchange rate types are Historic and Derived. These exchange rate types are complex in the sense that a given numeric exchange rate can be associated with members from many dimension types. Associating an exchange rate with members from additional dimension types narrows its range of application.

If you select either the Historic exchange rate type or the Derived exchange rate type, then the table has the following column structure:

- The first column on the left holds the exchange rates by which you must multiply values that are expressed in the **Currency** column currencies in order to yield equivalent values that are expressed in the selected **View currency** currency.
- The other columns identify members of dimension types that the numeric exchange rates depend on. For example, if the exchange rates vary by organization, then the table includes a column for the organization dimension type.

  The dimension types that are included in the table were chosen when the table was created, and cannot be changed. The dimension type columns are arranged from left to right in the precedence order that is used to resolve conflicts between rows.

Each row of the table associates an exchange rate with a combination of dimension members. However, a row does not have to specify a member for every dimension type column. If a member is specified for a given dimension type, then the exchange rate applies only to the specified member. If a member is not specified for a given dimension type, then the exchange rate applies to every member of that dimension type. For example:

- If an exchange rate is associated with both an account and an organization, then it applies only to the value of that account within that organization.
- If an exchange rate is associated with an account but not with an organization, then it applies to the value of that account in any organization.

Change Complex Exchange Rates

To change the exchange rate in any row of the table:

1. Select the row. You can do this by clicking any cell in the row, either the exchange rate cell or a member cell.

   If you want to enter the same exchange rate in several rows, you can select all those rows at once by clicking a cell in each row while holding down the CTRL key. You can also select a block of rows by dragging the mouse over all the rows in the block or by clicking a cell in the first row of the block and then clicking a cell in the last row of the block while holding down the SHIFT key.

2. Select **Change exchange rates value** () from the toolbar.

   The Edit Rate window appears.

3. Enter the numeric exchange rate in the **Rate** field of the Edit Rate window, and then click **OK**.

You can also change an exchange rate by selecting the cell that contains the exchange rate and typing a different exchange rate in that cell.

Add Rows

To add rows to the table, select **Add exchange rates** () from the toolbar. This launches the New Exchange Rates wizard, which enables you to select dimension members for new rows.

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

Delete Rows

To delete a row from the table:

1. Select the row. You can do this by clicking any cell in the row, either the exchange rate cell or a member cell.

   If you want to delete several rows, you can select all those rows at once by clicking a cell in each row while holding down the CTRL key. You can also select a block of rows by dragging the mouse over all the rows in the block or by clicking a cell in the first row of the block and then clicking a cell in the last row of the block while holding down the SHIFT key.

2. Select **Delete exchange rates** () from the toolbar.

See Also

- Introduction to Exchange Rates on page 67
- Resolving Conflicts between Two Rows of a Rate Table on page 74

Resolving Conflicts between Two Rows of a Rate Table

Overview of Rate Conflicts

Complex exchange rate tables and driver rate tables have a wildcard capability that makes it possible for a table to contain two or more rates that apply to the same crossing. When this is the case, conflict resolution rules are used to determine which rate to retrieve. There are two conflict resolution rules—the superset rule and the precedence rule.

Superset Rule

If one row uses a superset of the members in another row, then the row with the greater number of members prevails.

For example, consider a table that has an account column and an organization column. Suppose that one row assigns the rate of 5 to account Inventory, regardless of the organization, while another row assigns the rate of 3 to the combination of account Inventory and organization North. Both these rows match any crossing that contains both account Inventory and organization North, so there is a conflict at all such crossings. The conflict is resolved by retrieving the rate that is associated with the most members. In this case, 5 is associated with one member while 3 is associated with two members, so 3 is retrieved.

Precedence Rule

If two rows conflict but neither row uses a superset of the members in the other row, then the conflict is resolved by the precedence ordering for dimension types.

For example, consider a table that has an account column and an organization column. Suppose that one row assigns the rate of 2 to account Inventory, regardless of the organization, while another row assigns the rate of 9 organization North, regardless of the account. Both these rows match any crossing that contains both account Inventory and organization North, so there is a conflict at all such crossings. The conflict is resolved by

retrieving the rate that is associated with the member that belongs to the most highly ranked dimension type:

- If Account is ranked above Organization, then the 2 associated with account Inventory is retrieved.
- If Organization is ranked above Account, then the 9 associated with organization North is retrieved.

If a table contains more than two columns of members, then the precedence rule is generalized in the following way. Look first at the most highly ranked dimension type that has a member in either of the conflicting rows. If one row contains a member for that dimension type while the other row does not, then the row that contains the member prevails. If both rows contain a member for the most highly ranked dimension type, then look at the next most highly ranked dimension type that has a member in either row. Continue in this manner until you come to the tie-breaking dimension type.

In any complex exchange rate table or driver rate table, the left-to-right order of the columns shows you the precedence order of the dimension types. The leftmost column holds the dimension type with the highest rank, and so on.

For an exchange rate set, you can change the precedence order using the **Dimension Types** page of the exchange rate set's Properties window.

For a driver rate set, you can change the precedence order using the **Dimension Types** page of the driver rate set's Properties window.

Chapter 7

Driver Rates

Introduction to Driver Rates 77

Driver Rate Types View 78

Driver Rate Sets View 79

Driver Rates View 80
Overview of the Driver Rates View 80
Change Driver Rates 80
Add Rows 81
Delete Rows 81

Introduction to Driver Rates

Driver rates are stored numeric values that can be retrieved by certain functions for use in computations. Each stored driver rate is associated with a driver rate type and a combination of members from certain dimension types. For example, you might maintain driver rates that are associated with the Interest Rates driver rate type and combinations of time and organization. You can use driver rates to represent interest rates, tax rates, unit prices, or any set of values that depend on combinations of dimension members.

Driver rates are stored in driver rate sets. To work with driver rate sets, use the Driver Rate Sets view in the Rates workspace.

Ordinarily, you load driver rates into driver rate sets by using the **Load Driver Rates** option in the Driver Rate Sets view, or the corresponding SAS Data Integration Studio job. However, it is possible to enter driver rates manually.

To view or edit the driver rates in a driver rate set:

1. In the Driver Rate Sets view, select the driver rate set.
2. Select **Driver Rates**.

   The Driver Rates view opens for the selected driver rate set, enabling you to work with the driver rates.

To work with driver rate types, use the Driver Rate Types view. To open the Driver Rate Types view:

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

In summary, managing driver rates involves three views that have similar names:

- The Driver Rate Sets view lists the driver rate sets for the open cycle. Driver rate sets are the containers for driver rates.
- The Driver Rates view gives you access to the driver rates that are in a selected driver rate set.
- The Driver Rate Types view lists the driver rate types that are used in the driver rate sets.

When you create a model for a given cycle, all the driver rate sets that belong to that cycle are available for use in the model. In creating the model, you associate each analysis member in the model with a particular driver rate set. Therefore, when you create a driver rate set, you should have in mind the analysis member or members that you intend to associate it with. For example, for a given cycle you might maintain a set of actual past prices or interest rates, or a set of likely future prices or interest rates, or a set of prices or interest rates that would become likely if a certain event were to occur, and so on.

In a SAS Financial Management Studio formula, use the DRATE function to retrieve driver rates. The argument of the DRATE function is the relevant driver rate type. The analysis member of the crossing at which the formula is evaluated determines which driver rate set the DRATE function looks at. In that driver rate set, DRATE retrieves the driver rate that is associated with members that match the members in the crossing at which the formula is evaluated.

In the SAS Financial Management Add-In for Microsoft Excel, you can retrieve any driver rate using the CDA function CDARate or the calculated-member function fmRate.

See Also

Loading Data from the Cross Industry Detail Data Store on page 15

Driver Rate Types View

The Driver Rate Types view list the existing driver rate types.

The following options are available:

New driver rate type
launches the New Driver Rate Type wizard, which enables you to create a driver rate type.

Delete
deletes the selected driver rate types.

See Also

- Using Views on page 20
- Introduction to Driver Rates on page 77

Driver Rate Sets View

The Driver Rate Sets view lists all the driver rate sets that belong to the open cycle.

The following options are available:

New Driver Rate Set
launches the New Driver Rate Set wizard, which enables you to create a driver rate set.

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

Load Driver Rates
launches the Load Driver Rates wizard, which enables you to load driver rates from the Cross Industry Detail Data Store to the selected driver rate set. In the wizard, you specify which driver rate types to include in the load operation.

All driver rates in the selected driver rate set for the specified driver rate types are deleted before new driver rates are loaded.

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

This option is active only if exactly one driver rate set is selected.

Driver Rates
opens the Driver Rates view for the selected driver rate set, enabling you to view, edit, and delete the driver rates that it contains.

Copy Here
launches the Copy Driver Rate Set wizard, which enables you to create a copy of the selected driver rate set that belongs to the open cycle. The copy is identical to the original except for the new name and description that you give it.

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

This option is active only if exactly one driver rate set is selected.

After you create the copy, you can modify it by using the **Properties** and **Driver Rates** options.

Copy To
launches the Copy Driver Rate Set To wizard, which enables you to create a copy of the selected driver rate set and give it to another cycle. The copy is identical to the original driver rate set except for the new name and description that you give it.

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

This option is active only if exactly one driver rate set is selected.

After you create the copy, you can modify it by opening the target cycle and using the **Properties** option and the **Driver Rates** option in the Driver Rate Sets view.

Delete
deletes the selected driver rate sets.

Properties
displays the properties of the selected driver rate set.

This option is active only if exactly one driver rate set is selected.

See Also

- Using Views on page 20
- Introduction to Driver Rates on page 77

Driver Rates View

Overview of the Driver Rates View

Use the Driver Rates view to work with the driver rates in the selected driver rate set, which is identified below the view tab.

The view displays a table of driver rates for the selected driver rate type. To select a different driver rate type, use the **Driver rate type** field. To create another driver rate type, select **Tools** ⇨ **Driver Rate Types** and use the Driver Rate Types view.

Each table has a **Rate** column for the numeric driver rates and one or more other columns that hold dimension members. Each column that holds dimension members represents a dimension type, which is identified in the column heading. The dimension types that are included in the table were chosen when the table was created, and cannot be changed. The dimension type columns are arranged from left to right in the precedence order that is used to resolve conflicts between rows.

Each row of the table associates a numeric driver rate with a combination of dimension members.

A row does not have to specify a member in every dimension type column. If a member is specified for a given dimension type, then the driver rate applies only to the specified member. If a member is not specified for a given dimension type, then the driver rate applies to every member of that dimension type. For example:

- If a driver rate is associated with an account, an organization, and a time period, then it applies only to that account within that organization during that time period.
- If a driver rate is associated with an account and a time period but not with an organization, then it applies to that account during that time period, regardless of the organization.
- If a driver rate is associated with an account, but not with an organization or a time period, then it applies to that account across all organizations and all time periods.

Change Driver Rates

To change the driver rate in any row of the table:

1. Select the row. You can do this by clicking any cell in the row—either the driver rate cell or a member cell.

   If you want to enter the same driver rate in several rows, you can select all those rows at once by clicking a cell in each row while holding down the CTRL key. You can also select a block of rows by dragging the mouse over all the rows in the block or by clicking a cell in the first row of the block and then clicking a cell in the last row of the block while holding down the SHIFT key.

2. Select **Change driver rates value** () from the toolbar.

   The Edit Rate window appears.

3. Enter the numeric driver rate in the **Rate** field of the Edit Rate window, and then click **OK**.

You can also change a driver rate by selecting the cell that contains the driver rate and typing a different driver rate in that cell.

Add Rows

To add rows to the table, select **Add driver rates** () from the toolbar. This launches the New Driver Rates wizard, which enables you to select dimension members for new rows.

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

Delete Rows

To delete a row from the table:

1. Select the row. You can do this by clicking any cell in the row—either the driver rate cell or a member cell.

   If you want to delete several rows, you can select all those rows at once by clicking a cell in each row while holding down the CTRL key. You can also select a block of rows by dragging the mouse over all the rows in the block or by clicking a cell in the first row of the block and then clicking a cell in the last row of the block while holding down the SHIFT key.

2. Select **Delete driver rates** () from the toolbar.

See Also

- Introduction to Driver Rates on page 77
- Resolving Conflicts between Two Rows of a Rate Table on page 74

Chapter 8

Models: General Considerations

Introduction to Models **83**

Required Features of a Model **84**

Optional Features of a Model **85**

Models View **86**
Overview of the Models View 86
Options for All Models 86
Options for Financial Models Only 87

Introduction to Models

Each model belongs to a particular cycle. A model is a complex object that defines the computations that connect the data that is stored in the cycle to the numeric values that users enter or view.

A model that belongs to a financial cycle is a financial model. A model that belongs to an operational cycle is an operational model. Financial models and operational models are broadly similar but differ in the following important ways:

- Financial models are used to support financial form sets, which enable users to enter financial data through data-entry forms. Financial models are also used to generate reports in the SAS Financial Management Add-In for Microsoft Excel that contain read-only tables and/or CDA tables. Operational models are used exclusively to support operational form sets, which enable users to enter operational data through data-entry forms. Operational models do not support report generation. In order to create a report that includes data that was entered using an operational form set, you must load a summarized version of that data into a financial cycle, using the Load Operational Data wizard in the Periods workspace, and then use financial models to create the report.
- Financial models can include manual adjustments and adjustment rules, which play important roles in the report-generation process. Operational models cannot include manual adjustments or adjustment rules. The Models workspace is a much simpler place when an operational cycle is open than when a financial cycle is open, because the Manual Adjustments view, the Adjustment Rules view, and all the options that concern the adjustment process are missing.
- Two or more financial models can be brought together in a composite model. You can use a composite model to generate reports that cannot be generated with either of the component models alone. Operational models cannot be brought together with other

operational models or with financial models in a composite model. The composite model capability exists exclusively for financial models.

See Also

Introduction to Composite Models on page 113

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.

Following are the required features of a model:

- an association with a cycle

  The underlying cycle receives the data from any data-entry tables that are based on the model and 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 to identify the model

  Giving the model an identity is the first step in the New Model wizard. After the model is created, you can change the name and description using the **Identification** page of the model's Properties window.

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

  The collection of all the specified hierarchies determines the set of crossings that can appear in tables that are based on the model. Any crossing that consists of one member from each specified hierarchy is available. All the properties of the hierarchy members, such as formulas or security restrictions, become part of the model.

  The specification of each hierarchy includes both the name of the hierarchy and an as-of date. The model uses the version of the named hierarchy that is current at the as-of-date. There is one important exception: for every member of every hierarchy, access to associated data is always determined by the set of users that is specified on the member's **Security** page at the moment of use.

  Several pages of the New Model wizard are devoted to selecting the hierarchies. After the model is created, you can select different hierarchies and as-of dates on the **Dimensions** page of the model's Properties window.

- a time span, which is a set of adjacent time periods

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

  You specify the model's time span on the same wizard page on which you select the model's time hierarchy. You can change the model's time span on the **Dimensions** page of the model's Properties window.

- a rank ordering of the hierarchies for the purpose of resolving formula conflicts

  If a crossing is within the scope of two or more formulas that are associated with different members of the crossing, then the value at that crossing is determined by a formula that is associated with the member that belongs to the most highly ranked hierarchy.

A default rank ordering of hierarchies is set automatically. You can view it and change it using the **Formulas** page of the model's Properties window.

- the default read member and the default write member for each hierarchy

  A hierarchy's default read member is used to complete any query that does not include an explicit specification for that hierarchy. A hierarchy's default write member is used to complete any data record that is created through a data-entry form or generated by an adjustment rule that does not include an explicit specification for that hierarchy.

  You can specify some of these default members in the New Model wizard, as you select the hierarchies. You can also specify them on the **Dimensions** page of the model's Properties window, after you create the model. Some of these defaults are set by the software and cannot be changed.

- 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 applicable exchange rate 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 Actual crossings and Budget crossings, for example.

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

- 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 look-up in the driver rate set that is associated with the crossing's analysis member. This enables you, for example, 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 them on the **Rates** page of the model's Properties window.

- formatting specifications for debit and credit accounts

  You can specify whether the values of debit accounts will be displayed 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 them on the **Formats** page of the model's Properties window.

Optional Features of a Model

Any 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, use the Security page of the model's Properties window.

- 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.

A financial 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.

- a connection to another financial model that gives a query based on one model access to the other model's adjustments

  You can specify this connection on the Prior Period Adjustments page of the New Model wizard or on the **Adjustments** page of the model's Properties window.

In general, you specify a model's optional features after you create the model with the New Model wizard.

Models View

Overview of the Models View

The Models view lists the models that belong to the open cycle. If a financial cycle is open, then the listed models are financial models. If an operational cycle is open, then the listed models are operational models.

Options that concern the adjustment process are available only for financial models.

A model is sometimes marked with the Error icon  . To see detailed information about a model's errors, use the **Properties** option to display the model's properties.

A financial model is sometimes marked with the Warning icon (). See Warnings for Financial Models on page 94.

Options for All Models

The following options are available for financial models and operational models:

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

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

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

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

Delete
deletes the selected model or models.

The software prevents you from deleting any financial model that is part of an existing composite model.

Cell Protection Rules
Opens a workbook in Microsoft Excel, in which you can view, modify, or add cell protection rules for the selected model.

You can protect cell crossings in a data-entry form by creating one or more rules that are associated with a model or a form template. Users can also protect individual cells in a form. Rules that are associated with a model apply to all form sets that use the model. For more information, see the *SAS Financial Management: User's Guide* or the online Help for the SAS Financial Management Add-In for Microsoft Excel.

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.

Options for Financial Models Only

The following options are available for financial models but not for operational models:

Export Data Records
launches the Export Data Records wizard, which exports numeric data from the selected model to a target SAS library.

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

Lock
locks the selected financial model or models that are not currently locked.

This option is not available for operational models. However, an operational model is automatically locked if any operational form set that uses it is published.

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

When a model is locked, either manually or automatically, 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."

See Locked Objects on page 16.

Unlock
unlocks the selected financial model or models that are currently locked.

This option is not available for operational models. However, an operational model is automatically unlocked if the only published operational form set that uses it loses its Published status.

When a model is unlocked, either manually or automatically, the as-of dates do not change. To change the as-of date for a hierarchy, you must use the **Dimensions** page of the model's Properties window.

See Locked Objects on page 16.

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 appears when you select this option.

Clear Warnings
removes all warnings from the selected model or models.

This option is active only 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.

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. It is a combination of the following two administrative reports that you can run in the Document Manager:

- Manual Adjustments
- Rules Facts

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.

In the Document Manager, you can also run these administrative reports that provide additional perspectives on posted adjustment values:

- Facts
- Non Leaf
- Ownership Adjustments
- Rule

See Also

- Using Views on page 20
- Introduction to Models on page 83
- Adjustment Options in the Models View on page 90

Chapter 9

Models: The Adjustment Process

Overview of the Adjustment Process 89
Differences between Posting Manual Adjustments and Posting Adjustment Rules 90
Adjustment Options in the Models View 90
Posting Status of Manual Adjustments and Adjustment Rules 91
Being On Hold 92
What Happens When You Post Adjustments 92
Overview 92
Threshold for Adjustment Rules 93
Warnings for Financial Models 94
Working with Manual Adjustments 95
Manual Adjustments View 96
Working with Adjustment Rules 98
Adjustment Rules View 99

Overview of the Adjustment Process

An optional but potentially important part of a financial model is a set of specifications for an adjustment process. Numeric values that are loaded into a financial cycle from the Cross Industry Detail Data Store or another cycle, or entered into a financial cycle through a data-entry form, can be modified by adjustments that you specify in the Models workspace. The process of adding the values of the adjustments to the values that were loaded or entered through forms is controlled by the roll-up arithmetic of the Source hierarchy.

The Source hierarchy has three members for unadjusted values:

- Base
- BaseJourn
- BaseForm

And it has five members for various types of adjustment values:

- Manual
- Bal
- Reclass

- Alloc
- CPO

For a set of crossings that are identical in every dimension except Source, the values that are associated with these eight members are combined at the TotalBeforeElim member. For example, a certain TotalBeforeElim value might be the sum of the corresponding Base value, which was loaded from the Cross Industry Detail Data Store, and the corresponding Manual value, which was specified in a manual adjustment. In reports, you can see the unadjusted data, the amounts of various types of adjustments, and the fully adjusted values by looking at the appropriate members of the Source hierarchy.

To set the stage for the roll-up arithmetic of the Source hierarchy, you must populate the crossings that hold the adjustment values. This involves two main steps:

1. Create the appropriate adjustments and adjustment rules in a financial model.
2. Post the adjustments and adjustment rules that you have created.

   Posting places the adjustment values in the crossings where the roll-up arithmetic of the Source hierarchy can access them.

See Also

Introduction to the Source Dimension Type on page 52

Differences between Posting Manual Adjustments and Posting Adjustment Rules

A manual adjustment is an adjustment that you explicitly specify. The specification includes a set of numeric values and the dimension members that the values are associated with. When you post a manual adjustment, all this information is copied to crossings that have the Manual member in the Source dimension.

An adjustment rule is a set of instructions for generating adjustments, using preexisting numeric values as input. For example, a balancing rule compares certain preexisting values that are supposed to be equal and generates adjustments when it finds inequalities. When you post an adjustment rule, the rule generates its adjustments. The generated adjustments are copied to crossings that have one of the following members in the Source dimension, depending on the type of the adjustment rule:

- Bal (for reporting entity balancing rules, intercompany balancing rules, and net intercompany balancing rules)
- Reclass (for reclassification rules)
- Alloc (for allocation rules)
- CPO (for ownership rules)

Adjustment Options in the Models View

For a given financial model that is selected in the Models view, use the following options to work with adjustments:

- To create, edit, delete, or copy manual adjustments, select **Manual Adjustments** and work in the Manual Adjustments view.
- To create, edit, delete, or copy adjustment rules, select **Adjustment Rules** and work in the Adjustment Rules view.
- To post both manual adjustments and rule-generated adjustments, select **Post Adjustments** and use the Post Adjustments window to specify which time periods and analysis values to post adjustments for.
- To view a listing of currently posted adjustment values, select **Posted Adjustment Data**.

  You can also view a list of currently posted adjustment values for a selected set of adjustment rules by selecting the rules in the Adjustment Rules view and then selecting **Posted Adjustment Data** there.
- To see the history of past posts, select **Properties**, and then select the **History** page in the Properties window.

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 will 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 financial model includes any manual adjustment or adjustment rule that is not on hold and that 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 the model includes adjustment information that is not currently in position to be rolled up into reports. To put all the current adjustment information in position to be rolled up into reports, you must post the model's adjustments.

See Also

Warnings for Financial Models on page 94

Being On Hold

Every manual adjustment and every adjustment rule is either on hold or not on hold. When you post adjustments, all items that are not on hold are posted, while all items that are on hold are withheld from the posting process.

In the Manual Adjustments view and the Adjustment Rules view, any item that is on hold is marked with the On Hold icon . Also, any item that is on hold has the **On hold** check box selected on the **General** tab of its Properties window.

When you create a manual adjustment or an adjustment rule, it is not on hold. There are two ways to put a manual adjustment or an adjustment rule on hold:

- Select the item, and then select **Put On Hold**.
- Select the item, select **Properties**, and then select the **On hold** check box on the **General** tab of the Properties window.

There are two ways to take a manual adjustment or an adjustment rule off hold:

- Select the item, and then select **Take Off Hold**.
- Select the item, select **Properties**, and then deselect the **On hold** check box on the **General** tab of the Properties window.

See Also

What Happens When You Post Adjustments on page 92

What Happens When You Post Adjustments

Overview

The input to an adjustment rule can include adjustment values that are posted before the rule runs. This is why the Manual Adjustments view has the **Before rules** and **After rules** radio buttons. This is also why the Adjustment Rules view lists adjustment rules in a certain execution order.

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. See Threshold for Adjustment Rules on page 93.
4. All the **After rules** manual adjustment values are posted.

The only way to change the set of posted adjustments is to post adjustments again. To delete certain posted adjustment values:

1. Put on hold the manual adjustments or adjustments rules that are the sources of the records that you want to delete.
2. Post adjustments.

   The posting process starts by deleting all currently posted adjustment values. The posted adjustment values that you want to delete are not posted again because the manual adjustments and adjustment rules that they come from are on hold.

Note: Deleting a manual adjustment or adjustment rule immediately deletes its posted adjustments. If the output of the deleted adjustment provided input to an adjustment rule that runs later in the posting process, then all adjustments should be reposted.

Threshold for Adjustment Rules

The adjustments that are generated by the following rules are subject to an absolute value threshold:

- allocation rules
- reclassification rules
- intercompany balancing rules
- net intercompany balancing rules

The threshold is designed to ignore any adjustment values that are so small that they are not useful.

Note: Adjustments that are generated by reporting-entity balancing rules and ownership rules are not subject to a threshold.

Values lower than the threshold value are dropped (not posted). For example, assume the following adjustments:

- **`Account A`**: value 5.0 (target)
- **`Account B`**: value 0.1 (target)
- **`Account C`**: value –5.0 (offset)
- **`Account D`**: value –0.1 (offset)

With a threshold value of 0.2, any entry whose absolute value is less than 0.2 is dropped. In the previous example, there would be two line items posted rather than four:

- **`Account A`**: value 5.0 (target)
- **`Account C`**: value –5.0 (offset)

If the entry to the target crossing is below the threshold, the corresponding offset entry is dropped as well so that the entry continues to have a net value of zero. In this example, three line items are generated for posting:

- **`Account D`**: value 6.0 (target)
- **`Account E`**: value 0.15 (target)
- **`Account F`**: value –6.15 (offset)

With a threshold value of zero, all three of the line items are posted. With a threshold value of 0.2, only two line items are posted:

- **Account D**: value 6.0 (target)
- **Account F**: value –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.

The default threshold value is 0.001. For intercompany balancing rules and net intercompany balancing rules, you cannot modify the threshold value.

For allocation rules and reclassification rules, you can modify the threshold or effectively eliminate it (by setting its value to 0). See "Post-Configuration Steps" in the *SAS Solutions Services: System Administration Guide*. After modifying the threshold value (and restarting the managed servers), you can repost allocation and reclassification rules to use the new threshold.

See Also

Being On Hold on page 92

Warnings for Financial Models

In the Models view, a financial model can be marked with the Warning icon ().

A warning signals a possible discrepancy between the adjustments that are currently posted for the selected model and the adjustments that would be posted if you post adjustments for the selected model now. Such a discrepancy arises when you create the first manual adjustment or adjustment rule for a new financial model, which does not yet have any posted adjustments. It also arises for a financial model that has posted adjustments when you change any aspect of the model that can affect the result of posting adjustments.

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 is a list of the messages that can appear there:

- Data records have changed
- Exchange rates have changed
- Driver rates have changed
- Hierarchies have changed
- Hierarchy selections have changed
- Exchange rate selections have changed
- Driver rate selections have changed
- Default write member selections have changed
- The dimension evaluation order has changed
- The prior period adjustments model has changed

- Manual adjustments have changed
- Rules have changed
- The status of the prior period adjustments model has changed
- The prior period adjustments model has been posted

See Also

- What Happens When You Post Adjustments on page 92
- Posting Status of Manual Adjustments and Adjustment Rules on page 91

Working with Manual Adjustments

A manual adjustment specifies a set of numeric adjustment values for a designated set of crossings.

There are two types of manual adjustments—balanced and unbalanced:

- A balanced manual adjustment consists of a set of debit and credit line items in which the sum of the debits equals the sum of the credits. The equality of debits and credits is enforced by the software when you create the adjustment.
- An unbalanced manual adjustment consists of a set of debit and credit line items in which the sum of the debits might or might not equal the sum of the credits. All numeric values are permitted.

  Only a Finance Process Administrator can create, edit, or delete unbalanced manual adjustments.

For a balanced manual adjustment, you must designate an organizational reporting entity. For an unbalanced manual adjustment, you can designate a reporting entity, but you are not forced to. For any manual adjustment that has a designated reporting entity, each line item of the adjustment must be associated either with the reporting entity itself or with some organization that is hierarchically subordinate to the 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 financial model, do the following:

1. In the **Models** view, select the model.
2. Select **Manual Adjustments**.

   The Manual Adjustments view for the selected model appears.
3. In the Manual Adjustments view, make sure that the selections are correct in the **Analysis** and **Time** fields and that you have the correct radio button 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. A reversing entry is identical to the original adjustment except that debits become credits and credits become debits.

See Also

- Manual Adjustments View on page 96
- User Roles on page 14
- Organization Details on page 50

Manual Adjustments View

The Manual Adjustments view lists a subset of the manual adjustments that belong to the selected financial model. The selected model is identified immediately below the view's tab. The listed subset is defined by the following filter conditions that are specified near the top of the view:

- If the **Before rules** radio button is selected, then the view lists only manual adjustments that are posted before any adjustment rules are posted. If the **After rules** radio button is selected, then the view lists only manual adjustments that are posted after all adjustment rules are posted. Select the appropriate radio button.
- The view lists only manual adjustments that apply to the analysis member that is shown in the **Analysis** field. Select the appropriate analysis member from the field's drop-down list.
- The view lists only manual adjustments that apply to the time period that is shown in the **Time** field. Select the appropriate time period from the field's drop-down list.

By changing the filter conditions, you can access any manual adjustment that belongs to the selected financial model.

Any manual adjustment that is on hold is marked with the On Hold icon . Any manual adjustment that contains an error is marked with the Error icon .

The options that are available in the Manual Adjustments view can be restricted by properties of the open cycle. If the cycle imposes no restrictions, then 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.

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

New Unbalanced Manual Adjustment
launches the New Unbalanced Manual Adjustment wizard, which enables you to create an unbalanced manual adjustment.

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

Put On Hold
puts on hold all the selected manual adjustments that are not currently on hold.

This option is active only 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.

This option is active only 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 only if the selected manual adjustments are **After rules** manual adjustments.

Delete
deletes the selected manual adjustments.

When you delete a manual adjustment, all posted values that were produced from it are also deleted.

Copy Here
launches the Copy Manual Adjustment wizard, which enables you to make a differently named copy of the selected manual adjustment for the same analysis member, time period, and relationship to adjustment rules.

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

This option is active only if exactly one manual adjustment is selected.

Copy To
launches the Copy Manual Adjustment To wizard, which enables you to make a copy of the selected manual adjustment for any combination of a different financial model, analysis member, or time period. This wizard enables you to make reversing entries as well as ordinary copies. A reversing entry is identical to the original adjustment except that debits become credits and credits become debits.

You can copy a manual adjustment from any financial model to any other financial model. However, the result might not be an exact copy if the source and target models do not use exactly the same set of dimension types or exactly the same dimensions within each dimension type. If there is a dimension type in the source model that is not in the target model, then all references to it are lost. If there is a dimension type in the target model that is 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.

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

This option is active only if exactly one manual adjustment is selected.

Find Errors
displays the Manual Adjustment Errors window, which enables you to find manual adjustments that contain errors.

Properties
displays the properties of the selected manual adjustment, enabling you to edit the properties that can be changed.

This option is active only if exactly one manual adjustment is selected.

See Also

- Using Views on page 20
- Working with Manual Adjustments on page 95
- Being On Hold on page 92

Working with Adjustment Rules

An adjustment rule generates a set of adjustments that accomplish a specific purpose. The different purposes of adjustment rules correspond to the different rule types, which follow:

- Intercompany balancing rules on page 103
- Net intercompany balancing rules on page 104
- Reporting-entity balancing rules on page 105
- Reclassification rules on page 106
- Allocation rules on page 106
- Ownership rules on page 107

To create, edit, delete, or copy the adjustment rules that belong to a particular financial model:

1. In the Models view, select the model.
2. Select **Adjustment Rules**.

   The Adjustment Rules view for the selected model appears. In the Adjustment Rules view, all the adjustment rules for the selected model are displayed in a single list. They are not tied to particular analysis members or time periods. You specify which analysis members and time periods to run a rule for when you post the rule. The order of the list is important because it determines the posting sequence of the rules.
3. Use the Adjustment Rules view as explained in the online Help for that view.

You can promote allocation rules and ownership rules from one system to another by exporting the rules from the first system and then importing them to the second system. You cannot promote adjustment rules of other types.

Any rule–creating option that is not enabled for the cycle that the selected model belongs to is not available. To enable or disable a rule–creating option for a certain cycle:

1. Select **Tools** ⇨ **Cycle Manager**.
2. In the Cycle Manager window, select the cycle and then select **Properties**.
3. In the Properties window, select the **Adjustments** tab.
4. Make the necessary changes on the **Adjustments** tab.

See Also

- Adjustment Rules view on page 99
- Promoting Objects to Another System on page 19

Adjustment Rules View

The Adjustment Rules view lists the adjustment rules for the selected financial 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**.

Any adjustment rule that is on hold is marked with the On Hold icon  . Any adjustment rule that contains an error is marked with the Error icon  .

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, which enables you to create an intercompany balancing rule.

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

New Reporting-Entity Balancing Rule
launches the New Reporting-Entity Balancing Rule wizard, which enables you to create a reporting-entity balancing rule.

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

New Net Intercompany Balancing Rule
launches the New Net Intercompany Balancing Rule wizard, which enables you to create a net intercompany balancing rule.

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

New Allocation Rule
launches the New Allocation Rule wizard, which enables you to create an allocation rule.

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

New Reclassification Rule
launches the New Reclassification Rule wizard, which enables you to create a reclassification rule.

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

New Ownership Rule
launches the New Ownership Rule wizard, which enables you to create an ownership rule.

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

You can create only one ownership rule per financial 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.

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

Export Adjustment Rules
launches the Export Adjustment Rules wizard, which enables you to export the selected ownership rule or allocation rules to a SAS package file.

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

This option is active only if an ownership rule or an allocation rule is selected. Adjustment rules of other types cannot be exported.

Put On Hold
puts on hold all the selected adjustment rules that are not currently on hold.

This option is active only 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.

This option is active only if at least one selected adjustment rule is on hold.

Delete
deletes the selected adjustment rules. When you delete an adjustment rule, all posted values that were produced by that rule are also deleted.

Copy Here
launches the Copy Rule wizard, which enables you to make a differently named copy of the selected adjustment rule for the same model.

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

This option is active only if exactly one adjustment rule is selected.

Copy To
launches the Copy Rule To wizard, which enables you to make a copy of the selected adjustment rule for another model.

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

You can copy an adjustment rule from any financial model to any other financial model. However, the result might not be an exact copy if the source and target models do not use exactly the same set of dimension types or exactly the same dimensions within each dimension type. If there is a dimension type in the source model that is not in the target model, then all references to it are lost. If there is a dimension type in the target model that is 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.

This option is active only if exactly one adjustment rule is selected.

Reorder Adjustment Rules
displays the Reorder Adjustment Rules window, which 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 financial 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.

This option is active only if exactly one adjustment rule is selected.

See Also

- Using Views on page 20
- Working with Adjustment Rules on page 98
- Being On Hold on page 92

Chapter 10

Models: Adjustment Rule Types

Intercompany Balancing Rules **103**
Net Intercompany Balancing Rules **104**
Reporting-Entity Balancing Rules **105**
Allocation Rules **106**
Reclassification Rules **106**
Ownership Rules: Introduction **107**
Ownership Rules: Organization Hierarchy Requirements **108**
Ownership Rules: Holding Types **108**
Ownership Rules: Consolidation Methods **109**
Overview of Consolidation Methods 109
Cost Method 109
Equity Method 110
Full Consolidation Method 110
Full Consolidation with Minority Interest Method 110
Proportional Method 111
Treasury Stock Method 111
Parent Method 111
“No Consolidation” Method 111
“No Method” 112

Intercompany Balancing Rules

An intercompany balancing rule looks for inequalities between a sum of designated debit account values that belong to one organization and a sum of designated credit account values that belong to another organization. In many cases, the comparison involves only one debit account and one credit account.

For example, organization A's notes payable to organization B should have the same value as organization B's notes receivable from organization A. You can check that the values are equal by defining and posting an appropriate intercompany balancing rule.

An intercompany balancing rule always examines crossings whose Source hierarchy member is TotalBeforeElim. The values that the rule finds at these crossings incorporate relevant values from manual adjustments and rule-generated adjustments that were posted earlier in the posting sequence.

If the amounts that should be equal are not equal, then the rule generates an adjustment that removes the inequality. The generated adjustment places the value that is needed to achieve equality in two places—a designated balancing account and a designated offset account. The offset account is necessary in order to keep the balance sheet of the affected organization in balance. It can be a catch-all account such as Miscellaneous Liabilities or any other account that meets your needs.

The adjustment can be made either to the lowest common parent organization of the two compared organizations or to one of the compared organizations. You make this choice on a per-cycle basis using the "Intercompany Balancing Rules" property setting for each cycle. To see this property setting:

1. Select **Tools** ⇨ **Cycle Manager**.

   The Cycle Manager window appears.

2. In the Cycle Manager window, select the correct cycle.

3. Select **Properties**.

   The Properties window for the selected cycle appears.

4. In the Properties window, select the **Intercompany Trading** page.

If you define a rule that makes intercompany balancing adjustments to one of the two compared organizations, then the software determines which organization receives the adjustment by examining the designated balancing account:

- 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.

If an intercompany balancing rule is part of a financial 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. To do this, use the **Write Members** tab of the rule's Properties window.

Note: The adjustments that are generated by an intercompany balancing rule are subject to an absolute value threshold. See Threshold for Adjustment Rules on page 93.

See Also

- Introduction to the Source Dimension on page 52
- What Happens When You Post Adjustments on page 92

Net Intercompany Balancing Rules

A net intercompany balancing rule looks for inequalities between complementary net accounts that concern financial relationships between organizations. For example, organization A's net value of all notes payable to and all notes receivable from organization B should have the same absolute value as organization B's net value of all notes payable to and all notes receivable from organization A.

A net intercompany balancing rule always examines crossings whose Source hierarchy member is TotalBeforeElim. The values that the rule finds at these crossings incorporate

relevant values from manual and rule-generated adjustments that were posted earlier in the posting sequence.

If the amounts that should be equal are not equal, then the rule generates an adjustment that removes the inequality. The generated adjustment places the value that is needed to achieve equality in two places—a designated balancing account and a designated offset account. The offset account is necessary in order to keep the balance sheet of the affected organization in balance. It can be a catch-all account such as Miscellaneous Liabilities or any other account that meets your needs.

The adjustment is always made to the lowest common parent organization of the two compared organizations.

If a net intercompany balancing 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. To do this, use the **Write Members** tab of the rule's Properties window.

Note: The adjustments that are generated by a net intercompany balancing rule are subject to an absolute value threshold. See Threshold for Adjustment Rules on page 93.

See Also

- Introduction to the Source Dimension on page 52
- What Happens When You Post Adjustments on page 92

Reporting-Entity Balancing Rules

A reporting-entity balancing rule looks for inequalities among the accounts of all organizations that are reporting entities. The typical use of a reporting-entity balancing rule is to verify the standard balance-sheet equation, Total Assets = Total Liabilities + Total Equity.

A reporting-entity balancing rule always examines crossings whose Source hierarchy member is Total. The values that the rule finds at these crossings incorporate relevant values that are associated with all subordinate members of the Source hierarchy.

When the rule finds an inequality, it generates an adjustment that removes the inequality and brings the organization's balance sheet into balance. The generated adjustment places the value that is needed to achieve equality in a designated balancing account for the reporting-entity organization within which the inequality was detected.

If a reporting-entity balancing rule is part of a financial 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. To do this, use the **Write Members** tab of the rule's Properties window.

See Also

- Introduction to the Source Dimension on page 52
- What Happens When You Post Adjustments on page 92

Allocation Rules

An allocation rule allocates the value at a designated input crossing among a set of designated output crossings. A single allocation rule can allocate the values at many input crossings, provided that all the allocations have the same pattern. The distribution of each input value among its designated output crossings is governed by the numbers in the allocation table that is the heart of the rule.

Typically, an allocation rule also places values at designated offset crossings. The purpose of the offset values is to cancel out the input values in the hierarchical roll-up process, so that reports show only the outputs of the allocation process.

You can think of the process of creating an allocation rule as a process of constructing an allocation table. In designating the input crossings, you are defining the rows of the allocation table. In designating the output crossings, you are defining the columns of the allocation table. In placing numbers in the cells of the table, you are specifying how the rule computes the output values from the input values.

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 could distribute certain sales and marketing costs among several product lines. This would facilitate a profitability analysis of the product lines.
- A two-step allocation rule could distribute certain sales and marketing costs among several product lines, and then distribute each resulting product line amount among several sales channels. This would facilitate a profitability analysis of each combination of a product line with a sales channel.

When an allocation rule runs, it generates adjustments that make all the specified changes. All the output values of the last step in the rule are written to the Alloc member of the Source hierarchy.

The input values for the first step are associated with the member of the Source hierarchy that you specify in the rule. The default choice is the Total member. With this choice, there is a roll-up through the entire Source hierarchy, incorporating all adjustments that were posted earlier in the adjustment process, before the allocation rule runs.

Note: The adjustments that are generated by an allocation rule are subject to an absolute value threshold. See Threshold for Adjustment Rules on page 93.

See Also

- Introduction to the Source Dimension on page 52
- What Happens When You Post Adjustments on page 92

Reclassification Rules

A reclassification rule transfers the value at a designated input crossing to a designated output crossing. A single rule can transfer many input values, provided that all the reclassifications have the same pattern.

Typically, a reclassification rule also places offsets at designated crossings. The purpose of the offsets is to cancel out the input values in the hierarchical roll-up process, so that reports show only the outputs of the reclassification process.

Reclassification rules are very similar to allocation rules. You can think of a reclassification as the simplest possible allocation, in which 100% of the input value is allocated to a single output crossing.

Typical uses of a reclassification rule include the following:

- The amount of a certain outstanding debt is stored in the SAS Financial Management database as a long-term liability. As the maturity date of the debt approaches, you would like this amount to appear in reports as a short-term liability. You can use a reclassification rule to transfer the amount from the long-term liability account to a short-term liability account.
- The SAS Financial Management database includes data for organization A. Organization A has recently been merged into organization B. You would like all the organization A values to appear in reports as organization B values. You can use a reclassification rule to transfer all the organization A amounts to the corresponding accounts within organization B.

When a reclassification rule runs, it generates adjustments that make all the specified changes. All the output values are written to the Reclass member of the Source hierarchy.

The input values are associated with the member of the Source hierarchy that you specify in the rule. The default choice is the Total member. With this choice, there is a complete roll-up through the Source hierarchy, incorporating all adjustments that were posted earlier in the adjustment process, before the reclassification rule runs.

Note: The adjustments that are generated by a reclassification rule are subject to an absolute value threshold. See Threshold for Adjustment Rules on page 93.

See Also

- Introduction to the Source Dimension on page 52
- What Happens When You Post Adjustments on page 92

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 financial model's organization hierarchy. Without an ownership rule, each parent-child relationship in an organization hierarchy is taken to represent full ownership.

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 special needs 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 two members. One of these members, called the consolidation parent, will contain the real-world organization's complete consolidated results. The other member, called the standalone parent, will contain data for the real-world organization's separate financial statements as defined by international accounting standards.

Each standalone parent must be a child of its corresponding consolidation parent. The consolidation parent must be a reporting entity. The standalone parent must be a reporting entity and must have no descendants that are reporting entities. In addition, it is a good idea to adopt a naming convention for the members of the organization hierarchy that makes plain the relation between each consolidation parent and its corresponding standalone parent. The ownership rule uses the Direct Parent holding type to mark this special kind of relation.

The first step in defining an ownership rule is to designate the corresponding standalone parent for each consolidation parent. This step is not possible if the underlying organization hierarchy is not designed properly.

See Also

- Introduction to the Organization Dimension Type on page 49
- Ownership Rules: Holding Types on page 108

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
: is the special relationship that exists between a consolidation parent and its companion standalone parent. Both the consolidation parent and its companion standalone parent represent the same real-world organization. The consolidation parent contains the real-world organization's complete consolidated results. The standalone parent contains the data for the real-world organization's separate financial statements as defined by international accounting standards.

Direct Transacted
: is 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 standalone parent.

Direct Non-Transacted
: is 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 standalone 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 standalone 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
: is 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 standalone 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 general 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 a set of three accounts:

- The Source account contains the held organization's net income.
- The Target account is 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.
- The Offset account is 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. From the point of view of 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.

As a general rule, 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 is used to resolve 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 standalone parent. These two members of an organization hierarchy represent the same real-world organization; the consolidation parent contains its consolidated financial results and the standalone parent contains its separate financial results as defined by international accounting standards. The Parent method performs a full consolidation of the standalone parent into the consolidation parent, applying any necessary adjustments to prevent the double counting of data from held organizations that is reflected in the standalone parent.

"No Consolidation" Method

"No consolidation" 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 this method to investigate the impact of excluding all data from the held organization.

"No Method"

"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 11

Composite Models

Introduction to Composite Models **113**

Required Features of a Composite Model **114**

Optional Features of a Composite Model **115**

Composite Models View **116**
Overview of the Composite Models View 116
Options for Composite Models 116

Introduction to Composite Models

A composite model is a structure that combines two or more financial models. The financial models that make up a composite model can belong to different cycles or the same cycle. If they belong to different cycles, then all the cycles must use the same set of dimension types.

Operational models cannot be made part of a composite model.

In the SAS Financial Management Add-In for Microsoft Excel, you can use a composite model whose constituent models belong to different cycles to create a read-only table or a CDA table that retrieves data from all the cycles that underlie the constituent models. For example, if you keep actual performance data in cycle X and financial planning data in cycle Y, then you can use a composite model to create a report that shows actual performance values and planning values side by side.

A composite model with only one underlying cycle can also be useful. For example, if you create several models that focus on different time spans within the same cycle, then you can use a composite model to create a report that covers all the time periods that are covered by the constituent models.

Whether you need to use composite models depends on the nature of your data and the way you define your cycles and models.

If a composite model has two or more underlying cycles that contain numeric values for the same crossing, then the values for that crossing from all the underlying cycles are added together as part of the process of computing the displayed value for that crossing. It is not necessary for the crossing to be represented in all the constituent models.

A composite model uses all the posted adjustments of all its constituent models. You cannot create additional adjustments for the composite model itself.

To work with composite models:

1. Open any financial cycle.
2. Select the Models workspace.
3. Select **Tools** ⇨ **Composite Models** to open the **Composite Models** view.

See Also

- Introduction to Models on page 83
- Introduction to Cycles on page 55

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.

Following are the required features of a composite model:

- a code, name, and description to identify the composite model

  Giving the composite model an identity is the first step in the New Composite Model wizard. After the composite model is created, you can change the name and description, but not the code, using the **General** tab of the composite model's Properties window.

- a set of constituent financial models

  Selecting the constituent financial models is the second step in the New Composite Model wizard. After the composite model is created, you can view its constituent financial models on the **Models** tab of the composite model's Properties window.

- a hierarchy for each dimension type that is used in the constituent financial models

  Although all the constituent financial models must use the same set of dimension types, they do not need to use the same set of hierarchies. Therefore, you need to specify a set of hierarchies for the composite model.

  The collection of all the specified hierarchies determines the set of crossings that can appear in tables that are based on the composite model. Any crossing that consists of one member from each specified hierarchy is potentially available. All the properties of the hierarchy members, such as formulas or security restrictions, become part of the composite model.

  The specification of each hierarchy includes both the name of the hierarchy and an as-of date. The composite model uses the version of the named hierarchy that is current at the as-of-date. There is one important exception: for every member of every hierarchy, access to associated data is always determined by the set of users that is specified on the member's **Security** page at the moment of use.

  Several pages of the New Composite Model wizard are devoted to selecting these hierarchies. After the composite model is created, you can select different hierarchies and as-of dates on the **Dimensions** tab of the composite model's Properties window.

- a rank ordering of the hierarchies for the purpose of resolving formula conflicts

  If a crossing is within the scope of two or more formulas that are associated with different members of the crossing, then the value at that crossing is determined by a

formula that is associated with the member that belongs to the most highly ranked hierarchy.

For a composite model, the rank ordering of hierarchies is set automatically and cannot be changed. You can view it on the **Dimensions** tab of the composite model's Properties window.

- the default read member for each hierarchy

  A hierarchy's default read member is used to complete any report query that does not include an explicit specification for that hierarchy.

  For a composite model, the default read members are always the default members of the hierarchies. You can see these default members on the hierarchy pages of the New Composite Model wizard and on the **Dimensions** tab of the composite model's Properties window, but you cannot change them.

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

  For any crossing whose value is calculated by means of a currency conversion, the applicable exchange rate 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 Actual crossings and Budget crossings, for example.

  You set these associations on the Exchange Rates page of the New Composite Model wizard. You can change them on the **Rates** tab of the composite model's Properties window.

- an association of a driver rate set with each analysis member in the composite 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 look-up in the driver rate set that is associated with the crossing's analysis member. This enables you to use different sets of tax rates, interest rates, or prices for Actual crossings and Budget crossings, for example.

  You set these associations on the Driver Rates page of the New Composite Model wizard. You can change them on the **Rates** tab of the composite 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 Composite Model wizard. You can change them on the **Formats** tab of the composite model's Properties window.

See Also

Resolving Conflicts between Formulas on page 179

Optional Features of a Composite Model

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, use the **Security** tab of the composite model's Properties window.

Composite Models View

Overview of the Composite Models View

The Composite Models view lists every existing composite model, without exception. The list of composite models is not related to the open cycle.

A composite model is sometimes marked with the Error icon . This icon indicates that at least one of the constituent financial models has an error. To discover the nature of the errors, identify the constituent models using the **Models** tab of the composite model's Properties window, and then investigate the constituent models in the Models view.

A composite model is sometimes marked with the Warning icon ( ). This icon indicates that at least one of the constituent financial models has a warning. To discover the nature of the warnings, identify the constituent models using the **Models** tab of the composite model's Properties window, and then investigate the constituent models in the Models view.

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.

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

Copy Here
makes a complete copy of the selected composite model. A window appears to enable you to specify the code, name, and description of the copy.

Delete
deletes the selected composite model. You are asked to confirm your desire to delete the composite model before the deletion occurs.

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

See Also

- Using Views on page 20
- Introduction to Composite Models on page 113

Chapter 12

Forms and the Forms Workspace

Forms and Form Sets **117**

Workflow Types **117**
Overview 117
Bottom-Up Workflows 118
Top-Down Workflows 118

The Forms Workspace **118**

Comparison of Financial and Operational Form Sets **118**

Forms and Form Sets

A form is a document that is typically used for data entry purposes. A form set is a collection of forms that can be sent through a data-entry and approval process.

The route that a form takes is determined by the form set's workflow. For example, after a user submits a form, it might go to a manager for review and approval, to that manager's manager, and on up the review process. If a form is rejected, it might be returned to the previous user for additional editing.

When you create a form set, you select a target hierarchy. From that hierarchy, you select a subset of members to be included in the workflow.

Every form set has a deadline (a due date and time), which applies to each of its forms. The deadline for each form is displayed in the Web-based Form Manager.

Workflow Types

Overview

There are two types of workflows in SAS Financial Management:

- 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 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 Solutions Services: 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. Approval leads to aggregation of the approved data. Rejection leads to revision of the rejected data followed by another submission for review.

Top-Down Workflows

In a top-down workflow, an initial user enters data at the highest level of the form set's target hierarchy. Those values are allocated down to the next level of members of the target hierarchy. If they are roll-up members of the target hierarchy, lower-level forms can further allocate those values to their subordinate members.

The Forms Workspace

In the Forms workspace, you can create and manage form sets that belong to the open cycle. If the open cycle is a financial cycle, then the form sets are financial form sets. If the open cycle is an operational cycle, then the form sets are operational form sets.

Every form set, whether financial or operational, is a set of similarly structured data-entry forms that enable users to enter data into the associated cycle. The structure of the data-entry forms is defined by a template that you create in Microsoft Excel.

The Forms workspace contains three views:

- The Form Sets view opens when you select the Forms workspace. It lists all the form sets in the open cycle and the phases that contain the form sets. Phases are folders that enable you to group form sets.
- The Forms view opens when you select the **Forms** option for a selected form set. It displays all the forms that belong to the selected form set.
- The Attachments view opens when you select the **Attachments** option for a selected form set. It displays all 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.

Comparison of Financial and Operational Form Sets

Financial form sets and operational form sets are broadly similar. The following table summarizes the more important differences between the two.

Table 12.1 *Differences Between Financial and Operational Form Sets*

Feature	Financial Form Sets	Operational Form Sets
Form template	Can include data-entry tables, read-only tables, CDA tables, and supplemental schedules. Data can be directly entered into a financial form template.	Includes at least one data-entry table. It cannot include any of the other table types. Data cannot be directly entered into an operational form template.
Use of models	A financial form set can contain multiple models that share the same cycle.	An operational form set is tied to a single model, which you select in the New Form Set wizard.
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.	The as-of date and time are specified in the model properties. When the form set is published, it uses the as-of date and time from the locked model.
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 every member of the target hierarchy in the workflow. However, a form can optionally include writable descendants.	Can be an organization hierarchy or a hierarchy from any non-required dimension type in the underlying cycle. It cannot be an account hierarchy. There is one form for each member of the target hierarchy in the workflow.
Hierarchy levels	Every form has the same granularity of crossings. For example, exactly the same time periods appear in every form. If one form has years, quarters, and months, then all the forms in the form set have years, quarters, and months.	On the Form Detail page for an operational form set, you can divide the target hierarchy into several tiers and specify a different granularity of crossings for each tier. A crossing that is present in one tier must be present in all the tiers below it.
Form authors and reviewers	A bottom-up form can have multiple authors and reviewers. A top-down form can have multiple authors.	A bottom-up form can have only one author and one reviewer. A top-down form can have only one author.
Formula types	Formulas of all three types—reporting formulas, driver formulas, and modeling formulas—can be executed. You can use the **Run driver formulas** option to make sure that the stored output values of all driver formulas are current. (This option is not available when the target hierarchy is the Account dimension.)	Only driver formulas can be executed. The **Run driver formulas** option is not available. However, **Run driver formulas** is automatically executed when the form set is published. Any reporting formulas or modeling formulas that are associated with members in the form set are ignored. Driver formulas that include references to members other than time periods and accounts are also ignored. If you publish a form set with formulas that will be ignored, a message informs you of this fact. At any time, you can check for formula warnings on the Formulas page of the model properties window.

Feature	Financial Form Sets	Operational Form Sets
Data-entry facilities	Users can edit forms in Microsoft Excel or in the Web-based Form Editor, depending on the form set properties. If a form includes a supplemental schedule, it must be edited in Microsoft Excel.	Users edit forms using the Web-based Form Editor. (The form template is designed in Microsoft Excel.) Supplemental schedules are not available.

Chapter 13

Working with Forms and Form Sets

Creating and Publishing a Form Set **122**
Overview 122
Create the Form Set 122
Define the Form Template 123
Edit the Form Set Properties 123
Publish the Form Set 124

Form Sets and the Target Hierarchy **125**
(Financial Form Sets) Selecting Target Members 125
(Operational Form Sets) Selecting Target Members 127

Assigning Form Authors and Reviewers **129**
Overview 129
Assigning Authors and Reviewers to Financial Forms 129
Assigning Authors and Reviewers to Operational Forms 131

Enforcing Form Set Deadlines **133**
Overview 133
Setting the Deadline 134
Automatically Locking Forms 134
Sending Reminders 134
Overriding Auto-Locking 135

Time Periods in a Data-Entry Table **135**
Overview 135
Fixed Time Periods 135
Floating Time Periods 136

Cell Protection for Data-Entry Tables **138**
About Cell Protection 138
Cell Protection Rules 140
Defining Cell Protection for a Model 140
Defining Cell Protection for a Form Set 141
Editing Cell Protection Rules for a Model or a Form Set 141
Adding Cell Protection at the Form Level 143
Cell Protection and Supplemental Schedules 143

Allow Data to be Entered for Parent Members (Financial Forms Only) **144**

Allocating Using Predefined Weights (Financial Forms Only) **145**
Overview 145
Enable Automatic Allocation 145
Eligible Cells 145
How the Allocation Is Applied 146
Allocations to Non-Additive Members 147

Allocations to Members of the Target Hierarchy 148
Modeling and Driver Formulas and Automatic Allocation 148
Reallocation 149
Editing Allocation Weights 149
Creating a Separate Weight Table 151

Supplemental Schedules 151
Overview 151
Implementing a Custom Data Provider 152
Designing a Form Template 152
What Happens at Run Time 153

Creating and Publishing a Form Set

Overview

Each form set is associated with a cycle and a model. A financial form set can use multiple models, but an operational form set has a single model.

Before you create a form set, you must define the following entities:

- a cycle.
- an exchange rate set for that cycle.
- one or more models within that cycle.
- (operational form sets only) a member property to be used in selecting the form set's target hierarchy. (See "Selecting a Form Group" on page 128.)

Here is a high-level view of the steps that are required to create and publish a form set:

1. Create a form set.
2. Define the form template.
3. Edit the form set properties, as needed.
4. Publish the form set.

Create the Form Set

To create a form set:

1. Select **Tools** ⇨ **Cycle Manager** and select the cycle that this form set will be associated with.
2. Select **Forms** to open the Forms workspace.
3. Select a phase. You can use an existing phase, or click **Create a new phase** to create a phase to hold the form set.
4. Select **Create a new form set**.

   In the form set wizard, your selections include the following:

   - the workflow type, which can be bottom-up or top-down. See "Workflow Types" on page 117.
   - (operational forms only) the model for the form set.

- the members that will be included in the workflow for this form set. See "Form Sets and the Target Hierarchy" on page 125.
- the currency for the form set (financial form sets only).
- the writable members of the Analysis dimension.
- the date and time that the form set is due and the actions, if any, that will be taken regarding this deadline. See "Enforcing Form Set Deadlines" on page 133.

5. To go on to create the form template immediately, select **Open Excel to create the form template** on the form set wizard Summary page.

   Otherwise, you can later select the form set and select **Template** to open the template in Microsoft Excel.

Define the Form Template

1. (Financial form sets) With the form template open in Microsoft Excel, add any combination of data-entry tables, read-only tables, CDA tables, and supplemental schedules.

   (Optional form sets) With the form template open in Microsoft Excel, create at least one data-entry table that is based on the model.

   Note: The template for a financial form set is not required to contain a data-entry table. Operational form sets require at least one data-entry table and cannot contain any other type of table.

   To invoke the wizard, select **Insert** ⇨ **Data Entry Table**. See "Designing a Data-Entry Form Template" on page 214.

2. As necessary, apply formatting, calculated members, and Visual Basic code that operates on the data-entry form and calls methods in the SAS Financial Management API for Microsoft Excel.

3. Select **Save Template** to save the template on the server. (You do not need to save the template on your desktop.)

Edit the Form Set Properties

Edit the form set properties as needed:

1. In SAS Financial Management, the form set now has a status of Draft. (If the form set status is No template, click the Refresh button  .)

2. When you select the form set, the **Forms** tab displays the forms in the workflow. Each form has a status of Draft. You can edit the form set properties or individual form properties as necessary. For example, you might want to assign additional reviewers to a financial form, or you might want to select different members of the target hierarchy.

   Operational forms open only in the Web-based form editor.

   By default, financial forms open only in Microsoft Excel. If you want the forms to be editable in the Web-based form editor, you can make that selection in the form set properties after you create the form set.

   Note: Supplemental schedules can be used only in financial form sets, and data-entry for these forms can occur only in Microsoft Excel.

Publish the Form Set

1. In the Forms workspace of SAS Financial Management Studio, select the form set and select **Publish**.

   Note: You cannot publish an operational form set unless each of its forms has an author and a reviewer.

2. If you select **Notify users**, notifications are sent to each of the first set of users in the form set workflow. Users select their notification language and type (e-mail or alert, or both) via the **Preferences** option in the portal.

3. Users can log on to Form Manager and view their forms. Users open a form, enter data, and submit the form to the next state in the workflow. For a top-down workflow, users review forms and approve or reject them.

   Note: Form administrators can access all forms in the Form Manager.

4. When all the forms are submitted, mark the form set Complete.

5. If necessary, you can reset the form set and republish it.

   If you reset an operational form set, you are given the choice of deleting or preserving the associated data.

   If you reset a financial form set, the data associated with the form set is preserved. (To delete the data associated with a financial form set, use the **Delete** operation.)

For financial form sets: If data is loaded into the form set's cycle while the forms are available for data-entry, then the newly loaded data is immediately visible in the forms.

For operational form sets: When an operational form set is published, a separate data store is created for each published form. Subsequent changes to the form set's cycle are not reflected in the forms. When the form set is marked Complete, the form's data is added to the main SAS Financial Management database.

See Also

Using the Web-based Form Manager and Form Editor

- “Form Manager Page” on page 291
- “Introduction to Data Entry” on page 289

Editing forms in Microsoft Excel

- “Entering Data as Part of a Bottom-Up Workflow” on page 221
- “Entering Data as Part of a Top-Down Workflow” on page 222

Form Sets and the Target Hierarchy

(Financial Form Sets) Selecting Target Members

Overview

When you create a financial form set, you select a target hierarchy from the organization dimension, the account dimension, or any non-required dimension in the open cycle. From that hierarchy, you select a subset of members to be part of the workflow. When you publish the form set, there is a form for each target member in the workflow.

For a financial form set, two sets of rules are applied to determine the target members that will be part of the workflow:

- member selection rules
- descendant rules

When you publish the form set, the rules are applied based on the hierarchy's as-of date and time. Any members that are not writable at publish time are removed from the workflow.

Selecting a Hierarchy

To select a target hierarchy for a financial form set, you make these selections:

- **As-of date**: The target hierarchy can undergo changes as people edit it in SAS Financial Management Studio or load different members from the Cross Industry Detail Data Store. Only active members for the selected as-of date are included in the target hierarchy.

  The default as-of date is **Current until form set is published**. When you publish the form, the target hierarchy is based on that date. Alternatively, you can click the As-of Date button to select an earlier date and time.

  Note: The default set of authors for a form is not subject to the as-of date.

- **Dimension**: the dimension that contains the target hierarchy.

  You can select from the organization dimension, the account dimension, and all non-required dimensions in the open cycle.

  If you are creating a form set that will include an HCM supplemental schedule, then you must select the organization dimension.

- **Hierarchy**: the target hierarchy for the form set.

  You can select from all available hierarchies in the selected dimension. To be available, a hierarchy must be used in at least one model that belongs to the open cycle.

  If you are creating a form set that will include an HCM supplemental schedule, you must select an organization hierarchy that is used by SAS Human Capital Management at your site.

Applying Member Selection Rules

To select target members for the workflow, you apply the following rules:

Table 13.1 Member Selection Rules

Rule	Description
Member	The member is added to the **Target Member** list.
Member and its children	The member and all its immediately subordinate members are added to the **Target Member** list.
Member and its descendants	The member and all its subordinate members are added to the **Target Member** list.
Member and its leaf descendants	The member and its subordinate members that are leaf members are added to the **Target Member** list.
Children of member	The immediately subordinate members (but not the member itself) are added to the **Target Member** list.
Descendants of member	All the subordinate members (but not the member itself) are added to the **Target Member** list.
Leaf descendants of member	All the leaf descendants (but not the member itself) are added to the **Target Member** list.
Exclude	The member is excluded from the **Target member** list. All subordinate members are also excluded, unless you apply additional rules to one or more of these subordinate members.
No rule	No rule is directly applied to this member, although rules might be inherited from a parent member.

Applying Descendant Selection Rules

By default, a separate form is created for each member that is part of the workflow. In some cases, you might want a form author to be assigned one inclusive form that also applies to several subordinate members.

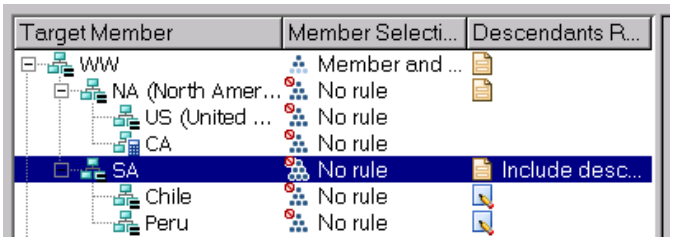
To include additional members in a parent form, use the member selection rules to select only the members that you want to be assigned forms. Then follow the descendant selection rules to select members that will be included in parent forms:

Table 13.2 Descendant Selection Rules

Rule	Description
Include member in parent form	This member is included with the closest parent that is selected to be assigned a form. This rule takes precedence over parent rules. It does not apply to descendants of this member.

Rule	Description
Exclude member from parent form	This member is excluded from parent forms. This rule takes precedence over parent rules. It does not apply to descendants of this member.
Include descendants in parent forms	All eligible descendants are included with the closest parent that is selected to be assigned a form.
Exclude descendants from parent forms	Descendants of this member are not included in parent forms.
Remove rule	Remove the rule from this item.

For example, assume that you have a member, SA, with two child members, Chile and Peru. You want to include both Chile and Peru in the form that is created for the SA member. You set the member selection rule for WW to **Member and its children**. As a result, forms are assigned to the WW, NA, and SA members. No child members of NA or SA are included. Then you set the descendant rule for SA to **Include descendants**.



As a result, the Chile and Peru members are included in the form for SA. Because you did not apply a similar descendant selection rule to NA, none of its child members are included on the NA form, as shown in the preview region:

(Operational Form Sets) Selecting Target Members

Overview

For an operational form set, you select a model and a target hierarchy. From that target hierarchy, you select a subset of members to be included in the workflow:

- First you select a member property that separates the target hierarchy into form groups. You select one or more form groups to be included in the workflow.
- Then you apply member selection rules to the members of the selected form groups. When you publish the form set, the rules are applied based on the hierarchy's as-of date and time. Any members that are not writable when the form set is published are removed from the workflow.

When you publish the form set, there is a form for each target member in the workflow.

The Target Hierarchy

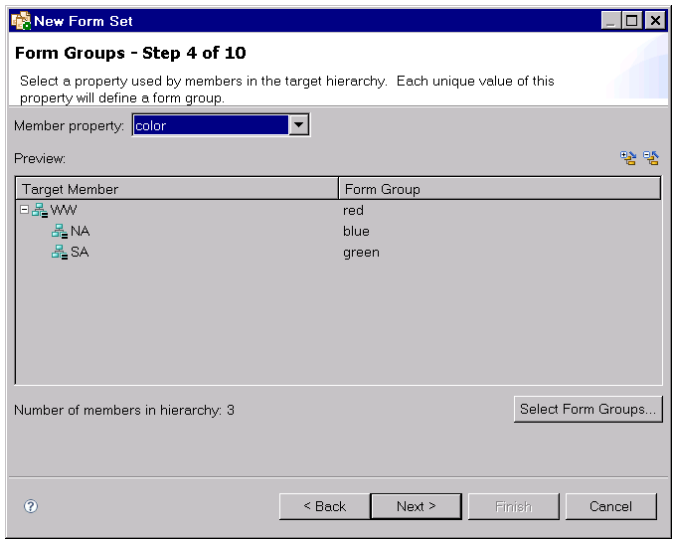
On the Target Hierarchy page for an operational form set, you select from the following hierarchies in the model that you selected:

- the organization hierarchy
- the hierarchies of any nonrequired dimension types that are in the model

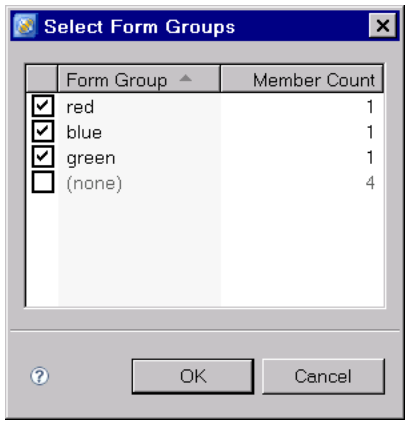
You do not select an as-of date; for operational planning, the as-of date is inherited from the as-of date in the selected model.

Selecting a Form Group

On the Form Groups page, you select a custom property that can be used to separate the target hierarchy into form groups. This property must be associated with at least one member of the target hierarchy.



The Form Group Selection window displays the form groups. Form groups are defined by the values that are assigned to the selected member property. The window also displays the number of target hierarchy members in each group.



If you select the check box for a form group, then its members are eligible to be part of the workflow (subject to the member selection rules).

A default form group of **(none)** represents all target hierarchy members that are not assigned a value for this member property.

Note: You can also use form groups to define a different level of detail for the forms in each group. See the online Help for the Form Detail page.

Applying Member Selection Rules

To select members to be included in the workflow, you apply member selection rules to the target hierarchy. See Table 13.1 on page 126.

The preview region displays the members that are included in the workflow. Each of these members corresponds to a form in the form set.

Assigning Form Authors and Reviewers

Overview

In both bottom-up and top-down workflows, a form author edits a form and sends it to the next state in the workflow.

In bottom-up workflows, a form reviewer examines the form's contents and approves or rejects the form.

A financial form can have multiple authors and reviewers. If a form has more than one author, any author can edit the form. If a form has more than one reviewer, any reviewer can approve or reject the form, unless you define an approval sequence. In that case, the form remains partially approved until all reviewers in the sequence have approved it.

An operational form can have only one author and one reviewer.

Assigning Authors and Reviewers to Financial Forms

Default Authors and Reviewers

Each form in a form set is assigned to a member of the target hierarchy. The default authors for a financial form are users who match these criteria:

- They are currently associated with this member. (You can view these user-member associations in the Dimensions workspace, on the **Users** tab of the member properties.)
- They have the Financial Management: Form Submitter role.

The default reviewers for a financial form are users who match these criteria:

- They are the default authors of the form's closest roll-up member.

  For example, assume a target hierarchy with members A, B, and C. A is the parent of B, and B is the parent of C. If only members A and C are included in the workflow, then the default reviewers for form C would come from the default authors of form A.
- They have the Financial Management: Form Approver role.

You can assign additional users, groups, and members to supplement or replace the default authors and reviewers.

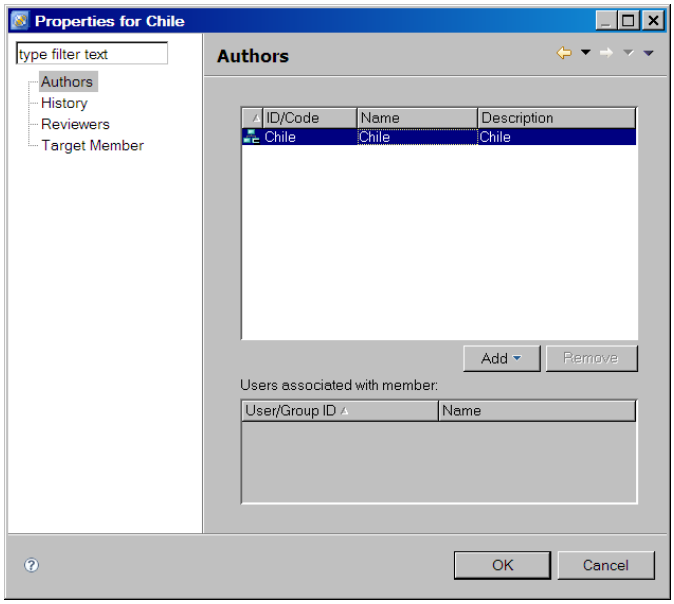
Adding Authors for a Financial Form

To modify the authors for a form, follow these steps:

1. Open the form properties.

Note: Instead of editing form properties, you can right-click a form set and select **Authors** or **Reviewers**. In the window that appears, you can select a form and view and modify its authors or reviewers.

2. Select the **Authors** tab to display the default authors for this form.



3. To add users or groups to the list of authors:

   a. Select **Add Users/Groups**.

   b. In the window that appears, type a search string and click **Search**. To view the entire list, leave the search string empty.

      The list is filtered to include users with a required role. The groups are not filtered by role membership.

   c. Click **Add**. The users and groups that you selected are displayed in the **Authors** list.

      If you added a group, you can select the group in the **Authors** list. At the bottom of the window, the **Users associated with this group** list displays group members who have the Financial Management: Form Submitter role.

4. To add members to the list of authors for this form:

   a. Select **Add Members**.

   b. From the **Dimension** drop-down list, select a dimension. The hierarchies for that dimension are displayed.

   c. Select a member and click **Add**.

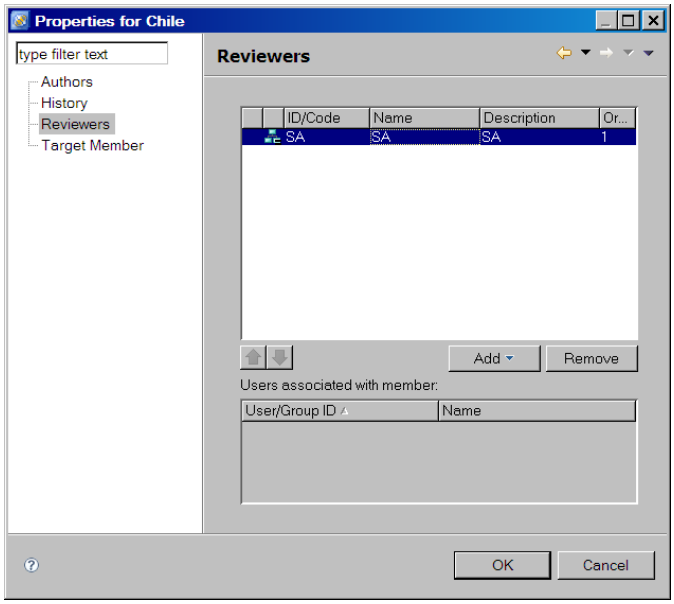
      The member is displayed in the **Authors** list. If you select the member, any associated users who have the Financial Management: Form Submitter role are displayed in the **Users associated with this member** list.

When a user opens Form Manager, a check is made to see whether the user has a required role (either directly or via group membership). For example, you add GroupA to the Authors list. Neither GroupA nor any of its users has a required role, and so no user appears in the **Users associated with this group** list. Later, you grant the Financial Management: Form Submitter role to User1, a member of GroupA. You log on to the portal and (in Document Manager) you run the utility to clear the role cache. In Form Manager, User1 can view and edit the form.

(Bottom-Up Forms Only) Adding Reviewers for a Financial Form

To modify the reviewers for a form, follow these steps:

1. Open the form properties.
2. Click the **Reviewers** tab, which contains the default reviewer: the parent member for this form in the workflow. In this example, the parent is the SA member.



3. Click **Add Users/Groups** or **Add Member** to add a user, group, or another member to the list of reviewers. See "Adding Authors for a Financial Form" on page 129 for details.

   If you select **Add Member** from the **Reviewers** tab, only members of the target dimension are available for selection.

4. (Optional) Create an approval sequence.

   If the **Reviewers** list contains more than one reviewer, the **Order** column displays the sequence in which this form must be reviewed, beginning with level 1. If reviewers are at the same level, only one of those reviewers must approve the form.

   Use the up and down arrows to change the sequence.

Assigning Authors and Reviewers to Operational Forms

Default Authors and Reviewers

Each form in a form set is assigned to a member of the target hierarchy. The default author for an operational form is a user who matches these criteria:

- The user is currently associated with this member. (You can view these user-member associations in the Dimensions workspace, in the member properties.)
- The user has the Financial Management: Planning Data Entry User role.

The default reviewer for an operational form is a user who matches these criteria:

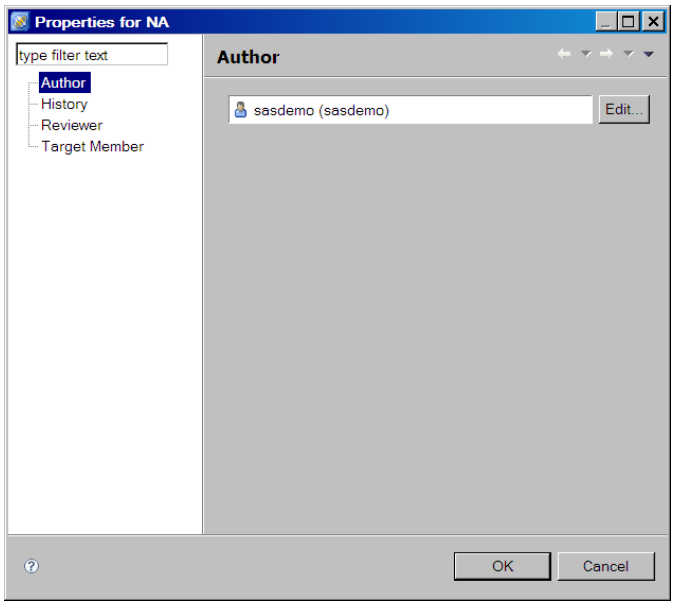
- The user is the default author of the form's closest roll-up member.
- The user has the Financial Management: Planning Data Entry User role.

Note: If there is more than one user with this role, only one user is selected.

Assigning an Author to an Operational Form

To modify the author for an operational form, follow these steps:

1. Open the form properties.
2. Click the **Author** tab, which displays the default author, if there is one.



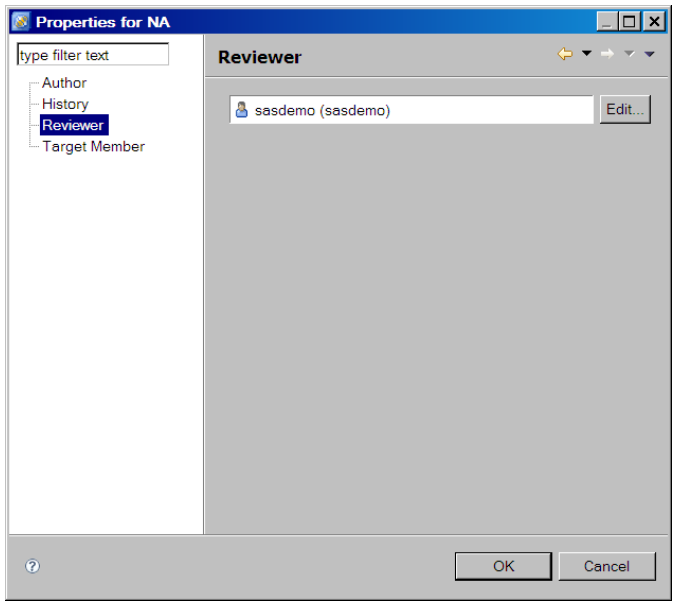
3. Click **Edit** to select a different user. In the Add Users window, you can select from available users. The list is filtered to include only users with the Financial Management: Planning Data Entry User role.

Note: Instead of editing form properties, you can right-click a form set and select **Authors** or **Reviewers**. In the window that appears, you can select a form and view and modify its author or reviewer.

(Bottom-up Workflows Only) Assigning a Reviewer to an Operational Form

To modify the reviewer for an operational form, follow these steps:

1. Open the form properties.
2. Click the **Reviewer** tab, which displays the default reviewer, if there is one.

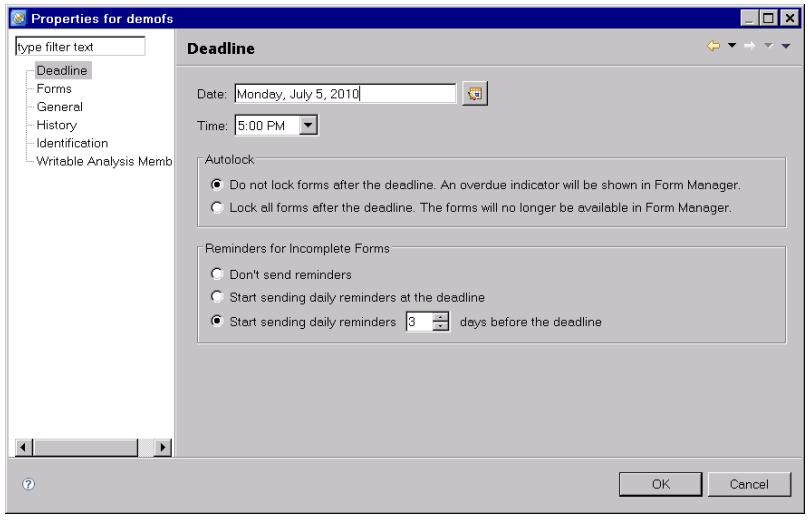


3. Click **Edit** to select a different user. In the Add Users window, you can select from available users. The list is filtered to include only users with the Financial Management: Planning Data Entry User role.

Enforcing Form Set Deadlines

Overview

In the Deadline properties for a form set, you can set the date and time that a form set is due and specify what action, if any, will be taken regarding the deadline.



Setting the Deadline

From the **Date** and **Time** drop-down lists, you select a due date and time to apply to all forms in the form set. The time selection is on the hour.

Note: In Form Manager and in notifications, deadlines are displayed in terms of the form user's time zone.

Automatically Locking Forms

In the **Autolock** section, you can set the way forms are treated once the deadline is reached:

- Select **Do not lock forms after the deadline** to suggest a date and time when all forms in a form set are expected to be submitted and approved.

  In Form Manager, an overdue indicator is shown for forms that are past due. However, forms remain available for data entry, submission, approval, and other workflow actions (subject to workflow rules) until they are explicitly locked or until the form set is completed.

- Select **Lock forms after the deadline** to enforce a due date and time when all forms in a form set must be submitted and approved.

  When the deadline arrives, the forms are locked. They are no longer available in the Form Manager list, and no further workflow actions are permitted. If a user is editing a form, the form status changes to read-only.

  Note: The form set itself is not locked.

Manual locking has the same effect on a form. It is no longer available in Form Manager. If the form is open, it becomes a read-only form, and a notification is sent to the user. (An administrator can manually lock a form by locking the form itself, its containing form set, or its containing cycle.)

Sending Reminders

In the **Reminders** section, you can choose whether to send automated reminders to form authors and reviewers. The following options are available:

- **Don't send reminders**: No reminders are sent.
- **Start sending daily reminders at the deadline**: Reminders are sent when the deadline is reached and continue to be sent every day until the form set is locked or completed.

  Note: This option applies only when automatic locking is disabled.

- **Start sending daily reminders before the deadline: __ days(s)**: Reminders are sent beginning the specified number of days before the deadline. The reminders continue to be sent every day until the form set is locked or completed.

Reminders are sent at the same time that you selected for the deadline. Reminders are sent to the following recipients:

- the author, if the form's state (as displayed in Form Manager) is Unedited, Edited, or Rejected
- the first reviewer, if the form's state is Submitted
- the next reviewer in the workflow, if the form's state is Partially approved

Note: After you publish a form set, you can send additional (non-automated) reminders. Select the form set or select one or more forms. Then select **Send Notification**.

Overriding Auto-Locking

As long as the form set itself is not locked or completed, you can override auto-locking using one of these methods:

- Disable the auto-lock option for a form set.
- Extend the deadline for a form set.

Disabling auto-lock or extending the deadline does not automatically unlock any forms. You must explicitly unlock the forms. This feature gives you the option to extend more time to selected users, while keeping other forms locked.

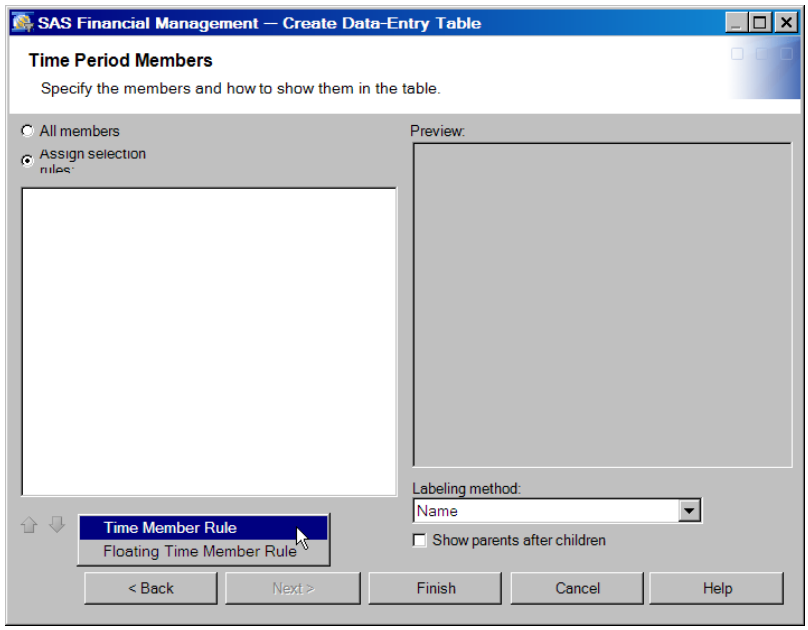
See Also

"Locked Objects" on page 16

Time Periods in a Data-Entry Table

Overview

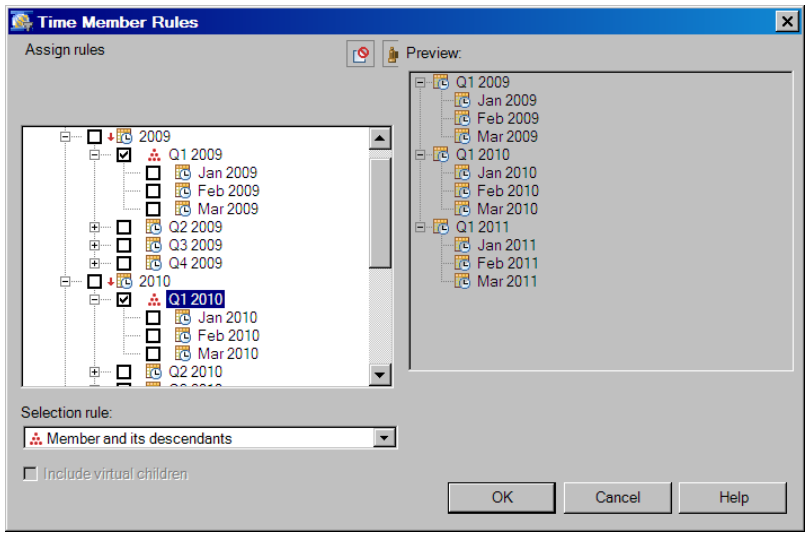
When you create a data-entry table in a form template, you select a time period from the time hierarchy. You can select **All members**, or you can select **Assign selection rule** and define a time member rule, which can be fixed or floating.



Fixed Time Periods

To select fixed time periods from the time hierarchy, select **Assign selection rules**. From the **Add** button's drop-down list, select **Time Member Rule**.

In the Time Member Rules window, you can select multiple time periods, consecutive or not. For each period, make a selection from the **Selection rule** drop-down list. The members that you select are included in the table. The preview region displays the results of your selections.



Floating Time Periods

About Floating Time Periods

In the Floating Time Member Rule window, you can specify a time period that is relative to the table default read member for the time hierarchy. By default, the table default read member inherits its value from the model for the table.

A floating time member rule enables form administrators to easily reuse form sets. If you design a form template to use fixed time periods, and you want to reuse the form set using different time periods, you must edit the member selection rules for the time hierarchy in the form template.

However, if you define a floating time member rule, the dates in the template (and the forms) are relative to the default read member for the time hierarchy, as set in the model. In that case, you could update the time hierarchy's default read member in the model properties, and the change would apply to any form sets that use that model.

In both cases (whether you update the form template or the model), the change applies immediately. You do not need to republish the form set.

Note: To view the table default read member, open the table properties and click the **Dimensions** tab. If you modify the table default read member in the table properties, this value is no longer updated by changes to the model.

Define the Viewable Time Period

To define a floating time period rule, select **Assign selection rules**. From the **Add** button's drop-down list, select **Floating Time Member Rule**.

To define the viewable time period, in the **Periods** region, make selections for **First period** and **Last period**. The selections are as follows:

Table 13.3 *Floating Time Member Rule: Viewable Periods*

Prompt	Selection	Description
First period	**Default period**	The table default read member
	Default period plus or Default period minus	The specified number of periods before or after the default period.
Last period	**First period**	The value that you specified as **First period**.
	First period plus	The specified number of periods after the first period.

If the time hierarchy's default read member is 2011, the **`Default period plus 3`** is 2014. If the time hierarchy's default read member is September 2011, the **`Default period plus 3`** is December 2011.

You must specify time periods that are at the same hierarchical level as the table default time period. If the table default time period is not the lowest level of the time hierarchy, all subordinate time periods are included. For example, if the time hierarchy includes years and their months, and the table default time period is a year, then each year that you specify in this window also includes the months for that year.

Define the Editable Time Period

The floating time member rule specifies both a viewable time period and an editable time period. An editable period can be the entire viewable period or a subset of the viewable period.

In a form, crossings with editable time periods are displayed in yellow (subject to other criteria such as cell protection rules and writable analysis members). If a crossing would otherwise be editable but contains a period that cannot be edited, it is displayed in purple and is protected against manual data entry and spread.

To define the editable time period, in the **Editable Periods** region, make selections for **First editable period** and **Last editable period**. The selections are as follows:

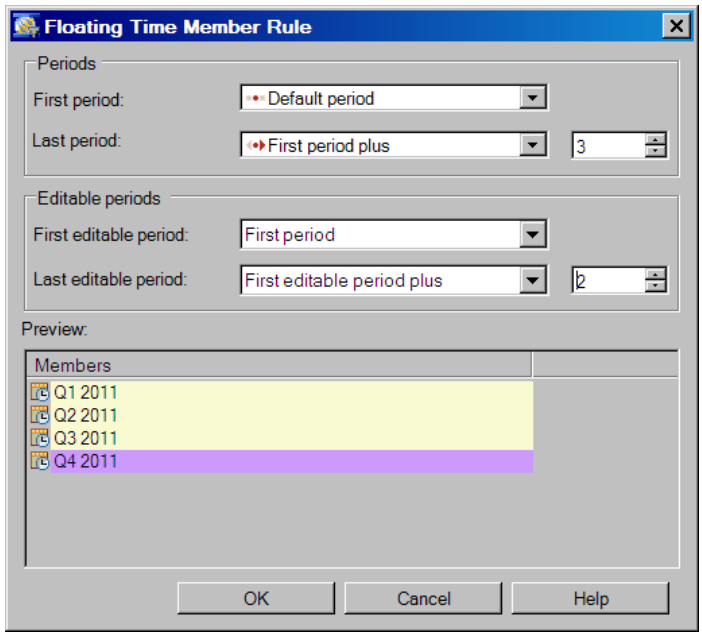
Table 13.4 *Floating Time Member Rule: Editable Periods*

Prompt	Selection	Description
First editable period	**First period**	The value that you specified as **First period** when you defined the viewable periods.
	First period plus	The specified number of periods after the first period.

Prompt	Selection	Description
Last editable period	**First editable period**	The value that you specified as **First editable period**.
	First editable period plus	The specified number of periods after the first editable period.
	Last period	The value that you specified as **Last period**.

The preview region shows the results of your selections. Editable periods are displayed in yellow. Protected periods are in purple. Only periods at the level of the table default read member are displayed in this window, even if there are subordinate time periods.

In the example below, all four quarters of 2011 are viewable, but only the first three quarters can be edited.



Cell Protection for Data-Entry Tables

About Cell Protection

You can protect cell crossings in a data-entry form by creating one or more rules that apply to the dimensions in the data-entry table.

Cells are protected against the following actions:

- manual data entry
- spread
- automatic allocation (applies only to financial forms in a bottom-up workflow)

However, the values of these protected cells can still change as the result of indirect actions, including the following:

- calculations
- changes in the values of descendants that roll up to the protected cell
- changes in exchange rates
- changes in previous periods when frequency is To Date (for example, Year To Date or Quarter To Date)
- data that is loaded via SAS Data Integration Studio jobs
- data that was seeded from other models
- rules-based adjustments and allocations
- manual adjustments

Cell protection is applied in the following order:

1. Rules that are defined in a model. These rules are inherited by every form set that uses the model.
2. Rules that are defined in a form template. These rules, as well as the rules from the model, are inherited by all forms in the form set.
3. Cell protection that is set in a data-entry form. This protection applies only to the form in which it is defined and applies only to financial forms. You must set form-based cell protection in Microsoft Excel, but the protected cells are visible (and honored) in the Web-based Form Editor as well.

A form cannot override the protection that was set in the form set or the model, and a form set cannot override the protection that was set in the model. For example, if the model rules protect a specific crossing, the form set and its forms cannot unprotect it. However, both the form template and individual forms can define additional cell protection.

Cell protection applies to all users, regardless of their group or role membership. In a data-entry table, cells that are protected are displayed in purple. However, their contents are still viewable.

Supplemental schedules in a data-entry form honor cell protection. Forms that are being edited offline honor cell protection, with the exception of changes to cell protection rules that occurred after the form was checked out.

Note: The member protection option has been replaced by cell protection. In form templates that were migrated from a previous release, member protection settings are converted to cell protection rules.

See Also

- "Cell Protection Rules" on page 140
- "Defining Cell Protection for a Model" on page 140
- "Defining Cell Protection for a Form Set" on page 141
- "Adding Cell Protection at the Form Level" on page 143
- "Cell Protection and Supplemental Schedules" on page 143

Cell Protection Rules

You can define cell protection rules for both models and form sets. When you define a cell protection rule, you make the following selections:

- the dimension members to be included in the rule. If a crossing includes a member from each selected dimension, the crossing participates in the rule. Other dimensions are ignored.
- whether crossings that participate in the rule are protected or unprotected. Unprotect rules can be used to undo the effect of a previous rule; for example, you might apply a protection rule to a broad range of crossings and then unprotect a subset of those crossings.

Here is a simple example:

Rule	Rule Type	PRODUCT	INTORG	TIME
1	Protect	P1	US, SA	Q1 2011
2	Unprotect	P1	Chile	Q1 2011
3	Protect	P3		2011

Within the same model or template, rules are applied based on their order, starting with the first rule in the list. For example:

- In the first rule above, all crossings for product P1 and its descendants are protected for US and SA and their descendants (including Chile), in the first quarter of 2011.
- The second rule overrides rule 1's protection for crossings that include Chile, which otherwise would have the same protections as its parent, the SA member.

The third rule protects all crossings for product P3 and its descendants for all of 2011, for all members of the INTORG dimension.

See Also

- "Defining Cell Protection for a Model" on page 140
- "Defining Cell Protection for a Form Set" on page 141

Defining Cell Protection for a Model

Cell protection rules for a model are typically loaded to the SASSDM database by a SAS Data Integration Studio job. The rules apply to all form sets that use the selected model.

From the Models workspace of SAS Financial Management Studio, you can open a Microsoft Excel workbook to view and modify these rules, or to add new rules. Be aware that loading the rules again from the SAS Data Integration Studio job will delete any rules that you added or changed via Microsoft Excel.

To view, modify, or add cell protection rules for a model via Microsoft Excel, follow these steps:

1. In SAS Financial Management Studio, select a model and select **Show cell protection rules**.

A worksheet opens in Microsoft Excel, with the Cell Protection window open.

2. To confirm that your rules are being applied correctly, select **Insert** ⇨ **Data-Entry Table** to create a data-entry table. Make sure the cells that should be protected are displayed in purple.
3. You can modify these rules or define new rules. See "Editing Cell Protection Rules for a Model or a Form Set" on page 141.

When you copy a model, its cell protection rules are automatically copied as well.

Defining Cell Protection for a Form Set

In a form template, you can define cell protection rules for a data-entry table. The rules apply to all forms in the form set.

- **For financial form sets:** If the form template contains more than one data-entry table, tables that use the same model share the same rules. If the data-entry tables use different models, they have different sets of cell protection rules.
- **For operational form sets:** All the data-entry tables in a form template use the same model and share the same rules.

Follow these steps:

1. In the form template, insert a data-entry table (or edit an existing data-entry table).
2. Select any cell in the table and click **Edit Protection Rules**. See "Editing Cell Protection Rules for a Model or a Form Set" on page 141.

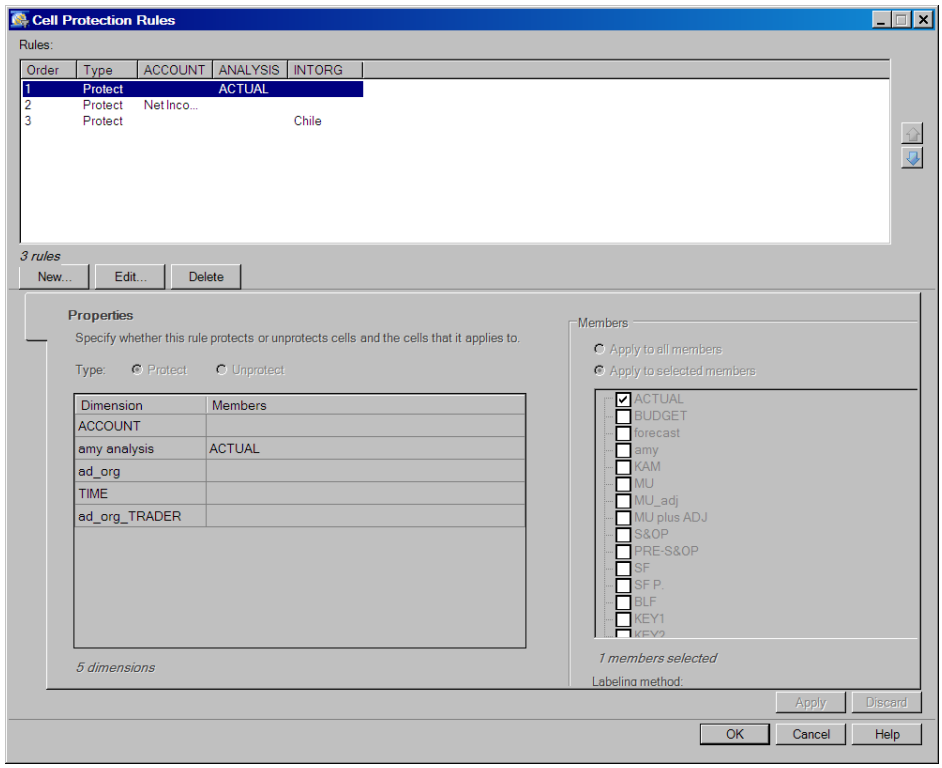
   Note: In the data-entry table, you will see the effect of any cell protection rules that were defined in the model. The model rules are maintained separately; you can view them by selecting **Cell Protection Rules** for a model in SAS Financial Management Studio.

When you modify the cell protection rules for a form template, you do not need to save the template (unless you make other changes). The new protection rules apply to all forms that use that template, even if the form set has been published. You do not need to republish the form set.

When you copy a form set, you have the option of including the form set's cell protection rules in the copy.

Editing Cell Protection Rules for a Model or a Form Set

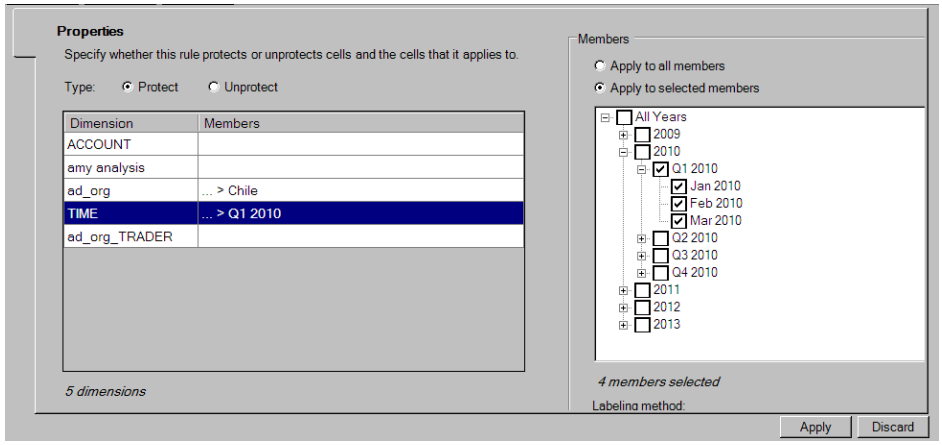
For a model or form set, you can define one or more rules to protect crossings in data-entry tables.



The following options are available:

- To create a rule, click **New**.
- To edit an existing rule, select the rule and click **Edit**.
- To remove an existing rule, select the rule and click **Delete**.
- To reorder the list of rules, use the up and down arrows. Rule number 1 is applied first, then rule number 2, and so on.

To add or edit a cell protection rule:



1. In the **Create a rule** region, select the **Protect** radio button to protect crossings that participate in the rule.

   Select the **Unprotect** radio button to undo protection for crossings that participate in the rule.

2. For each dimension that will be considered for this rule, do the following:

   a. Select a dimension.

   b. In the **Members** region, select one of the following radio buttons:

      - **All members within rule crossings**: all members of this dimension that meet the other criteria
      - **Selected members within rule crossings**: in the hierarchy, select the check boxes for one or more members. Member selection always includes the member's descendants and virtual children (.VC).

        To affect only the parent member, select the .VC member instead of the parent.

   A crossing participates in the rule if it contains a selected member from each selected dimension. The remaining dimensions are ignored (which is the same as selecting all members from each of the remaining dimensions).

3. Click **Apply** to complete the rule definition.

   Click **Discard** to discard any changes you made.

4. When you are finished, click **OK**.

   The rules are automatically written to the database, and any data-entry tables that are associated with these rules are refreshed to display the current protection status.

   If you click **Cancel** instead, your changes are discarded.

Adding Cell Protection at the Form Level

In a data-entry table, a protected cell (that would otherwise be writable) is displayed in purple and is read-only.

If cells have been protected by an administrator, you cannot undo that protection in an individual form. However, you can add further protection to a form, as follows:

- To protect one or more cells, select the cells and select **Protect Cell**.

  Only writable cells (displayed in yellow) can be protected.

  Note: If you protect a cell that includes a roll-up member, you protect both the roll-up and its .VC member. However, the protection does not include the descendants of that roll-up member. The cells of its leaf descendants are still writable (assuming they are not otherwise protected).

- To undo protection that was set in this form for one or more cells, select the cells and select **Unprotect Cell**.
- To undo all cell protection that was set in this form, click anywhere in the table and select **Unprotect All Cells**.

The changes that you make are saved automatically and apply to any users who open the same form.

Cell Protection and Supplemental Schedules

You cannot apply cell protection directly to the cells in a supplemental schedule. However, the measures in a supplemental schedule can inherit cell protection that is applied to the associated data-entry table, form set, or model:

- You protect an account that is used as a measure in a supplemental schedule.

That measure is also protected in the supplemental schedule. If you make the protection rule more granular (for example, to include specific time periods), then the protection in the supplemental schedule is also more restricted.

- You protect a member of the dimension that is used for the detail rows in a supplemental schedule.

  At run time, if the user selects that member in the FM table, the corresponding cells are protected in the supplemental schedule.

Allow Data to be Entered for Parent Members (Financial Forms Only)

By default, when users edit a data-entry table, they enter values in cells that contain leaf members. Those values are rolled up to the parent member in the hierarchy, subject to certain restrictions.

In financial forms, depending on the table settings, form users can directly enter values into the cell for a parent member (a roll-up point). The change in the parent's value is added to the .VC member for the parent.

This option applies only to financial forms in a bottom-up form set. It is disabled by default. To enable form users to enter values at roll-up points:

1. Create a template for a data-entry form.

   Notice that the cells for rolled-up values are gray.

2. Open the table properties and click the **Data-Entry** tab.

3. Select **Allow data to be entered for parent members** and save your changes.

Notice that the color for the roll-up cells has changed to yellow, signifying that the cells are writable. If a dimension is on the table: If a user enters a value into a parent cell, the difference in value is added to the .VC member. If a dimension is off the table, the operation is applied to the default write member (or its .VC member).

Note: For the time dimension, which does not have .VC members, data entry must occur at a leaf member.

Enabling this option can have the following side effects:

- The **Spread** and **Adjust Values** options are disabled when the range includes roll-up cells.
- Pasting into an area that is a mixture of roll-ups and leaf members can have results that the user might not expect. The results depend on the order of the writeback operation. Either the roll-ups or the leaf members might be written first, and the results might be different in each case.

Allocating Using Predefined Weights (Financial Forms Only)

Overview

In a financial form, depending on the table settings, users can enter values directly into a parent (roll-up) cell. This can be a quick way to enter data for planning and budgeting, using relative weights. If **Allocate using predefined weights** is selected, the value that is entered in the parent cell is automatically distributed among eligible leaf members. The allocation takes place according to rules that are explained later in this section.

Automatic allocation applies only to financial forms that are part of a bottom-up workflow. This option is disabled by default and can be enabled in the table properties of the form set template.

See Also

- "Allow Data to be Entered for Parent Members (Financial Forms Only)" on page 144
- "How the Allocation Is Applied" on page 146

Enable Automatic Allocation

To enable automatic allocation: In the table properties, click the **Data Entry** tab and select **Allocate using predefined weights**. Save the table properties.

Note: If automatic allocation is enabled for a data-entry table, the **Spread** and **Adjust Values** options are disabled when the range includes roll-up cells. If automatic allocation is enabled for any of the data-entry tables on a form, the form cannot be edited offline.

Eligible Cells

For automatic allocation to take place, the parent cell must contain a roll-up point in at least one dimension other than Time. (If the parent cell includes a roll-up point in Time, the cell is not writable. You cannot perform an automatic allocation across the Time dimension.)

To receive an automatic allocation from the parent cell, a leaf descendant must meet the following requirements:

- It must be included in the table on the rows, columns, or slicers.
  - If a member is part of a slicer, you must be able to switch to the member using the **Select Member** option.
  - If a member is part of a row or column, you must be able to navigate to the member by expanding the hierarchy or drilling down in the hierarchy.

    Note: System filters, member filter combinations, and member property filtering are used for display purposes only and do not affect a leaf descendant's eligibility for automatic allocation.
  - If a dimension is off the table, the default write member is used as the only eligible leaf descendant for that dimension.

- If the member is hidden using the Excel add-in's **Hide** option, it is not eligible to receive an automatic allocation.
- The member must be writable. For example, it cannot be protected, it cannot be a target of a server-side formula, and the user must have security access to that member. Other examples of non-writable members include account types such as Retained Earnings and Cumulative Translation Adjustment, as well as Analysis members that are not designated as writable in the form set.

There is a special case of eligible cells, when member selection rules for a hierarchy include a roll-up point but not its descendants. In these cases, the roll-up point is the last member of its branch to be displayed in the table. The roll-up point is treated as if it were a leaf member and the allocation is made to the .VC member. The weight for the roll-up point is used (not the weight for the .VC member). The cell must otherwise be eligible for an allocation: for example, it cannot be hidden or protected, and it cannot include certain account types (see the list above).

See Also

- "Allocations to Members of the Target Hierarchy" on page 148
- Chapter 16, "Security in SAS Financial Management," on page 199

How the Allocation Is Applied

When a form user writes directly to a roll-up with automatic allocation enabled, the value is distributed among all eligible cells, according to the relative weights for those cells. If each cell had the same weight, then the value entered at the roll-up point would be equally distributed among the eligible cells.

A weight table defines the weights for each dimension in the data-entry table. The default weight is **Same as Target**, meaning that the relative weight for a target dimension member is its current value. For more information about the weight table, see "Editing Allocation Weights" on page 149.

Here is a simple example of automatic allocation using the current cell values as weights. In this example, the allocation takes place across two dimensions, and each member in the roll-up hierarchy is additive—that is, it rolls up to its parent. (This property is set in the Dimensions workspace in SAS Financial Management Studio.) The form looks like this:

CW_TIME	...	Jun 2010				
CW_Analysis	...	Budget				
CW_IntOrg	...	Central				
		Video Games	Action	Simulation	Arcade	Puzzle
Administrative Expense		12.00	4.00	4.00	2.00	2.00
Office Supplies		6.00	2.00	2.00	1.00	1.00
Postage		3.00	1.00	2.00	0.00	0.00
Other Administrative Expense		3.00	1.00	0.00	1.00	1.00

The `Puzzle` member of the Product dimension is displayed in purple, because it is protected.The remaining cells, including parent cells, are in yellow, indicating that they are writable.

If you double the value in the (Video Games, Administrative Expense) cell (to 24.00), results are as follows:

- The values for eligible cells increase in proportion to their weights. (Recall that the weights were **Same as Target**.)
- The values for Puzzle remain the same, because they were protected. Those values were subtracted from the totals before the allocations occurred.
- For any dimension that is not on the table (that is, not on the columns, rows, or slicers), no allocation is performed. All data is written to the default write member of that hierarchy.

CW_TIME ...	Jun 2010				
CW_Analysis ...	Budget				
CW_IntOrg ...	Central				
	Video Games	Action	Simulation	Arcade	Puzzle
Administrative Expense	24.00	8.80	8.80	4.40	2.00
Office Supplies	12.00	4.40	4.40	2.20	1.00
Postage	6.60	2.20	4.40	0.00	0.00
Other Administrative Expense	5.40	2.20	0.00	2.20	1.00

The automatic allocation is to all eligible leaf members that roll up (directly or indirectly) to the cell where the value is entered. Allocations are always made directly to all eligible leaf members. They do not take place first across one dimension first and then across another dimension.

If the hierarchy contains members that do not roll up to the parent, those cells might still receive an allocation. See "Allocations to Non-Additive Members" on page 147.

Allocations to Non-Additive Members

If the roll-up hierarchy contains any members that are non-additive, they can still change as a result of the allocation. If there are no additive cells in the roll-up hierarchy, then no allocation occurs at all. Otherwise, non-additive cells are updated in proportion to the changes in the additive cells.

In this example, the Flight Stick member does not roll up to its parent in the Product dimension.

Account	Units		
Analysis	BUDGET		
	Customer Total	Buy Best	SW Mart
Total Product	12.00	3.00	9.00
Game Controller	5.00	1.00	4.00
Joy Stick	7.00	2.00	5.00
Flight Stick (nonadditive)	9.00	3.00	6.00

If you enter a value at the roll-up point of (Total Product, Customer Total), the allocation occurs across all eligible members, including non-additive members. In this example, the user enters a value of 24 in the (Customer Total, Total Product) cell.

Account	Units		
Analysis	BUDGET		
	Customer Total	Buy Best	SW Mart
Total Product	24.00	6.00	18.00
Game Controller	10.00	2.00	8.00
Joy Stick	14.00	4.00	10.00
Flight Stick (nonadditive)	18.00	6.00	12.00

Notice that the values for Flight Stick, the non-additive member, also change, in proportion to the changes in the other cells. However, the Flight Stick values do not affect the parent cells.

Allocations to Members of the Target Hierarchy

Typically, allocations are not performed down the target hierarchy, because the children's values are not writable in the parent's form. If the .VC member for the parent is included on the table, the values are written to that .VC member (using the weight from the .VC member).

If the .VC member for the parent is not included on the table, there are no allocatable descendants and no allocation takes place.

Note: If a form set uses the descendants rule to include writable members, the allocation does occur down the target hierarchy for those writable members in a form. In a form template, which can display all members of the target hierarchy as writable, the allocation occurs down the target hierarchy.

Modeling and Driver Formulas and Automatic Allocation

A crossing that is the target of a modeling formula or a driver formula is not writable and therefore cannot receive allocation values. However, an automatic allocation might be made to a cell that is the trigger for a formula. In that case, the formula executes in the same manner as if the user had entered data directly into the cell. The side effects can be unexpected unless you are aware of this behavior.

In this example, the value of Other Selling Expenses is computed by this formula: `["ACCOUNT"="Travel"]*2`. The cells for Other Selling Expenses are gray and cannot receive an allocation.

CW_Analysis	...	Budget				
CW_TIME	...	May 2010				
CW_IntOrg	...	Eastern				
		Video Games	Action	Simulation	Arcade	Puzzle
Staff Expenses		50.00	21.00	7.00	12.00	10.00
Salaries		21.00	10.00	5.00	4.00	2.00
Benefits		11.00	5.00	2.00	2.00	2.00
Travel		6.00	2.00	0.00	2.00	2.00
Other Selling Expenses		12.00	4.00	0.00	4.00	4.00

If the user doubles the value at (Video Games, Staff Expenses) to 100, the following things happen:

1. The current value for Other Selling Expenses is subtracted from the amount to be allocated, because those cells do not participate in the allocation.

2. The remainder of the automatic allocation is applied to the writable (yellow) cells, including the Travel cells.
3. Writing to Travel causes the formula to be executed for Other Selling Expenses, which now has a larger value.
4. The values from Other Selling Expenses roll up to the parent. As a result, the value at (Video Games, Staff Expenses) ends up being greater than the value that the user originally entered.

CW_Analysis	...	Budget
CW_TIME	...	May 2010
CW_IntOrg	...	Eastern

	Video Games	Action	Simulation	Arcade	Puzzle
Staff Expenses	115.79	48.63	16.21	27.79	23.16
Salaries	48.63	23.16	11.58	9.26	4.63
Benefits	25.47	11.58	4.63	4.63	4.63
Travel	13.89	4.63	0.00	4.63	4.63
Other Selling Expenses	27.79	9.26	0.00	9.26	9.26

Note: You might notice that the value for (Video Games, Other Selling Expenses) does not appear to be exactly double the value for (Video Games, Travel). This reflects rounding after the automatic allocation.

Reallocation

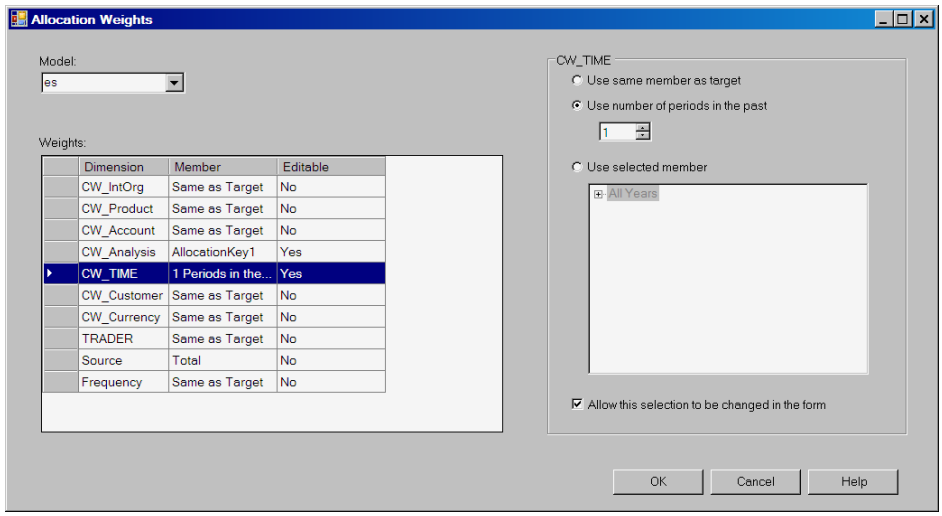
For a form template or form that is being edited in Microsoft Excel, a **Reallocate** option is available. Here is one scenario:

1. With **Allocate using predefined weights** enabled, a user enters a value in a parent cell. The value is allocated to eligible subordinate cells, dependent on the allocation weights table.
2. The user right-clicks anywhere in the table and selects **Edit** ⇨ **Allocation Weights**.
3. Because one of the allocation weights is editable, the user selects a different member for that weight and saves the changes.
4. The values in the form do not change until the user right-clicks the parent cell and selects **Edit** ⇨ **Reallocate** to perform the allocation again using the updated weights.

Editing Allocation Weights

To edit the allocation weights for a data-entry table, follow these steps:

1. In the form template, click anywhere in the table and select the **Edit** ⇨ **Allocation Weights** option.



2. For each dimension in the data-entry table, select a member for the weight table. When it comes time to perform the allocation, the crossing at each data-entry cell has a corresponding crossing (based on your selections here) that represents the weight for that cell.

   a. From the **Model** drop-down list, select the current model or a compatible model.

      Note: For the purposes of automatic allocation, a compatible model is one that uses the same hierarchies as the model used by the table. The as-of date might be different, but the hierarchies must be the same.

   b. For each dimension, select a member to provide the weight. Some possible selections are as follows:

      - If you select **Same as Target** for each dimension, the value at the roll-up point is distributed to eligible cells in proportion to their current values. The current value of each cell acts as its weight.
      - As another possibility, you might want to use the values of a previous period (such as last month or last year) as allocation weights. You could select **Number of periods in the past: 1** for the time dimension, and select **Same as Target** for the remaining dimensions. The allocation would be made in proportion to the relative values of the same crossings during the previous period.
      - If you were introducing a new product and wanted to base your projections on existing data from an established product, you might select the established product from the Product dimension and **Number of periods in the past** from the time dimension.
      - Rather than using existing values as weights, you could define a new member of one of the dimensions and assign weights to that member. The typical dimension for this purpose is the Analysis dimension. See "Creating a Separate Weight Table" on page 151.

The weights for the target hierarchy are always set to **Same as Target** and cannot be changed. In most cases, a form user has access only to the target member, and changing the weight to reference a different member would prevent the allocation from occurring.

The selection allocated weight member for a dimension can be a leaf member or a roll-up member. For example, although the roll-up point must contain a leaf member of the Time dimension (such as January 2010), the weight for the Time dimension can be associated with a non-leaf member (such as Q1 2010).

Weights apply to the entire table. By default, weights are defined in the form template and cannot be changed by form users. On a dimension-by-dimension basis, you can permit users to select a different weight when they edit a form. (The ability to change weights is available only in Microsoft Excel.)

Creating a Separate Weight Table

In addition to using existing values, you can create a new dimension member for use as allocation weights. Follow these steps:

1. In the Dimensions workspace, create a dimension member to use as the allocation key. The usual dimension for this purpose is the Analysis dimension. However, it can be any dimension.
2. Create a form set and a data-entry table that include this member, so that you can load its values.

   Use a model that is the same as or compatible with the data-entry table that will use the weights.
3. Open the weight table in Microsoft Excel and assign the appropriate values.

   Typically, leaf members are used for weights. However, a roll-up point can also be used as a weight. In that case, you would write the weight value to the .VC member.

If you want users to be able to view the weight table, include it in the template for the form that will use these weights, as a read-only table or as a data-entry table. Alternatively, you can create a separate form set and template for the weights and make your form users aware of it. If the Analysis member that is used for the weights is writable, users can also modify the weights.

If you do not want users to view the weight table, you can create a separate form set and template for the weights and enter the weights in the template. In that case, you would not need to publish the form set that contains the weight table.

Note: Instead of creating a form set and data-entry table, you can use a SAS Data Integration Studio job to load the weights.

Supplemental Schedules

Overview

A supplemental schedule is an additional table that can be added to a financial form template. Supplemental schedules allow users to reference detailed information outside the financial model.

Supplemental detail is typically confidential and specific to individual cost centers (or other dimension members). For a budgeting exercise, it might include details such as salary, bonus, and travel expenses for each employee.

The overall procedure is as follows:

1. A custom data provider is implemented.
2. A process administrator creates a form template and inserts a data-entry table and one or more supplemental schedules. These supplemental schedules can draw on an external data source, as defined by the custom data provider.

3. During data entry, a user interacts with both the data-entry table and the supplemental schedules. The user can add new line items to a supplemental schedule without requiring the Process Administrator to modify the associated hierarchy.

   When the user saves the supplemental data:

   - Data from the supplemental schedule is saved in a separate table in the external database.
   - In a bottom-up workflow, SAS Financial Management totals are rolled up to the data-entry table and saved in the SAS Financial Management database.

Note: Supplemental schedules can be used only in Microsoft Excel and apply only to financial planning.

Implementing a Custom Data Provider

Supplemental schedules require a data provider. The data provider identifies the data source for supplemental measures and how those measures can be used in a supplemental schedule.

One or more custom data providers can be implemented at a site. Contact your SAS consultant.

A site with SAS for Workforce Planning & Budgeting (which includes both SAS Financial Management and SAS Human Capital Management) has the built-in SAS Human Capital Management data provider for supplemental schedules.

Designing a Form Template

The Data-Entry Table

Design the data-entry table carefully. It should contain only the information that is needed. When you insert a supplemental schedule, you select its fields from the set of leaf members in the data-entry table and the set of supplemental measures. This selection is easier if you limit the members in the data-entry table.

Do not copy or import a form set that contains a supplemental schedule. You must create a new form set instead.

Note: The SAS Human Capital Management data provider puts some restrictions on the data-entry table. For more information, see the *SAS Human Capital Management: Administrator's Guide*.

The Supplemental Schedule

When you insert a supplemental schedule, you make these selections:

- a data provider. If you see no data providers in the list, click **Add** and browse to the configuration file that contains information about custom data providers. The consultant who implements the custom data providers should tell you this location.
- the table that the supplemental schedule will be associated with, if the form template contains more than one data-entry table.
- the dimension to be used for the detail rows.
- the dimension that you are measuring, to be used for the columns.

  Note: If you do not select the Account dimension for the rows or the columns, you are asked to select accounts that require supplemental information.

- SAS Financial Management measures and supplemental measures for the supplemental schedule that you are creating.

For detailed instructions, see the online Help for the supplemental schedule wizard.

After you insert the supplemental schedule, you can modify it. For example, you can add or remove measures, and you can add or remove accounts that need supplemental information. You can also add text columns and calculated members. Keep the following points in mind:

- In a bottom-up workflow, when the user saves the supplemental data, the totals for SAS Financial Management member columns in the supplemental schedule are rolled up to the data-entry table. To include data from supplemental measures in the roll-ups, you need to make one or more SAS Financial Management member columns (in the supplemental schedule) dependent on the supplemental measures. You do so via the **Calculated Members** option.
- The following dimensions are not permitted as slicers: Currency, Trader, Source, and Frequency.
- A supplemental schedule honors both member protection and member-level security.
- Formatting options that are available for the data-entry table can also be used for supplemental schedules (**Format Members** for member formatting; **Format Cells** for cell formatting). However, you cannot use the SAS Financial Management Add-In API for Microsoft Excel for a supplemental schedule.

See Also

- Creating Tables on page 236
- Using Calculated Members on page 243
- Supplemental Schedule option on page 268

What Happens at Run Time

During data entry, the cells for supplemental measures are populated by the custom data source, and the cells for SAS Financial Management measures are populated with SAS Financial Management data. As a user makes a selection in the data-entry table, the supplemental schedule is updated to match the member that was selected. For example, if the detail dimension in the supplemental schedule is Account, then selecting a particular account member in the data-entry table causes the supplemental table to display detail rows that are associated with that member.

If a cell in the supplemental schedule is writeable, the user can update that cell. The user can also add rows to the supplemental schedule. For example, in a budgeting exercise, the user might want to create a set of additional positions to represent potential new hires. These positions exist only in the supplemental schedule data; they are not added to the associated hierarchy.

Note: If a numeric field has an error (for example, resulting from a failed calculation or a failed query from an external table), a zero value is displayed. If the data returned from the server has a null or empty value for a date field, Excel renders it as `1/0/1900`.

When the user saves the supplemental data, the following actions occur:

- In a bottom-up workflow, totals from SAS Financial Management measures in the supplemental schedule are rolled up to the corresponding measures in the data-entry table.

- Values from the data-entry table are saved to the SAS Financial Management database.
- Changes in the supplemental measures are saved to a separate table in the external data source.

  The source tables for the supplemental schedule are not affected. These changes are written to an additional table that records supplemental-schedule data.

See Also

- Entering Data into a Supplemental Schedule on page 224
- Save All Supplemental Data on page 266

Chapter 14

Built-In Computations

Survey of Computations **155**

Generating Adjustments from Adjustment Rules **156**

Generating Intercompany Eliminations **156**

Frequency Conversion **157**
Frequency Conversion Overview 157
Frequency Conversion and the Flow Account Types 158
Frequency Conversion and the Balance Account Types 158
Frequency Conversion and the Statistical Account Type 159

Currency Conversion **159**

Combining Frequency Conversion and Currency Conversion **160**

Computing Retained Earnings **161**
Overview of Computing Retained Earnings 161
A Simple Example 161
Second Example 162
Third Example 163
Fourth Example 163
Summary of Design Principles 163

Computing Cumulative Translation Adjustments **163**

Hierarchical Roll-Ups **164**
Overview of Hierarchical Roll-Ups 164
Relation to Other Computations 164
Exceptions to the General Pattern 165

Survey of Computations

SAS Financial Management stores data in compartments called cycles and displays numeric values in Microsoft Excel, using the SAS Financial Management Add-In for Microsoft Excel, or in the Web-based Form Editor. The displayed values can include stored values, but many displayed values are computed from values that are permanently stored, and are not stored themselves. The computations can include operations of the following kinds:

- Generating adjustments from adjustment rules on page 156
- Generating intercompany eliminations on page 156

- Frequency conversion on page 157
- Currency conversion on page 159
- Retained Earnings computations on page 161
- Cumulative Translation Adjustment (CTA) computations on page 163
- Hierarchical roll-ups on page 164
- Evaluation of formulas on page 167

Generating Adjustments from Adjustment Rules

When an adjustment rule runs, it generates adjustments that depend on the data that exists when the rule starts to run. Adjustment rules of different types look for different things in the preexisting data and generate different types of adjustments when they find what they are looking for.

The adjustment rules in a given model run in a defined order. The data that exists when a given adjustment rule starts to run includes the data that is stored in the cycle, the before-rule manual adjustments that are part of the model, and the adjustments that have been generated by the adjustment rules that have already run.

To provide inputs to adjustment rules, it is sometimes necessary to subject preexisting data values to currency conversion and/or frequency conversion. For example, the definition of an allocation rule includes a designated allocation currency. All inputs to the rule that are not stored in the allocation currency must be converted to the allocation currency.

The inputs to adjustment rules can include the values at crossings that have one or more non-leaf members. To provide these inputs, partial hierarchical roll-ups are performed. The final and comprehensive hierarchical roll-up incorporates all before-rule manual adjustments, all adjustments that are generated by adjustment rules, and all after-rule manual adjustments, in addition to the unadjusted data that is permanently stored in the cycle.

See Also

- What Happens When You Post Adjustments on page 92
- Working with Manual Adjustments on page 95
- Working with Adjustment Rules on page 98
- Currency Conversion on page 159
- Frequency Conversion on page 157
- Hierarchical Roll-Ups on page 164

Generating Intercompany Eliminations

Certain accounts are marked as intercompany accounts. In stored cycle data, any crossing that contains an intercompany account should contain two different organizations, one for the Organization dimension type and one for the Trader dimension type. The numeric value at such a crossing represents a specific financial relationship between the two organizations,

such as sales from one organization to the other or funds that one organization owes to the other.

Numeric values of this sort are not meaningful for any organization that both of the participating organizations are hierarchically subordinate to. For example, a sale that a German subsidiary makes to a French subsidiary is not a sale that is made by the European division that both subsidiaries are part of. For the European division, it is an internal transfer.

To prevent such values from rolling up to organizations where they have no meaning, offsetting elimination values are generated. Each offsetting elimination value is associated with a crossing whose Source member is Elim and whose Organization member is the lowest-level organization at which the value that needs to be eliminated loses its meaning. In the final hierarchical roll-up process, the value that needs to be eliminated and the offsetting elimination value sum to zero.

The process of generating intercompany elimination values is very similar to the process of generating adjustments from an adjustment rule. The main difference is that the process of generating any necessary intercompany elimination values always occurs, so there is no need to define an adjustment rule for this purpose.

Values that need to be eliminated typically occur in pairs. For example, if there is a crossing for a sale that a German subsidiary makes to a French subsidiary, then there should be a complementary crossing for a purchase that the French subsidiary makes from the German subsidiary. Because the two values are eliminated together, it is important that they be equal before they are eliminated. Otherwise, the elimination introduces an imbalance. Ensuring the equality of all such pairs of values that are eliminated together is the role of the intercompany balancing rules.

See Also

- Intercompany Property of Accounts on page 49
- Introduction to the Source Dimension Type on page 52
- Hierarchical Roll-Ups on page 164
- Generating Adjustments from Adjustment Rules on page 156
- Intercompany Balancing Rules on page 103
- Net Intercompany Balancing Rules on page 104

Frequency Conversion

Frequency Conversion Overview

All stored numeric values are implicitly associated with the Period Activity member of the Frequency dimension. In order to compute values for certain other members of the Frequency dimension, the software sums the stored Period Activity values over a series of time periods. The need for such a summation varies by account type. There are three cases:

- flow account types
- balance account types
- the Statistical account type

Frequency conversion can occur either before or after currency conversion. The order of frequency conversion and currency conversion depends on the account type, the exchange rate type, and the frequency to be displayed.

See Also

- Introduction to the Frequency Dimension Type on page 51
- Currency Conversion on page 159
- Combining Frequency Conversion and Currency Conversion on page 160

Frequency Conversion and the Flow Account Types

The flow account types are Revenue, Expense, and Statistical Flow. For an account that belongs to one of these account types, each stored Period Activity value is the flow volume for the associated time period. For example, the valued stored for (Salary Expense, August 2009) is the amount that is paid for salaries during August 2009. To compute the flow volume over a longer span of time that ends with the given time period, a summation over a series of time periods is required.

If the stored Period Activity values are associated with months, then the different values that can be displayed for a given month are computed in the following way:

- For Period Activity, Period To Date, Month Activity, and Month To Date, the stored value is displayed. No frequency conversion is required. For example, the stored salary expense for August 2009 is displayed for all these Frequency members.
- For Quarter Activity and Quarter To Date, a summation over some or all of the months in the quarter is required. For example, salary expense for the third quarter of 2009 through August 2009 is computed as the sum of salary expense for July 2009 and salary expense for August 2009.
- For Year Activity and Year To Date, a summation over some or all of the months in the year is required. For example, salary expense for 2009 through August 2009 is computed as the sum of salary expense for January 2009, salary expense for February 2009, and so on, through salary expense for August 2009.

Note that for flow accounts the value that is displayed for a given Frequency member never depends on the Activity/To Date distinction; all that matters is the length of time that the Frequency member represents.

Frequency Conversion and the Balance Account Types

The balance account types are Asset, Liability, Equity, and Statistical Balance. For an account that belongs to one of these account types, the stored Period Activity value for a given time period is the difference between the end-of-period balance for that period and the end-of-period balance for the previous period. The end-of-period balance for a given time period is computed by summing a series of stored Period Activity values that begins with the first time period in the governing model's time hierarchy. The end-of-period balance is the value that users are most often interested in.

The end-of-period balance is displayed in association with any To Date member of the Frequency dimension. For example, if the stored Period Activity values are associated with months, then the end-of-period balance for a given month is displayed for all of the following Frequency members:

- Period To Date

- Month To Date
- Quarter To Date
- Year To Date
- Lifetime To Date

Frequency Conversion and the Statistical Account Type

Accounts that belong to the Statistical account type have the special property of frequency invariance. Frequency conversion never occurs for the values of such accounts. Two Statistical-account crossings that differ only in their Frequency members always display the same numeric value. This property of Statistical accounts suits them for storing values such as prices, for which summation over a series of time periods is not a meaningful operation.

See Also

Statistical Account Type on page 47

Currency Conversion

If a value is stored in one currency and a user requests to see that value in another currency, then a currency conversion step is required. In order to perform a currency conversion, the software retrieves the stored value and the relevant exchange rate and computes their product.

In some cases, a frequency conversion is performed by summing a series of stored values and then the output of the frequency conversion undergoes currency conversion. In other cases, individual stored values undergo currency conversion and then the outputs of the currency conversion are summed to effect a frequency conversion. The order of frequency conversion and currency conversion depends on the account type, the exchange rate type, and the frequency to be displayed.

The relevant exchange rate for a given currency conversion is determined by the following factors:

- the currency of the input value
- the currency of the output value
- the time period that the two values are associated with
- the analysis member that the two values are associated with
- the association in the governing model between the analysis member and an exchange rate set
- the exchange rate type for the account that the values belong to

For simple exchange rate types, these factors uniquely determine the correct exchange rate: it is the exchange rate of the relevant exchange rate type in the relevant exchange rate set that connects the relevant currencies in the relevant time period. For complex exchange rate types, the correct exchange rate also depends on one or more other members of the crossing whose value must be converted to another currency. For example, complex exchange rates can vary by organization. In this case, the exchange rate that is retrieved depends on the organization that the stored value belongs to, in addition to all the previously mentioned factors.

Currency conversion occurs before the final hierarchical roll-up process. Each rolled-up value that you see in a report is the result of roll-ups that occur in the currency of the displayed value.

The relation of currency conversion to formula evaluation depends on formula type as follows:

- All input values for driver formulas and modeling formulas are in the currency of the stored data. The output values of driver formulas and modeling formulas are converted to other currencies as necessary before the hierarchical roll-up process begins.
- Reporting formulas are evaluated after the hierarchical roll-up process, and hence after all currency conversion is complete. They can have input values in any currency.

See Also

- Frequency Conversion on page 157
- Combining Frequency Conversion and Currency Conversion on page 160
- Survey of Exchange Rate Types on page 68
- Hierarchical Roll-Ups on page 164
- Formula Types on page 169

Combining Frequency Conversion and Currency Conversion

All stored values are explicitly associated with a particular currency and implicitly associated with the Period Activity member of the Frequency dimension. If a value for a different currency and frequency needs to be computed, then the computation must combine currency conversion and frequency conversion.

The way in which the software combines currency conversion and frequency conversion depends on the account type of the relevant account and the exchange rate type that is associated with the relevant account. Following are the two methods and the circumstances in which each method is used:

- **Use exchange rates for all the participating time periods.**Retrieve an exchange rate for each time period that is needed to perform the conversion to the target frequency. Convert each stored value to the target currency using the period-specific exchange rate. Finally, sum the resulting values to yield the value for the target time period in the target currency and frequency. Because many period-specific exchange rates are used, the currency conversion must occur before the frequency-converting summation.

  This method is used in the following circumstances:

  - The stored values are for a flow account (Revenue or Expense account type). For example, to compute Year To Date sales in a target currency, the stored Period Activity sales figures for each month of the year are first converted to the target currency, and then the monthly figures in the target currency are summed to yield the requested Year To Date value. In general, the exchange rate type associated with a flow account is Period Average, but the computation is done in the same way regardless of the exchange rate type.
  - The stored values are for an account that has a complex exchange rate type (Derived or Historic) associated with it. The account type of the account makes no difference.

Any account that uses a complex exchange rate is treated in the same manner as a flow account.

- **Use a single exchange rate for the target time period.**Retrieve a single exchange rate that is associated with the target time period. Sum the stored values for the relevant series of time periods that ends with the target time period, and then apply the exchange rate to the resulting sum.

  This method is used if the stored values are for a balance account (Asset, Equity, or Liability account type) that has a simple exchange rate type (Period Average, Period Open, Period Close) associated with it. For example, to compute any "To Date" value of inventory in a target currency, the stored Period Activity inventory figures for each month in the model are first summed to yield an inventory balance in the storage currency, and then that balance is converted to the target currency. In general, the exchange rate type associated with a balance account is Period Close, but the computation is done in the same way for any simple exchange rate type.

See Also

- Introduction to the Frequency Dimension Type on page 51
- Currency Conversion on page 159
- Frequency Conversion on page 157
- Survey of Exchange Rate Types on page 68

Computing Retained Earnings

Overview of Computing Retained Earnings

SAS Financial Management provides a special mechanism for updating a balance sheet retained earnings account on a periodic basis. The core of this mechanism is the Retained Earnings account type, whose purpose is to compute a change in retained earnings by combining the values of relevant source accounts such as Net Income and Dividends. The rest of the mechanism is an account hierarchy that is structured in such a way that the periodic update of the balance sheet retained earnings account is achieved by means of the hierarchical roll-up process.

See Also

- Retained Earnings Account Type on page 46
- Hierarchical Roll-Ups on page 164

A Simple Example

The account hierarchy can be structured in various ways in accordance with your needs. The simplest possible account structure for updating a balance sheet retained earnings account involves a parent and two children:

```
Retained Earnings (account type Equity)
   Retained Earnings - Initial Balance (account type Equity)
   Change in Retained Earnings (account type Retained Earnings)
```

The significance of these three accounts is as follows:

Retained Earnings
: is the current, updated retained earnings account that is the goal of the computation. Its value is the sum of the values of the two subordinate accounts, as determined by the hierarchical roll-up process.

Retained Earnings - Initial Balance
: holds a previous retained earnings balance that you import from your General Ledger or other source system.

  For each leaf organization of interest, import the correct retained earnings balance as of the end of the first time period of the governing model's time hierarchy. Do not import any subsequent retained earnings balances. The balance that you import carries forward to all subsequent time periods because the Period Activity of the Retained Earnings - Initial Balance account for all subsequent time periods is zero.

Change in Retained Earnings
: is the total change in retained earnings from the time of the initial balance to the end of the current period.

  To compute this value correctly, you must define the account that belongs to the Retained Earnings account type to be all-encompassing. It must encompass all accounts that contribute to retained earnings, all relevant members of the Source hierarchy, and all time periods from the first time period in the model to the current time period. To encompass all time periods, select the “From beginning of time through end of current period” roll-forward method.

Second Example

Using more complicated account structures, you can compute and display various components of the complete change in retained earnings. For example, one account can hold the current year's change in retained earnings and another account can hold the prior years' change:

```
Retained Earnings (account type Equity)
   Retained Earnings - Initial Balance (account type Equity)
   Change in Retained Earnings - Current Year (account type Retained Earnings)
   Change in Retained Earnings - Prior Years (account type Retained Earnings)
```

In this example, the significance of the last two accounts is as follows:

Change in Retained Earnings - Current Year
: is the change in retained earnings from the beginning of the current year to the end of the current period.

  To compute this value correctly, you must select the “From beginning of year through end of current period” roll-forward method.

Change in Retained Earnings - Prior Years
: is the change in retained earnings from the time of the initial balance to the beginning of the current year.

  To compute this value correctly, you must select the “From beginning of time to beginning of year” roll-forward method.

Third Example

Components of the change in retained earnings can be distinguished by source account in addition to time span, as in the following example:

```
Retained Earnings (account type Equity)
   Retained Earnings - Initial Balance (account type Equity)
   Net Income - Current Year (account type Retained Earnings)
   Dividends - Current Year (account type Retained Earnings)
   Net Income - Prior Years (account type Retained Earnings)
   Dividends - Prior Years (account type Retained Earnings)
```

Fourth Example

As the number of displayed components of the change in retained earnings grows, it is natural to introduce accounts to hold subtotals. For example, you could reorganize the accounts in the previous example as follows:

```
Retained Earnings (account type Equity)
   Net Income - Current Year (account type Retained Earnings)
   Dividends - Current Year (account type Retained Earnings)
   Retained Earnings at Beginning of Year (account type Equity)
      Retained Earnings - Initial Balance (account type Equity)
      Net Income - Prior Years (account type Retained Earnings)
      Dividends - Prior Years (account type Retained Earnings)
```

Summary of Design Principles

In all these examples, the basic principles are the same:

- For each leaf organization of interest, import a single retained earnings balance to use as the starting point of all retained earnings computations in SAS Financial Management.
- For each component of the complete change in retained earnings that you want to display separately, define an account that belongs to the Retained Earnings account type.
- Construct an account hierarchy whose roll-up process adds all the components of retained earnings change to the initial retained earnings balance, yielding the current retained earnings balance. Include additional subtotal accounts as appropriate.

Computing Cumulative Translation Adjustments

A cumulative translation adjustment (CTA) is often needed when an organization's balance sheet is presented in a currency other than the organization's functional currency. This need arises because different balance sheet accounts are converted from the functional currency to the target currency using different exchange rates. If the balance sheet is in balance in the functional currency, then it is no longer in balance in the target currency, unless a cumulative translation adjustment is included.

To compute a cumulative translation adjustment for a balance sheet, use an account that belongs to the Cumulative Translation Adjustment (CTA) account type.

Each output crossing of a CTA computation contains the following members:

- the particular CTA account that governs the computation
- the CTA member of the Source hierarchy
- the organization whose balance sheet is being balanced

  Cumulative translation adjustments are computed only for leaf organizations. Balance sheets of higher-level organizations are compiled through the hierarchical roll-up process, possibly including an ownership rule.
- the target currency

  An organization's functional currency is never a target currency for that organization. There is no cumulative translation adjustment for a crossing that contains an organization and its own functional currency.
- the time period of the balance sheet that is being balanced

See Also

- Cumulative Translation Adjustment Account Type on page 44
- Introduction to the Source Dimension Type on page 52
- Hierarchical Roll-Ups on page 164

Hierarchical Roll-Ups

Overview of Hierarchical Roll-Ups

Hierarchical roll-ups are the summations that are implicit in the structure of a model's hierarchies. In general, the numeric value of a crossing X that contains one or more non-leaf members is computed by summing the numeric values of a set of crossings that are related to crossing X in the following way:

- In every dimension type except one, they have the same members as crossing X.
- In that one remaining dimension type, each of them has a member that is a hierarchical child of the member in crossing X.

Relation to Other Computations

A final and comprehensive hierarchical roll-up occurs before the evaluation of reporting formulas and after most other computations, including the following:

- generation of adjustments from adjustment rules
- generation of intercompany eliminations
- evaluation of driver formulas
- evaluation of modeling formulas
- frequency conversions
- currency conversions
- computations of retained earnings

- computations of cumulative translation adjustments

For any crossing that receives a value as a result of the hierarchical roll-up process and that is also in the scope of a reporting formula, the computed value of the reporting formula overwrites the roll-up value. In general, however, you should define the scopes of reporting formulas so that the formula evaluation does not overwrite any roll-up values. For example, if there is no base data associated with the Variance member of an Analysis dimension, then a reporting formula that is associated with the Variance member does not overwrite any roll-up values.

In addition to the final and comprehensive hierarchical roll-up process, partial hierarchical roll-ups can occur in order to provide inputs for computations that occur before the final and comprehensive hierarchical roll-up process. Here are two examples:

- An allocation rule can allocate a roll-up value. In this case, the input to the allocation rule must be computed by a hierarchical roll-up process before the allocation rule can run.
- An account that belongs to the Retained Earnings account type can have a Net Income account as a source account. In this case, the value of Net Income must be computed by a hierarchical roll-up process before the retained earnings computation can take place.

Exceptions to the General Pattern

In the final and comprehensive hierarchical roll-up process, the summation proceeds up the framework of hierarchies level by level. When the process is complete, all the crossings that include non-leaf members have values that are the sums of the values of all the all-leaf crossings that are hierarchically subordinate to them.

There are two important exceptions to this general pattern:

- A crossing is omitted from the summation if the relevant child member has the **This member rolls up into its parent** check box on the **General** tab of the member properties window deselected.
- The roll-up behavior of time hierarchies varies with account types and Frequency members, as explained below.

For display crossings whose Frequency member involves frequency conversion through the summation of a series of stored Period Activity values, there is no hierarchical roll-up process in a time hierarchy. Instead, each displayed value reflects the summation of the appropriate time series of stored Periodic Activity values. Here are two examples:

- For a balance account and any To Date display frequency, tables show the end-of-period balance for every time period. Each end-of-period balance is computed by summing a series of stored Period Activity values from the first time period in the governing time hierarchy through the time period in question.
- For a flow account and the Year To Date frequency, reports show year-to-date aggregate flows for every time period. Each year-to-date aggregate flow is computed by summing a series of stored Period Activity values from the first time period in the year through the time period in question.

On the other hand, for a flow account and the Period Activity frequency, a time hierarchy rolls up just like a hierarchy in any other dimension type. For example, the sales expense value for a quarter is the sum of the sales expense values for the three months of that quarter, and the sales expense value for a year is the sum of the sales expense values for the four quarters of that year.

See Also

- Generating Adjustments from Adjustment Rules on page 156
- Generating Intercompany Eliminations on page 156
- Frequency Conversion on page 157
- Currency Conversion on page 159
- Combining Frequency Conversion and Currency Conversion on page 160
- Computing Retained Earnings on page 161
- Computing Cumulative Translation Adjustments on page 163
- Formula Types on page 169

Chapter 15

Formulas

Introduction to Formulas **167**

Creating Formulas **168**
Overview of Creating Formulas 168
Create Formulas for an Existing Member 168
Create a Member That Carries Formulas 169

Formula Types **169**
Overview of Formula Types 169
Formula Types and Account Types 170
Formula Types and Form Sets 170
Formula Types and Constant Formulas 171
Reporting Formulas versus Driver Formulas and Modeling Formulas 171
Driver Formulas versus Modeling Formulas 172
Details of Driver-Formula Computations 173

Formula Scopes **175**
Overview of Formula Scopes 175
Factors That Determine the Scope of a Formula 175
Specifying Scope Definitions 176

Formula Expressions **176**
Overview of Formula Expressions 176
Expression Text and Context: the Same-Member Rule 176
Expression Text and Context: Specifying Default Members 178
Syntax of Expression Text 178

Resolving Conflicts between Formulas **179**

Avoiding Overuse of the IF and NESTIF Functions **180**

Dictionary of Functions **181**

Introduction to Formulas

Formulas enable you to specify dependencies between crossings that are not handled by the built-in computations. In some cases, you can achieve the same result with either a built-in computation or a formula. Where that is true, the built-in computation is always simpler and more efficient. Reserve formulas for computational needs that are not met by any built-in computation.

SAS Financial Management has two complementary formula facilities. There is a formula facility in SAS Financial Management Studio, and there is another formula facility in the

SAS Financial Management Add-In for Microsoft Excel. The Excel Add-In formula facility gives Excel Add-In users the ability to define and modify their own formulas. Use the SAS Financial Management Studio formula facility to create formulas that are needed by all users and formulas that are beyond the capabilities of the Excel Add-In formula facility.

When you define a formula in SAS Financial Management Studio, you associate it with a particular member of a particular dimension. You can associate any number of formulas with the same member. Associating several formulas with the same member can be an efficient alternative to defining a single formula that uses IF or NESTIF to test the current crossing.

If you create a member using the Member wizard, there are wizard pages that enable you to associate formulas with the member as you create it. You can also associate formulas with any existing member, using the **Formulas** tab of the member's Properties window.

Each formula belongs to one of three formula types. All formulas that are associated with a given member must belong to the same formula type.

Associating a formula with a member begins the process of defining the formula's scope. The scope of a formula is the set of crossings where the formula is evaluated.

You can associate formulas with most members of most dimensions. The exceptions are members for which formulas would not make sense:

- You cannot associate formulas with any member of the Currency, Source, or Time dimension types.
- In the Account dimension type, you cannot associate formulas with any Retained Earnings account or any Cumulative Translation Adjustment account. These account types implement specific built-in computations.

See Also

- Formula Types on page 169
- Formula Scopes on page 175
- Avoiding Overuse of the IF and NESTIF Functions on page 180

Creating Formulas

Overview of Creating Formulas

Using the **Formulas** tab of a member's Properties window, you can create formulas for an existing member. Using the Member wizard, you can create a member that carries formulas.

Both the **Formulas** tab and the Member wizard enable you to specify every aspect of a formula, including the formula type, the formula scope, and the formula expression.

Create Formulas for an Existing Member

To associate one or more formulas with an existing member, or to edit a formula that is associated with an existing member, do the following:

1. In the Dimensions view, select the dimension that contains the member.

2. Either select **Members** to open the Members view for the selected dimension, or select **Hierarchies** to open the Hierarchies view.
3. In the Members view or the Hierarchies view, select the member.
4. Select **Properties**.

   The Properties window for the selected member appears.
5. Work on the **Formulas** tab of the Properties window, consulting the online Help for that tab as necessary.

Create a Member That Carries Formulas

To create a member that carries one or more formulas:

1. In the Dimensions view, select the dimension in which you want to create the member.
2. If you want to create a formula-carrying member in the selected dimension without inserting it into a hierarchy, select **Members** to open the Members view for the selected dimension.

   If you want to create a formula-carrying member in the selected dimension while inserting the new member into a hierarchy, select **Hierarchies** to open the Hierarchies view.
3. If you use the Members view, select **New Member** to launch the New Member wizard. On the Creation Method page of the wizard, select **Create a new member with formulas**. Continue through the wizard, consulting the online Help for the individual wizard pages as necessary.

   If you use the Hierarchies view, select the existing member that marks the location where you want to insert the new member, and then select **Add Member** to launch the Add Member wizard. On the Method page of the wizard, select **Create a new member with formulas**. Continue through the wizard, consulting the online Help for the individual wizard pages as necessary.

See Also

- Formula Types on page 169
- Formula Scopes on page 175
- Formula Expressions on page 176

Formula Types

Overview of Formula Types

The behavior of a formula that you create in SAS Financial Management Studio depends on what type of formula it is. There are three types—reporting formulas, driver formulas, and modeling formulas.

If you associate a formula with a member of any dimension type other than Account, then the formula is automatically a reporting formula. For example, if you associate a formula

that computes variances with the Variance member of an analysis dimension, then that formula is automatically a reporting formula.

If you associate a formula with an account, then you must specify what type of formula it is. If you associate more than one formula with the same account, then all those formulas must be of the same type. You cannot mix formula types on a single account.

Deciding which type of formula to associate with an account involves two decisions, which are based on different criteria. Your first decision is whether to create a reporting formula or a formula of one of the other types. If you need to create a formula of one of the other types, then you have a second decision: whether to create a driver formula or a modeling formula. You can switch the formulas that are associated with a given account from one formula type to another after the formulas are created.

You cannot associate a formula of any type with any member of the Currency, Source, or Time dimension types.

Formula Types and Account Types

Formula types are related to account types in the following ways:

- You cannot associate a formula of any type with a Retained Earnings account or a Cumulative Translation Adjustment account.
- Reporting formulas can be appropriate for accounts of the following types: Statistical, Revenue, Expense, Statistical Flow, Asset, Equity, Liability, Statistical Balance. Many reporting formulas, such as those that compute financial ratios, are most appropriately associated with Statistical accounts.
- Driver formulas and modeling formulas can be appropriate for accounts of the following types: Revenue, Expense, Statistical Flow, Asset, Equity, Liability, Statistical Balance.

  In general, it is not appropriate to associate a driver formula or a modeling formula with a Statistical account. This is because the computed values of driver formulas and modeling formulas normally serve as inputs to the hierarchical roll-up process, and Statistical accounts do not participate in the hierarchical roll-up process.

  It is often appropriate for a driver formula or a modeling formula to get input values from a Statistical account. For example, in a driver formula that computes monetary sales as the product of unit sales and unit price, the unit price can be kept as a value of a Statistical account. This prevents the unit price from participating in the hierarchical roll-up process. Alternatively, the unit price can be kept as a driver rate.

Formula Types and Form Sets

In a financial form set, any well-formed formula of any type is evaluated. In an operational form set, only driver formulas are evaluated. Moreover, not all driver formulas are evaluated in an operational form set. The only formulas that are evaluated in an operational form set are driver formulas that contain references to no members other than accounts and time periods. These are the driver formulas whose evaluation can be triggered by data-entry events.

When you publish an operational form set, a warning message appears if the model that the form set is based on includes any formulas that cannot be evaluated in an operational form set. The warning message gives you access to a list of all the formulas that cannot be evaluated in an operational form set. This warning message is not necessarily a problem. You might want to leave the listed formulas as they are, so that they can be evaluated in

other contexts. You can proceed to publish the form set, knowing that the listed formulas will be ignored in this context.

The **Run driver formulas** option is available for financial form sets but not for operational form sets. For financial form sets, this option is needed to evaluate driver formulas that contain references to members other than accounts and time periods. In operational form sets, driver formulas that contain references to members other than accounts and time periods are ignored, so this option is not needed.

For a discussion of driver-formula syntax and the **Run driver formulas** option, seeDetails of Driver-Formula Computations on page 173.

Formula Types and Constant Formulas

Constant formulas are evaluated only if they are reporting formulas. A constant formula that is a driver formula or a modeling formula is ignored.

A constant formula is a formula that contains no references to members. Its computed value does not depend on the values at any crossings. Here are some examples of constant formulas:

- `2/3`
- `DRATE("price")`
- `1.5*DRATE("price")`

For a detailed discussion of formula syntax, see Formula Expressions on page 176.

Reporting Formulas versus Driver Formulas and Modeling Formulas

Reporting formulas differ from driver formulas and modeling formulas in their relationship to the hierarchical roll-up process:

- The evaluation of driver formulas and modeling formulas is an early step in the computation of values for display. It happens before the hierarchical roll-up process begins.

  The evaluation of reporting formulas is the last step in the computation of values for display. It happens after the hierarchical roll-up process is complete.

- Driver formulas and modeling formulas are used to feed the hierarchical roll-up process. They compute inputs to the hierarchical roll-up process that depend on other inputs to the hierarchical roll-up process. For example, a planning exercise might make the assumption that the monetary value of employee benefits is 30% of the monetary value of employee salaries. You can incorporate this assumption into a model by associating with the Benefits leaf account a driver formula or modeling formula that refers to the Salary leaf account. For each Salary value, the formula yields a corresponding Benefits value. The Salary values and the corresponding Benefits values are all inputs to the hierarchical roll-up process.

  Reporting formulas are used to compute analytically useful values that depend on roll-up values. The computation of any standard financial ratio makes a good example. Consider the computation of debt-to-equity ratios for various organizations. Total debt and total equity are computed by the roll-up process in the account hierarchy. For any organization that is above the leaf level of the organization hierarchy, the roll-up process in the organization hierarchy is also involved. You can include debt-to-equity ratios in read-only tables, CDA tables, or data-entry tables in the SAS Financial Management Add-In for Microsoft Excel by having a Debt-to-Equity-Ratio account that carries a reporting formula that refers to the Total Debt account and the Total Equity account.

It is easy to decide whether a given formula should be a reporting formula. If the inputs to the formula include outputs of the hierarchical roll-up process, then the formula must be a reporting formula. If the outputs of the formula are inputs to the hierarchical roll-up process, then the formula must be a driver formula or a modeling formula. The decision always depends on the relationship of the formula to the hierarchical roll-up process.

Here are some other ways in which reporting formulas differ from driver formulas and modeling formulas:

- The scope of a driver formula or modeling formula is automatically limited to crossings whose members are all leaf members.

  The scope of a reporting formula can include any crossing that contains the member that carries the formula. If the hierarchical roll-up process produces a value for a crossing that is in the scope of a reporting formula, then the value that is computed by the reporting formula overwrites the previously computed roll-up value.

  See Formula Scopes on page 175.

- Driver-formula computations and modeling-formula computations are done in the functional currency of the relevant organization. The output values of driver formulas and modeling formulas undergo conversion to other currencies as necessary.

  Reporting-formula computations are done in the currency of the display crossing. To produce the inputs to a reporting formula, values undergo conversion to the display currency as necessary.

- The input crossings of a driver formula or modeling formula must consist entirely of leaf members. Moreover, a driver formula or modeling formula must not have an input crossing that contains a Retained Earnings account or a Cumulative Translation Adjustment account.

  A reporting formula has no restrictions on its input crossings.

Driver Formulas versus Modeling Formulas

The main difference between driver formulas and modeling formulas concerns the circumstances in which the formulas are evaluated:

- A modeling formula is evaluated each time a user of the SAS Financial Management Add-In for Microsoft Excel or the Web-based Form Editor opens a table that depends on the formula. In this respect, a modeling formula is like a reporting formula.
- A driver formula is never evaluated when a user opens a table. Instead, driver-formula computations are triggered by certain other events, and the computed output values are stored as data. When a user opens a table that depends on a driver formula, the relevant stored output values are retrieved and used.

  Driver-formula computations can be triggered when a form set is published, when a user enters data into a cell of a selected form, and when an administrator selects **Run driver formulas** for a selected financial form set. In each case, a specific set of driver-formula computations is performed, as explained in the next section.

  Note: The **Run driver formulas** option is available only for financial form sets. It is not available for operational form sets. See Formula Types and Form Sets on page 170.

Another difference between driver formulas and modeling formulas concerns scope constraints in the Source hierarchy:

- The crossings in the scope of a modeling formula can contain any Base leaf member (Base, BaseJourn, BaseForm) or any adjustment leaf member (Manual, Bal, Reclass, Alloc, CPO) of the Source hierarchy.
- The scope of a driver formula is automatically limited to the BaseForm member of the Source hierarchy; there is no choice.

For driver formulas and modeling formulas, input crossings can contain any Base leaf member (Base, BaseJourn, BaseForm) or any adjustment leaf member (Manual, Bal, Reclass, Alloc, CPO) of the Source hierarchy.

The adjustment members of the Source hierarchy are rarely needed for either input crossings or output crossings. A driver formula always has BaseForm outputs and typically has BaseForm inputs. A modeling formula typically has either Base inputs and Base outputs or BaseForm inputs and BaseForm outputs.

Driver formulas are intended primarily for use with input values that users enter through forms. However, driver formulas can also have input values that are loaded into a cycle from the Cross Industry Detail Data Store or from another cycle. When you load data into a financial cycle, you can choose to associate the data with the BaseForm member of the Source hierarchy. Moreover, you can create a one-form financial form set for the sole purpose of applying the **Run driver formulas** option to a certain collection of BaseForm cycle data. It is not necessary to use the financial form set to enter data.

The key advantage of driver formulas over modeling formulas is the speed with which formula-dependent values can be displayed when a table is opened in the SAS Financial Management Add-In for Microsoft Excel or the Web-based Form Editor. Opening a table can take a long time if the operation includes the evaluation of many modeling formulas at many crossings. Using stored output values of driver formulas enables tables to open more quickly.

With this performance advantage of driver formulas comes the risk that the stored output values might be out of date. This can happen if non-form inputs to the driver formulas such as cycle data, driver rates, or adjustments have changed since the formulas were last evaluated. It can also happen if the driver formulas themselves have been modified. For a financial form set, you can make sure that all the stored output values of driver formulas are up-to-date through appropriate use of the **Run driver formulas** option.

If driver formulas or non-form inputs to driver formulas change frequently, then you must use the **Run driver formulas** option frequently to make sure that all driver-formula output values are up to date. At some point, this can become an inconvenience, making modeling formulas the better choice.

Details of Driver-Formula Computations

Events of these types can trigger driver-formula computations:

- a user entering data into a cell of a form
- an administrator publishing a form set
- an administrator selecting the **Run driver formulas** option for a selected financial form set

  Note: The **Run driver formulas** option is available only for financial form sets. It is not available for operational form sets. See Formula Types and Form Sets on page 170.

Each type of event triggers driver-formula computations at a specific set of crossings.

When a user enters data into a cell of a form, driver-formula computations occur at those crossings that satisfy all the following conditions:

- The crossing's account has a driver formula associated with it, and the crossing is in that formula's scope.
- The crossing's account is in the Account hierarchy that is used by the form set that the form belongs to.
- The crossing's time period is in the form set that the form belongs to.
- For all dimension types other than Account and Time, the crossing's members match the members of the cell into which the user enters data.

The following points about data-entry-triggered driver-formula computations deserve special attention:

- The restriction on accounts is different from the restriction on time periods. The crossing's account must be in the Account hierarchy that is used by the form set. The crossing's time period must be in the form set itself.
- A driver formula typically contains explicit references to one or more accounts. Explicit time references such as ["TIME"=CURRENT("TIME")–1] are also common. If these are the only types of explicit references in your driver formulas, then user-entered values are fed immediately into all driver-formula computations that depend on them.

  However, the situation is different for any driver formula that contains explicit references to members of other dimension types. For example, consider a driver formula that contains an explicit reference to [ORGA]. In an operational form set, this driver formula is ignored. In a financial form set, when a user enters a value into an ORGA cell, this driver formula is evaluated only at ORGA crossings. It is not evaluated at crossings for any other organization, even though there are crossings for other organizations whose values depend on the entered ORGA value. To perform all the driver-formula computations that depend on the entered ORGA value, you must select the **Run driver formulas** option.

When a finance process administrator publishes a selected financial form set or selects **Run driver formulas** for a selected financial form set, driver-formula computations occur at those crossings that satisfy all the following conditions:

- The crossing's account has a driver formula associated with it, and the crossing is in that formula's scope.
- The crossing's account is in the Account hierarchy that is used by the selected form set.
- The crossing's time period is in the selected form set.
- Every other member of the crossing is in the selected form set.

When a finance process administrator publishes a selected operational form set, driver-formula computations occur at those crossings that satisfy all the following conditions:

- The crossing's account has a driver formula associated with it, and the crossing is in that formula's scope.
- The crossing's account is in the Account hierarchy that is used by the selected form set.
- The crossing's time period is in the selected form set.
- Every other member of the crossing is in the selected form set.
- The driver formula includes no references to members other than accounts and time periods.

  If the model that the form set is based on includes driver formulas that refer to members of other dimension types, then a warning message appears. See Formula Types and Form Sets on page 170.

See Also

- Hierarchical Roll-Ups on page 164
- Formula Scopes on page 175

Formula Scopes

Overview of Formula Scopes

If the scope of a formula does not overlap the scope of any other formula, then the formula's scope is the set of crossings where the formula is evaluated. If the scopes of two formulas overlap, then only one of the formulas is evaluated at each crossing in the area of overlap. The procedures for resolving conflicts between formulas determine which formula is evaluated.

Factors That Determine the Scope of a Formula

The scope of a formula is determined by several factors. Some of these factors are specific to models. It is therefore possible for a formula to have different scopes in different models. The scope of a particular formula in a particular model is determined by the following:

- whether the member that carries the formula is a leaf member in the model

  If the member that carries the formula is not a leaf member in the model, then the formula is ignored in that model. In effect, its scope is empty.

- the identity of the member that carries the formula

  A formula can be evaluated only at crossings that contain the member that carries the formula.

- the total set of crossings in the model

  The scope of a formula in a given model is a subset of all the crossings in that model.

- the formula type of the formula

  The scopes of driver formulas and modeling formulas are limited to crossings that consist entirely of leaf members. This limitation does not apply to reporting formulas; they can be evaluated at any crossing in which the member that carries the formula is a leaf member.

- user-specified scope definition

  When you create or edit a formula, you can narrow its scope by specifying members for various dimension types. For a dimension type with no scope-narrowing members specified, the scope can include crossings that contain any member from that dimension type. For a dimension type with one or more scope-narrowing members specified, the scope is limited to crossings that contain the specified members.

  At one extreme, if you do not specify any scope-narrowing members for any dimension type, then the scope of the formula in a given model consists of all crossings in that model that contain the member the carries the formula. In a model that has several hierarchies that have a few hundred members each, such a maximum scope consists of billions of crossings.

  At the other extreme, if you specify a single member for each dimension type, then the scope of the formula consists of only one crossing.

Specifying Scope Definitions

When you create a formula with the Formula wizard, you can use the Formula Scope page to narrow the scope. When you edit a formula, you can use the **Scope** tab of the formula's Properties window to change the scope definition.

Scope definitions are important. Ideally, the scope of a formula should be the precise set of crossings where the formula yields meaningful results that you are interested in. If your formula scopes are larger than they need to be, then your system wastes time on unnecessary computations.

If you associate several formulas with the same member, then it is important to give each formula a different scope definition. It is all right for the scope definitions to overlap. However, if formula scopes overlap, then formula conflicts arise at crossings that are in the scope of more than one formula. For this reason, you must place the formulas in a rank order. Each conflict is resolved in favor of the conflicting formula that has the highest rank. If a member carries formulas whose scopes do not overlap, then the formulas never conflict and the specified rank order is never used.

Sometimes you can specify a given computation over a given scope in more than one way. For example, if a scope consists of a single crossing, you can in principle associate an appropriate formula with any member in that crossing and then narrow the scope by specifying a single member for each of the other dimension types. In such a case, associate the formula with a member that makes the required formula simpler and easier to maintain. In general, think first of the set of crossings where a formula should be evaluated, and from that determine which member of which dimension to associate the formula with.

See Also

- Resolving Conflicts between Formulas on page 179
- Formula Types on page 169

Formula Expressions

Overview of Formula Expressions

The text of every formula expression is accompanied by a context. Working with formula expressions requires an understanding of both the text and the context.

Expression Text and Context: the Same-Member Rule

A formula expression includes one or more references to crossings that contain input values for the computation. The references to input crossings consist of relevant members. However, most of the members that are needed to refer to specific input crossings are not present in the text of the expression. Instead, they are part of the expression's context.

The relationship between text and context is best approached through examples. Suppose that the following formula is associated with the PROFIT_MARGIN account:

```
["ACCOUNT"="PROFIT"]/["ACCOUNT"="SALES"]
```

The scope of this formula consists of crossings that combine the PROFIT_MARGIN account with various organizations, time periods, analysis members, and so on. For each output crossing, a profit margin is computed as the ratio of profit to sales. In each case, the appropriate input crossings are determined by combining the account members that are present in the expression text with other members that are components of the output crossing. In other words, apart from the members that are present in the expression text, each input crossing contains the same members as the output crossing that it feeds.

To compute the actual profit margin for organization A in 2007, the actual profit for organization A in 2007 is divided by the actual sales of organization A in 2007. To compute the forecast profit margin for organization B in 2009, the forecast profit for organization B in 2009 is divided by the forecast sales of organization B in 2009. And so on. The formula expression represents a sheaf or bundle of similar computations that are carried out across the formula's scope, using members from each output crossing to complete the specification of the associated input crossings.

Microsoft Excel formulas also have this contextual aspect. In Microsoft Excel, cell references that specify only a row and a column are assumed to refer to cells that are in the same sheet as the cell that contains the formula. For example, the formula expression 2*B6 refers to cell Sheet1!B6 if it is in a Sheet1 cell, but it refers to cell Sheet2!B6 if it is in a Sheet2 cell. You can think of the same-member rule in SAS Financial Management formulas as an extension to multiple dimensions of the same-sheet rule in Microsoft Excel formulas.

Consider another example. Suppose that the following formula is associated with the SALES_PERCENT account:

```
["ACCOUNT"="SALES"]/["ACCOUNT"="SALES"]["INTORG"="WORLDWIDE"]
```

The SALES_PERCENT formula is similar to the PROFIT_MARGIN formula. In both formulas, the value at each output crossing in the formula's scope is computed as the ratio of the values at two input crossings. The key difference is that the input crossing reference in the denominator of the SALES_PERCENT formula consists of two members instead of one; it contains an account and an organization instead of just an account. As the SALES_PERCENT formula is evaluated across its scope, the organizations in the numerator input crossings are the same as the organizations in the corresponding output crossings, but the organization in all the denominator input crossings is WORLDWIDE.

To compute the actual sales percentage for organization A in 2007, the actual sales for organization A in 2007 is divided by the actual sales for the WORLDWIDE organization in 2007. To compute the forecast sales percentage for organization B in 2009, the forecast sales for organization B in 2009 is divided by the forecast sales for the WORLDWIDE organization in 2009. And so on.

This example illustrates several points about the relationship between the text and the context of a formula expression:

- An input crossing reference can contain members for any subset of the set of dimension types in the governing model. You can create an input crossing reference that is only an account. You can create an input crossing reference that combines an account and an organization. You can create an input crossing reference that combines an account and an organization and an analysis member. And so on.
- Different input crossing references in the same formula can contain members for different dimension types. In the SALES_PERCENT formula, the numerator input crossing reference is an account while the denominator input crossing reference combines an account and an organization.
- The context fills out each input crossing reference as needed. For the numerator input crossing reference of the SALES_PERCENT formula, the context supplies every member except the account. For the denominator input crossing reference of the

SALES_PERCENT formula, the context supplies every member except the account and the organization.

Expression Text and Context: Specifying Default Members

By default, the context of a formula expression is fully defined by the same-member rule as explained in the previous section. However, there is an alternative. For any dimension type, you can designate a single specific member as a constant component of all input crossings. The constant member can be either the default member of the model hierarchy for that dimension type or any member that you select.

Here is an example that illustrates the potential usefulness of designating a default member for an expression. Suppose that you project that the salary expense for each month of one year will be 5% greater than the salary expense for the same month of the previous year. To compute budgeted salary expenses according to this relationship, you could define a driver formula in the following way:

- Associate the following formula expression with the Salaries account:

```
1.05*["ACCOUNT"="Salaries"]["TIME"=CURRENT("TIME")-12]
```

- Designate the Actual member of the Analysis dimension type as a default member for the formula's input crossings.
- Limit the scope of the formula to the Budget member of the Analysis dimension type.

In this example, the 5% growth factor and the 12-month time shift are specified in the text of the expression, while the transfer from Actual to Budget is handled by specifying a default expression member and a scope restriction.

To specify a constant member for all input crossings for one or more dimension types, select the **Specify default members for crossings in the expression** check box in one of these locations:

- Expression page of the Formula wizard
- **Expression** tab of a formula's Properties window

This check box activates a work area in which you can designate constant members for selected dimension types.

Note: You cannot select the Frequency dimension type here. The Frequency member in an input crossing is always the same Frequency member that is in the associated output crossing.

Syntax of Expression Text

To build the text of a formula expression, use the Formula Editor window. To display the Formula Editor window, click **Edit** on the Expression page of the Formula wizard or the **Expression** tab of a formula's Properties window.

SAS Financial Management Studio formulas have a conventional syntax that combines input crossing references, operators, and SAS Financial Management functions. The **Validate Expression** button on the Formula Editor window enables you to check whether the expression that you are building violates any syntactic rules.

The syntax of input crossing references is illustrated by the formula examples in the preceding discussions of the context of formula expressions. Each input crossing reference consists of a single member or a set of members from different dimension types. Here is a one-member reference:

```
["ACCOUNT"="PROFIT"]
```

Here is a two-member reference:

```
["ACCOUNT"="SALES"]["INTORG"="WORLDWIDE"]
```

Note the following syntactic details:

- Each reference to a member is enclosed in square brackets.
- Inside the square brackets, there is a dimension type code surrounded by quotation marks, an equal sign (=), and a member code surrounded by quotation marks.

This syntax is generated automatically when you insert a member into an expression using the **Members** tab of the Formula Editor window.

In most dimension types, each non-leaf member of a hierarchy has a special leaf member subordinate to it, which is known as its virtual child. Virtual child members are used to load or enter data for the non-leaf members that they are subordinate to without interfering with the hierarchical roll-up that occurs for the non-leaf member itself. To refer to a virtual child member in a formula, you can use either the VIRTUALCHILD function or the .vc suffix:

```
["INTORG"=VIRTUALCHILD("MKTG")]

["INTORG"="MKTG.vc"]
```

The Formula Editor window has a button for each of the following standard operators:

- arithmetical: addition, subtraction, multiplication, division, exponentiation
- Boolean: AND, OR, NOT
- comparison: equal, not equal, greater than, less than, greater than or equal to, less than or equal to
- string: concatenation

To insert a function into an expression, use the **Functions** tab of the Formula Editor window.

See Also

- Formula Scopes on page 175
- Formula Types on page 169
- Dictionary of Functions on page 181

Resolving Conflicts between Formulas

If a crossing is in the scopes of two or more formulas, then the formulas conflict at that crossing. The numeric value of that crossing must be computed using one of the conflicting formulas, but which one?

Formula conflicts can arise in two ways:

single-member formula conflict
: Two or more formulas that are associated with the same member have overlapping scopes. A formula conflict exists at each crossing that is in the area of overlap.

Such conflicts are resolved by the ranking of the formulas on the **Formulas** tab of the member's Properties window. The conflicting formula with the highest rank is evaluated and all other formulas are ignored.

This type of conflict cannot arise if you never associate more than one formula with the same member.

multi-member formula conflict
Formulas are associated with members of two or more dimension types. For example, a formula is associated with account ABC and another formula is associated with analysis member Variance. A conflict exists at any crossing that combines account ABC with analysis member Variance, if the crossing is in the scopes of both formulas.

Such conflicts are resolved by the ranking of the dimension types on the **Formulas** tab of the governing model's Properties window. The conflicting formula whose member belongs to the most highly ranked dimension type is evaluated and all other formulas are ignored.

This type of conflict cannot arise if you only associate formulas with accounts.

It is possible for both types of conflict to exist at the same crossing. In that case, the conflict resolution process has two steps:

1. The software uses the rank order on the **Formulas** tab of the governing model's Properties window to determine which member of the crossing is the winning member.
2. The software uses the rank order on the **Formulas** tab of the winning member's Properties window to determine which of the winning member's formulas is the winning formula.

See Also

Formula Scopes on page 175

Avoiding Overuse of the IF and NESTIF Functions

You can reduce computation time by avoiding formulas such as the following, which test the current crossing by using CURRENT or ISCURRENT inside IF or NESTIF:

```
IF(CURRENT("ANALYSIS")="Forecast",1,3)

IF(ISCURRENT("ANALYSIS","Forecast"),1,3)

NESTIF(CURRENT("ANALYSIS")="Forecast",1,
     CURRENT("ANALYSIS")="Budget",3,
          CURRENT("ANALYSIS")="Actual",5)

NESTIF(ISCURRENT("ANALYSIS","Forecast"),1,
     ISCURRENT("ANALYSIS","Budget"), 3,
          ISCURRENT("ANALYSIS","Actual", 5)
```

Instead of defining a conditional formula of this sort, define a set of formulas that are all associated with the same member. Define a separate formula for each conditional computation, giving it a scope that corresponds to the associated condition.

Each IF statement shown here can be replaced with a pair of formulas: one that has its scope limited to the Forecast member of the Analysis dimension type and one that has its scope

limited to all the other members of the Analysis dimension type. Each NESTIF statement shown here can be replaced with a set of three formulas, one that has its scope limited to Forecast, one that has its scope limited to Budget, and one that has its scope limited to Actual.

See Also

Formula Scopes on page 175

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.

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
: returns the code of the member that is the designated hierarchical ancestor of the member that is specified either explicitly or implicitly by the first argument of the function.

  The ANCESTOR function takes two arguments:

  - 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:
    - 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.
    - If the first argument is the time dimension type code or a function that returns a time period code, then you can use one of the following period type values to indicate a particular level of the time hierarchy:

      ```
      AllYears

      Year

      HalfYear
      ```

```
QuarterYear

Month

Week

Day
```

Consider a year-quarter-month time hierarchy and the following two references:

```
["ACCOUNT"="Headcount"]["TIME"=ANCESTOR("TIME",2)]

["ACCOUNT"="Headcount"]["TIME"=ANCESTOR("TIME","Year")]
```

When evaluated at a month crossing, both of these references retrieve the value of the Headcount account for the year that contains that month.

Instead of a dimension type code, the first argument can be a function that returns a member code, as in the following examples:

```
["ACCOUNT"="Headcount"]["TIME"=ANCESTOR(PREVIOUS("TIME"),2)]

["ACCOUNT"="Headcount"]["TIME"=ANCESTOR(PARENT("TIME"),"Year")]
```

If the specified ancestor does not exist, then the operand that the ANCESTOR function is part of receives a value of 0.

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.

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, 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, 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]
```

See Avoiding Overuse of the IF and NESTIF Functions on page 180.

CURRENTNAME

returns the name 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 CURRENTNAME function takes one argument: a dimension type code. You might find the CURRENTNAME function useful as part of the Boolean expression of an IF function. For example, the following IF function returns **`0`** for the account named CASH and **`1`** for all other accounts:

```
IF(CURRENTNAME("ACCOUNT")="CASH",0,1)
```

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()
```

You can use the PUTN function to convert the value returned by the DATETIME function to a readable exact time, as follows:

```
PUTN((DATETIME(),"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 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.

EXP
returns the result of raising **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 examples:

```
IF("A"="B","right","wrong")

IF(1=2,"right","wrong")

IF("A"="B",1,0)

IF(1=2,1,0)
```

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.

See Avoiding Overuse of the IF and NESTIF Functions on page 180.

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)
```

  See Avoiding Overuse of the IF and NESTIF Functions on page 180.

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, all of which 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, all of which 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 any number of arguments, all of which 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**

See Avoiding Overuse of the IF and NESTIF Functions on page 180.

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:

```
["ACCOUNT"="Headcount"]["TIME"=OPENINGPERIOD(PREVIOUS(("TIME")))]
```

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.

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. 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:

AccountType	is 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	is 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	is 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`**
Functional Currency	is the functional currency of an organization. This property is valid only

	if the member is an organization. Any currency code can be returned.
Intercompany	indicates 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	is 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	indicates 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.

For example:

- **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.

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.

For example:

- **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, which represents an angle measure in radians. Since all the arc 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, which represents an angle measure in degrees. Since 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`**.

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

  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:

  ```
  ["INTORG"=VIRTUALCHILD(PARENT("INTORG"))]
  ```

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 one of the following exchange rate type codes:

```
PeriodAverage

PeriodClose

PeriodOpen

Custom1

Custom2

Derived

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 16

Security in SAS Financial Management

About Security in SAS Financial Management **199**

Object-Level Security **200**
Why Use Object-Level Security? 200
Effects of Denying Access to a Cycle 200
Effects of Denying Access to a Model 200
Precedence Rules for Object-Level Security 201
Denying Access to an Object 201
Superuser Access to Cycles and Models 202

Member-Level Security **202**
Why Use Member-Level Security? 202
Effects of Member-Level Security 202
Precedence Rules for Member-Level Security 202
Denying Read Access 203

About Security in SAS Financial Management

Security features in SAS Financial Management include the following:

- **User and group management** can be used to define permission settings for folders and files.
- **Role-based security** determines a user's ability to perform various tasks, such as creating a dimension or editing a form.

  For information about groups and roles for SAS Financial Management, see "Assigning Groups and Roles" in the *SAS Solutions Services: System Administration Guide*.
- **Object-level security** consists of permissions that you apply to models and cycles, in order to restrict users' access to those objects.
- **Member-level security** consists of permissions that you apply to members of one or more dimensions, to restrict users' access to data.

Object-Level Security

Why Use Object-Level Security?

Object-level security refers to permissions that are applied to cycles or models.

Your site might require multiple cycles. For example, you might need to access data in terms of different period types (such as months and quarters), or you might need to segregate data for a special project. In such cases, you might want to protect access to certain cycles.

At a site with multiple models, one model might be used for management reporting, allowing managers to view data by product. Another model might be used for legal (external) reporting, allowing users to view data by country. In that case, you could establish security settings so that managers had access only to the management model, while financial administrators had access to both models.

Effects of Denying Access to a Cycle

If no permission is set on a cycle, the default permission is grant. If a user is denied access to a cycle, these are the results:

- SAS Financial Management Studio:
  - The cycle is not listed in the Cycles workspace. Accordingly, the user cannot view or edit the cycle's properties.
  - The user is denied access to any models that use that cycle.
  - Because rate sets are subordinate to cycles, the user cannot view or edit exchange rates or driver rates that are associated with the cycle.
  - The user is denied access to all form sets and forms that are associated with the cycle.
- SAS Financial Management Add-In for Microsoft Excel:
  - Models that are based on the cycle are not available for selection in the table wizard. Accordingly, the user cannot create a table that shows data from the cycle or a table that enables users to enter data into the cycle.
  - The user cannot view any table that shows data from the cycle or enter data in any table that stores data in the cycle.
- Document Manager: The user cannot run any secondary financial reports that show the cycle data or numbers that are generated from the cycle data.
- Form Manager: A form that is associated with a denied cycle is not listed.

Effects of Denying Access to a Model

If no permission is set on a model, the default permission is grant. If a user is denied access to a model, these are the results:

- The user is also denied access to a composite model that includes that model.

  Note: The remainder of the items in this list also apply to composite models.

- SAS Financial Management Studio: The model does not appear in the list of available models.
- Form Manager: A form that is associated with a denied model is not listed.
- SAS Financial Management Add-In for Microsoft Excel:
  - The model is not available for selection in the table wizard. Accordingly, the user cannot create a table that is based on the model.
  - If the user tries to open a read-only table that uses the model, SAS Financial Management tries to substitute a compatible model and displays an alert message that warns of the substitution. (In terms of security, a compatible model is one that contains the same dimension types that are used in the table as a column, row, or slicer.) If no compatible model is found, SAS Financial Management displays an error message.
  - If the user tries to open a CDA table that uses the model, SAS Financial Management displays #NUM and Error strings in the cells and headings.
  - If the user tries to open a data entry table that uses the model, SAS Financial Management displays an error message.

Precedence Rules for Object-Level Security

Precedence rules for object-level security are established based on the answers to these questions:

- Who are the permissions assigned 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.

- Do two groups with the same relationship to the user (identity precedence) apply conflicting permissions to the object?

  If so, the grant permission wins. When there is no specific permission, the default is to grant access.

Denying Access to an Object

To deny a user or group access to a cycle, model, or composite model:

1. Open the properties for the cycle, model, or composite model.
2. Select the **Security** tab.
3. If necessary, add the user or group.
4. Highlight the user or group and make sure that the **Can access this [cycle,model,composite model]** check box is not selected.

   For detailed instructions, click the **Help** button.
5. Click **OK**.

   Your changes are written to the Solutions Data Mart. They are applied the next time the user logs on to the middle tier.

Superuser Access to Cycles and Models

There is one exception to object-level security: members of the Administrators group can access all models and cycles, regardless of the security settings. (These users still require specific roles in order to manipulate cycles and models.)

This superuser status does not apply to member-level security. You can still apply member-level permissions that deny the Administrators group access to data.

Member-Level Security

Why Use Member-Level Security?

Member-level 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 permission for all members of each dimension. You assign member-level security in SAS Financial Management Studio.

Member-level security affects read permission only. Regardless of the as-of date for a member, the permissions that are currently set on the member apply. 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 Member-Level Security

In Microsoft Excel, if a user is denied read access to a member, all crossings that include that member are displayed as red cells. If the user right-clicks one of these red cells and selects **Tools ⇨ Cell information**, a pop-up message says that the query failed because of security. When a user opens an existing report that contains a read-only table, the slicer opens to the top-level member to which the user has access

Subordinate members inherit their parents' security settings unless the settings are overridden. However, siblings of the member are not affected. Roll-ups are also unaffected by grants or denials to subordinate members.

Member-level security also applies in SAS programs (such as a stored process) and in planning forms. In SAS programs, values are returned as **`NaN`** if member-level security prevents read access. In a form, users can see only the data to which they have read access.

Note: Filtering takes place on the server. Only the data that the user is entitled to view is sent to a client.

Precedence Rules for Member-Level Security

By default, all users and groups have read permission for all members of each dimension. Authorization for member-level permissions takes the following path:

1. Are any permissions directly assigned to the member? If so:
   - Who are the permissions assigned 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.

- Do two groups with the same relationship to the user apply conflicting permissions to the member? If so, the grant permission wins.

2. If no permissions are directly assigned to the member: Are any permissions directly assigned to the parent member in the hierarchy?

   The authorization process looks at the parent and works its way up the hierarchy.

Denying Read Access

In the Dimensions workspace of SAS Financial Management Studio, you can deny a user read access to a member by denying access to the user individually or by denying access to a group that the user belongs to.

To deny a user or group access to a selected member, follow these steps:

1. Open the properties for the member and click the **Security** tab.
2. If necessary, add the user or group.
3. Highlight the user or group and make sure that the **Read data using this member** check box is not selected.

   For detailed instructions, click the **Help** button.
4. Click **OK**.

   Your changes to member-level security are written to the Solutions Data Mart. They are applied the next time the user logs on to the middle tier.

See Also

"Cell Protection for Data-Entry Tables" on page 138

Part 3

SAS Financial Management Add-In for Microsoft Excel

Chapter 17
Getting Started with the Excel Add-In *207*

Chapter 18
Uses of the Excel Add-In *211*

Chapter 19
Creating a Table with the Table Wizard *227*

Chapter 20
Working with Tables *235*

Chapter 21
Dictionary of Menu Options *249*

Chapter 22
Dictionary of CDA Functions *273*

Chapter 23
Dictionary of Calculated-Member Functions *279*

Chapter 17

Getting Started with the Excel Add-In

Overview of the SAS Financial Management Add-In for Microsoft Excel **207**
Summary of Tasks **207**
Ways to Access the Excel Add-In **208**
Combining Features of the Add-In with Features of Excel **208**
Logging On and Logging Off **208**
Accessibility Features **209**

Overview of the SAS Financial Management Add-In for Microsoft Excel

The SAS Financial Management Add-In for Microsoft Excel connects your desktop copy of Microsoft Excel to the SAS Financial Management database. Through this connection, data can flow from the SAS Financial Management database to a Microsoft Excel worksheet and from a Microsoft Excel worksheet to the SAS Financial Management database.

Summary of Tasks

This add-in is an essential tool for the following important tasks:

- designing a financial report on page 211
- viewing a financial report on page 213
- designing a data-entry form template on page 214
- entering data through a form as part of a bottom-up workflow on page 221
- reviewing the data in a form as part of a bottom-up workflow on page 222
- entering data through a form as part of a top-down workflow on page 222

Ways to Access the Excel Add-In

The way that you access Microsoft Excel and this add-in depends on the task that you are performing:

- To design a financial report, you typically open Microsoft Excel on your desktop and then use the **Log On** option.
- To view a financial report, you typically select a report that is posted in the Web-based Document Manager. In response, Microsoft Excel opens and you are automatically logged on to this add-in.
- To design a data-entry form, you use one of the appropriate options in the **Forms** workspace of SAS Financial Management Studio. In response, Microsoft Excel opens and you are automatically logged on to this add-in.
- To participate in any way in a data-entry workflow, you select a form that is listed in the Web-based Form Manager. In response, Microsoft Excel opens and you are automatically logged on to this add-in.

Some menu options that are part of this add-in are active no matter how you access it. Other menu options might be active or inactive depending on how you access the add-in. For details, see the dictionary of menu options.

Combining Features of the Add-In with Features of Excel

This add-in enables you to build worksheets that combine access to the SAS Financial Management database with all the features of Microsoft Excel. Here are two examples:

- A worksheet that presents numeric values from the SAS Financial Management database in a read-only table or a CDA table can also include supplementary work areas that further analyze values that appear in these tables. A cell in a supplementary work area can contain a Microsoft Excel formula that refers to one or more cells in a read-only table or a CDA table.
- A worksheet that collects user-entered data in a data-entry table and feeds it to the SAS Financial Management database can also include supplementary work areas that compute numeric values that are needed in the data-entry table. In this case, a cell in the data-entry table can contain a Microsoft Excel formula that refers to one or more cells in a supplementary work area.

Logging On and Logging Off

If you are not currently logged on to the SAS Financial Management Add-In for Microsoft Excel, then the **Log On** option is available. When you select **Log On**, the SAS Log On window appears, enabling you to enter the necessary information to establish a connection between Microsoft Excel and your SAS Financial Management server.

If you are currently logged on, then the **Log Off** option is available. Select **Log Off** to end the connection between Microsoft Excel and your SAS Financial Management server.

Accessibility Features

The SAS Financial Management Add-In for Microsoft Excel includes accessibility and compatibility features that improve the usability of the product for users with disabilities. These features are related to accessibility standards for electronic information technology adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended.

In general, the SAS Financial Management Add-In for Microsoft Excel has the accessibility controls that are standard in a Windows environment. Note the following:

- In read-only tables, data-entry tables, and supplemental schedules, color alone is used to indicate the roles of data cells. For example, editable cells are yellow and only their yellow color indicates that they are editable.

If you have questions or concerns about the accessibility of SAS products, send an e-mail message to **accessibility@sas.com**.

Chapter 18

Uses of the Excel Add-In

Designing a Financial Report **211**
Overview of Financial Reports 211
Read-Only Tables 212
CDA Tables 212
Single-Cell Uses of the CDAGet Function 213

Viewing a Financial Report **213**

Designing a Data-Entry Form Template **214**
Overview of the Design Process 214
Creating a Data-Entry Table 215
Frequency Members for Entering Data 215
Setting a Refresh Policy for Financial Forms in Excel 216
Designing Cell Protection for Data-Entry Tables 216
Enabling Users to Enter Data at Roll-Up Points 218
Designing Automatic Allocation for Data-Entry Tables 219
Including Tables of Other Types in Financial Forms 219
Creating a Supplemental Schedule 220

Entering Data as Part of a Bottom-Up Workflow **221**
Overview 221
Entering Data into Parent Cells 222

Reviewing Data as Part of a Bottom-Up Workflow **222**

Entering Data as Part of a Top-Down Workflow **222**

Entering Data Offline **223**

Entering Data into a Supplemental Schedule **224**

Designing a Financial Report

Overview of Financial Reports

A financial report displays numeric values that are either stored in the SAS Financial Management database or computed from values that are stored in the SAS Financial Management database. There are three ways to retrieve numeric values:

- read-only tables
- CDA tables

- single-cell uses of the CDAGet function

You can include any number of these SAS Financial Management structures in a single worksheet. You can also include any number of ordinary Microsoft Excel structures and use all the capabilities of Microsoft Excel. For example, you can define supplementary work areas that perform further analysis on the values that are retrieved from the SAS Financial Management database. A cell in a supplementary work area can contain a Microsoft Excel formula that refers to cells in SAS Financial Management structures. You can also complement your numeric output with Microsoft Excel graphs and charts.

After you design a financial report, you can publish it in various file formats to either your desktop or the Web-based Document Manager. To publish a report, select **Publish** and work through the Publish Report wizard. For details, see the online Help for the individual wizard pages.

Read-Only Tables

Read-only tables are multi-dimensional tables that provide special interactive capabilities such as expanding and collapsing hierarchies, drill-down, filtering, and pivoting. To define a read-only table:

1. Select the cell that will be the upper left corner of the table.
2. Select **Read-only Table**.
3. Work through the wizard, consulting the online Help for individual wizard pages as necessary.

When you create a read-only table, you can include either fixed time periods, floating time periods, or a combination of both fixed and floating time periods. Floating time periods give users a movable window on the time dimension. They are defined relative to the table default time period, which a user can change with the **Table Properties** option.

In addition to its row and column dimensions, a read-only table can include slicer dimensions. These are displayed immediately above the upper left corner of the rectangle of rows and columns. A slicer dimension has one member selected at a time. All the numeric values that are displayed in a table at a given time are associated with the currently selected members of the slicer dimensions. When a user selects a different member for a slicer dimension, the table displays a different slice of data.

It is possible to link slicer dimensions across tables so that a user can make a single selection that affects two or more tables in a coordinated manner.

See Also

Coordinating Slicers between Tables on page 245

CDA Tables

CDA tables are collections of cells without special interactive capabilities that provide more formatting flexibility, such as the ability to insert extra rows and columns. To define a CDA table:

1. Select the cell that will be the upper left corner of the table.
2. Select **CDA Table**.
3. Work through the wizard, consulting the online Help for individual wizard pages as necessary.

When you create a CDA table that has slicer dimensions, you can choose whether to give users the ability to select slicer members.

Single-Cell Uses of the CDAGet Function

A single-cell use of the CDAGet function places a single number in a single cell. To use the CDAGet function, follow the general instructions for inserting a CDA function.

If you use the CDAGet function to place a number in a cell, then you probably want to place a suitable label in a nearby cell. There are two approaches:

- Type appropriate text in a nearby cell.
- Select **Member Labels**. Use the Member Labels window to find an appropriate label and place it in a nearby cell.

  You can use any one of the available labeling methods. However, if you want to use a Microsoft Excel reference to the cell that contains the label as an argument of a CDA function, then you must select the **Code** labeling method. Member names and descriptions are not valid argument values for CDA functions.

By placing instances of the CDAGet function and associated labels in many cells, you can build a custom table one cell at a time.

See Also

- Using CDA Functions in Microsoft Excel Formulas on page 273
- CDAGet on page 275

Viewing a Financial Report

If you open a financial report from the Web-based Document Manager, the report is immediately available for viewing.

If the report includes a read-only table, then you can change your view of that table in a variety of ways, including the following:

- Select another member from a slicer dimension to display another slice of numeric values. Slicer dimensions are listed above the table itself. In some worksheets, slicer dimensions of different tables are linked.
- Click an underlined row heading or column heading to expand or collapse the portion of the hierarchy that is subordinate to it.
- Select a table cell and then select an option such as **Filter Member Combination**, **Format Members**, or **Pivot** that affects your view of the table.

See Also

Coordinating Slicers between Tables on page 245

Designing a Data-Entry Form Template

Overview of the Design Process

After you use the New Form Set wizard in SAS Financial Management Studio to create a form set, you are given the option of opening the form template in Microsoft Excel. After you design the form template, select **Save Template**.

You can continue working on the template later. To make changes to an existing form template, select the form set in SAS Financial Management Studio and then select **Template**. The system logs on to the middle tier and displays the form template that is associated with the selected form set. Again, select **Save Template** to save your work.

When a form template is complete and ready for use, publish the associated form set to make the forms available to the users who enter data. To publish a form set, select the form set in SAS Financial Management Studio and then select **Publish**.

The heart of any form template is a data-entry table, which enables users to enter data into the forms that are based on the template. Without at least one data-entry table, a form cannot perform its basic function of receiving user-entered data. In general, a data-entry table should include the target hierarchy of the form set. If a data-entry table does not include the target hierarchy, then no user can see numeric values that other users have entered.

If you are designing a template for a financial form set, then you can also include tables of other types: read-only tables, CDA tables, and supplemental schedules. If you are designing a template for an operational form set, the other table types are not permitted.

When you create a data-entry table in a form template, you select a time period from the time hierarchy. You can select **All members**, or you can define a time member rule, which can be fixed or floating. Floating time periods are defined relative to the table default time period. The floating time member rule specifies both a viewable time period and an editable time period.

Some of the features that you can add to a form template are supported by the Excel add-in but not by the Web-based Form Editor. If you are designing a template for forms that can be opened in the Web-based Form Editor, do not include any of these Excel-only features. There are two cases:

- The forms of an operational form set can be opened only in the Web-based Form Editor. If you are designing an operational form template, then the **Save Template** option checks that the template is valid for the Web. It is impossible to save an operational form template that is not valid for the Web. The **Validate Web Form** option is available as a convenience; you can use it at any time to make sure that you have not introduced any Excel-only features.
- A financial form set can be designed to permit data entry exclusively in the Excel add-in, exclusively in the Web-based Form Editor, or in both places. If you are designing a financial form template, then the **Save Template** option does not check that the template is valid for the Web. If you are working on a financial form set that permits Web data entry, always select **Validate Web Form** before you select **Save Template**.

See Also

- Validate Web Form on page 270

- Save Template on page 267

Creating a Data-Entry Table

To create a data-entry table that maps to the SAS Financial Management database:

1. Select the cell that will be the upper left corner of the table.
2. Select **Data-entry Table**.

   This launches the Create Data-Entry Table wizard.
3. Work through the wizard, consulting the online Help for individual wizard pages as necessary.

In addition to its row and column dimensions, a data-entry table can include slicer dimensions. The slicers are displayed immediately above the upper left corner of the rectangle of rows and columns. A slicer has one member selected at a time. All the numeric values that a data-entry table can receive at a given time are associated with the currently selected members of the slicers. When a user selects a different member in a slicer, the table can receive a different slice of data.

It is possible to link slicer dimensions across tables so that a user can make a single selection that affects two or more tables in a coordinated manner.

See Also

Coordinating Slicers between Tables on page 245

Frequency Members for Entering Data

The numeric values that a user enters into a data-entry table must correspond to one of these members of the Frequency dimension:

- Period Activity (PA)
- Period to Date (PTD)
- Year To Date (YTD)

You can design a data-entry table that receives numeric values for all of these frequencies, or for any two of them, or for any one of them.

There are two ways to design a data-entry table that receives numeric values for only one frequency:

- Include Frequency in the table layout as a slicer dimension. Make PA or PTD or YTD the only available member for that slicer. The single-member Frequency slicer tells users what type of values they can enter, but users cannot select different Frequency members.
- Do not include the Frequency dimension in the table layout. After you define the table layout, select **Table Properties**. On the **Dimensions** tab of the Table Properties window, select PA, PTD, or YTD as the default write member for the Frequency dimension. With this design, the relevant Frequency member is not a visible part of the table.

To design a data-entry table that accepts numeric values for two or three Frequency members, include Frequency in the table layout as a slicer dimension and make two or three members available for selection. This enables users to switch from one Frequency slice to another.

See Also

Using the Frequency Dimension on page 243

Setting a Refresh Policy for Financial Forms in Excel

If you are designing a template for financial forms that can be opened in this Excel add-in, then it is important to select a refresh policy. Using **Automatically Refresh Tables**, you can choose between these alternatives:

- Refresh each form automatically each time a user enters a numeric value (checkmark next to the option).
- Refresh each form only when a user selects **Refresh** or **Refresh All** (no checkmark next to the option).

The Web-based Form Editor does not have an automatic refresh capability. If you are designing an operational form template, the **Automatically Refresh Tables** option is not available.

See Also

Automatically Refresh Tables on page 252

Designing Cell Protection for Data-Entry Tables

About Cell Protection

You can protect cell crossings in a data-entry form by creating one or more rules that apply to the dimensions in the data-entry table.

Cells are protected against the following actions:

- manual data entry
- spread
- automatic allocation (applies only to financial forms in a bottom-up workflow)

However, the values of these protected cells can still change as the result of indirect actions, including the following:

- calculations
- changes in the values of descendants that roll up to the protected cell
- changes in exchange rates
- changes in previous periods when frequency is To Date (for example, Year To Date or Quarter To Date)
- data that is loaded via SAS Data Integration Studio jobs
- data that was seeded from other models
- rules-based adjustments and allocations
- manual adjustments

Cell protection is applied in the following order:

1. Rules that are defined in a model. These rules are inherited by every form set that uses the model.
2. Rules that are defined in a form template. These rules, as well as the rules from the model, are inherited by all forms in the form set.

3. Cell protection that is set in a data-entry form. This protection applies only to the form in which it is defined and applies only to financial forms. You must set form-based cell protection in Microsoft Excel, but the protected cells are visible (and honored) in the Web-based Form Editor as well.

A form cannot override the protection that was set in the form set or the model, and a form set cannot override the protection that was set in the model. For example, if the model rules protect a specific crossing, the form set and its forms cannot unprotect it. However, both the form template and individual forms can define additional cell protection.

Cell protection applies to all users, regardless of their group or role membership. In a data-entry table, cells that are protected are displayed in purple. However, their contents are still viewable.

Supplemental schedules in a data-entry form honor cell protection. Forms that are being edited offline honor cell protection, with the exception of changes to cell protection rules that occurred after the form was checked out.

Note: The member protection option has been replaced by cell protection. In form templates that were migrated from a previous release, member protection settings are converted to cell protection rules.

Cell Protection Rules

You can define cell protection rules for both models and form sets. When you define a cell protection rule, you make the following selections:

- the dimension members to be included in the rule. If a crossing includes a member from each selected dimension, the crossing participates in the rule. Other dimensions are ignored.
- whether crossings that participate in the rule are protected or unprotected. Within the same set of rules, unprotect rules can be used to undo the effect of a previous rule; for example, you might apply a protection rule to a broad range of crossings and then unprotect a subset of those crossings.

Model-Based Cell Protection Rules

Cell protection rules that are associated with a model are typically loaded to the SASSDM database by a SAS Data Integration Studio job. The rules apply to all form sets that use the selected model.

From the Models workspace of SAS Financial Management Studio, you can open a Microsoft Excel workbook to view and modify these rules, or to add new rules. Be aware that loading the rules again from the SAS Data Integration Studio job deletes any rules that you added or changed via Microsoft Excel.

To view or define cell protection rules for a model, follow these steps:

1. In SAS Financial Management Studio, select a model and select **Show cell protection rules**.

   A worksheet opens in Microsoft Excel, with the Cell Protection window open.

2. Click **Edit Protection Rules** to view the current protection rules for the model.
3. To confirm that your rules are being applied correctly, select **Insert** ⇨ **Data-Entry Table** to create a data-entry table. Make sure the cells that should be protected are displayed in purple.
4. You can modify these rules or define new rules.

When you copy a model, its cell protection rules are automatically copied as well.

Template-Based Cell Protection Rules

In a form template, you can define cell protection rules that apply to all forms in the form set.

- **For financial form sets:** If the form template contains more than one data-entry table, tables that use the same model share the same rules. If the data-entry tables use different models, they have different sets of cell protection rules.
- **For operational form sets:** All the data-entry tables in a form template use the same model and share the same rules.

Follow these steps:

1. In the form template, insert a data-entry table.
2. Select any cell in the table and click **Edit Protection Rules**.

   Note: In the data-entry table, you will see the effect of any cell protection rules that were defined in the model. The model rules are maintained separately; you can view them by selecting **Cell Protection Rules** for a model in SAS Financial Management Studio.

When you modify the cell protection rules in a form template, you do not need to save the template (unless you make other changes). The new protection rules apply to all forms that use that template, even if the form set has been published. You do not need to republish the form set.

When you copy a form set, you have the option of including the form set's cell protection rules in the copy.

Note: The measures in a supplemental schedule cannot be protected directly, but they can inherit cell protection from the associated data-entry table. For more information, see the *SAS Financial Management: User's Guide*.

See Also

"Protecting Cells in a Data-Entry Table" on page 247

Enabling Users to Enter Data at Roll-Up Points

Note: This option applies only to financial forms in a bottom-up workflow.

By default, when users edit a data-entry table, they enter values in cells that contain leaf members. Those values are rolled up to the parent member in the hierarchy, subject to certain restrictions.

An administrator can enable form users to enter values into the cell for a parent member (a roll-up point). The change in the parent's value is added to the .VC member for the parent. If a dimension is on the table: If a user enters a value into a parent cell, the difference in value is added to the .VC member. If a dimension is off the table, the operation is applied to the default write member (or its .VC member).

Note: For the time dimension, which does not have .VC members, data entry must occur at a leaf member.

This option (**Allow data to be entered for parent members**) is disabled by default. In a form template, you set the option in the table properties for a data-entry table, on the **Data-Entry** tab.

Designing Automatic Allocation for Data-Entry Tables

Note: The automatic allocation option applies only to financial forms in a bottom-up workflow.

If automatic allocation is enabled for a data-entry table, then a value that is entered in a parent cell is automatically distributed based on the relative weights of eligible leaf descendants. The parent cell must contain a roll-up point in at least one dimension other than Time. (If the parent cell includes a roll-up point in Time, the cell is not writable. You cannot perform an automatic allocation across the Time dimension.)

Weights apply to the entire table. In the Allocation Weights window, an administrator can edit the assigned weights in a form template. If a weight is listed as **Editable**, it can also be edited in a form.

To receive an automatic allocation from the parent cell, a leaf descendant must meet the following requirements:

- It must be included in the table on the rows, columns, or slicers.
  - If a member is part of a slicer, you must be able to switch to the member using the **Select Member** option.
  - If a member is part of a row or column, you must be able to navigate to the member by expanding the hierarchy or drilling down in the hierarchy.

    Note: System filters, member filter combinations, and member property filtering are used for display purposes only and do not affect a leaf descendant's eligibility for automatic allocation.
  - If a dimension is off the table, the default write member is used as the only eligible leaf descendant for that dimension.
- If the member is hidden using the Excel add-in's **Hide** option, it is not eligible to receive an automatic allocation.
- The member must be writable. For example, it cannot be protected, it cannot be a target of a server-side formula, and the user must have security access to that member. Other examples of non-writable members include account types such as Retained Earnings and Cumulative Translation Adjustment, and Analysis members that are not designated as writable in the form set.

Allocations are made to all eligible leaf descendants of the parent cell that are displayed in the form. If the .VC member has a weight, it is included in the distribution, If a cell is not eligible to receive an allocation, the allocation value is distributed to the remaining cells.

You enable automatic allocation on the **Data-Entry** tab of the table properties.

For more information about automatic allocation, see the *SAS Financial Management: User's Guide*.

Including Tables of Other Types in Financial Forms

For a financial form set, you can include read-only tables or CDA tables to display numeric values that can guide the data-entry process.

For a financial form set, you can include a SAS Human Capital Management supplemental schedule that feeds a data-entry table, provided that you have SAS Human Capital Management installed at your site. This arrangement facilitates salary planning by retrieving salary data for employees from the SAS Human Capital Management database.

It is likely that a single SAS Human Capital Management supplemental schedule that feeds a single data-entry table will meet your salary-planning needs. However, it is possible to create two or more supplemental schedules that feed the same data-entry table or different data-entry tables.

If custom work has been done at your site to support other data providers for supplemental schedules, then you can add other types of supplemental schedules.

In a form template that includes more than one table, you must stack the tables vertically. Do not place two tables side by side.

In addition to SAS Financial Management tables, you can include any number of ordinary Microsoft Excel structures and make use of all the capabilities of Microsoft Excel. For example, a form template whose focus is a data-entry table can also include supplementary work areas that compute values that are needed in the data-entry table. A cell in the data-entry table can contain a Microsoft Excel formula that refers to a cell in the supplementary work area.

To see all the Microsoft Excel formulas that a data-entry table contains, select **Table Properties**, and then select the **Calculated Values** tab of the Table Properties window.

Creating a Supplemental Schedule

If SAS Human Capital Management is installed at your site, then you can insert a SAS Human Capital Management supplemental schedule into a template for a financial form set. A SAS Human Capital Management supplemental schedule can do all of the following:

- retrieve data from the SAS Human Capital Management database, such as salary data for individual employees
- receive numeric values that are entered by users
- calculate numeric values from other numeric values in the supplemental schedule, using calculated-member functions
- feed numeric values into the associated data-entry table
- save the numeric values that are displayed in the supplemental schedule to the SAS Human Capital Management database

If a custom data provider has been implemented at your site, then you can insert a custom supplemental schedule into a template for a financial form set. A custom supplemental schedule can do all of the following:

- retrieve data from the database that is served by the custom data provider
- receive numeric values that are entered by users
- calculate numeric values from other numeric values in the supplemental schedule, using calculated-member functions
- feed numeric values into the associated data-entry table
- save the numeric values that are displayed in the supplemental schedule to the database that is served by the custom data provider

To insert a supplemental schedule:

1. Create the data-entry table that the supplemental schedule will feed. Make sure that the data-entry table includes all the crossings that the supplemental schedule must feed. In the case of a SAS Human Capital Management supplemental schedule, this means that the data-entry table must include an organization hierarchy that is used by SAS Human Capital Management.

The option that inserts a supplemental schedule is not active until you create a data-entry table.

2. Below the data-entry table, select the cell that will be the upper left corner of the supplemental schedule.
3. Select **Supplemental Schedule** to launch the Create Supplemental Schedule wizard.
4. Work through the wizard, consulting the online Help for individual wizard pages as necessary.

   When you finish the wizard, the supplemental schedule appears. If the form template contains only one data-entry table, then the new supplemental schedule is automatically associated with that data-entry table. If the form template contains more than one data-entry table, then the wizard presents a page that enables you to select the data-entry table to associate the supplemental schedule with.

   If the associated data-entry table has slicers, then its slicers also apply to the supplemental schedule. However, by default the data-entry table's slicers are not displayed as part of the supplemental schedule. To change the way a supplemental schedule's slicers are displayed, select **Table Properties** and use the **Hide Slicers** region on the **General** tab.

Entering Data as Part of a Bottom-Up Workflow

Overview

In a SAS Financial Management data-entry table, the cells that you can enter data into are yellow. Enter data into the yellow cells in the same way that you enter data into any ordinary Microsoft Excel worksheet cell.

In general, the cells that are yellow belong to the organization that you are responsible for. Another participant in the same workflow might see the same data-entry table with a different set of cells shown in yellow.

If you are entering data for an organization that is not a leaf member in the organization hierarchy, then you must use the virtual child of that organization (marked **vc** and subordinate to the organization). The values for the organization itself are computed from the values for all its children, including its virtual child.

If the **Automatically Refresh Tables** option is set, then the numeric values in functionally related cells are automatically updated as you enter data into the yellow cells. Otherwise, you must select **Refresh** or **Refresh All** to update the values in functionally related cells.

In yellow cells, you can also place Microsoft Excel formulas that refer to cells that are outside the data-entry table. This enables you to create supplementary work areas that feed the data-entry table. To see all the Microsoft Excel formulas that a data-entry table contains, select **Table Properties**, and then select the **Calculated Values** tab of the Table Properties window.

A form can include a supplemental schedule.

See Also

- Entering Data into a Supplemental Schedule on page 224
- Entering Data Offline on page 223

Entering Data into Parent Cells

Note: These options apply only to financial forms in a bottom-up workflow.

In most cases, a parent cell is gray, indicating that its value is derived from the values of its subordinate cells. (A gray cell might also indicate that its value comes from a calculation.)

If a parent cell is yellow, it is writable. When you enter data into a writable parent cell, the value is distributed in one of two ways:

- It is written to the virtual child (vc) cell for the parent. This behavior occurs if **Allow data to be entered for parent members** is selected in the table properties.
- It is automatically allocated to subordinate cells, according to a weight table that is defined by administrators. This behavior can occur if **Allocate using predefined weights** is selected in the table properties.

  To view the weight table that determines the allocation, right-click anywhere in the table and select **Edit** ⇨ **Allocation Weights**.

Note: These options are mutually exclusive. They can be enabled or disabled only by an administrator of the form template.

Reviewing Data as Part of a Bottom-Up Workflow

Reviewing the data in a form is basically the same task as viewing a financial report.

See Also

Viewing a Financial Report on page 213

Entering Data as Part of a Top-Down Workflow

In a data-entry table, the cells that you can enter data into are yellow. Enter data into the yellow cells in the same way that you enter data into any ordinary Microsoft Excel worksheet cell.

In general, the cells that are yellow belong to the organization that you are responsible for. Another participant in the same workflow might see the same data-entry table with a different set of cells shown in yellow.

A user who enters data for the top member in a top-down workflow has more latitude in entering data than a user who enters data for any other member. At the top, you must enter all the amounts that will cascade down the target hierarchy for the workflow. To do this, use the virtual child of the top member. At any lower level, you receive certain amounts that have been pushed down to you, and your role is limited to allocating and distributing the amounts that you receive.

At any level, you can allocate amounts to lower levels manually or you can select **Allocate** to perform even or proportional allocations.

If the **Automatically Refresh Tables** option is set, then the numeric values in functionally related cells are automatically updated as you enter data into the yellow cells. Otherwise, you must select **Refresh** or **Refresh All** to update the values in functionally related cells.

In yellow cells, you can also place Microsoft Excel formulas that refer to cells that are outside the data-entry table. This enables you to create supplementary work areas that feed the data-entry table. To see all the Microsoft Excel formulas that a data-entry table contains, select **Table Properties**, and then select the **Calculated Values** tab of the Table Properties window.

A form can include a supplemental schedule.

See Also

- Allocate on page 251
- Entering Data into a Supplemental Schedule on page 224
- Entering Data Offline on page 223

Entering Data Offline

Normally, you enter data into a form while the SAS Financial Management Add-In for Microsoft Excel is connected to the SAS Financial Management server. The data that you enter is saved in the SAS Financial Management database as you enter it. Cells whose values depend on the data that you enter are updated, either automatically or when you select **Refresh** or **Refresh All**.

You can also enter data into a form while the SAS Financial Management Add-In for Microsoft Excel is not connected to the SAS Financial Management server. To make this possible, check out the form and save it as a local Excel file. Each time you enter data, you must save it in the local file. Cells whose values depend on the data that you enter are not updated as you work. Eventually, you check the form in again, reconnecting it to the server. At that point, all the data that you entered offline is saved in the SAS Financial Management database, and all cells whose values depend on the data that you entered offline are automatically updated.

You cannot check out a form that is already checked out. You cannot check out a form that contains a supplemental schedule.

To check out a form for offline data-entry:

1. Open the form from the Form Manager as if you were going to enter data online.
2. Select **Check Out Form**.

   The Check Out Form window appears.
3. In the Check Out Form window, click **Yes**.

   The Save As window appears.
4. In the Save As window, specify the location where you want to save the file, and then click **Save**.

To check in an offline form:

1. Open your local copy of the file.
2. Select **Check In Form**.

The SAS Log On window appears.

3. In the SAS Log On window, specify the correct environment, user name, and password, and then click **OK**.

   The Check In Form window appears.

4. In the Check In Form window, click **Yes**.

   The SAS Financial Management Add-In for Microsoft Excel connects to the SAS Financial Management server. All the data in the local copy of the form is uploaded to the SAS Financial Management database.

Entering Data into a Supplemental Schedule

A form can include a supplemental schedule. Each supplemental schedule is associated with a data-entry table and a data provider. The associated data provider retrieves data from an external database for display in the supplemental schedule. Numeric values that are computed in the supplemental schedule can feed the associated data-entry table. The data provider can also save data that is displayed in the supplemental schedule back to the external database. For example, the SAS Human Capital Management data provider is a two-way data bridge between a SAS Human Capital Management supplemental schedule and the SAS Human Capital Management database.

You can recognize a supplemental schedule by its last row, which displays column totals in green cells.

Entering data into a supplemental schedule is like entering data into a data-entry table. The cells that you can enter data into are yellow. In addition to entering data into existing rows, you can add new rows. To add new rows to a SAS Human Capital Management supplemental schedule, use the **New Position**, **Repeat New Position**, **Delete Position**, and **Change Description** options.

After you enter data into a supplemental schedule, you must select **Save All Supplemental Data**. This important option does all of the following for a selected supplemental schedule:

- It feeds numeric values from the totals row of the supplemental schedule into the associated data-entry table.
- It saves the data-entry table values to the SAS Financial Management database.
- It saves values from the supplemental schedule back to the external database that is served by the data provider.

Note: When you enter a number at roll-up via a supplemental table, the total from the supplemental table might not match the value in the FM table, because the number in the FM table includes any contributing values from its descendants.

Data that is saved to the external database might not be qualified by all the slicer members that qualify the supplemental schedule. No data provider retains Source, Trader, or Frequency members in the externally saved data. In addition, there can be other dimensions whose members are not saved by a particular data provider. For example, the SAS Human Capital Management data provider does not save currency members to the SAS Human Capital Management database.

See Also

- Entering Data as Part of a Bottom-Up Workflow on page 221

- Entering Data as Part of a Top-Down Workflow on page 222
- New Position on page 262
- Repeat New Position on page 265
- Delete Position on page 258
- Change Description on page 254
- Save All Supplemental Data on page 266

Chapter 19

Creating a Table with the Table Wizard

Model Page **227**

Table Layout Page **228**
Overview of the Table Layout Page 228
Give a Dimension Type or a Member Property a Role in a Table 228
Remove a Dimension Type or a Member Property from a Table 228
Change the Display Positions of Dimension Types and Member Properties 229
Selections for a Data-Entry Table That Is Fed by an HCM Supplemental Schedule 229

Members Page **229**
Overview of the Members Page 229
Time Dimension Type 230
Multiple-Level Hierarchies 230
Flat Hierarchies or Member Properties 231
Target Members of the Workflow 232

Selection Summary Page **233**

Selection Rules **233**

Model Page

Use the Model page to select the model that the table will be based on. For a read-only table or a CDA table, the model provides a pool of base data along with the hierarchies, driver rates, exchange rates, adjustments, and adjustment rules that are needed to generate displayed results from the base data. For a data-entry table, the model provides everything that is needed to generate the data that is stored from the data that users enter.

To select a model, click it in the displayed list of models. The selected model is highlighted.

To sort the list of models by the text in a certain column, click the column heading. Click the same column heading again to reverse the sort order.

Table Layout Page

Overview of the Table Layout Page

Use the Table Layout page to specify the basic layout of the table that you are creating. After you create the table, you can change its basic layout using either drag-and-drop techniques or the **Pivot** option. The **Pivot** option displays the Pivot window, which looks and works like this wizard page.

See Also

Table Layout Principles on page 237

Give a Dimension Type or a Member Property a Role in a Table

To give a dimension type or a member property a role in a table:

1. Click the dimension type or member property in the **Available** region.

   There are member properties that belong to the Account, Organization, and Time dimension types. To display the member properties that belong to one of these dimension types, click the + sign next to the dimension type. You cannot give a role to a member property until you have given a role to the dimension type that the member property belongs to.

   The item that you click is highlighted.

2. Click the + arrow that points to the appropriate region on the right.

   The selected item is added to the appropriate region. It is also bolded in the **Available** region to show that it is being used.

Alternatively, you can use the drag-and-drop technique:

1. Click the dimension type or member property in the **Available** region and keep the mouse button down.
2. With the mouse button down, move the cursor to the appropriate region on the right.
3. Release the mouse button.

Remove a Dimension Type or a Member Property from a Table

To remove a dimension type or a member property from a table:

1. Click the dimension type or member property in the region where it currently resides.

   The item that you click is highlighted.

2. Click the – arrow that points from the highlighted item back to the **Available** region.

   The highlighted item is removed from the region on the right. It is also unbolded in the **Available** region to show that it is not being used.

Alternatively, you can use the drag-and-drop technique:

1. Click the dimension type or member property in the region where it currently resides and keep the mouse button down.
2. With the mouse button down, move the cursor to the **Available** region.
3. Release the mouse button.

You can also use the drag-and-drop technique to move a dimension type or member property directly from one role to another. For example, you can drag-and-drop a dimension type from the **Columns** region to the **Rows** region.

Change the Display Positions of Dimension Types and Member Properties

If two or more items are in the same region on the right, then the order in which they are listed in that region determines the order in which they are displayed in the table. For column headings and slicers, the list order on this wizard page becomes the top-to-bottom display order in the table. For row headings, the list order on this wizard page becomes the left-to-right display order in the table.

To change the list order of the items in any region on the right, use the up and down arrow buttons that are to the right of the region. To move an item up one position in the list, select the item and click the up arrow button. To move an item down one position in the list, select the item and click the down arrow button.

Selections for a Data-Entry Table That Is Fed by an HCM Supplemental Schedule

If you are creating a data-entry table that will be fed by an HCM supplemental schedule, then make the following selections:

- One column dimension: Account
- One row dimension: Organization
- Two slicer dimensions: Time and Analysis

Members Page

Overview of the Members Page

In the Create Table wizard, the Members page appears once for each dimension type or member property that has a role in the table. See Using Member Properties as Filters, Groupers, or Labels on page 239.

The fields on this page depend on the dimension type or contents:

- multiple-level hierarchies, excluding the Time dimension type and the target hierarchy (see "Multiple-Level Hierarchies" on page 230)
- Time dimension type (see "Time Dimension Type" on page 230)
- dimension types with flat hierarchies, as well as member properties (see "Flat Hierarchies or Member Properties" on page 231)
- members of the target hierarchy (see "Target Members of the Workflow" on page 232)

Time Dimension Type

To select the entire time hierarchy, select the **All members** radio button. The table includes the entire time hierarchy, even if the hierarchy changes after you create the table.

Alternatively, you can use selection rules to select one or more time period ranges. For a data-entry table, you can define a floating time member rule or a fixed time member rule. For a read-only table, you can define multiple floating time member rules, in addition to or in place of one fixed time member rule. The preview region of the page lists the currently selected time periods.

Follow these steps:

1. Select the **Assign selection rules** radio button. This radio button activates the rest of the page
2. Click **Add**.
3. From the pop-up menu, select **Time Member Rule** or **Floating Time Member Rule**.
4. In the window that appears, define the rule.

Make the following additional selections as necessary:

- To modify a rule, select the rule and click **Edit**.
- To delete a rule, select the rule and click **Remove**.
- Select from the **Labeling method** drop-down list to change the way time periods are identified in the table. Time periods can be identified by code, name, description, or combinations of these values.

  Alternatively, right-click anywhere in the display region and select a labeling method from the pop-up menu.
- If you select the **Show parents after children** check box, parent members are displayed to the right of child members in a column, or below child members in a row.
- To change the order of the rules, use the arrows next to the **Add** button. The preview region shows the new time period order.

Note: For the Time dimension type in a form template that includes an HCM supplemental schedule, you must select exactly one time member. It must be a leaf time period of the year that includes the HCM salary data. For example, if the data-entry form will include HCM salary data for the year 2010 and your leaf time periods are months, then you might select the January 2010 time period or the December 2010 time period.

Multiple-Level Hierarchies

For multiple-level hierarchies, you can select all available members, or you can define rules to select a subset of members. The preview region of the page lists the currently selected members.

Note: These instructions do not apply to the Time dimension type (see "Time Dimension Type" on page 230) or to the target hierarchy in a form template (see "Target Members of the Workflow" on page 232).

To select the entire hierarchy, select the **All members** radio button. The table includes the entire hierarchy, even if the hierarchy changes after you create the table.

To select a subset of the hierarchy, follow these steps:

1. Select the **Assign selection rules** radio button This activates the hierarchy display below the radio buttons.
2. Select a check box in the hierarchy display to select a member.
   - Use the plus and minus buttons to expand or collapse the hierarchy.
   - To find a member, click the Find button  and enter a search string in the Find window. You can search only for text that is currently displayed, as determined by the current selection in the **Labeling method** field.
3. From the **Selection rule** drop-down list, select a rule to apply to that member. For details, see "Selection Rules" on page 233.

   The members in the defined subset are added to the preview region.

Repeat steps 2–3 as necessary.

Make the following additional selections as necessary:

- **Include virtual children:** If you select this check box, all virtual children that satisfy your selection criteria are displayed in the preview region and are included in the table. See "Selection Rules" on page 233 for the effect of this check box on the results of different selection rules.
- **Labeling method:** To specify the way members are identified in the table, select from this drop-down list. Members can be identified by code, name, description, or combinations of these values. Alternatively, right-click anywhere in the display region and select a labeling method from the pop-up menu.
- **Show parents after children:** If you select this check box, parent members are displayed to the right of child members in a column, or below child members in a row.
- **Show slicer members list:** This check box is present if you are working on a slicer dimension of a CDA table. Select this option if you want users to be able to select different members for this slicer dimension in the CDA table.

  By default, the check box is deselected, and this slicer dimension is permanently set to the member that you select on this page. (If you selected more than one member, this slicer dimension is permanently set to the first member in the list.)

To modify the table's initial expansion state: In the preview region, expand and collapse the hierarchy to specify its initial state in the table. This feature applies only to rows and columns of read-only and data-entry tables.

To remove a member from the table:

- Clear the check box for a member to remove the member from the table hierarchy.
- If the member inherited its rule from a parent, select the member and select **Exclude** from the **Selection rules** drop-down list.

Either of these actions removes the member that is associated with the check box. It also removes all subordinate members, unless they have their own rules.

To undo all selections: Click the Clear all button  .

Flat Hierarchies or Member Properties

For flat hierarchies or for member properties, you can select all available members or property values, or you can define rules to select a subset of members or values. The preview region of the page displays the results.

To include all members of the flat hierarchy or all possible values of the member property, select the **All members** radio button. For a hierarchy, this means that the table always includes the entire hierarchy, even if the hierarchy changes after you create the table.

To define a subset of members or values, follow these steps:

1. Select the **Selected members** radio button.
2. To include a member or property value, select its check box.

Make the following additional selections as necessary:

- **Labeling method:** To specify the way members are identified in the table, select from this drop-down list. Members can be identified by code, name, description, or combinations of these values. Alternatively, right-click anywhere in the display region and select a labeling method from the pop-up menu.
- **Show slicer members list:** This check box is present if you are working on a slicer dimension of a CDA table. Select this check box if you want to be able to select different members for this slicer dimension in the CDA table.

  By default, the check box is deselected, and this slicer dimension is permanently set to the member that you select on this page. (If you selected more than one member, this slicer dimension is permanently set to the first member in the list.)

To find a member or property value, click the Find button  and enter a search string in the Find window. You can search only for text that is currently displayed, as determined by the current selection in the **Labeling method** field.

To select all current members or property values, click the Check all button  . For a hierarchy, this selection includes every member that is currently in the hierarchy, but not members that are added to the hierarchy after you create the table. (To incorporate possible future changes to the hierarchy, use the **All members** radio button instead.)

To deselect all members or property values, click the Clear all button  .

Target Members of the Workflow

In a form template, the **Preview and expansion state** region shows target hierarchy members as they will appear in the data-entry table as a result of the following selections:

- **Labelling method:** Select from this drop-down list to specify the way members are identified in the table: Members can be identified by code, name, description, or combinations of these values.

  Alternatively, right-click anywhere in the display region and select a labeling method from the pop-up menu.
- **Include virtual children:** If you select this check box, all virtual children that satisfy your selection criteria are displayed in the preview region and are included in the table. See "Selection Rules" on page 233 for the effect of this check box on the results of different selection rules.
- **Show parents after children:** If you select this check box, parent members are displayed to the right of child members in a column, or below child members in a row.

Note: You can affect the appearance of the target hierarchy, but you cannot apply selection rules. The member selection rules for the target hierarchy are a property of the form set, not the template.

Selection Summary Page

The Selection Summary page lists the specifications that you have just made, enabling you to review them. If you notice a specification that you want to change, click **Back** as many times as necessary to return to the page where you can change it. If everything is in order, click **Finish**.

Selection Rules

The **Selection rule** drop-down list provides the following choices:


Member
the designated member.


Member and its children
the designated member and all members that are immediately subordinate to it.

If **Include virtual children** is also selected, the member's virtual child is included.


Member and its descendants
includes the entire sub-hierarchy from the designated member down.

If **Include virtual children** is also selected, all virtual children that are descendants of the designated member are included.


Member and its leaf descendants
includes the designated member and all members that are subordinate to it but have no members subordinate to them.

If **Include virtual children** is also selected, all virtual children that are descendants of the designated member are included.


Children of member
includes all members that are immediately subordinate to the designated member, but not the member itself.

If **Include virtual children** is also selected, the member's virtual child is included.


Descendants of member
includes the entire sub-hierarchy that is headed by the designated member, but not including the designated member itself.

If **Include virtual children** is also selected, all virtual children that are descendants of the designated member are included.


Leaf descendants of member
includes all members that are subordinate to the designated member but have no members subordinate to them.

If **Include virtual children** is also selected, all virtual children that are descendants of the designated member are included.

Exclude
excludes the designated member from the subset. All members that are subordinate to the designated member are also excluded, unless you apply additional rules to one or more of these subordinate members.

No rule
applies no rule to the designated member, although the member might inherit a rule from a parent.

Chapter 20

Working with Tables

Creating Tables **236**

Deleting Tables **237**

Table Layout Principles **237**
Overview of Table Layout Principles 237
User Interface for Defining a Table Layout 237
Dimension Types and Member Properties 238
Column Headings 238
Row Headings 238
Slicers 238

Using Member Properties as Filters, Groupers, or Labels **239**

Using the Table Pop-Up Menu **240**

Changing the View Given by a Table **240**

Pivoting a Table **241**
Overview of Pivoting 241
Using the Pivot Option 241
Drag-and-Drop Pivoting 241
Removing Dimensions with the DELETE Key 242

Using Calculated Members **243**

Using the Frequency Dimension **243**

Associating Comments with Table Cells **244**

Formatting Table Cells **244**
Specifying Cell Formats 244
Resolving Conflicts between Format Specifications 245

Coordinating Slicers between Tables **245**

Exporting Values to a Metric Table **246**

Converting a Read-Only Table to a CDA Table **246**

Converting a CDA Table to a Read-Only Table **247**

Protecting Cells in a Data-Entry Table **247**

Creating Tables

To create a SAS Financial Management table:

1. Select a location for the new table by selecting the cell that will be the table's upper-left corner.
2. Select the appropriate table option and work through the wizard that the option launches, consulting the online Help for the individual wizard pages as necessary.

There are four table options:

- **Insert ⇨ Data-entry Table** creates a data-entry table, which is the heart of a data-entry form.

  This option is active only when you launch the Excel add-in from the Forms workspace of SAS Financial Management Studio.

- **Insert ⇨ Read-only Table** creates a read-only table, which displays numeric values that are computed from data in the SAS Financial Management database. A read-only table can be expanded, collapsed, pivoted, sliced, and manipulated in other ways. A read-only table can be either the centerpiece of a report or a supporting table in a financial data-entry form.

  This option is active when you launch the Excel add-in from your desktop. It is also active when you launch the Excel add-in from the Forms workspace of SAS Financial Management Studio, if you are working on a financial form set.

- **Insert ⇨ CDA Table** creates a CDA table, which displays numeric values that are computed from data in the SAS Financial Management database. A CDA table cannot be expanded, collapsed, or pivoted, but it can be sliced. Its cells can be formatted in all the ways that are supported by Microsoft Excel. A CDA table can be either the centerpiece of a report or a supporting table in a financial data-entry form.

  This option is active when you launch the Excel add-in from your desktop. It is also active when you launch the Excel add-in from the Forms workspace of SAS Financial Management Studio, if you are working on a financial form set.

- **Insert ⇨ Supplemental Schedule** creates a supplemental schedule, which displays numeric values that are imported from a data provider and feeds the imported numeric values into a data-entry table. For example, if you have SAS Human Capital Management, then you can create a SAS Human Capital Management supplemental schedule that imports salary data from SAS Human Capital Management and feeds it into a data-entry table.

  This option is active only when you launch the Excel add-in from the Forms workspace of SAS Financial Management Studio in order to work on the template for a financial form set, and only after you have created a data-entry table in the template.

If you create more than one SAS Financial Management table in the same worksheet, put the tables one below the other in a single vertical stack. Do not put any other information in any row that is used by a SAS Financial Management table. A row that runs through a SAS Financial Management table can behave in unpredictable ways outside the table.

See Also

- Designing a Financial Report on page 211

- Designing a Data-Entry Form Template on page 214

Deleting Tables

If you are using the Excel add-in to design a report or a form template, then you can delete any table that you create.

To delete a table, use the Microsoft Excel **Delete** option to delete a range of cells that includes the table:

1. Select a range of cells that completely includes the table that you want to delete.
2. With the cursor anywhere inside the selected range of cells, right-click to display the pop-up menu.
3. Select **Delete** from the pop-up menu.
4. In the Delete window, select **Entire row**, and then click **OK**.

   This instruction assumes that you have followed the recommendation to put no other information in any row that is used by a SAS Financial Management table.

Table Layout Principles

Overview of Table Layout Principles

To define the layout of a new table, use the Table Layout page of the Create Table wizard. To change the layout of an existing table, you can either use drag-and-drop techniques or select **Pivot** and use the Pivot window. The Table Layout page and the Pivot window work in the same way, as described here.

See Also

Pivoting a Table on page 241

User Interface for Defining a Table Layout

The **Available** region lists all the dimension types that have hierarchies in the governing model. Three of these dimension types—Account, Organization, and Time—have plus signs next to them. When you click a plus sign next to a dimension type, a list of properties of the members of that dimension type is displayed.

The three regions on the right—**Columns**, **Rows**, and **Slicers**—represent, respectively, a table's column headings, row headings, and slicers. You must use at least one dimension type as a source of column headings and at least one dimension type as a source of row headings. You can also use dimension types to provide slicers, additional column headings, and additional row headings. Using a dimension type in the table makes the members of the model hierarchy for that dimension type available for display in the table.

Dimension Types and Member Properties

If you use the Account, Organization, or Time dimension type, then you can also use any listed property of the members of that dimension type. Using a member property in a table makes the possible values of that property available for display in the table.

The role that one of these dimension types plays in a table determines how its member properties can be used.

See Also

Using Member Properties in Tables on page 239

Column Headings

If you use only a single dimension type as a source of column headings, then the table columns have simple headings that identify members of that dimension type. If you use two or more sources of column headings—some combination of dimension types and member properties—then the table columns have nested headings. The top-to-bottom order in which dimension types and member properties are listed in the **Columns** region determines the top-to-bottom order of the nested column headings in the table.

Row Headings

If you use only a single dimension type as a source of row headings, then the table rows have simple headings that identify members of that dimension type. If you use two or more sources of row headings—some combination of dimension types and member properties—then the table rows have nested headings. The top-to-bottom order in which dimension types and member properties are listed in the **Rows** region determines the left-to-right order of the nested row headings in the table.

Slicers

The dimension types and member properties that you use as slicers appear above the rectangular grid of rows and columns. Each slicer includes a field that displays the currently selected member or property value for that slicer. In general, it is possible to select different members or property values for a slicer. However, you can define static slicers that are permanently set to a fixed selection.

By selecting different combinations of values for a table's slicers, you can display different slices of data. The displayed slice of data is the slice that is associated with the currently selected values of the slicers, together with the table default read members of any dimension types that are not used in the table.

A table that has no slicers can be used to view only one slice of data. In this case, the displayed slice is the slice that is associated with all the table default read members of all the dimension types that are not used in the table.

After you create a read-only table or a data-entry table, you can change the table default read member of any unused dimension type with the **Table Properties** option.

Using Member Properties as Filters, Groupers, or Labels

At a minimum, there are three dimension types for which a table can use member properties:

- A table that uses the Account dimension type can also use properties of accounts.
- A table that uses the Organization dimension type can also use properties of organizations.
- A table that uses the Time dimension type can also use properties of time periods.

If custom properties have been defined for the members of other dimension types, then a table can also use member properties of those other dimension types.

The member properties that are available for use in tables are the properties whose values can be returned by the CDAProperty function. The property values that can appear in tables are the values that this function can return.

Member properties are available for inclusion in a table layout whenever you use the Create Table wizard to create a table and whenever you use **Pivot** to change the layout of a table. You can add a given member property to a table only if the table already includes the dimension type to which the property applies. In addition, the role that a member property can play in a table depends on the role that the corresponding dimension type plays. The dependencies are as follows:

- A member property can be a source of row headings only if the companion dimension type is also a source of row headings.
- A member property can be a source of column headings only if the companion dimension type is also a source of column headings.
- A member property can be a slicer only if the companion dimension type is a source of row headings or a source of column headings or a slicer.

Member properties can enhance tables in three ways. A member property can serve as a filter, as a grouper, or as a label, depending on how it is positioned relative to the corresponding dimension type:

Filter
: Make the corresponding dimension type either a source of row headings or a source of column headings. Make the member property a slicer. You can then use the slicer to restrict the table to various subsets of the members of the corresponding dimension type.

  For example, make the Account dimension type a source of row headings and make the Account Type property a slicer. Selecting a particular account type in the slicer then restricts the table rows to accounts that belong to the selected account type. If you select Asset in the slicer, only asset accounts are displayed. If you select Revenue in the slicer, only revenue accounts are displayed.

Grouper
: Make the corresponding dimension type and the member property both sources of row headings or both sources of column headings, and nest the corresponding dimension type inside the member property. The members of the corresponding dimension type are then grouped by the values of the member property. The members are displayed without any hierarchical structure because members can be separated from their parents and children when they are grouped in this way.

For example, make the Account dimension type and the Account Type property both sources of row headings, and nest the Account dimension type inside the Account Type property. The accounts are then grouped by account type. All the asset accounts are grouped together within the Asset value of the Account Type property. All the revenue accounts are grouped together within the Revenue value of the Account Type property.

Label
Make the corresponding dimension type and the member property both sources of row headings or both sources of column headings, and nest the member property inside the corresponding dimension type. The members of the dimension type are then labeled, but not grouped, by the adjacent values of the member property.

For example, make the Account dimension type and the Account Type property both sources of row headings, and nest the Account Type property inside the Account dimension type. Each account is labeled by the adjacent account type. Each asset account has the label "Asset" next to it. Each revenue account has the label "Revenue" next to it.

See Also

CDAProperty on page 275

Using the Table Pop-Up Menu

If you select any cell in a Microsoft Excel spreadsheet and click the right mouse button, a pop-up menu appears. If the selected cell is in a SAS Financial Management table, then the pop-up menu includes additional options that are provided by SAS Financial Management. The set of additional options depends on the type of table that you are working with and the type of cell that you select within the table.

Every option that is on the table pop-up menu is also on one of the menus above the workbook display. The table pop-up menu gives you an alternative way to access the options that manipulate existing tables. The dictionary of menu options includes an entry for every menu option. Each entry specifies all the ways in which the option can be accessed.

Changing the View Given by a Table

Here are some ways in which you can change the set of crossings whose values are displayed in a table:

- Double-click any underlined row heading or column heading. The underlined headings are non-leaf members of the hierarchy that they belong to. Double-clicking an underlined heading expands or collapses the portion of the hierarchy that is subordinate to it.
- Select a **Drill**, **Collapse**, or **Expand** option.
- Select a different member of a slicer dimension, if the table has slicer dimensions. This switches the display to a slice of values that is associated with the newly selected slicer member.
- Pivot the table, using either the **Pivot** option or drag-and-drop techniques.

- Select a dimension type by selecting one of its members and then use the **Show Members** option to define a different subset of the members of that dimension type to include in the table.
- Use the **Filter Member Combination** and **Filters** options to suppress and restore the display of certain rows or columns.
- Use the **Table Properties** option to set a different default member for a dimension that is not part of the table layout, or to switch to a different model.

See Also

Pivoting a Table on page 241

Pivoting a Table

Overview of Pivoting

Any change in the role that any dimension plays in a table is a case of pivoting the table. Pivoting a table includes all of the following:

- adding a row dimension, column dimension, or slicer dimension
- removing a row dimension, column dimension, or slicer dimension
- moving a dimension from one part of the table to another: from row to column or slicer, from column to row or slicer, from slicer to row or column
- changing the display order of the slicer dimensions
- changing the nesting order of the row dimensions or the column dimensions

If you are working with a read-only table or designing a data-entry table, then the full set of pivoting possibilities is available. If you are editing or reviewing a data-entry form, then you can change the roles that dimensions play in a data-entry table but you cannot add or remove dimensions.

Using the Pivot Option

You can do every available type of pivoting with the **Pivot** option, which displays the Pivot window.

Drag-and-Drop Pivoting

You can do many types of pivoting by dragging-and-dropping a table cell onto a target cell.

To perform a drag-and-drop pivot operation, do the following:

1. Select the dimension that you want to drag by clicking any member cell of the dimension. For a slicer dimension, you can also click the cell that holds the name of the dimension.
2. Without pressing any mouse button, move the cursor to the border of the selected cell.

   The four-arrow symbol appears. Make sure that this symbol is visible before you continue.
3. Press the left mouse button, hold it down, and drag the selected cell to a target cell.

4. When the selected cell coincides with the target cell, release the mouse button.

   A Microsoft Office Excel window with the following message appears: “Do you want to replace the contents of the destination cells?”

5. In response to this question, click **OK**.

   The role of the dragged dimension changes as specified by the following table.

Target Cell	Resulting Role of the Dragged Dimension
slicer cell	slicer dimension immediately before the slicer dimension that contains the target cell
row header cell	row dimension immediately to the left of the row dimension that contains the target cell
data cell adjacent to a row header cell	row dimension adjacent to the data cells
column header cell	column dimension immediately above the column dimension that contains the target cell
data cell adjacent to a column header cell	column dimension adjacent to the data cells
cell outside the table	none (dimension is removed from the table)

Drag-and-drop pivot operations have the following limitations:

- The target table cell must occupy only a single Excel spreadsheet cell. A large row header cell or column header cell that results from nesting two or more row dimensions or column dimensions cannot be the target cell of a drag-and-drop pivot operation.
- If the dragged table cell occupies more than one Excel spreadsheet cell, then you must drop the first spreadsheet cell that is part of the dragged table cell onto the target table cell.
- You cannot use drag-and-drop to add a dimension to a table.
- You cannot use drag-and-drop to pivot a row or column dimension into a slicer dimension if the table does not already have a slicer dimension.
- You cannot use drag-and-drop to pivot a dimension into the last slicer position, after all existing slicers.

Drag-and-drop pivoting can be a convenient alternative for the cases to which it applies. The **Pivot** option is always available and has no limitations.

Removing Dimensions with the DELETE Key

You can remove a dimension from a table in the following way:

1. Select any member of a row or column dimension, or the dimension label of a slicer dimension.
2. Press **DELETE**.

You can remove a row dimension only if the table has two or more row dimensions. You can remove a column dimension only if the table has two or more column dimensions.

Using Calculated Members

You can add calculated members to a read-only table, a data-entry table, or a supplemental schedule. Each calculated member carries a formula that is used to calculate the values that are associated with that calculated member. Calculated-member formulas use the syntax of Microsoft Excel formulas, supplemented as appropriate with the calculated-member functions that are provided by the SAS Financial Management Add-In for Microsoft Excel.

Calculated members in supplemental schedules differ from calculated members in read-only tables and data-entry tables in the following ways:

- In read-only tables and data-entry tables, a calculated member can be inserted between any two members in any dimension. In supplemental schedules, a calculated member can be added only to the measure dimension, where it can either be inserted between two items or substituted for an item.

  By replacing a member in a supplemental schedule's measure dimension with a calculated member that depends on a measure, you can make data in the SAS Financial Management database functionally dependent on data in the database that is served by the supplemental schedule's data provider. Conversely, by replacing a measure in a supplemental schedule's measure dimension with a calculated member that depends on a member, you can make data in the database that is served by the supplemental schedule's data provider functionally dependent on data in the SAS Financial Management database.
- In read-only tables and data-entry tables, a calculated-member formula can use any calculated-member function. In supplemental schedules, the only calculated-member function that a calculated-member formula can use is fmValue.
- In read-only tables and data-entry tables, a calculated-member formula continues to work correctly if a member that it refers to is hidden in the table. In supplemental schedules, a calculated-member formula works correctly only if every member or measure that it refers to is visible in the supplemental schedule.

To add, delete, or edit a calculated member, select **Calculated Members**. In a supplemental schedule, deleting a calculated member that had replaced a member or measure restores the original member or measure.

If you select a calculated member in a row or column heading of a table or a supplemental schedule, the formula that the calculated member carries is displayed.

Using the Frequency Dimension

The frequency dimension provides a supplementary perspective on the dimension of time. It is not a fully independent dimension. The frequency dimension is predefined and cannot be revised. It consists of a single flat set of members.

By combining a given time period with different members of the frequency dimension, you can display different but related numeric values that are associated with that time period. For example, consider a revenue account such as Total Sales and a month such as June 2010. You can combine these with various frequency members as follows:

- Combine them with Period Activity to represent total sales for June 2010. This is the default if you do not include the frequency dimension in your table.

- Combine them with Year To Date to represent total sales for the year 2010 through June 2010.
- Combine them with Quarter To Date to represent total sales for the quarter that includes June 2010, through June 2010.
- Combine them with Life To Date to represent total sales for the time span that is covered by your SAS Financial Management data, through June 2010.

Associating Comments with Table Cells

Microsoft Excel gives you the ability to associate a comment with any spreadsheet cell. However, to save cell comments with your SAS Financial Management tables, you must take an additional step.

To associate a comment with a cell:

1. Select the cell.
2. Select **Insert Comment** from the pop-up menu. A comment box appears.
3. Type the comment in the comment box.
4. Select any other cell.

To save comments that you insert with the **Insert Comment** option, you must take the following additional step:

1. Select a range of cells that includes the cells with comments.
2. Select **Attach Modifications**.

To undo the effect of **Attach Modifications** on a cell, select the cell, and then select **Clear ⇨ Modifications**.

Changes to comments are saved when you do one of the following:

- Select **Save Template** while working on a form template.
- Select **Save Form Design** while editing a form.
- Select the Microsoft **Save** option or publish a report with **Publish** while working on a read-only table.

Formatting Table Cells

Specifying Cell Formats

You can specify Microsoft Excel formatting for cells in a SAS Financial Management table in three different ways:

- Select **Table Properties**. On the **Styles** tab of the Table Properties window, assign formatting styles to functional components of a table such as row headings, column headings, or data cells.
- Select **Format Members**. Use the Format Members window to assign formatting styles to members. A formatting style that is assigned to a member is applied to all the data cells whose crossings contain that member.

- Use the Microsoft **Format Cells** option to apply ad hoc formatting to a selected cell or range of cells.

To save ad hoc formatting that you set with the Microsoft **Format Cells** option, you must take the following additional step:

1. Select a range of cells that includes the formatted cells.
2. Select **Attach Modifications**.

To undo the effect of **Attach Modifications** on a cell, select the cell, and then select **Clear ⇨ Modifications**.

Changes to ad hoc formatting are saved when you do one of the following:

- Select **Save Template** while working on a form template.
- Select **Save Form Design** while editing a form.
- Select the Microsoft **Save** option or publish a report with **Publish** while working on a read-only table.

Resolving Conflicts between Format Specifications

If there are conflicts between formatting that you set in these different ways, then the conflicts are resolved by the following precedence ordering:

- Formatting that is set with the Microsoft **Format Cells** option has the highest precedence.
- Formatting that is set with **Format Members** has intermediate precedence.
- Formatting that is set with **Table Properties** has lowest precedence.

It is also possible to have conflicts within **Format Members** between members of different dimensions. These conflicts are resolved by the **Dimension Precedence** tab of the Format Members window.

Coordinating Slicers between Tables

If a certain hierarchy plays the role of a slicer hierarchy in two or more read-only tables or data-entry tables in the same workbook, then it is possible to connect these slicers in such a way that selecting a slicer member in one table automatically selects the same slicer member in other tables. This is a one-way relationship; a slicer in one table controls a slicer in a second table, but not the reverse.

For example, you can give the organization slicer of table Y control over the organization slicer of table Z. If a user selects Headquarters in the organization slicer of table Y, then Headquarters is automatically selected in the organization slicer of table Z. The organization slicer of table Z does not permit direct selection of organizations; it merely reflects the selections that are made in table Y.

A table can have a mix of controlling slicers and stand-alone slicers or a mix of controlled slicers and stand-alone slicers. However, a table cannot have a mix of controlling slicers and controlled slicers.

To establish a control connection between slicers in different tables:

1. Select any cell in the table that you want to put under the control of another table.
2. Select **Table Properties**.

3. In the Table Properties window, select the **Slicers** tab.
4. Use the **Shared Slicers** region of the **Slicers** tab to define the connection between tables. For details, see the online Help for the Table Properties window.

Exporting Values to a Metric Table

When you export numeric values to a metric table, they become associated with labels that are known as measures, in addition to all the dimension members that they are already associated with. Metric tables are used to supply values to performance scorecards that are displayed in SAS Strategy Management or the KPI Viewer.

To export numeric values from a SAS Financial Management table to a metric table:

1. In the source table, select the cell that represents the values to export. In addition to the value that is currently displayed in the selected cell, you can export values that would appear in the selected cell if you selected different members for slicer dimensions.
2. Select **Export as Measure**.
3. In the Export as Measure window, specify the values to export, the measure to associate the exported values with, and a description for the target table.

Converting a Read-Only Table to a CDA Table

To convert the currently displayed slice of a read-only table to a CDA table:

1. Select the read-only table. To do this, click the extreme upper-left cell of the table. If the table has slicers, then this is the cell that contains the name of the top slicer. If the table does not have slicers, this is the cell at the intersection of the top row of column headings and the leftmost column of row headings.

   The entire table is highlighted when you select it.
2. Select **Copy as CDA**.
3. In the **Table position** field of the Copy as CDA window, type the cell reference that will become the upper-left corner of the CDA table. For example, if you want the upper-left corner of the CDA table to be in cell H24, type **`H24`** in this field.
4. Click **OK**.

   The original read-only table remains in existence. A corresponding CDA table is created at the location that you specified.

You can also select any range of cells within a read-only table and convert the selected range to a CDA table.

Note: The **Copy as CDA** option is intended primarily for tables in which all the row, column, and slicer headings are members. If you apply it to a table that has any row, column, or slicer headings that are the values of member properties, then the resulting CDA table might be imperfect.

Converting a CDA Table to a Read-Only Table

To convert a CDA table to a read-only table:

1. Select the exact range of cells that contains the header of the CDA table. These are all the cells above the column headings that contain general information about the table.
2. Select **Copy as Read-Only Table**.

   The CDA table disappears and a corresponding read-only table is created. The upper-left cell of the new read-only table is A1. The new read-only table includes all the members of its hierarchies, regardless of any member restrictions in the source CDA table.

Protecting Cells in a Data-Entry Table

In a data-entry table, protected cells are displayed in purple and are read-only.

If cells are protected by rules that were defined by an administrator, you cannot undo that protection in an individual form. However, you can add further protection to a form, as follows:

- To protect one or more cells, select the cells and select **Protect Cell**.

  Only writable cells (displayed in yellow) can be protected.
- To undo protection that was set in this form for one or more cells, select the cells and select **Unprotect Cell**.
- To undo all cell protection that was set in this form, click anywhere in the table and select **Unprotect All Cells**.

The changes that you make are saved automatically and apply to any users who open the same form.

Chapter 21

Dictionary of Menu Options

About SAS Financial Management Add-In 251
About SAS Solutions Add-In 251
Adjust Values 251
Allocate 251
Allocation Weights 252
Attach Modifications 252
Automatically Refresh Tables 252
Calculated Members 253
CDA Table 253
Cell Information 253
Change Description 254
Check In Form 254
Check Out Form 254
Clear Modifications 255
Clear Table's Modifications 255
Collapse 255
Contributing Data 256
Convert to CDA 256
Copy as CDA 256
Copy as Read-Only Table 257
Copy as Table 257
Data-Entry Table 257
Delete Position 258
Document 258
Drill Down 258
Drill Up 258
Edit Protection Rules 259
Expand 259

Expand All 259
Export as Measure 259
Filter Member Combination 260
Filters 260
Format Members 261
Hide Member 261
Log Off 261
Log On 262
Member Labels 262
New Position 262
Pivot 263
Prompt for File Format 263
Properties 263
Protect Cell 263
Publish 264
Read-Only Table 264
Reallocate 265
Refresh 265
Refresh All 265
Repeat New Position 265
Repeat Spread 266
SAS Financial Management Add-In Help 266
SAS Solutions Add-In Help 266
Save All Supplemental Data 266
Save Form Design 267
Save Template 267
Show Members 268
Spread 268
Supplemental Schedule 268
Table Properties 269
Unassign 269
Unprotect All Cells 269
Unprotect Cell 270
Validate Web Form 270

About SAS Financial Management Add-In

The **About SAS Financial Management Add-In** option displays information about the SAS Financial Management Add-In for Microsoft Excel.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Help** ⇨ **About SAS Financial Management Add-In**
- Excel 2003: **SAS Solutions** ⇨ **Help** ⇨ **About SAS Financial Management Add-In**

About SAS Solutions Add-In

The **About SAS Solutions Add-In** option displays information about the SAS Solutions Services Add-In for Microsoft Office.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Help** ⇨ **About SAS Solutions Add-In**
- Excel 2003: **SAS Solutions** ⇨ **Help** ⇨ **About SAS Solutions Add-In**

Adjust Values

The **Adjust Values** option displays the Adjust Values window, which enables you to adjust the numeric values in the selected cells by a specified percentage or absolute amount.

This option is active only if one or more yellow cells are selected in a data-entry table.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Adjust Values**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Adjust Values**

Allocate

The **Allocate** option launches the Allocate wizard, which enables you to allocate the value in the selected cell, or a part of that value, to a set of corresponding cells that belong to specified members of the target hierarchy. Corresponding cells represent crossings that differ only in their target hierarchy member.

For detailed information about the Allocate wizard, see the online Help for the individual wizard pages.

This option is active only if a yellow cell is selected in a data-entry table for a form set that has a top-down workflow.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Allocate**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Allocate**

Allocation Weights

The **Allocation Weights** option opens a window for selecting the weights for automatic allocation. These weights determine the way values are distributed when a user enters a value in a parent cell of a data-entry table. It applies only when **Allocate using predefined weights** is selected from the table options.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Allocation Weights**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Allocation Weights**

Attach Modifications

The **Attach Modifications** option attaches the following to the SAS Financial Management table that contains the currently selected cells:

- any comments that you have specified for currently selected cells using the Microsoft Excel **Insert Comment** option
- any format settings that you have specified for currently selected cells using the Microsoft Excel **Format Cells** option

You must attach comments and format settings to the table in order to save them with the table. They are saved when you do one of the following:

- Select **Save Template** while working on a form template.
- Select **Save Form Design** while editing a form.
- Select the Microsoft **Save** option or publish a report with **Publish** while working on a read-only table.

To undo the effect of **Attach Modifications**, use **Clear** ⇨ **Modifications** or **Clear** ⇨ **Table's Modifications**.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Attach Modifications**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Attach Modifications**
- Pop-up menu: **Edit** ⇨ **Attach Modifications**

Automatically Refresh Tables

The **Automatically Refresh Tables** option determines whether the displayed numbers in a data-entry table are automatically refreshed each time a user enters a number. If there is a checkmark next to this option, then the displayed numbers are automatically refreshed. If there is no checkmark next to this option, then the displayed numbers are refreshed only when the user selects **Refresh** or **Refresh All**.

This option is active only if you are designing a data-entry form template for a financial form set. Operational forms can be edited only in the Web-based Form Editor, which has no automatic refresh capability.

Each time you select this option, the adjacent checkmark is either added or removed. The choice that you make applies to all forms that are based on the open form template.

If you are working with a data-entry form, then this option is grayed out and it might or might not have a checkmark next to it. If you are viewing a report, then this option is grayed out and has a checkmark next to it.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Automatically Refresh Tables**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Automatically Refresh Tables**

Calculated Members

The **Calculated Members** option displays the Calculated Members window, which enables you to add calculated members to tables and modify or delete previously added calculated members.

This option is active when you select any cell of a read-only table, a data-entry table, or a supplemental schedule.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Members** ⇨ **Calculated Members**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Calculated Members**
- Pop-up menu: **Members** ⇨ **Calculated Members**

CDA Table

The **CDA Table** option launches the Create CDA Table wizard, which guides you through the process of creating a CDA table. Read-only tables and CDA tables are the central components of financial reports. See Designing a Financial Report on page 211.

For detailed information about the Create CDA Table wizard, see the online Help for the individual wizard pages.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **CDA Table**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **CDA Table**

Cell Information

The **Cell Information** option displays information about the selected cell. If the cell value is the computed value of a formula, then the display includes information about the formula.

This option is active only if a numeric cell is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Cell Information**
- Excel 2003: **SAS Solutions** ⇨ **Tools** ⇨ **Cell Information**
- Pop-up menu: **Tools** ⇨ **Cell Information**

Change Description

The **Change Description** option displays the Change Description window, which enables you to change the row heading of the currently selected row of the currently selected supplemental schedule. In the Change Description window, enter the text of the new description in the **Name** field.

This option is active only if a row heading that has been inserted manually is selected in a supplemental schedule.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (SAS Human Capital Management)** ⇨ **Change Description**
- Excel 2003: **SAS Solutions** ⇨ **Change Description**
- Pop-up menu: **Change Description**

Check In Form

The **Check In Form** option checks in the open form, connecting it to the SAS Financial Management server and updating all server-controlled calculations. After you select this option, you must log on to the server.

This option is active only if the form that you have open is currently checked out.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Check In Form**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Check In Form**

Check Out Form

The **Check Out Form** option checks out and closes the open form, saving it to a specified location on your desktop. A checked-out form is disconnected from the SAS Financial Management server. You can enter data into it without seeing the results of any server-controlled calculations.

This option is active only if the form that you have open is not currently checked out. You cannot check out a form that contains a supplemental schedule.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Check Out Form**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Check Out Form**

Clear Modifications

The **Clear Modifications** option removes the following from all cells in the selected range:

- any comments that you have specified using the Microsoft Excel **Insert Comment** option
- any format settings that you have specified using the Microsoft Excel **Format Cells** option

Clear Modifications undoes the effect of **Attach Modifications** for the selected cells.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Clear** ⇨ **Modifications**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Clear** ⇨ **Modifications**
- Pop-up menu: **Edit** ⇨ **Clear** ⇨ **Modifications**

Clear Table's Modifications

The **Clear** ⇨ **Table's Modifications** option removes the following from all cells in the selected table:

- any comments that you have specified using the Microsoft Excel **Insert Comment** option
- any format settings that you have specified using the Microsoft Excel **Format Cells** option

Clear ⇨ **Table's Modifications** undoes the effect of **Attach Modifications** for the entire table.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Clear** ⇨ **Table's Modifications**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Clear** ⇨ **Table's Modifications**
- Pop-up menu: **Edit** ⇨ **Clear** ⇨ **Table's Modifications**

Collapse

The **Collapse** option hides all the descendants of the selected row or column heading.

This option is active only if a row or column heading that is not currently collapsed is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Navigate** ⇨ **Collapse**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Collapse**
- Pop-up menu: **Collapse**

Contributing Data

The **Contributing Data** option displays, in a separate Microsoft Excel workbook, the stored data that was used to compute the value in the selected cell.

If the display of contributing value contains values that are summarized versions of data that was loaded from an operational cycle, then the display has a **Details** column with buttons that you can click to display the fully detailed operational data.

This option is active only if a numeric cell is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Contributing Data**
- Excel 2003: **SAS Solutions** ⇨ **Tools** ⇨ **Contributing Data**
- Pop-up menu: **Tools** ⇨ **Contributing Data**

Convert to CDA

The **Convert to CDA** option converts the Microsoft Excel formula in the selected cell so that all of its references to cells in SAS Financial Management tables are replaced by equivalent CDA references.

A Microsoft Excel formula refers to cells by Microsoft Excel row and column designations. Any change in the layout of a SAS Financial Management table can misdirect these references. A formula that has been converted to use only CDA references is insulated from changes in the layout of tables.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Convert to CDA**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Convert to CDA**

Copy as CDA

The **Copy as CDA** option displays the Copy as CDA window, which enables you to specify a target location for a copy of the selected read-only table or range of cells. The copy that this option creates consists of CDA cells, which can be moved and formatted independently of each other.

To select an entire table for copying, double-click the cell in the upper-left corner of the table.

Note: The **Copy as CDA** option is intended primarily for tables in which all the row, column, and slicer headings are members. If you apply it to a table that has any row, column, or slicer headings that are the values of member properties, then the resulting CDA table might be imperfect.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Copy as CDA**

- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Copy as CDA**
- Pop-up menu: **Edit** ⇨ **Copy as CDA**

Copy as Read-Only Table

The **Copy as Read-Only Table** option creates a read-only table that corresponds to the selected CDA table. For details, see Converting a CDA Table to a Read-Only Table on page 247.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Copy as Read-only Table**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Copy as Read-only Table**
- Pop-up menu: **Edit** ⇨ **Copy as Read-only Table**

Copy as Table

The **Copy as Table** option makes a copy of the currently selected table in a new worksheet.

Modifying a copy of an existing table is an alternative to creating a table using **Data-entry Table** or **Read-only Table**.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Copy as Table**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Copy as Table**
- Pop-up menu: **Edit** ⇨ **Copy as Table**

Data-Entry Table

The **Data-Entry Table** option launches the Create Data-Entry Table wizard, which guides you through the process of creating a data-entry table. Data-entry tables are the central components of data-entry form templates.

For detailed information about the Create Data-Entry Table wizard, see the online help for the individual wizard pages.

This option is active only if you have opened Microsoft Excel from the **Forms** workspace of SAS Financial Management Studio in order to design a form template. After you have designed the form template, you must select **Save Template**.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **Data-Entry Table**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **Data-Entry Table**

Delete Position

The **Delete Position** option deletes the currently selected employee row from the currently selected SAS Human Capital Management supplemental schedule.

This option is active only if a row heading that has been inserted manually is selected in a SAS Human Capital Management supplemental schedule. For a custom supplemental schedule, there is an equivalent option with a different name.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (SAS Human Capital Management)** ⇨ **Delete Position**
- Excel 2003: **SAS Solutions** ⇨ **Delete Position**
- Pop-up menu: **Delete Position**

Document

The **Document** option displays the Insert Document window, which enables you to select a document from the Document Manager and insert it into the current worksheet.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **Document**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **Document**

Drill Down

The **Drill Down** option removes the selected row or column heading from the table display and instead displays all of that member's children.

This option is active only if a row or column heading is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Navigate** ⇨ **Drill Down**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Drill Down**
- Pop-up menu: **Drill Down**

Drill Up

The **Drill Up** option removes the selected row or column heading and all its siblings from the table display and instead displays their common parent.

This option is active only if a row or column heading is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Navigate** ⇨ **Drill Up**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Drill Up**
- Pop-up menu: **Drill Up**

Edit Protection Rules

For a model or a form template, edit rules to protect crossings in data-entry tables.

Click anywhere in a data-entry table and use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Edit Protection Rules**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Edit Protection Rules**

Expand

The **Expand** option leaves the selected row or column heading in the table display and in addition displays all its children.

This option is active only if a row or column heading that is not currently expanded is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Navigate** ⇨ **Expand**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Expand**
- Pop-up menu: **Expand**

Expand All

The **Expand All** option leaves the selected row or column heading in the table display and in addition displays all its descendants.

This option is active only if a row or column heading that is not currently expanded is selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Navigate** ⇨ **Expand All**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Expand All**
- Pop-up menu: **Expand All**

Export as Measure

The **Export as Measure** option displays the Export as Measure window, which enables you to export the values in the selected cells to a metric table. Metric tables associate numeric values with labels that are known as measures. They can be used to supply values

to scorecards that are displayed in SAS Strategic Performance Management or the KPI Viewer.

This option is active only if numeric cells are selected.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Export as Measure**
- Excel 2003: **SAS Solutions** ⇨ **Tools** ⇨ **Export as Measure**
- Pop-up menu: **Tools** ⇨ **Export as Measure**

Filter Member Combination

The **Filter Member Combination** option creates and applies one or more table filters based on the selected row or column headings. You can check the table filters that you have created by selecting the **Filters** option. The **Table** tab of the Filters window displays a complete list of current table filters.

A table filter can consist of a single member or a combination of members from two or more dimensions. A single-member filter suppresses any row or column that has that member in one of its heading cells. A multiple-member filter suppresses any row or column that has each of the filter's members in one of its heading cells.

If you create multiple-member filters and later use the **Pivot** option, be aware of the following:

- If you use the **Pivot** option to convert a row or column dimension into a slicer dimension and you select a slicer member that is part of a multiple-member filter, then that multiple-member filter suppresses any row or column that has each of the filter's other members in one of its heading cells.
- If you use the **Pivot** option to divide the members of a multiple-member filter between row headings and column headings, then the filter has no effect on the table. This is because there is no row or column to which all the members in the filter apply.

This option is active only if at least one member is selected in at least one row heading or column heading.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Filter** ⇨ **Filter Member Combination**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Filter Member Combination**
- Pop-up menu: **Members** ⇨ **Filter Member Combination**

Filters

The **Filters** option displays the Filters window, which enables you to do the following:

- view and remove table filters for the selected table
- view, enable, and disable system filters for the selected table

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Filter** ⇨ **Filters**

- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Filters**
- Pop-up menu: **Members** ⇨ **Filters**

Format Members

The **Format Members** option displays the Format Members window, which enables you to manage formatting styles for the numeric values in cells that are associated with specified members or property values.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Members** ⇨ **Format Members**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Format Members**
- Pop-up menu: **Members** ⇨ **Format Members**

This is one of several ways in which you can format table cells. For a comprehensive discussion, see Formatting Table Cells on page 244.

Hide Member

The **Hide Member** option applies to read-only tables and data-entry templates. It removes the selected column heading or row heading from the table. To restore a hidden column or row heading, select another heading from the same dimension, and then select **Show Members**.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Members** ⇨ **Hide Member**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Hide Member**
- Pop-up menu: **Members** ⇨ **Hide Member**

Log Off

The **Log Off** option ends the connection between your desktop copy of Microsoft Excel and the SAS Financial Management database.

This option is active only if you are currently logged on.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Log Off**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Log Off**

Log On

The **Log On** option displays the SAS Log On window, which accepts your username and password. After you enter your username and password and click **OK**, your desktop copy of Microsoft Excel is connected to the SAS Financial Management database.

This option is active only if you are not currently logged on.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Log On**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Log On**

Member Labels

The **Member Labels** option displays the Member Labels window, which enables you to search for members in any dimension of any model and drag the identifying text into spreadsheet cells.

Member labels have two main uses:

- Identify, for the reader of a report, the numeric values that are retrieved with the CDAGet function.
- Provide argument values for CDA functions. A function argument that refers to a member must use a member code (not a name or description). You can use a member code in a spreadsheet cell as an argument value by placing a Microsoft Excel reference to that cell in the argument position of a CDA function.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **Member Labels**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **Member Labels**

See Also

- Using CDA Functions in Microsoft Excel Formulas on page 273
- CDAGet on page 275

New Position

The **New Position** option displays the Add New Detail window, which enables you to add one or more employee rows to the selected SAS Human Capital Management supplemental schedule.

This option is active only if a row heading is selected in a SAS Human Capital Management supplemental schedule. For a custom supplemental schedule, there is an equivalent option with a different name.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (SAS Human Capital Management)** ⇨ **New Position**
- Excel 2003: **SAS Solutions** ⇨ **New Position**
- Pop-up menu: **New Position**

Pivot

The **Pivot** option displays the Pivot window, which enables you to change any specification that you made on the Table Layout page of the Create Table wizard. This includes the following changes:

- Change the role that is played by a certain dimension type or member property.

  See Using Member Properties as Filters, Groupers, or Labels on page 239.
- Add to the table layout a dimension type or a member property that is not currently playing any role.
- Remove a dimension type or a member property from the table layout.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Pivot**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Pivot**
- Pop-up menu: **Members** ⇨ **Pivot**

Prompt for File Format

The **Prompt for File Format** option is available only in Microsoft Excel 2007, where it controls the behavior of the **Save Template** option. Initially, the **Save Template** option displays a window that enables you to choose whether to save a template in Excel 2007 format or Excel 2003 format. This window also contains a check box that enables you to choose not to display it any more. If you select that check box and later want to display the format-choice window again, select **Prompt for File Format**.

This option is active only if the ability to choose the file format of a saved template is turned off. Its Excel 2007 selection path is **SAS Solutions (File)** ⇨ **Save** ⇨ **Prompt for File Format**.

Properties

See Table Properties on page 269.

Protect Cell

To protect selected cells in a data-entry table, use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Protect Cell**
- Excel 2003: **SAS Solutions** ⇨ **Protect Cell**
- Pop-up menu: **Edit** ⇨ **Protect Cell**

Only writable cells (displayed in yellow) can be protected. When they are protected, the cells are displayed in purple and are read-only.

If cells are protected by rules that were defined by an administrator, you cannot undo that protection. However, you can protect additional cells.

The changes that you make are saved automatically and apply to any users who open the same form.

See Also

- "Unprotect Cell" on page 270
- "Unprotect All Cells" on page 269
- "About Cell Protection" on page 138

Publish

The **Publish** option launches the Publish Report wizard, which enables you to create different types of reports and save them in the Document Manager or on your desktop. You can create either static or dynamic Excel reports. You can also create SAS reports that can be viewed with SAS Web Report Studio.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Publish**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Publish**

Read-Only Table

The **Read-Only Table** option launches the Create Read-Only Table wizard, which guides you through the process of creating a read-only table. Read-only tables and CDA tables are the central components of financial reports.

For detailed information about the Create Read-Only Table wizard, see the online help for the individual wizard pages.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **Read-Only Table**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **Read-Only Table**

Reallocate

The **Reallocate** option is available if Allocate using predefined weights is enabled for the current data-entry table, and if you have selected a parent cell in the table. It redistributes the value currently in the selected cell, according to the relative weights in the allocation weights table. This option is useful if the parent cell value did not change, but some selections in the allocation weights table changed or the weights themselves changed.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Reallocate**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Reallocate**

Refresh

The **Refresh** option refreshes numbers but not members. This option recomputes and redisplays all numeric values in the current worksheet. However, it does not retrieve any changes that have been made to hierarchies since you opened the file.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Refresh**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Refresh**

Refresh All

The **Refresh All** option refreshes numbers and members. This option recomputes and redisplays all numeric values in the current worksheet. It also retrieves any changes that have been made to hierarchies since you opened the file.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Refresh All**
- Excel 2003: **SAS Solutions** ⇨ **View** ⇨ **Refresh All**

Repeat New Position

The **Repeat New Position** option adds an employee row to the selected SAS Human Capital Management supplemental schedule. The new row is placed between the last currently existing employee row and the "Detail averages" row. The employee name in the new row is built by placing "Copy of" in front of the name in the selected employee row. You can change the employee name with the **Change Description** option.

This option is active only if a row heading that has been inserted manually is selected in a SAS Human Capital Management supplemental schedule. For a custom supplemental schedule, there is an equivalent option with a different name.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (SAS Human Capital Management)** ⇨ **Repeat New Position**
- Excel 2003: **SAS Solutions** ⇨ **Repeat New Position**
- Pop-up menu: **Repeat New Position**

Repeat Spread

The **Repeat Spread** option applies to the selected range the same spread pattern that you have just applied to another range with the Spread option.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Repeat Spread**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Repeat Spread**

SAS Financial Management Add-In Help

The **SAS Financial Management Add-In Help** option opens the online Help system for the SAS Financial Management Add-In for Microsoft Excel.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Help** ⇨ **SAS Financial Management Add-In Help**
- Excel 2003: **SAS Solutions** ⇨ **Help** ⇨ **SAS Financial Management Add-In Help**

SAS Solutions Add-In Help

The **SAS Solutions Add-In Help** option opens the online Help system for the SAS Solutions Services Add-In for Microsoft Office. That Add-In provides a set of services that are available to you as a user of the SAS Financial Management Add-In for Microsoft Excel.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tools)** ⇨ **Help** ⇨ **SAS Solutions Add-In Help**
- Excel 2003: **SAS Solutions** ⇨ **Help** ⇨ **SAS Solutions Add-In Help**

Save All Supplemental Data

The **Save All Supplemental Data** option does all of the following for a selected supplemental schedule:

- It feeds numeric values from the totals row of the supplemental schedule into the associated data-entry table.

- It saves the data-entry table values to the SAS Financial Management database.
- It saves values from the supplemental schedule back to the external database that is served by the data provider.

This option is active only if you are designing a form template that includes a supplemental schedule or using a data-entry form that includes a supplemental schedule.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (SAS Human Capital Management)** ⇨ **Save All Supplemental Data**

  Note: This is the path name for the SAS Human Capital Management data provider. A custom data provider has a custom path name.
- Excel 2003: **SAS Solutions** ⇨ **Save All Supplemental Data**
- Pop-up menu: **Save All Supplemental Data**

See Also

Entering Data into a Supplemental Schedule on page 224

Save Form Design

The **Save Form Design** option saves all features of the current form other than the numeric data. The numeric data is saved automatically as you enter it. Use this option to save changes to comments, format settings, filters, and so on.

If you make design changes that you could save with this option and then exit from the spreadsheet without using this option, a window appears that enables you to choose whether to save your design changes before exiting.

This option is active only if you have opened Microsoft Excel from the Form Manager in order to enter data into a data-entry form.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Save** ⇨ **Save Form Design**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Save Form Design**

Save Template

The **Save Template** option saves the current form template.

This option is active only if you have opened Microsoft Excel from the **Forms** workspace of SAS Financial Management Studio. You must use this option whenever you create or edit a form template.

If you are working on an operational form template, then **Save Template** checks that the template is valid for the Web-based Form Editor. If you are working on a financial form template, then **Save Template** does not make this check. If you are working on a financial form template for forms that can be edited in the Web-based Form Editor, then select **Validate Web Form** to check that the template is valid for the Web-based Form Editor.

Select one of the following options:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Save** ⇨ **Save Template**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Save Template**

See Also

- Designing a Data-Entry Form Template on page 214
- Validate Web Form on page 270

Show Members

The **Show Members** option displays the Show Members window for the dimension that contains the selected column heading, row heading, or slicer member. The Show Members window enables you to specify which members from that dimension to include in the table. There are several versions of this window—one for flat dimensions (Analysis, Currency, Frequency), one for time dimensions, and one for all other dimensions.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Members** ⇨ **Show Members**
- Excel 2003: **SAS Solutions** ⇨ **Members** ⇨ **Show Members**
- Pop-up menu: **Members** ⇨ **Show Members**

Spread

The **Spread** option displays the Spread window, which enables you to spread numeric values within the selected range of cells. Many spread patterns are available. In general, values spread toward the right from the left column of the selected range, or downward from the top row of the selected range.

If you need to apply a certain spread pattern to several ranges of cells, you might be able to save time by using **Spread** once and then continuing with **Repeat Spread**.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Spread**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Spread**

Supplemental Schedule

The **Supplemental Schedule** option launches the Supplemental Schedule wizard, which guides you through the process of creating a supplemental schedule. If SAS Human Capital Management is installed at your site, then you can create a supplemental schedule that contains data for individual employees. If a custom supplemental data provider has been implemented at your site, then you can create a supplemental schedule that contains data from the custom provider.

This option is active only if you have opened Microsoft Excel from the **Forms** workspace of SAS Financial Management Studio in order to design a form template. In addition, your

site must include either SAS Human Capital Management or a custom supplemental data provider. After you design the form template, you must select **File** ⇨ **Save Template** from the **SAS Solutions** menu.

For detailed information about the Supplemental Schedule wizard, see the online Help for the individual wizard pages.

Select one of the following options:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Insert** ⇨ **Supplemental Schedule**
- Excel 2003: **SAS Solutions** ⇨ **Insert** ⇨ **Supplemental Schedule**

Table Properties

The **Table Properties** option displays the Table Properties window, which enables you to view or change various properties of the currently selected table.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Properties**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Table Properties**
- Pop-up menu: **Edit** ⇨ **Table Properties**

Unassign

The **Unassign** option computes the sum of the values in all the yellow cells in the selected range of cells, places the sum in the active cell, and places zeros in all the other cells that contributed to the sum. The selected range of cells must be part of a single column or a single row. You can use this option to undo a spread operation.

This option is active only if a range of cells is selected in a data-entry table and the active cell is a yellow cell.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (Tables)** ⇨ **Edit** ⇨ **Unassign**
- Excel 2003: **SAS Solutions** ⇨ **Edit** ⇨ **Unassign**

Unprotect All Cells

The **Unprotect All Cells** option removes cell protection from all cells in a data-entry table, whether the cells are visible or not.

Note: This option applies only to cell protection that was set in the current form. If cells are protected by rules that were defined by an administrator, you cannot undo that protection.

Click anywhere in the table and select one of the following options:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Unprotect All Cells**
- Excel 2003: **SAS Solutions** ⇨ **Unprotect All Cells**

See Also

- "Protect Cell" on page 263
- "Unprotect Cell" on page 270
- "About Cell Protection" on page 138

Unprotect Cell

The **Unprotect Cell** option removes cell protection from all selected cells of a data-entry table.

Note: This option applies only to cell protection that was set in the current form. If cells are protected by rules that were defined by an administrator, you cannot undo that protection.

Select one of the following options:

- Excel 2007 or later: **SAS Solutions (Navigate)** ⇨ **Unprotect Cell**
- Excel 2003: **SAS Solutions** ⇨ **Unprotect Cell**
- Pop-up menu: **Edit** ⇨ **Unprotect Cell**

See Also

- "Protect Cell" on page 263
- "Unprotect All Cells" on page 269
- "About Cell Protection" on page 138

Validate Web Form

The **Validate Web Form** option checks that all the features of the open form template are supported by the Web-based Form Editor. The role of this option varies with the type of form template that you are working on, in the following ways:

- If you are working on an operational form template, then menu options that add only Excel-only features are not available. In addition, the **Save Template** option makes the complete set of Web checks before it saves the template. For both these reasons, the **Validate Web Form** option is a convenience rather than a necessity.
- If you are working on a financial form template, then menu options that add only Excel-only features are available and the **Save Template** option does not make Web checks. Therefore, if the forms based on the template can be edited on the Web, the **Validate Web Form** option plays an essential role.

The Web-based Form Editor does not support the following:

- read-only tables, CDA tables, and supplemental schedules
- CDA functions
- references to cells that are not in a SAS Financial Management table.
- cell formatting

- use of a member property as a grouper
- filtering by member combinations

The Web-based Form Editor supports the following Microsoft Excel functions but no others:

- AVG
- DATE
- DATEVALUE
- DAY
- IF
- MAX
- MIN
- MONTH
- SUM
- TODAY
- WEEKDAY
- YEAR

This option is active only if you are working on a form template.

Use one of the following selection paths:

- Excel 2007 or later: **SAS Solutions (File)** ⇨ **Save** ⇨ **Validate Web Form**
- Excel 2003: **SAS Solutions** ⇨ **File** ⇨ **Validate Web Form**

See Also

- Designing a Data-Entry Form Template on page 214
- Using Member Properties in Tables on page 239
- Filter Member Combination on page 260

Chapter 22

Dictionary of CDA Functions

Using CDA Functions in Microsoft Excel Formulas 273
CDACXRate 274
CDADesc 274
CDAGet 275
CDAName 275
CDAProperty 275
CDARate 277
CDAXRate 278

Using CDA Functions in Microsoft Excel Formulas

You can use a CDA function as all or part of a Microsoft Excel formula that is associated with a particular cell. Each CDA function returns a specific item of information from the SAS Financial Management database. “CDA” stands for “cell data access.”

To insert a CDA function into a cell:

1. Select the target cell.
2. Select **Function** from the **Insert** menu.

   The Insert Function window appears.
3. In the **Or select a category** field, select **User Defined**.

   A list of functions appears in the **Select a function** region.
4. Click one of the following functions to select it:
   - CDACXRate on page 274
   - CDADesc on page 274
   - CDAGet on page 275
   - CDAName on page 275
   - CDAProperty on page 275
   - CDARate on page 277
   - CDAXRate on page 278

The other listed functions have special purposes. Do not select them.

5. Click **OK**.

   The Function Arguments window appears. The function that you selected is identified at the top of the window, followed by a field for each of its arguments.

6. In each argument field, type the appropriate value for that argument.

   Alternatively, you can type a Microsoft Excel reference to a cell that contains the value for that argument. For example, for an argument that is a member code, you can type a reference to a cell that contains the appropriate member code. You can use the **Member Labels** option to look up member codes and place them in spreadsheet cells.

7. Click **OK**. The window closes and the function that you have specified is inserted in the selected cell.

CDACXRate

The CDACXRate function returns a specified complex exchange rate. A complex exchange rate is an exchange rate that can vary with accounts, organizations, and members of other dimensions, in addition to exchange rate type and time period.

The syntax is:

```
CDACXRate(model, RATE_TYPE, exch_rate_type, from_curr, to_curr,analysis,
time, dim_1, mem_1, dim_2, mem_2, ...)
```

- **`model`** is the code of a model.
- **`RATE_TYPE`** is the string **`RATE_TYPE`**. You must specify this string as the second parameter even though it never varies.
- **`exch_rate_type`** is one of the two complex exchange rate types: Derived, Historic.

  Use uppercase and lowercase letters exactly as shown.
- **`from_curr`** is the code of the input currency.
- **`to_curr`** is the code of the output currency.
- **`analysis`** is the code of an analysis member that the specified model associates with the exchange rate set that contains the exchange rate that you want to retrieve.
- **`time`** is the code of the time period of the exchange rate that you want to retrieve.
- Each **`dim`** variable (for example, **`dim_1`**, **`dim_2`**, and so on) is the code of an additional dimension that the exchange rate that you want to retrieve depends on.
- Each **`mem`** variable (for example, **`mem_1`**, **`mem_2`**, and so on) is the code of the member of the preceding dimension that the exchange rate is associated with.

The dimension-and-member pairs can be specified in any order.

CDADesc

The CDADesc function returns the description of a specified dimension member.

The syntax is:

```
CDADesc(model, dim, mem)
```

- **model** is the code of a model.
- **dim** is the code of a dimension in the specified model.
- **mem** is the code of a member in the specified dimension.

CDAGet

The CDAGet function returns the numeric value for a specified crossing of a specified model.

The syntax is:

```
CDAGet(model, dim_1, mem_1, dim_2, mem_2, ...)
```

- **model** is the code of the model.
- Each **dim** is the code of a dimension in the model.
- Each **mem** is the code of a member of the preceding dimension.

Each argument can be specified either as the code itself or as a Microsoft Excel reference to a cell that contains the code. In a CDA table that is created with the Create CDA Table wizard, the cells that display numeric values contain the CDAGet function with references to nearby cells that contain the appropriate codes. Here is an example:

```
=CDAGet($E$10, $E$11, G$13, $E$12, $D16)*$F$9
```

The dimension-and-member pairs can be specified in any order. For any model dimension that you omit, the CDAGet function uses the default read member of the model hierarchy in that dimension. The combination of the explicitly specified members and the default read members determines the crossing whose value is returned.

CDAName

The CDAName function returns the name of a specified dimension member.

The syntax is:

```
CDAName(model, dim, mem)
```

- **model** is the code of a model.
- **dim** is the code of a dimension in the specified model.
- **mem** is the code of a member in the specified dimension.

CDAProperty

The CDAProperty function returns the value of a specified property of a specified dimension member.

The syntax is:

```
CDAProperty("modelcode", "dimcode", "memcode", "propertycode")
```

- **modelcode** is the code of a model.
- **dimcode** is the code of a dimension in the model. The dimension must belong to a dimension type whose members have a retrievable property, as explained below.
- **memcode** is the code of a member in the dimension.
- **propertycode** is the code of a property of the member. The property can be either a property supplied by SAS or a custom property.

Following are some property codes supplied by SAS and the values that the CDAProperty function can return for each of them:

AccountBehavior
is the account type category to which the account type of an account belongs. The following values can be returned:

- **Balance** spans the Asset, Liability, Equity, and Statistical Balance account types.
- **Flow** spans the Revenue, Expense, and Statistical Flow account types.
- **Hybrid** corresponds to the Retained Earnings account type.
- **CTA** corresponds to the CTA account type.
- **NonFrequency** corresponds to the NonFrequency account type, which is also known as the Statistical account type.

This property is valid only if the member is an account.

AccountType
is the account type of 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)

This property is valid only if the member is an account.

BalanceType
is the balance type of an account. The following values can be returned:

- **Credit**
- **Debit**

This property is valid only if the member is an account.

ExchangeRateType
is the exchange rate type of an account. The following values can be returned:

- **PeriodAverage**
- **PeriodClose**

- **PeriodOpen**
- **Custom1**
- **Custom2**
- **Derived**
- **Historic**

This property is valid only if the member is an account.

Functional Currency
is the functional currency of an organization. Any currency code can be returned.

This property is valid only if the member is an organization.

Intercompany
indicates whether an account is an intercompany account. The following values can be returned:

- **True**
- **False**

This property is valid only if the member is an account.

Level
is the period type of a time period. The following values can be returned:

- **AllYears**
- **Year**
- **HalfYear**
- **QuarterYear**
- **Month**
- **Week**
- **Day**

This property is valid only if the member is a time period.

ReportingEntity
indicates whether an organization is a reporting entity. The following values can be returned:

- **True**
- **False**

This property is valid only if the member is an organization.

CDARate

The CDARate function returns a numeric driver rate from a driver rate set.

The syntax is:

```
CDARate(model, RATE_TYPE, rate_type, analysis, time, dim_1, mem_1, dim_2, mem_2, ...)
```

- **model** is the code of a model.

- **RATE_TYPE** is the string **RATE_TYPE**. You must specify this string as the second parameter even though it never varies.
- **rate_type** is the code of the rate type for the driver rate that you want to retrieve.
- **analysis** is the code of an analysis member that the specified model associates with the driver rate set that contains the driver rate that you want to retrieve.
- **time** is the code of the time period of the driver rate that you want to retrieve.
- Each **dim** is the code of an additional dimension that the driver rate that you want to retrieve depends on.
- Each **mem** is the code of the member of the preceding dimension that the driver rate is associated with.

The dimension-and-member pairs can be specified in any order.

CDAXRate

The CDAXRate function returns a simple exchange rate from an exchange rate set. A simple exchange rate is an exchange rate that depends only on an exchange rate type and a time period. For a given exchange rate type and a given time period, there is only one simple exchange rate across all accounts and organizations.

The syntax is:

```
CDAXRate(model, exchratetype, fromcurr, tocurr, analysismemcode, timememcode)
```

- **model** is the code of a model.
- **exchratetype** is one of the following simple exchange rate types:

  ```
  PeriodAverage
  PeriodClose
  PeriodOpen
  Custom1
  Custom2
  ```

  Use uppercase and lowercase letters exactly as shown. There are no spaces between letters.
- **fromcurr** is the code of the input currency.
- **tocurr** is the code of the output currency.
- **analysismemcode** is the code of an analysis member that the specified model associates with the exchange rate set that contains the exchange rate that you want to retrieve.
- **timememcode** is the code of the time period of the exchange rate that you want to retrieve.

Chapter 23

Dictionary of Calculated-Member Functions

Using Calculated-Member Functions **279**

fmCode **280**
Overview 280
Syntax with Required Arguments Only 280
Syntax Including Optional Arguments 280

fmCXRate **281**
Overview 281
Syntax with Required Arguments Only 281
Syntax Including Optional Arguments 282

fmProperty **282**

fmRate **283**
Overview 283
Syntax with Required Arguments Only 283
Syntax with Optional Arguments 283

fmValue **284**
Overview 284
Syntax with Required Arguments Only 284
Syntax Including Optional Arguments 285

fmXRate **285**
Overview 285
Syntax with Required Arguments Only 285
Syntax Including Optional Arguments 286

Using Calculated-Member Functions

Calculated-member functions can be used in calculated-member formulas, the formulas that are associated with calculated members. In general, to be useful a calculated-member formula must contain at least one calculated-member function.

See Also

Using Calculated Members on page 243

fmCode

Overview

The fmCode function returns the code of the member of a specified dimension that is at a specified crossing. By default, the specified crossing is the crossing where the function is evaluated.

Syntax with Required Arguments Only

The fmCode function can be used with one argument:

```
fmCode("dimensioncode")
```

dimensioncode is the code of a dimension in the model.

This form of the fmCode function returns the code of the member of the **dimensioncode** dimension that is at the crossing where the function is evaluated.

For example, if the code of the account dimension is ACCOUNT, then the following formula returns 1 if it is evaluated at a crossing that includes the XYZ account, and otherwise returns zero:

```
IF(fmCode("ACCOUNT")="XYZ",1,0)
```

Syntax Including Optional Arguments

The fmCode function can be used with three arguments to return the code of the time period of a crossing that is displaced in time relative to the crossing at which the function is evaluated:

```
fmCode("timedimensioncode","timedimensioncode",n)
```

timedimensioncode is the code of the model's time dimension. Note that this dimension code has to be specified in the first argument and the second argument.

n is a positive or negative integer that indicates a time displacement measured in periods.

Here are two examples, using TIME as the code of the time dimension:

- `fmCode("TIME","TIME",-3)`

  If the time member of the crossing where this function is evaluated is April 2010, then this function returns the code of the January-2010 member.

- `fmCode("TIME","TIME",+2)`

  If the time member of the crossing where this function is evaluated is the first quarter of 2010, then this function returns the code of the third-quarter-of-2010 member.

Note: You cannot use this form of the fmCode function on a calculated time member.

fmCXRate

Overview

The fmCXRate function returns a complex exchange rate.

You can use a formula that consists of this function alone to display complex exchange rates in a table. In general, there is no need to write formulas that perform computations with exchange rates because the currency conversion process occurs automatically. However, displaying the exchange rates that are used in the automatic currency conversion process can be useful.

Syntax with Required Arguments Only

The fmCXRate function can be used with three arguments:

```
fmCXRate("exchangeratetype","fromcurrency","tocurrency")
```

`exchangeratetype` is one of the following complex exchange rate types:

- Derived
- Historic

`fromcurrency` and **`tocurrency`** are currency codes.

This form of the fmCXRate function returns a complex exchange rate that satisfies the following conditions:

- It belongs to the exchange rate type that is specified by the first argument.
- It converts a value in the **`fromcurrency`** currency to a value in the **`tocurrency`** currency.
- It is in the exchange rate set that is associated with the analysis member of the crossing where the function is evaluated.
- It is associated with the time member of the crossing where the function is evaluated.

Other members of the crossing where the function is evaluated are used as necessary to complete the look-up. If complex exchange rates are associated with organizations in the relevant exchange rate set, then the organization member of the crossing is used in the look-up.

For example, if accounts play no role in the look-up, then the following formula, associated with a calculated account member, can be used to display derived exchange rates from United States dollars to euros:

```
fmXRate("Derived","USD","EUR")
```

Note: With this three-argument syntax, the fmCXRate function performs its look-up using only members of the crossing where it is evaluated. Therefore, this syntax does not work if the function is associated with a calculated time member, a calculated analysis member, or a calculated member of any other dimension that plays a role in the look-up.

Syntax Including Optional Arguments

The fmCXRate function can be used with 3+2n arguments, where n is a positive integer:

```
fmCXRate("exchangeratetype","fromcurrency","tocurrency",
"dimensioncode1","membercode1", ...)
```

Each **dimensioncode** is the code of a dimension in the model.

Each **membercode** is the code of a member of the dimension that immediately precedes it.

You must add optional arguments in pairs.

You can use a pair of time arguments to specify a time member to use instead of the time member of the crossing where the function is evaluated. If the function is associated with a calculated time member, then you must use a pair of time arguments.

For a **dimensioncode** from the Time dimension type, you can also use a positive or negative integer to specify a displacement relative to the crossing where the function is evaluated. However, you cannot do this if the function is associated with a calculated time member. For examples, see fmCode on page 280.

You can use a pair of analysis arguments to specify an analysis member to use instead of the analysis member of the crossing where the function is evaluated. If the function is associated with a calculated analysis member, then you must use a pair of analysis arguments.

You can use a pair of arguments for another dimension to specify a member to use in the look-up instead of the member of that dimension that is at the crossing where the function is evaluated.

For example, the following formula, associated with a calculated account, can be used to display historic exchange rates from United States dollars to euros that are associated with the COMMON account. The retrieved exchange rates might also be associated with members of other dimensions that are part of the crossing where the function is evaluated:

```
fmXRate("Historic","USD","EUR","ACCOUNT","COMMON")
```

fmProperty

The fmProperty function returns the value of a specified property of the member of a specified dimension that is at the crossing where the function is evaluated. The syntax has two arguments:

```
fmProperty("dimensioncode","propertycode")
```

dimensioncode is the code of a dimension in the model.

propertycode is a property code. The property code can be the code of any property supplied by SAS or the code of any custom property. For a list of property codes supplied by SAS, see CDAProperty on page 275.

For example, the following function returns the value of the **BalanceType** property of the member of the **ACCOUNT** dimension that is at the crossing where the function is evaluated:

```
fmProperty("ACCOUNT","BalanceType")
```

In the following formula, associated with the calculated analysis member VARIANCE, the fmProperty function is used to determine the correct way to compute a variance for each account:

```
IF(fmProperty("ACCOUNT","BalanceType")="CREDIT",
fmValue("ACTUAL")-fmValue("BUDGET"),fmValue("BUDGET")-fmValue("ACTUAL"))
```

fmRate

Overview

The fmRate function returns a driver rate.

You can use a formula that consists of this function alone to display driver rates in a table. In general, driver rates are used in computations only in formulas that are defined in SAS Financial Management Studio. However, displaying the driver rates that are used in those computations can be useful.

Syntax with Required Arguments Only

The fmRate function can be used with one argument:

```
fmRate("driverratetype")
```

`driverratetype` is one of the driver rate types that is defined in SAS Financial Management Studio.

This form of the fmRate function returns a driver rate that satisfies the following conditions:

- It belongs to the driver rate type that is specified in the argument.
- It is in the driver rate set that is associated with the analysis member of the crossing where the function is evaluated.
- It is associated with the time member of the crossing where the function is evaluated.

Other members of the crossing where the function is evaluated are used as necessary to complete the look-up. If driver rates are associated with organizations in the relevant driver rate set, then the organization member of the crossing is used in the look-up.

For example, if accounts play no role in the look-up, then the following formula, associated with a calculated account member, can be used to display driver rates of type Percent:

```
fmRate("Percent")
```

Note: With this one-argument syntax, the fmRate function performs its look-up using only members of the crossing where it is evaluated. Therefore, this syntax does not work if the function is associated with a calculated time member, a calculated analysis member, or a calculated member of any other dimension that plays a role in the look-up.

Syntax with Optional Arguments

The fmRate function can be used with 1+2n arguments, where n is a positive integer:

```
fmRate("driverratetype","dimensioncode1","membercode1",...)
```

Each **`dimensioncode`** is the code of a dimension in the model.

Each **membercode** is the code of a member of the dimension that immediately precedes it.

You must add optional arguments in pairs.

You can use a pair of time arguments to specify a time member to use instead of the time member of the crossing where the function is evaluated. If the function is associated with a calculated time member, then you must use a pair of time arguments.

For a **dimensioncode** from the Time dimension type, you can also use a positive or negative integer to specify a displacement relative to the crossing where the function is evaluated. However, you cannot do this if the function is associated with a calculated time member. For examples, seefmCode on page 280.

You can use a pair of analysis arguments to specify an analysis member to use instead of the analysis member of the crossing where the function is evaluated. If the function is associated with a calculated analysis member, then you must use a pair of analysis arguments.

You can use a pair of arguments for another dimension to specify a member to use in the look-up instead of the member of that dimension that is at the crossing where the function is evaluated.

For example, the following formula, associated with a calculated account, can be used to display driver rates of type Percent that are associated with the BENEFITS account. The retrieved driver rates might also be associated with members of other dimensions that are part of the crossing where the function is evaluated:

```
fmRate("Percent","ACCOUNT","BENEFITS")
```

fmValue

Overview

The fmValue function returns the numeric value that is at a specified crossing.

Syntax with Required Arguments Only

The fmValue function can be used with one argument:

```
fmValue("membercode1")
```

membercode1 is the code of a member of the calculated-member dimension. This is the dimension that contains the calculated member with which the formula that contains the function is associated.

This form of the fmValue function returns the value that is at the crossing that satisfies the following conditions:

- It contains the **membercode1** member of the calculated-member dimension.
- In all other dimensions, it matches the crossing where the function is evaluated.

For example, the following formula, associated with the calculated analysis member VARIANCE, computes variances from corresponding pairs of ACTUAL and BUDGET values:

```
fmValue("ACTUAL")-fmValue("BUDGET")
```

This form of the fmValue function appears automatically when you click a member while building a formula with the Calculated Member wizard.

Syntax Including Optional Arguments

The fmValue function can be used with 1+2n arguments, where n is a positive integer:

```
fmValue("membercode1","dimensioncode2","membercode2", ... )
```

membercode1 is the code of a member of the calculated-member dimension.

Each **dimensioncode** is the code of a dimension in the model.

Each **membercode** is the code of a member of the dimension that immediately precedes it.

You must add optional arguments in pairs. Each pair of optional arguments specifies a member to use instead of the member of the specified dimension that is in the crossing where the function is evaluated.

This form of the fmValue function returns the numeric value at the crossing that satisfies the following conditions:

- It contains the **membercode1** member of the calculated-member dimension.
- It contains the specified members of the specified dimensions.
- In all other dimensions, it matches the crossing where the function is evaluated.

For a **dimensioncode** from the Time dimension type, you can also use a positive or negative integer to specify a displacement relative to the crossing where the function is evaluated. For examples, seefmCode on page 280.

fmXRate

Overview

The fmXRate function returns a simple exchange rate.

You can use a formula that consists of this function alone to display simple exchange rates in a table. In general, there is no need to write formulas that perform computations with exchange rates because the currency conversion process occurs automatically. However, displaying the exchange rates that are used in the automatic currency conversion process can be useful.

Syntax with Required Arguments Only

The fmXRate function can be used with three arguments:

```
fmXRate("exchangeratetype","fromcurrency","tocurrency")
```

exchangeratetype is one of the following simple exchange rate types:

- PeriodAverage
- PeriodClose
- PeriodOpen
- Custom1

- Custom2

fromcurrency and **tocurrency** are currency codes.

This form of the fmXRate function returns the simple exchange rate that satisfies the following conditions:

- It belongs to the exchange rate type that is specified by the first argument.
- It converts a value in the **fromcurrency** currency to a value in the **tocurrency** currency.
- It is in the exchange rate set that is associated with the analysis member of the crossing where the function is evaluated.
- It is associated with the time member of the crossing where the function is evaluated.

For example, the following formula, associated with a calculated account member, can be used to display period-average exchange rates from United States dollars to euros:

```
fmXRate("PeriodAverage","USD","EUR")
```

Note: With this one-argument syntax, the fmXRate function performs its look-up using the analysis member and the time member of the crossing where it is evaluated. Therefore, this syntax does not work if the function is associated with a calculated analysis member or a calculated time member.

Syntax Including Optional Arguments

The fmXRate function can be used with five or seven arguments:

```
fmXRate("exchangeratetype","fromcurrency","tocurrency","timedimensioncode",
"timemembercode","analysisdimensioncode","analysismembercode" )
```

timedimensioncode is the code of the model's time dimension.

timemembercode is the code of a member of the model's time dimension.

analysisdimensioncode is the code of the model's analysis dimension.

analysismembercode is the code of a member of the model's analysis dimension.

You can use the two time arguments to specify a time member to use instead of the time member of the crossing where the function is evaluated. If the function is associated with a calculated time member, then you must use the two time arguments.

For a **dimensioncode** from the Time dimension type, you can also use a positive or negative integer to specify a displacement relative to the crossing where the function is evaluated. However, you cannot do this if the function is associated with a calculated time member. For examples, seefmCode on page 280.

You can use the two analysis arguments to specify an analysis member to use instead of the analysis member of the crossing where the function is evaluated. If the function is associated with a calculated analysis member, then you must use the two analysis arguments.

For example, the following formula, associated with a calculated analysis member, can be used to display period-close exchange rates from United States dollars to euros that are in the exchange rate table that is associated with the ACTUAL analysis member:

```
fmXRate("PeriodClose","USD","EUR","ANALYSIS","ACTUAL")
```

Part 4

SAS Financial Management on the Web

Chapter 24
Data Entry: General Considerations *289*

Chapter 25
Data Entry: Bottom-Up Workflow *299*

Chapter 26
Data Entry: Top-Down Workflow *305*

Chapter 27
Reports in the Document Manager *309*

Chapter 24

Data Entry: General Considerations

Introduction to Data Entry **289**

Form Set Types **290**

Facilities for Data Entry **290**

Security Roles for Data Entry **291**

Form Manager Page **291**
Overview of the Form Manager Page 291
List View 291
Hierarchy View 291
Columns 292

Form Editor Page **293**
Overview of the Form Editor Page 293
Entering Data into Parent Cells 294
Options below the Title Banner 294
Options on the Table Pop-Up Menu 295

Actions for Forms **297**

Status Values for Forms **297**

Introduction to Data Entry

You enter data through forms that have been designed in SAS Financial Management Studio and published from there to the Web. The forms that you are responsible for are available to you when you log on to the Form Manager.

A form set is a collection of forms. It has a target hierarchy (for example, a hierarchy of organizational departments or managers, or a hierarchy of accounts). Each form is associated with a member of the target hierarchy, although not all members of the hierarchy are selected to receive forms.

Each form in the form set is part of a workflow that provides the framework for a collaborative data-entry project that can involve any number of individuals. Different individuals are responsible for different forms. When a user completes a form, the workflow determines the form's next state.

Form Set Types

Every form set is either a financial form set or an operational form set. Every form set is either a bottom-up form set or a top-down form set. All combinations are possible:

- bottom-up financial form sets
- top-down financial form sets
- bottom-up operational form sets
- top-down operational form sets

For each form that is listed on the Form Manager page, the **Type** column indicates which type of form set it belongs to.

In general, financial form sets are for high-level planning that involves monetary values, whereas operational form sets are for detailed planning that can involve either monetary values or non-monetary values. For example, an operational form set for salary planning would involve monetary values, but an operational form set for unit sales planning would involve non-monetary values.

In a bottom-up form set, data-entry proceeds up the target hierarchy. Data is entered at the lowest hierarchical level first. At each level, data is entered, submitted to the next level up for approval, and aggregated if approved. In a top-down form set, data-entry proceeds down the target hierarchy. Data is entered at the highest hierarchical level, and then allocated and pushed down a level. At other levels, data received from a higher level is allocated further and then pushed down another level.

Facilities for Data Entry

SAS Financial Management provides two data-entry facilities:

- Web-based Form Editor
- SAS Financial Management Add-In for Microsoft Excel

Forms that belong to operational form sets can be edited only in the Web-based Form Editor.

Forms that belong to financial form sets can be edited in the way or ways that were specified when the form set was designed in SAS Financial Management Studio. There are three possibilities—Web-based Form Editor only, Excel Add-In only, or both.

The Form Manager page provides an option for each data-entry facility:

- The **Open** option opens a form in the Web-based Form Editor.
- The **Open in Excel** option opens a form in the Excel Add-In.

A particular form might offer both of these data-entry options or either one of them alone, depending on what type of form set it belongs and how its form set was designed.

Security Roles for Data Entry

To be assigned as a form author or reviewer, users must have the appropriate roles. If you cannot access a form, check with your administrator to make sure that you have the roles you need. Your role and group memberships determine the forms you can access and the data you can view and edit.

Form Manager Page

Overview of the Form Manager Page

Use the Form Manager page to manage data-entry forms.

You can toggle this page between a list view and a hierarchy view:

- **List View** displays a list of forms that you can sort and filter.
- **Hierarchy View** shows forms in the hierarchical context of their form sets.

In both the list view and the hierarchy view, the display includes only the forms that your security role and form-specific assignments entitle you to edit or review.

List View

The list view of the Form Manager page displays a list of data-entry forms. You can scroll, sort, filter, and refresh the display.

If the list is long, then you can scroll through it by using the four arrows immediately below the display area. The arrows that are accompanied by horizontal bars take you to the top and the bottom of the list, respectively. The plain arrows take you to the next block of forms in the indicated direction.

To sort the list by the information in a particular column, click the column heading. To reverse the sort order, click the same column heading again.

To set, change, or clear filter settings, click **Filter** and use the Filter window. If there are filter settings currently in force, then they are described in a line of text between the title banner and the list of forms.

To ensure that the display is current, click **Refresh**. Refreshing this page adds any forms that were published after you opened the page, removes any forms that were withdrawn from publication, and reflects any status changes to forms. The refreshed display continues to be restricted by the current filter settings.

Hierarchy View

The hierarchy view of the Form Manager page shows forms in the hierarchical context of their form sets. You can scroll, expand, collapse, and refresh the display.

To scroll the display, use the scroll bar on the right.

To expand a collapsed portion of a hierarchy, do one of the following to its root form:

- Click the plus sign in the **Target** column.

- Click the menu icon in the first column, and then select **Expand All** from the pop-up menu.

To collapse an expanded portion of a hierarchy, do one of the following to its root form:

- Click the minus sign in the **Target** column.
- Click the menu icon in the first column, and then select **Collapse All** from the pop-up menu.

To collapse all hierarchies completely, click **Collapse All** at the top of the display.

To ensure that the display is current, click **Refresh**. Refreshing this page adds any forms that were published after you opened the page, removes any forms that were withdrawn from publication, and reflects any status changes to forms.

Columns

In both the list view and the hierarchy view, the display includes the following columns, from left to right:

(action menu)
contains the pop-up menu of actions on a single form. Click the action menu and select an action from the pop-up menu.

See "Actions for Forms" on page 297.

(type)
indicates the type of form set the form belongs to. The arrow indicates the direction of the workflow (bottom-up or top-down). A small green circle signifies an operational form set.

Icon	Type
	Bottom-up form (financial planning)
	Top-down form (financial planning)
	Bottom-up form (operational planning)
	Top-down form (operational planning)

Click the column heading to reverse the ordering of bottom-up forms and top-down forms.

See Form Set Types on page 290.

(attachments)
displays a paper clip icon for each form that has attachments. Click the icon to display the Attachments page, which gives you access to the attachments.

Target
contains the name of the member of the target hierarchy that the form is assigned to. Typically, this is the name of an organization in an organization hierarchy.

You can click the name of the target member to open the form. If the form can be opened in either the Web-based Form Editor or the SAS Financial Management Add-In for Microsoft Excel, then clicking the name in this column opens the form in the Web-based Form Editor.

Name
contains the name of the form set that the form belongs to.

 (overdue)
indicates whether the form's due date has passed. The red flag icon is displayed for each form whose due date has passed.

Deadline
contains the date the form is due, displayed in the user's time zone.

All forms that belong to the same form set have the same deadline. A deadline that has passed is highlighted in red.

Some forms are locked when the deadline is reached. A locked form is removed from the list. If you have a form open for editing, its status changes to read-only.

Status
displays the current status of the form.

See "Status Values for Forms" on page 297.

 (multi-form workflow actions)
enables you to perform workflow actions, such as **Submit** or **Approve**, for several forms at once.

To select a form, click its check box. To select every form on the page, click the check box in the column heading. Then click the action menu in the column heading and select an action from the pop-up menu.

See "Actions for Forms" on page 297.

Form Editor Page

Overview of the Form Editor Page

Use the Form Editor page to enter data into the currently selected form.

The page title includes the name of the form set that the selected form belongs to and the hierarchy member that the selected form is assigned to. For example, if the form set is ABC and the hierarchy member is XYZ, then the page title is "Form Editor – ABC – XYZ."

A form contains one or more data-entry tables. Each data-entry table has at least one row dimension and at least one column dimension. It can also have slicer dimensions and additional row or column dimensions. The slicer dimensions appear above the grid of rows and columns. The displayed numeric values make up the slice of data that corresponds to the currently selected members of the slicer dimensions. To display a different slice of data, select different members for one or more slicer dimensions. To select a different member for a slicer dimension, click the selection icon  next to the currently selected member of that dimension. Any slicer dimension that does not have this icon is limited to the displayed member.

In column heading cells and row heading cells, leaf members are displayed in black and non-leaf members are displayed in blue. You can expand or collapse any non-leaf member by clicking it.

Cells that you can enter data into are yellow. To format recently entered numeric values correctly and to recompute values that depend on them, select **Refresh**.

Protected cells are displayed in purple. You can view their contents but you cannot enter data into those cells.

If there are comments associated with the selected form, you can display a complete list of the associated comments by expanding the **Comments** region. In the expanded **Comments** region, click any listed comment to display its content.

If you right-click any cell in a table grid, a pop-up menu appears. The options on the pop-up menu depend on which cell you right-click.

Entering Data into Parent Cells

Note: These options apply only to financial forms in a bottom-up workflow.

In most cases, a parent cell is gray, indicating that its value is derived from the values of its subordinate cells. (A gray cell might also indicate that its value comes from a calculation.)

If a parent cell is yellow, it is writable. When you enter data into a writable parent cell, the value might be distributed in one of two ways:

- It is written to the virtual child (vc) cell for the parent.
- It is automatically allocated to eligible subordinate cells, according to an allocation weights table that is defined by administrators.

Note: If parent cells are writable, the **Spread** and **Adjust Values** options are disabled when the range includes parent cells.

These options are mutually exclusive. They can be enabled or disabled only by an administrator of the form template.

Options below the Title Banner

The set of options immediately below the title banner changes as appropriate.

The following options are always present:

Refresh
formats all recently entered numeric values and recomputes all computed values to reflect all recently entered numeric values.

Add Comment
displays the Add Comment window, which enables you to write a comment and associate it with the currently displayed form.

After the comment is added, it can be viewed in the **Comments** region by anyone who works on the form.

If you are working on an operational form that has one or more child forms, then the **Open Child Form** option is present. The children of an operational form often include members that are not in the parent form. This option gives you a convenient way to view the values that are associated with such members. It displays the Open Child Form window, which enables you to select a child form and display it in another browser window.

Finally, appropriate workflow actions are present. For a bottom-up form set, you might see **Submit**, **Recall**, **Approve**, or **Reject**. For a top-down form set, you might see **Push**, **Push to All**, or **Recall**.

Options on the Table Pop-Up Menu

A pop-up menu appears when you right-click a table cell. The set of active options on the pop-up menu depends on where you right-click:

Pivot
swaps the table's row dimensions with the table's column dimensions.

This option is active when you right-click any row heading cell or any column heading cell.

Export to Excel
displays windows that enable you to export the data in the selected table to a Microsoft Excel file.

This option is active when you right-click the upper left corner of the table.

Drill Up
hides the selected row or column, all its siblings, all the descendants of the selected row or column, and all the descendants of the siblings. The resulting display includes the parent of the selected row or column and the parent's siblings, if there are any.

This option is active when you right-click a row heading or a column heading that has a parent in the table.

Drill Down
shows all rows that are children of the selected row or all columns that are children of the selected column. All other rows or columns are hidden.

This option is active when you right-click a row heading or a column heading that has children in the table.

Expand All
fully expands the portion of a hierarchy that is subordinate to the selected row or column member.

Copy
captures the numeric values in the selected range of cells so that you can paste them into another range of cells.

This option is active when you select one or more cells that contain numeric values.

Cut
captures the numeric values in the selected range of cells so that you can paste them into another range of cells, and also replaces the values in the selected range of cells with zeros.

This option is active when you select one or more cells that contain numeric values.

Paste
places the most recently copied or cut range of values into the selected range of cells.

If the source range and the target range have the same size and shape, then each copied or cut value is pasted into one cell.

If the target range fits inside the source range, then the target range is filled with values from the upper left portion of the source range. The other values in the source range are not pasted.

If the target range is large enough to contain two or more copies of the source range, then the pattern of values in the source range is pasted repeatedly into the target range, as many times as possible.

If one or more whole copies of the source range are pasted, then no partial copies of the source range are pasted. For example:

- If you paste a 2 x 2 source range into a 3 x 3 target range, then one whole copy of the source range is created and the remaining five cells of the target range are not changed.
- If you paste a 2 x 2 source range into a 2 x 5 target range, then two whole copies of the source range are created and the remaining two cells are not changed.

This option is active when you select one or more cells that contain numeric values, and you have already used **Copy** or **Cut** to capture values.

Adjust Values
displays the Adjust Values window, which enables you to change the values in the selected cells in any of the following ways:

- Increase or decrease each selected value by a specified percentage.
- Increase or decrease each selected value by a specified amount.
- Allocate a specified amount over all the selected cells in proportion to their current values.

This option is active when you select one or more cells that contain numeric values.

Spread
displays the Spread window, which enables you to spread one or more selected values over selected cells in various ways.

This option is active when you select any range of cells that contain numeric values. However, you can complete a spread operation only if the selected range of cells satisfies one of these conditions:

- For a horizontal spread, each row of the selected range must contain exactly one nonzero value and at least one zero value.
- For a vertical spread, each column of the selected range must contain exactly one nonzero value and at least one zero value.

If the selected range satisfies both conditions, then before the Spread window appears another window appears, in which you must specify whether you are performing a horizontal spread or a vertical spread.

If the selected range satisfies neither condition, then a message tells you that the selected range is not valid.

Repeat Spread
applies the last spread operation that you specified using the **Spread** option to the currently selected range of cells.

This option is active when you select one or more cells that contain numeric values, and you have already used the **Spread** option. However, the operation can be completed only if the last spread operation can be applied to the currently selected range of cells. If the last spread operation cannot be applied to the currently selected range of cells, then an explanatory message is displayed.

Clear ⇨ Selected Cells
replaces the values in all selected cells with zeros.

This option is active when you select one or more cells that contain numeric values.

Clear ⇨ Table
replaces the values in all cells of the table with zeros.

This option is active when you select one or more cells that contain numeric values.

Actions for Forms

The available actions for a form depend on the following factors:

- the workflow direction of the form set that the form belongs to (top-down or bottom-up).
- the status of the form within its workflow. There are different status values for top-down and bottom-up workflows.
- the type of data-entry that is permitted: in Microsoft Excel, in the Web-based Form Editor, or both. For an operational form set, data entry always occurs in the Web-based Form Editor.

Two additional actions are available for forms, regardless of their workflow type or their status:

Comments
opens the Comment Manager, in which you can view or create comments to accompany the form. For more information, see the online Help for Comment Manager.

Note: When you take action on a form, you are also given the opportunity to add a comment about that action.

Details
opens the Form Details page, which displays information such as form history and status.

See Also

- Bottom-Up Actions on page 300
- Bottom-Up Status Values on page 302
- Top-Down Actions on page 306
- Top-Down Status Values on page 307

Status Values for Forms

The status of a form depends on the following factors:

- The workflow direction of the form set that the form belongs to.
- The most recent action that was performed on the form. There are different actions for top-down and bottom-up workflows.

See Also

- “Bottom-Up Actions” on page 300
- “Bottom-Up Status Values” on page 302
- “Top-Down Actions” on page 306

- “Top-Down Status Values” on page 307

Chapter 25

Data Entry: Bottom-Up Workflow

Description of the Bottom-Up Workflow 299
Bottom-Up Actions 300
Bottom-Up Status Values 302

Description of the Bottom-Up Workflow

A bottom-up workflow begins when a bottom-up form set is published from SAS Financial Management Studio. Publishing makes all the forms of the bottom-up form set available in the Form Manager.

In a bottom-up workflow, data flows up the target hierarchy in the following way:

1. Users with the appropriate data-entry responsibilities enter data into the bottom-level forms.
2. The data in each bottom-level form flows up to its parent form, where it is aggregated with data that flows up from that parent's other children.
3. Data continues to flow up the hierarchy of forms level by level, with each parent form aggregating the data from its children.
4. The bottom-up flow of data ends when the top form receives and aggregates all the data.

You can enter data into a parent form if the data-entry table includes the virtual child of the parent member that the form is assigned to. Enter the data in association with the virtual child. The parent form then aggregates its virtual child data along with the data coming from the forms that are its real children.

The bottom-up flow of data is controlled by a repeating pattern of user actions, which you select on the Form Manager page:

- To enter data, you edit a form. You can edit a form as many times as necessary in order to get the numbers the way you want them.
- When you have finished editing a bottom-level form, you submit the form for review.
- If you submit a form for review and then decide that you need to make changes, you can recall the form, edit it some more, and then submit it again. Recalling a form can be useful in certain circumstances, but it is not part of the mainstream workflow.
- As a reviewer of a submitted form, you view the form and then either approve the form or reject it.

A rejected form, like a recalled form, must be submitted for review again after suitable changes are made to the data. A form can go through a loop of rejection, more editing, and resubmission as many times as necessary until it is approved.

An operational form always has a single reviewer. A financial form that has been approved by one reviewer might or might not have to go to other reviewers, depending on how the form set is defined. A financial form that has been approved by at least one reviewer but requires the approval of at least one more reviewer is in a partially approved state. A financial form that has been approved by the entire sequence of assigned reviewers is approved. When and only when a form is fully approved, its data flows up into its parent form.

A form that has been rejected by any reviewer must be submitted again and reviewed again by the entire sequence of assigned reviewers.

- A parent form cannot be submitted for review until all the child forms that contribute data to it have been approved.

  If you are responsible for submitting a certain parent form but other people are responsible for approving its child forms, then you must wait for all the approvals to occur.

  If you are responsible for submitting a certain parent form and you also have sole responsibility for approving all its child forms, then you can submit the parent form as soon as all the child forms have been submitted for review. In this case, selecting **Submit** launches a special compound action called "Approve Children and Submit."

A bottom-up workflow ends when a form administrator completes the form set, using SAS Financial Management Studio. This removes all the forms of the form set from the Form Manager. The form administrator can complete a form set at any time; it is not necessary to wait for the bottom-up flow of data to reach the top of the hierarchy. If the form administrator completes a form set prematurely, he can reactivate the form set. This returns the forms to the Form Manager for further work.

Each time a user applies an action to a form, the status of the form changes. As a bottom-up workflow proceeds, each form traces a simple life cycle in which a series of bottom-up actions takes the form through a series of bottom-up status values. At all times, each form's current status value is shown in the **Status** column of the Form Manager page, and the valid actions are available on the pop-up menu.

See Also

- Bottom-Up Actions on page 300
- Bottom-Up Status Values on page 302

Bottom-Up Actions

A form in a bottom-up workflow is subject to the following actions:

Approve
: moves a form's data up into its parent form. An operational form always requires one approval by its one assigned reviewer. Depending on how a financial form set is defined, a form might require one approval by one reviewer or a sequence of approvals by two or more reviewers. If a sequence of approvals is required, then the form's data moves up into its parent form after the last required approval.

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 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.

Check In
: reconnects a checked-out financial form (Microsoft Excel file) with the server.

  The status of the form changes from Checked Out to Edited.

  This action is available only for financial forms that can be edited in Microsoft Excel. It is not available for any operational forms.

Check Out
: saves on your desktop a form that you are currently editing in Microsoft Excel.

  The status of the form becomes Checked Out. No one else can edit it.

  This action is available only for financial forms that can be edited in Microsoft Excel. It is not available for any operational forms.

Open
: opens a form in the Web-based Form Editor. Whether you can enter data or only view data depends on the status of the form.

  The first time someone opens a form, the form's status changes from Unedited to Edited. The first time someone opens a newly rejected form, the form's status changes from Rejected to Edited. Any form that has the status of Edited can be edited again without a change of status.

  This option is available for all operational forms and for financial forms that belong to a form set that is enabled for Web data entry.

Open in Excel
: opens a form in Microsoft Excel. Whether you can enter data or only view data depends on the status of the form.

  The first time someone opens a form, the form's status changes from Unedited to Edited. The first time someone opens a newly rejected form, the form's status changes from Rejected to Edited. Any form that has the status of Edited can be edited again without a change of status.

  This option is available for financial forms that belong to a form set that is enabled for Excel data entry. It is not available for any operational forms.

Recall
: 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.

Reject
: 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
: makes a form available for review.

  Submitting a form changes its status from Edited to Submitted.

  Submitting a form that has unapproved children also changes the status of each unapproved child from Submitted to Approved.

See Also

- Bottom-Up Status Values on page 302
- Description of the Bottom-Up Workflow on page 299

Bottom-Up Status Values

A form that is part of a bottom-up workflow has a life cycle that takes it through a series of status values. Here are the status values, listed in the order in which a form typically passes through them:

Unedited
: The form has not been edited yet.

  There is a longer version of this status that is displayed to a specific subset of users:

Unedited – ready to edit	You see this status if you are responsible for editing and submitting the form.

Checked Out
: Someone has the form checked out and saved on their desktop for offline data entry. You can open the form in read-only mode, but you cannot see changes that have been made to the offline copy until the form is checked in.

  Checking a form out for offline data-entry is an optional step for forms that are edited in Microsoft Excel. A form that remains connected to the server throughout its life cycle will never have this status value. When a checked-out form is checked back in, its status changes to Edited.

Edited
: The form has been edited but has not yet been submitted for review.

  There are two longer versions of this status that are displayed to specific subsets of users:

Edited – ready to submit	You see this version if you are responsible for editing and submitting the form and the software allows you to submit the form now.
Edited – awaiting child forms	You see this version if you are responsible for editing and submitting the form but the software does not allow you to submit the form now. The software does not allow you to submit a parent form that has at least one child that must first be submitted or approved by another user.

Submitted
: The form has been submitted for review but has not been approved or rejected yet.

  There is a longer version of this status that is displayed to a specific subset of users:

Submitted – ready to review	You see this version if you are responsible for reviewing the submitted form.

Rejected
: The form has been submitted for review and has been rejected by one of its reviewers. More editing and a resubmission are in order. A form can complete its life cycle without passing through this status.

  There is a longer version of this status that is displayed to a specific subset of users:

Rejected – ready to edit	You see this version if you are responsible for editing and resubmitting the rejected form.

Partially approved
: The form has more than one required reviewer. It has been submitted for review and has been approved by at least one reviewer, but it still needs to be approved by at least one other reviewer. In a form set that does not assign multiple reviewers to forms, a form can complete its life cycle without passing through this status.

  There is a longer version of this status that is displayed to a specific subset of users:

Partially approved – ready to review	You see this version if you are the next reviewer in line.

Approved
: The form has been submitted for review and has been approved by all required reviewers.

See Also

- Bottom-Up Actions on page 300
- Description of the Bottom-Up Workflow on page 299

Chapter 26

Data Entry: Top-Down Workflow

Description of the Top-Down Workflow 305
Top-Down Actions 306
Top-Down Status Values 307

Description of the Top-Down Workflow

A top-down workflow begins when a top-down form set is published from SAS Financial Management Studio. Publishing makes the top form of the top-down form set available in the Form Manager.

In a top-down workflow, data flows down the hierarchy of forms in the following way:

1. A user with data-entry responsibility for the top form enters total amounts into the top form, using the virtual child of the top member of the target hierarchy. The user then allocates these total amounts among the actual children of the top member.
2. Each user who has data-entry responsibility for a form that is a child of the top form allocates the amounts that were received by that child among the child's children. A virtual child can receive a share of an allocation, if it is in the table.
3. Data continues to flow down the hierarchy of forms level by level. For each form, the user who has data-entry responsibility for it allocates among the form's children the data that the form received from its parent. A virtual child can receive a share of an allocation, if it is in the table.
4. The top-down flow of data ends when all the bottom-level forms receive their allocations.

In each form, changes other than hierarchical allocations can be made, such as redistributing amounts among accounts, time periods, or members of other dimensions. All such changes are subject to the enforced constraint that the total monetary value in the form is constant. In bottom-level forms, only these other types of changes can be made because there is no lower level to allocate to.

The top-down flow of data is controlled by a repeating pattern of user actions, which you select on the Form Manager page:

- To enter data, you edit a form. You can edit a form as many times as necessary in order to get the numbers the way you want them.

- When you have finished editing a form that has children, you push the form. Pushing a form moves the data that you allocated to its children into the child forms. It also makes those child forms available for editing.
- As an alternative to pushing a form that has children, you can use the special “Push to All” action. This action copies allocated data into all the form's descendant forms, not just its immediate children. It also makes the results of the entire multi-level allocation final. There is nothing for any other user to do except view the results of the “Push to All” action.
- If you push a form and then decide that you need to make changes, you can recall the form, edit it some more, and then push it again. Recalling a form can be useful in certain circumstances, but it is not part of the mainstream workflow.
- When you have finished editing a form that has no children, you are done with it.

A top-down workflow ends when a form administrator completes the form set, using SAS Financial Management Studio. This removes all the forms of the form set from the Form Manager. The form administrator can complete a form set at any time; it is not necessary to wait for the top-down flow of data to reach the bottom of the hierarchy. If the form administrator completes a form set prematurely, he can reactivate the form set. This returns the forms to the Form Manager for further work.

Each time a user applies an action to a form, the status of the form changes. As a top-down workflow proceeds, each form traces a simple life cycle in which a series of top-down actions takes the form through a series of top-down status values. At all times, each form's current status value is shown in the **Status** column of the Form Manager page and the currently valid actions are available on the pop-up menu.

See Also

- Top-Down Actions on page 306
- Top-Down Status Values on page 307

Top-Down Actions

A form in a top-down workflow is subject to the following actions:

Check In
: reconnects a checked-out financial form (Microsoft Excel file) with the server.

  The status of the form changes from Checked Out to Edited.

  This action is available only for financial forms that can be edited in Microsoft Excel. It is not available for any operational forms.

Check Out
: saves on your desktop a form that you are currently editing in Microsoft Excel.

  The status of the form becomes Checked Out. No one else can edit it.

  This action is available only for financial forms that can be edited in Microsoft Excel. It is not available for any operational forms.

Open
: opens a form in the Web-based Form Editor. Whether you can enter data or only view data depends on the status of the form.

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.

Below the top level, editing is subject to the constraint that you cannot change the total monetary value that is allocated to the form that you are editing.

This option is available for all operational forms and for financial forms that belong to a form set that is enabled for Web data entry.

Open in Excel
: opens a form in Microsoft Excel. Whether you can enter data or only view data depends on the status of the form.

  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.

  Below the top level, editing is subject to the constraint that you cannot change the total monetary value that is allocated to the form that you are editing.

  This option is available for financial forms that belong to a form set that is enabled for Excel data entry. It is not available for any operational forms.

Push
: 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 from Edited to Pushed and its child forms appear in the Form Manager with the status of Unedited.

Push to All
: 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 from Edited to Pushed, and the status of all the descendant forms changes to Completed.

Recall
: 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 Form Manager all the subordinate forms that had received data as a result of the previous Push or Push to All.

See Also

- Description of the Top-Down Workflow on page 305
- Top-Down Status Values on page 307

Top-Down Status Values

A form that is part of a top-down workflow has a life cycle that takes it through a series of status values. Here are the status values, listed in the order in which a form typically passes through them:

Unedited
: The form has not been edited yet.

There is a longer version of this status that is displayed to a specific subset of users:

Unedited – ready to edit	You see this status if the form is available for editing and you are responsible for editing it.

Checked Out

Someone has the form checked out and saved on their desktop for offline data-entry. You can open the form in read-only mode, but you cannot see changes that have been made to the offline copy until the form is checked in.

Checking a form out for offline data-entry is an optional step for forms that are edited in Microsoft Excel. A form that remains connected to the server throughout its life cycle never has this status value. When a checked-out form is checked back in, its status changes to Edited.

Edited

The form has been edited but has not yet been pushed. A form also has this status if it has been edited, pushed, and recalled.

There is a longer version of this status that is displayed to specific subsets of users:

Edited – ready to push	You see this version if you are responsible for pushing the form.

Pushed

The form has either been simply pushed or pushed to all its descendants. If the form was simply pushed, then the data that was allocated to the form's children when the form was edited has been copied into the child forms. If the form was pushed to all its descendants, then the data that was allocated to the form's descendants when the form was edited has been copied into the descendant forms.

Completed

The form has received an allocation of data as a result of a “Push to All” action on a higher-level form. The result of the allocation can be viewed but not changed.

See Also

- Description of the Top-Down Workflow on page 305
- Top-Down Actions on page 306

Chapter 27

Reports in the Document Manager

Types of Reports **309**

Viewing Reports **309**
Overview of Viewing Reports 309
Viewing Financial Reports 310
Viewing Administrative Reports 310

List of Administrative Reports **310**

Types of Reports

SAS Financial Management produces two broad types of reports—financial reports and administrative reports.

Financial reports are for end users. All financial reports are designed with the SAS Financial Management Add-In for Microsoft Excel. They consist of SAS Financial Management read-only tables, SAS Financial Management CDA tables, and associated structures that can be created in Microsoft Excel. Typical examples of financial reports are income statements, balance sheets, and budget summaries. The number and variety of financial reports depends on the needs and imagination of your user community.

Administrative reports focus on specific aspects of the data that is stored in cycles and the processing that this data undergoes. Except for the Audit report, administrative reports apply only to financial cycles. You can run the Audit report for any cycle. All of the administrative reports have been designed at SAS and are presented as HTML pages. An administrator can use them to better understand how the financial reports are produced and to solve any problems that might arise.

Viewing Reports

Overview of Viewing Reports

To view a report, use the Document Manager. In the Document Manager, you can view any report that is in a folder that you have access to.

Viewing Financial Reports

Financial reports are published to Document Manager folders from the SAS Financial Management Add-In for Microsoft Excel. If you do not know which folder to look in, check with the person who publishes financial reports.

To open a financial report in Microsoft Excel:

1. Select the folder that contains the report.

   An alphabetized list of the reports in the selected folder is displayed.

2. Click the report name in the **Name** column.

   The report file opens.

Viewing Administrative Reports

The full set of SAS Financial Management administrative reports is in the **Products > SAS Financial Management > 5.2 Standard Reports** folder.

To run an administrative report and display the output:

1. Select the **Products > SAS Financial Management > 5.2 Standard Reports** folder.

   An alphabetized list of administrative reports is displayed.

2. Click the report name in the **Name** column.

   A set of fields for the report parameters appears.

3. Enter any appropriate parameters, and then click **Run**. All parameters are optional.

   The report runs and its output is displayed.

List of Administrative Reports

The following administrative reports are available. You can run the Audit report for any cycle, financial or operational. All other administrative reports apply only to financial cycles.

Audit
: lists actions that have been completed by the SAS Financial Management software.

  You can limit an Audit report to an object type, an initiating user, or a range of dates.

Data Entry
: lists data records that were entered through financial forms using a specified financial model.

  You can limit a Data Entry report to a time period, an organization, an analysis member, or a financial form set.

Eliminations
: lists, for a specified financial model, data records for all accounts that have the Intercompany attribute but that are not specified in any intercompany balancing rule or net intercompany balancing rule. There should not be any such accounts, so this report should not list any data records. If the report does list data records, then you need

to edit the rules that look for imbalances in intercompany accounts, or add more such rules.

You can limit an Eliminations report to a time period, an organization, or an analysis member.

ETL Facts
lists data records that have been loaded from SAS Data Integration Studio to a specified time period and analysis member within a specified financial cycle.

You can further limit an ETL Facts report to a specified organization.

Facts
lists data records that are associated with a specified financial model.

You can limit a Facts report to a time period or an analysis member, and in several other ways.

ICAccounts
lists, for a specified financial model, accounts that have the Intercompany attribute but that are not specified in any intercompany balancing rule or net intercompany balancing rule.

You can limit an ICAccounts report to accounts that belong to a particular account type or accounts that have a particular balance type.

Intercompany
lists, for a specified financial model, data records in which the account member has the Intercompany attribute and the trader member is either EXT or identical to the organization member. No records should satisfy this condition.

You can limit an Intercompany report to a time period, an organization, or an analysis member.

Manual Adjustments
lists all the currently posted manual adjustments for a specified financial model.

You can limit a Manual Adjustments report to a time period, an organization, or an analysis member. You can also limit the report to a range of adjustment amounts.

Non Intercompany
lists, for a specified financial model, data records in which the account member does not have the Intercompany attribute and the trader member is neither EXT nor identical to the organization member. No records should satisfy this condition.

You can limit a Non Intercompany report to a time period, an organization, or an analysis member.

Non Leaf
lists all the non-leaf data records (also known as virtual-child data records) for a specified financial model.

You can limit a Non Leaf report to a time period, an organization, or an analysis member.

Ownership Adjustments
lists all the adjustments that are generated by the ownership rule for a specified financial model.

You can limit an Ownership Adjustments report to a time period, an analysis member, a holding organization, or a held organization.

Ownership Methods
lists all the ownership relations that are specified in the ownership rule for a specified financial model, showing the consolidation method for each relation.

You can limit an Ownership Methods report to a holding organization, a held organization, or a consolidation method.

Ownership Transactions
: lists all the asset purchases and asset sales that are specified in the ownership rule for a specified financial model.

  You can limit an Ownership Transactions report to a holding organization, a held organization, or a transaction type.

Rule
: lists all the adjustments that are generated by a specified adjustment rule within a specified financial model.

  You can limit a Rule report to a time period or an analysis member.

Rules Facts
: lists all the adjustments that are generated by all the adjustment rules that are part of a specified financial model.

  You can limit a Rules Facts report to a time period, an organization, or an analysis member. You can also limit the report to a range of adjustment amounts.

Trial Balance
: lists data records that are associated with a specified financial model and that were loaded from the Detail Data Store.

  You can limit a Trial Balance report to a time period, an organization, or an analysis member.

In all administrative reports that list numeric data, credits are shown as negative numbers and debits are shown as positive numbers.

Index

A

About options 251
ABS function 181
access restrictions
  composite models 115
  cycles 55
  members 85
  models 85
  roles of users 14, 291
accessibility 209
Account dimension type 42
account types
  Asset 44
  Cumulative Translation Adjustment 44
  Equity 45
  Expense 45
  Liability 46
  overview 43
  Retained Earnings 46
  Revenue 47
  Statistical 47
  Statistical Balance 48
  Statistical Flow 48
ACOS function 181
Adjust Values option 251, 296
adjustment rules
  generation of adjustments from 156
  managing 98
  overview 98
  promoting 19
  threshold 93
Adjustment Rules view 99
adjustments
  balanced manual 95
  generation from rules 156
  manual 95
  overview 89, 90
  posting 90, 91, 92
  unbalanced manual 95
administrative reports 309, 310
ALL member of a trader hierarchy 54
Allocate option 222, 251
Allocate using predefined weights 145, 222
allocation in a top-down workflow 222, 251
allocation rules
  managing 98
  overview 106
  promoting 19
Allocation Weights option 252
Allow data to be entered for parent members option 218, 222
Analysis dimension type 50
ANCESTOR function 181
Approved status of forms 302
approving
  forms 299, 300
as-of date
  hierarchy in a composite model 114
  hierarchy in a model 84
  target hierarchy of a form set 117
ASIN function 182
Asset account type 44
ATAN function 182
Attach Modifications option 252
attachments
  composite models 115
  cycles 55
  models 85
Audit administrative report 310
authors
  assigning to forms 129
automatic allocation 145, 219, 294
  driver formulas 148
  eligible cells 145
  enabling 145
  example 146
  how values are distributed 146
  modeling formulas 148
  non-additive members 147

opening the Allocation Weights table 252
Reallocate option 265
reallocation 149
separate weight table 151
target hierarchy 148
weights 149
automatic locking of forms 134
Automatically Refresh Tables option 216, 252

B

balance account types 43
balance type property of accounts 48
balanced manual adjustments
managing 95
overview 95
bottom-up workflow 117
creating form sets 117
entering data in Excel 221
overview 290
reviewing data in Excel 222
status values 302
steps 299, 300

C

calculated members 243
Calculated Members option 253
calculated-member functions
fmCode 280
fmCXRate 67, 281
fmProperty 282
fmRate 77, 283
fmValue 284
fmXRate 67, 285
calculations using formulas 167, 273, 279
calendar templates 38
Calendar Templates view 39
CDA functions
CDACXRate 67, 274
CDADesc 274
CDAGet 211, 275
CDAName 275
CDAProperty 275
CDARate 77, 277
CDAXRate 67, 278
overview 273
CDA Table option 253
CDA tables
converting read-only tables to 246
converting to read-only tables 247
deleting 237
inserting 211, 236
CEIL function 182
Cell Information option 253
cell protection 138, 216, 247, 259, 263
and supplemental schedules 143
at the form level 143
rules 140
rules for a form set 141
rules for a model 140
Unprotect All Cells option 269
Unprotect Cell option 270
Change Description option 254
Check In Form option 223, 254, 300, 306
Check Out Form option 223, 254, 300, 306
Checked Out status of forms 302, 307
Clear Modifications option 255
Clear option 296
Clear Table's Modifications option 255
clearing from a table
comments 255
formats 255
CLOSINGPERIOD function 182
Collapse option 240, 255
column headings in tables 228
columns of a view
customizing 20
comments in tables 244, 252, 267
Completed status of forms 307
complex exchange rate types 68, 71
composite models
attachments 115
copying 116
creating 116
deleting 116
optional features 115
overview 113
properties 116
required features 114
security 115, 200
status 116
Composite Models view 116
COMPRESS function 183
consolidation methods in ownership rules 109
consolidation parents in ownership rules 108
Contributing Data option 256
Convert to CDA option 256
converting
CDA tables to read-only tables 247, 257
currencies 159, 160
frequencies 157, 160
Microsoft Excel formulas to CDA formulas 256
read-only tables to CDA tables 246, 256
Copy as CDA option 246, 256
Copy as Read-Only Table option 247, 257
Copy as Table option 257

Copy option 295
copying
  adjustment rules 98
  composite models 116
  cycles 59
  dimensions 27
  driver rate sets 79
  exchange rate sets 70
  hierarchies 26
  manual adjustments 95
  members 26
  models 86
COS function 183
cost method of consolidation 109
creating
  adjustment rules 98
  CDA tables 211, 236
  composite models 116
  custom properties 34
  cycles 60
  data-entry tables 214, 236
  dimension types 25
  dimensions 27
  driver rate sets 79
  exchange rate sets 70
  formulas 168
  hierarchies 26
  manual adjustments 95
  members 26
  models 86
  read-only tables 211, 236
  supplemental schedules 236
Credit balance type of accounts 48
Cross Industry Detail Data Store 15, 34
Cumulative Translation Adjustment
    account type 44
cumulative translation adjustment
    computations 163
Currency dimension type 51, 159, 160
current data locale 36
CURRENT function 180, 183
CURRENTNAME function 183
custom properties
  creating 34
  overview 34
Custom Properties view 35
customizing columns of a view 20
Cut option 295
cycles
  attachments 55
  copying 59
  creating 60
  data management for 62
  data record structure 4, 61
  deleting 59
  dimension types required 42
  loading data into 15, 62
  locking 16, 59
  number of 59
  opening 60
  overview 55
  properties 59
  security 55, 200
  types 57
  unlocking 16, 59

D

data entry
  See also data-entry tables
  automatic allocation 145, 294
  parent cells 294
Data Entry administrative report 310
data locales 36
Data Locales view 37
data management for cycles 62
data record structure 4, 61
Data-Entry Table option 257
data-entry tables
  automatic allocation 145, 219
  calculated members 243
  cell protection 138, 216, 247, 263
  comments in 244
  deleting 237
  entering data for parent members 218
  entering data into parent cells 222
  exporting values to metric tables 246
  frequency members 215
  inserting 214, 236
  pop-up menu 240
  time periods 135
  with supplemental schedules 152
DATE function 183
DATETIME function 184
DAYOFMONTH function 184
DAYOFWEEK function 184
DAYOFYEAR function 184
deadline 291
deadlines
  for form sets 133
  for forms 18
Debit balance type of accounts 48
default data locale 36
default read members in models 84
default write members in models 84
Delete Position option 258
deleting
  adjustment rules 98
  composite models 116
  cycles 59
  data from cycles 62, 64
  dimension types 25

dimensions 27
driver rate sets 79
driver rates 79
exchange rate sets 70
exchange rates 70
hierarchies 26
manual adjustments 95
members 26
models 86
tables 237
Derived exchange rate type 68
descendant selection rules 126
designing
financial reports 211
form templates 214
Detail Data Store 15, 34
dimension types
Account 42
Analysis 50
creating 25
Currency 51, 159, 160
deleting 25
Frequency 51, 157, 160
Organization 49
overview 24
properties 25
required in a financial cycle 42
required in an operational cycle 42
role in tables 237
Source 52
Time 53
Trader 54
Dimension Types view 25
dimensions
copying 27
creating 27
deleting 27
exporting 27
loading members into 15
members 26
overview 24
promoting 19
properties 27
Dimensions view 27
Dimensions workspace 11, 24
Direct Non-Transacted holding type 108
Direct Parent holding type 108
Direct Transacted holding type 108
Document Manager 208, 211, 213, 258, 309
Document option 258
documents
inserting 258
DRATE function 77, 184
Drill Down option 240, 258, 295
Drill Up option 240, 258, 295
driver formulas 169
driver rate sets
copying 79
creating 79
deleting 79
properties 79
Driver Rate Sets view 79
driver rate types 77
Driver Rate Types view 78
driver rates
deleting 79
editing 79
loading 15
overview 77
resolving conflicts 74
Driver Rates view 80

E

Edit Protection Rules option 259
Edited status of forms 302, 307
editing
adjustment rules 98
driver rates 79
exchange rates 70
forms 299, 300, 305, 306
manual adjustments 95
elimination of intercompany trades 156
Eliminations administrative report 310
Equity account type 45
equity method of consolidation 109
ETL Facts administrative report 311
exchange rate sets
copying 70
creating 70
deleting 70
properties 70
Exchange Rate Sets view 70
exchange rate type property of accounts 48
exchange rate types
complex 68, 71
overview 68
simple 68, 71
Exchange Rate Types view 70
exchange rates
deleting 70
editing 70
loading 15
overview 67
resolving conflicts 74
Exchange Rates view 71
EXP function 184
Expand All option 240, 259, 295
Expand option 240, 259
Expense account type 45
Export as Measure option 246, 259

Export to Excel option 295
exporting
  data records to staging tables 86
  dimensions 27
  values to metric tables 246
EXT member of a trader hierarchy 54

F

Facts administrative report 311
Filter Member Combination option 240, 260
Filters option 240, 260
financial cycles
  dimension types required 42
  overview 57
financial form sets
  member selection rules 125
  overview 118, 290
Financial Management: Form Approver role 14
Financial Management: Form Submitter role 14
Financial Management: Planning Data Entry User role 14
Financial Management: Process Administrator role 14
financial models 83, 85, 86
financial reports 309
FIRSTCHILD function 185
FIRSTSIBLING function 185
fixed time member rules 230
fixed time periods 135
floating time member rules 230
floating time periods 135
  in data-entry tables 214
  in financial reports 211
FLOOR function 186
flow account types 43
fmCode function 280
fmCXRate function 281
fmProperty function 282
fmRate function 283
fmValue function 284
fmXRate function 285
Form Approver role 118
Form Editor 293
form groups 128
Form Manager 291
form sets
  See also financial form sets
  See also operational form sets
  Autolock 134
  creating 122
  deadline 16
  deadlines 133
  descendant selection rules 126
  financial 118
  locking 16
  operational 118
  overview 118
  promoting 19
  publishing 122, 214
  target hierarchy 125
  types 290
  unlocking 16
Form Submitter role 118
form templates
  designing 214
  saving 214
  supplemental schedules 220
  validating for the Web 214
Format Members option 261
formats for table cells 244, 252, 267
forms
  approving 299, 300
  authors 129
  auto-lock option 135
  automatic locking 134
  bottom-up workflow 221, 299, 300
  checking in 223, 254, 300, 306
  checking out 223, 254, 300, 306
  data entry 221
  editing 221, 222, 299, 300, 305, 306
  entering data 221
  entering data for parent members 144
  financial 118
  locking 16
  operational 118
  overview 3, 118
  pushing 305, 306
  recalling 299, 300, 305, 306
  refreshing tables 216
  rejecting 299, 300
  reminders, sending 134
  reviewers 129
  reviewing 222, 299, 300
  status 297
  submitting 299, 300
  supplemental schedules 224
  top-down workflow 222, 305, 306
  unlocking 16
Forms workspace 11, 118
formulas
  creating 168
  displaying 253
  functions in 181
  member references 176
  overview 167
  resolving conflicts 179
  scope 175
  syntax 176

types 169
Frequency dimension type 51, 157, 160, 243
frequency members in data-entry tables 215
full consolidation 109
full consolidation with minority interest 109
functional currencies 50
functions 181, 243, 273

G
green cells 224

H
Help system 21, 266
Hide Member option 261
hierarchies
as-of date in a composite model 114
as-of date in a model 84
overview 24, 26
roll-up arithmetic 4, 164
target hierarchy of a form set 117
Hierarchies view 28
hierarchy view in the Form Manager 291
Historic exchange rate type 68
holding types in ownership rules 108

I
ICAccounts administrative report 311
IF function 180, 186
importing
data into cycles 15, 62
data locales 36
driver rates 15
exchange rates 15
members into a dimension 15
INDEX function 186
INDEXC function 186
Indirect holding type 108
inserting
CDA tables 211, 236
data-entry tables 214, 236
read-only tables 211, 236
supplemental schedules 220, 236
Intercompany administrative report 311
intercompany balancing rules
managing 98
overview 103
intercompany property of accounts 49
intercompany trades
elimination of 156
INTORG dimension type 49
ISCURRENT function 180, 187
ISLEAF function 187
ISLEAFCROSSING function 187
ISVIRTUALCHILD function 187

K
KPI Viewer 246

L
LASTCHILD function 187
LASTSIBLING function 188
LEFT function 188
LENGTH function 188
Liability account type 46
linking slicers between tables 245
list view in the Form Manager 291
loading
custom properties 34
data into cycles 15, 62
data locales 36
driver rates 15
exchange rates 15
members into a dimension 15
locales (data) 36
locking
cycles 16, 59, 62
form sets 16
forms 16
models 16, 85, 86
periods in a cycle 16, 62, 64
LOG function 188
Log Off option 208, 261
Log On option 208, 262
LOWCASE function 188

M
manual adjustments
managing 95
overview 95
Manual Adjustments administrative report 311
Manual Adjustments view 96
MAX function 189
MEAN function 189
Member Labels option 262
member properties in tables 237, 239
member selection rules
applying to financial form sets 125
for operational form sets 129
member-level security 202
members of dimensions and hierarchies
formula expressions 176
importing 15

loading 15
overview 24
virtual child 42, 49, 176
working with 26
Members page of the Table wizard 229
Members view 32
metric tables
exporting values to 246
MIN function 189
MOD function 189
Model page of the Table wizard 227
modeling formulas 169
models
attachments 85
composite 113
copying 86
creating 86
deleting 86
exporting data records to staging tables 86
locking 16, 85, 86
optional features 85
overview 83
required features 84
security 85, 200
selecting for tables 227
status 94
unlocking 16
warnings 94
Models view 86
Models workspace 11, 83

N

NESTIF function 180, 189
net intercompany balancing rules
managing 98
overview 104
New Position option 262
NEXT function 190
Non Intercompany administrative report 311
Non Leaf administrative report 311
NONE currency 51
None exchange rate type 68

O

object-level security 200
offline data entry 223, 254
opening cycles 60
OPENINGPERIOD function 190
operational cycles
dimension types required 42
overview 57
operational form sets
form groups 128
member selection rules 129
overview 118, 290
target hierarchy 127
operational models 83
Organization dimension type 49
organizations
functional currencies 50
ownership rule requirements for 108
reporting entities 50
Ownership Adjustments administrative report 311
Ownership Methods administrative report 311
ownership rules
consolidation methods 109
consolidation parents 108
holding types 108
holdings by geography 107
holdings by line of business 107
organization hierarchies for 108
overview 107
promoting 19
standalone parents 108
write members 107
Ownership Transactions administrative report 312

P

parent cells
entering data 144, 222
PARENT function 191
parent method of consolidation 109
Partially Approved status of forms 302
Paste option 295
performance scorecards 246
Period Average exchange rate type 68
Period Close exchange rate type 68
periods in a cycle
deleting data from 64
loading data into 64
locking 16, 64
properties 64
unlocking 16, 64
Periods view 64
Periods workspace 11, 64
PI function 191
Pivot option 240, 241, 263, 295
pivoting tables 241
Planning Data Entry User role 118
planning overview 6, 289
posting adjustments 86, 90, 91, 92
POW function 191
PREVIOUS function 191
promoting

adjustment rules 19
dimensions 19
form sets 19
Prompt for File Format option 263
properties
composite models 116
cycles 59
dimension types 25
dimensions 27
driver rate sets 79
exchange rate sets 70
hierarchies 26
members 26, 237, 239
organizations 50
periods in a cycle 64
tables 269
PROPERTY function 192
proportional method of consolidation 109
Protect Cell option 143, 247, 263
Publish option 211, 264
publishing
financial reports 211, 264, 309
form sets 214
Pushed status of forms 307
pushing
forms 305, 306

R

Rates workspace 11, 67, 77
Read-Only Table option 264
read-only tables
calculated members 243
collapsing 240
comments in 244
converting CDA tables to 247
converting to CDA tables 246
deleting 237
expanding 240
exporting values to metric tables 246
filtering 240
inserting 211, 236
pivoting 240
pop-up menu 240
Reallocate option 265
reallocation 149
recalling
forms 305, 306
reclassification rules
managing 98
overview 106
Refresh All option 216, 265
Refresh option 216, 265
refreshing forms 216
Rejected status of forms 302
rejecting
forms 299, 300
reminders
of form deadlines 134
REPEAT function 193
Repeat New Position option 265
Repeat Spread option 266, 296
report types 309
reporting entities 50
reporting formulas 169
reporting-entity balancing rules
managing 98
overview 105
reports
designing 211
floating time periods 211
overview 3
publishing 211, 264
sharing 211
viewing 213, 309
Retained Earnings account type 46
retained earnings computations 161
Revenue account type 47
REVERSE function 193
reversing entries for manual adjustments 95
reviewers
assigning to forms 129
reviewing
forms 222, 299, 300
RIGHT function 193
roles of users for security 14, 291
roll-forward method for retained earnings updates 46
roll-up arithmetic in hierarchies 4, 164
ROUND function 193
row headings in tables 228
Rule administrative report 312
Rules Facts administrative report 312
rules that generate adjustments 98, 156

S

salary planning 220
SAS Human Capital Management 220
SAS Strategy Management 246
Save All Supplemental Data option 266
Save Form Design option 267
Save Template option 214, 267
saving
comments for table cells 244, 252, 267
form templates 214
formats for table cells 244, 252, 267
supplemental schedule data 224
SCAN function 194
scope of a formula 175
scorecards 246

security
composite models 115
cycles 55
features in SAS Financial Management 199
for cycles 200
for models 200
member-level 202
members 85
models 85
object-level 200
object-level security 200
roles of users 14, 291
superuser access to cycles and models 202
sharing financial reports 211
Show Members option 240, 268
simple exchange rate types 68, 71
SIN function 194
slicers in tables
defining 228
linking 245
Source dimension type and hierarchy 52
Spread option 268, 296
SQRT function 194
staging tables
exporting data records to 86
standalone parents in ownership rules 108
Statistical account type 47
Statistical Balance account type 48
Statistical Flow account type 48
status
composite models 116
forms 297
models 94
styles for formatting table cells 244
Submitted status of forms 302
submitting
forms 299, 300
SUBSTR function 194
SUM function 194
Supplemental Schedule option 268
supplemental schedules
calculated members 243
cell protection 143
creating 152
creating a data-entry table 152
custom data provider 152
deleting 237
entering data 224
inserting 220, 236
overview 58, 118, 151, 219
runtime 153
SAS Human Capital Management data provider 152
saving data 224

T

Table Layout page of the Table wizard 228
Table Properties option 240, 269
Table wizard
Members page 229
Model page 227
Table Layout page 228
TABLEC function 195
TABLEN function 195
TAN function 195
target hierarchy
for financial form sets 125
for operational form sets 127
target hierarchy of a form set 117, 125
task pane 11
templates for forms
designing 214
refreshing tables 216
saving 214
supplemental schedules 220
validating for the Web 214
threshold
for adjustment rules 93
time dimension
fixed time member rules 230
floating time member rules 230
Time dimension type 53
TIME function 195
time periods
in data-entry tables 135, 214
in financial reports 211
time periods in a cycle
loading data into 15
TODAY function 195
TODEGREES function 195
top-down workflow 117
creating form sets 117
entering data in Excel 222
overview 290
status values 307
steps 305, 306
TORADIANS function 195
Trader dimension type 54
TRANSLATE function 195
treasury stock method of consolidation 109
Trial Balance administrative report 312
TRIM function 195

U

Unassign option 269
unbalanced manual adjustments
managing 95
overview 95
Unedited status of forms 302, 307

unlocking
  cycles 16, 59, 62
  form sets 16
  forms 16
  models 16, 85, 86
  periods in a cycle 16, 62, 64
Unprotect All Cells option 269
Unprotect Cell option 270
UPCASE function 196
user roles 14, 291

V
Validate Web Form option 214, 270
validating form templates for the Web 214
VERIFY function 196
View menu 11
views
  Adjustment Rules 99
  Calendar Templates 39
  Composite Models 116
  Custom Properties 35
  Data Locales 37
  Dimension Types 25
  Dimensions 27
  Driver Rate Sets 79
  Driver Rate Types 78
  Driver Rates 80
  Exchange Rate Sets 70
  Exchange Rate Types 70
  Exchange Rates 71
  Hierarchies 28
  Manual Adjustments 96
  Members 32
  Models 86
  overview 20
  Periods 64
virtual child members 42, 49, 176
VIRTUALCHILD function 196

W
warnings
  models 94
wizards
  overview 21
workflow of a form set 117, 289, 290
workflow types 117
workspaces
  Dimensions 11, 24
  Forms 11, 118
  Models 11, 83
  overview 11
  Periods 11, 64
  Rates 11, 67, 77
write members in ownership rules 107

X
XRATE function 196

Y
yellow cells 221, 222, 224, 293

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.