SAS® Financial Management 5.3

Process Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Financial Management 5.3: Process Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Financial Management 5.3: Process Administrator's Guide

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

Electronic book 1, April 2012

Contents

Who Should Read This Book?

Audience

This book is intended for SAS Financial Management administrators and power users who perform the following tasks:

- all SAS Financial Management Studio tasks
- creating and managing form sets for financial and operational planning
- creating and publishing financial reports

For information about Web data entry, see the *SAS Financial Management User's Guide*.

What’s New in SAS Financial Management 5.3

Overview

SAS Financial Management 5.3 includes the following changes and new features.

SAS Financial Management Web Applications

The form manager, form editor, and document manager have been replaced by SAS Web applications that are displayed with the Adobe Flash Player. When they log on to SAS Financial Management, users with appropriate capabilities can select workspaces for the following tasks:

- managing and editing forms
- viewing reports
- viewing security information

A related addition to SAS Financial Management is the Form Status dashboard, a graphical display of form status (for example, forms in progress or overdue). The dashboard can be viewed in a SAS BI Dashboard portlet (in the SAS Information Delivery Portal) or in the SAS BI Dashboard Viewer.

Planning and Process Management

- In a financial form, an administrator can generate a forecast that is based on historical data. SAS High-Performance Forecasting is used to generate the forecast data. If forecasting is enabled for data entry, a user who is editing a form can also generate a forecast.
- The functional currency of the Organization dimension for a crossing is always used for writing data, even if the target hierarchy is from another dimension.
- As part of the audit trail for budgeting and planning processes, author and date information is stored and is available via the contributing data records for a cell crossing.

- Microsoft Excel formulas can no longer be directly entered in a data-entry table. The **Calculated Values** option has been removed. Similar functionality is available via the **Calculated Member** option and supplemental schedules.
- A new table property enables users to enter data in parent members for the Time dimension. By default, this option is disabled.
- A new table property delays sending data to the server until a refresh action takes place. The default is automatic writeback.
- Forms can now be submitted without first being edited.
- Individual forms in the same form set can have different deadlines.
- Target members can be added to a form set after publish, without resetting and republishing the form set.
- Support for supplemental schedules:
  - SAS Financial Management includes a data provider for supplemental schedules.
  - Supplemental schedules are now available in Web data entry.
  - In a read-only table or data-entry table, users can drill through to supplemental data.

The following features apply both to data entry and reports:

- Comments can be attached to cell crossings in a report or a form. Depending on their scope, cell comments can be included in financial reports.
- Member properties can be used to filter the displayed members in a table.

  Member selection rules can no longer be applied to member properties. Instead, property selection rules are applied to the associated dimension members.
- Dimension members can be reordered in tables, within the boundaries of their parent in the displayed hierarchy.
- Filter member combination is now supported in Web data entry, as well as in the SAS Financial Management Add-in for Microsoft Excel.

Reporting

- The SAS Financial Management Add-in for Microsoft Excel includes enhanced support for Excel charts. One use for these charts is to display forecasted data.
- From a SAS Financial Management report, a user can create an information map that can be opened in SAS Web Report Studio as a dynamic report.
- There are several new features for publishing financial reports:
  - Financial reports can be published as dynamic Excel files or as static Excel files, PDFs, or SAS Web Report Studio reports.
  - Static reports can be published to a SAS folder or a local folder, or they can be sent as e-mail attachments. There can be separate outputs for different hierarchy members.
  - Report settings can be saved and used to generate scheduled reports.

Administration

- Dimension member security includes separate Read and Write permissions.
- Object-level security is extended to hierarchies, dimensions, dimension types, and custom properties. Administrators can set security permissions for multiple objects (of the same kind) at the same time. Security settings can be exported and imported along with a dimension.
- A security report displays the level of access that users and groups have to dimension members and objects such as cycles, models, and dimensions. The report can be viewed online in the Administration workspace of SAS Financial Management on the Web. It can also be saved as a CSV file.
- It is possible to update the default read and write members programmatically for the time hierarchy for a specified model.
- Validation can be applied to custom properties.
- Administrators can lock specific analysis members for one or more periods. They do not need to lock the entire period.
- Administrators can delete a contiguous block of periods at the end of a cycle, if the periods are not locked and contain no data.
- Models can be added to or removed from a composite model.
- It is possible to copy an operational form set. (The form set data is not included in the copy.)
- Member selection rules were already used during form set creation. They are now used in many other operations in SAS Financial Management Studio, including formula scoping and wizards for loading and deleting data.
- In the previous release, adjustments that are generated by certain adjustment rules were subject to a materiality threshold that ignores very small values that are considered immaterial. This materiality threshold now applies to additional adjustment rules, automatic allocation, and explicit allocation in forms and form templates, and driver formulas. A JVM option, **`-Dfms.threshold.materiality`**, replaces the **`-Dfms.threshold.allocation`** option from the previous release.
- Formula scope can include property values as well as dimension members.
- Driver formulas now support the following:
  - cross-dimension references for all dimensions at writeback
  - execution in the context of a form and not the entire hierarchy
- SAS Financial Management supports capabilities, defined in metadata, that permit specific actions such as submitting a financial form or administering a dimension. A basic set of roles with these capabilities is provided. Sites can use these roles or create and assign additional roles to meet their needs.
- SAS Financial Management supports the JBoss application server.
- SAS Financial Management supports the Linux operating system.

For additional information, see “New Data Administration Features in SAS Financial Management” in the *SAS Financial Management: Data Administrator’s Guide*.

Documentation Changes

The following books are new (or have new titles) in SAS Financial Management 5.3. They are available on the SAS Financial Management documentation site at `http://support.sas.com/documentation/onlinedoc/fm/index.html`.

- *SAS Financial Management: User's Guide*

  This book is designed primarily for end users. It is also available from the Help menu to users who are logged on to SAS Financial Management in a Web browser.

- This book, *SAS Financial Management: Process Administrator's Guide*

  In previous releases, this book was known as the user's guide.

- *SAS Financial Management: Performance Guide*

  This document explains how to tune SAS Financial Management for performance and scalability.

For more documentation titles, see "Additional Documentation" on page 6.

Part 1

Introduction

Chapter 1
Overview of SAS Financial Management

SAS Financial Management Activities

SAS Financial Management is an advanced system for financial reporting and for financial and operational planning. It is designed to support the following financial management activities:

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

Client Applications

Users interact with SAS Financial Management by means of these client applications:

- In SAS Financial Management Studio, administrators build and manage the infrastructure that is required by the other two clients.
- The SAS Financial Management Add-In for Microsoft Excel connects a desktop copy of Microsoft Excel to the SAS Financial Management database. With this client, users build and view financial reports and enter data for financial planning purposes. Administrators also use this client to create form templates for financial and operational planning.
- From a Web browser, administrators and end users can log on to a SAS Financial Management Web application that is displayed with the Adobe Flash Player. The application includes a workspace for managing and editing data-entry forms for financial and operational planning. The flow of data is controlled by a workflow that is defined in SAS Financial Management Studio.

  The application also includes workspaces for accessing financial reports and (for administrators) viewing security information.

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

SAS Financial Management Data

Cycles and Dimension Types

In SAS Financial Management, a cycle is a structured pool of stored data. There are two types of cycles: financial and operational.

In a financial cycle, there are seven required dimension types:

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

In an operational cycle, there are only five required dimension types:

- Account
- Analysis
- Currency
- Organization
- Time

To meet its needs, a site can define additional dimension types for both financial and operational planning cycles.

Dimension Members and Crossings

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

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

Hierarchies

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

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

Models

A model is a structure for viewing and interacting with the data in a cycle. There are both financial and operational models.

A model is the basis of both forms and reports. Its structure includes a set of hierarchies (from required and optional dimension types) as well as rates and formulas. A financial model can also have its own set of adjustments and rules.

Comparing Financial and Operational Planning

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

- Administrators define form sets in SAS Financial Management Studio.
- Each form set has a target hierarchy and is associated with a bottom-up or top-down workflow.
- Administrators 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 flow of data is controlled by a bottom-up or top-down workflow.

However, there are many differences between financial and operational planning, including the following:

- Financial planning data is stored in financial cycles. Operational planning data is stored in operational cycles.
- Financial cycles require the Source and Trader dimension types. Operational cycles do not.
- You can load data from an operational cycle into a financial cycle, but not vice versa.
- A financial model can include manual adjustments and adjustment rules, provided that these options are enabled in the associated cycle. Operational models do not support manual adjustments and adjustment rules.
- 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.
- In the SAS Financial Management Add-In for Excel, read-only tables and CDA tables can be based only on financial models, which retrieve data from a financial cycle. Operational models are not available for selection when you create these tables.

Additional Documentation

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

- *SAS Financial Management: User's Guide*

  This book is designed primarily for end users and is also available from the Help menu to users who are logged on to SAS Financial Management in a Web browser.
- *SAS Financial Management: Process Administrator's Guide*

  In previous releases, this book was known as the *SAS Financial Management: User's Guide*
- *SAS Financial Management: System Administrator's Guide*
- *SAS Financial Management: Migration Guide*
- *SAS Financial Management: Customization Guide*
- *SAS Financial Management: Data Administrator's Guide*
- *SAS Financial Management: Data Model Reference*
- *SAS Financial Management: Formula Guide*
- *SAS Financial Management: Performance Guide*

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

For information about the SAS Intelligence Platform, see **http://support.sas.com/93administration**.

For information about administering third-party software, such as the Web application servers, see **http://support.sas.com/resources/thirdpartysupport/v93**.

Part 2

Administering SAS Financial Management

Chapter 2
Getting Started with SAS Financial Management Studio

Working in SAS Financial Management Studio

Overview

SAS Financial Management Studio has five workspaces:

- **Dimensions:** for defining dimensions, hierarchies, members, and custom properties
- **Periods:** for managing the time periods and the data in the open cycle
- **Rates:** for managing exchange rates and driver rates

- **Models:** for defining and managing models, composite models, manual adjustments, adjustment rules, and cell protection rules
- **Forms:** for creating and managing form sets

The objects in the Dimensions workspace apply to all cycles. The objects in the other four workspaces apply to the cycle that is currently open.

To select a cycle, select **Tools** ⇨ **Cycle Manager** or use the cycle options on the **File** menu.

Dimensions Workspace

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

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

Periods Workspace

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

In the Periods workspace, you can perform these tasks:

- load data from SAS Data Integration Studio
- load data from another model (available only in financial cycles)
- load operational data (available only in financial cycles)
- delete data
- delete cell comments
- lock time periods

For details about loading data using SAS Data Integration Studio, see *SAS Financial Management: Data Administrator's Guide*. (See “Additional Documentation” on page 6.)

Rates Workspace

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

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

- view and manage exchange and driver rates
- load exchange rates and driver rates
- copy rate sets
- delete rate sets

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

Models Workspace

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

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

- create and manage manual adjustments
- create and manage adjustment rules such as intercompany rules, allocation rules, reclassification rules, and ownership rules
- post adjustments
- view and manage cell protection rules

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

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

Forms Workspace

Use the Forms workspace to create and manage form sets. A form set is a collection of forms, which are primarily used for data entry.

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

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

User Roles and Capabilities

Capabilities for SAS Financial Management Studio

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

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

Roles for Editing and Approving Forms

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

Loading Data to the SAS Financial Management Data Mart

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

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

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

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

  Corresponding task in SAS Data Integration Studio: Load Dimension.

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

  This option launches the Load New Data wizard, which has different versions for financial and operational cycles. Loading data into a cycle from the SAS Financial Management staging area is part of the job of managing the data in a cycle. There are other ways in which data can enter a cycle:

  - entering data in a form
  - loading data from one financial cycle to another with the **Load Model Data** option
  - data from an operational cycle to a financial cycle using the **Load Operational Data** option

  Corresponding tasks in SAS Data Integration Studio: Load Base Data and Load OP Data.

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

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

  This option launches the Load Exchange Rates wizard.

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

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

  This option launches the Load Driver Rates wizard.

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

For details, see the online Help for each wizard.

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

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

See Also

- “Managing Data in a Cycle” on page 69
- “Promoting Objects to Another System” on page 16

Locked Objects

Overview of Locked Objects

SAS Financial management supports the locking of the following objects:

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

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

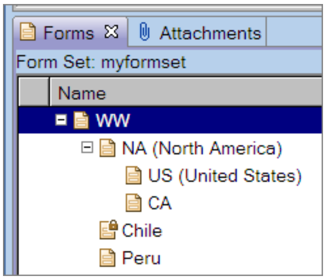
To unlock an object, reverse the process. If you locked a form set with the **Lock** button, click the **Unlock** button to unlock it.

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

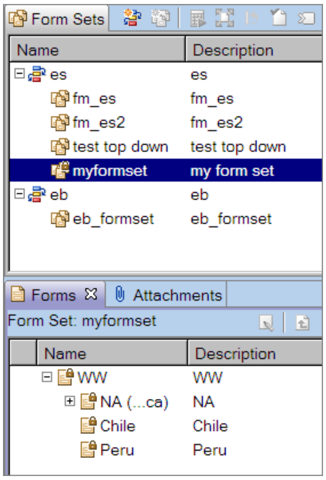
In some cases, an object is affected by more than one lock. For example, there might be a lock on a form, the related form set, and the related cycle. To undo those locks, begin by unlocking the most comprehensive lock. The subordinate locks still apply until you unlock them.

Here is an example:

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



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



Name	Description
⊟ es	es
fm_es	fm_es
fm_es2	fm_es2
test top down	test top down
myformset	my form set
⊟ eb	eb
eb_formset	eb_formset

Form Set: myformset

Name	Description
⊟ WW	WW
⊞ NA (...ca)	NA
Chile	Chile
Peru	Peru

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

Locked Cycles

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

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

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

Locked Cycle Periods

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

Note: Locking a cycle period does not affect the management of exchange rate and driver rate sets.

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 Analysis Members of Cycle Periods

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

Note: Locking an analysis member of a cycle period does not affect the management of exchange rates or driver rates.

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

- Open the period’s properties and select the **Analysis Member Locks** tab. Select the analysis members to lock.
- Select multiple periods, right-click, and select **Properties**. In the Analysis Member Locks window, select the analysis members to lock.

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

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. A locked form set implicitly locks the associated forms so that no data entry can be performed.

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, data entry is not permitted. 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 form set deadline.

Promoting Objects to Another System

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

The following object promotion facilities are available:

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

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

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

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

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

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

- **Financial form sets: 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 each wizard.

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

Using Views

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

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

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

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

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

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

Using the Help System

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

Help is context-sensitive:

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

A list of related topics appears.

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

Chapter 3
The Dimensions Workspace

Introduction to Dimensions, Hierarchies, and Members

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

- Dimension types represent the categories that are used to define your data. Some basic dimension types are Organization, Account, Time, Analysis, and Currency.
- A dimension is a set of members. Multiple dimensions can share the same dimension type.

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

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

Working with Dimension Types

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

The following dimension types are predefined:

Dimension Type	Description
Account	Account
Analysis	Analysis
CostCtr	Cost center
ExtOrg	External organization
Frequency	Frequency
IntOrg	Organization
ItemCat	Item category
Periods	Periods
ProfitCtr	Profit center
Source	Source

Dimension Type	Description
Time	Time
Trader	Trader

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

The following options are available:

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

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

To create a new dimension type that can be used to load data from the SAS Financial Management staging area, you must also 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 Financial Management: Data Administrator's Guide*.

Delete
deletes the selected dimension type.

Note: Deletion is possible only if the dimension type is not referenced by another object.

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

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 16

Working with Dimensions, Hierarchies, and Members

To work with dimensions, use the Dimensions view.

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

To work with hierarchies:

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

The Hierarchies view for the selected dimension opens.

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

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

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

To work with a member outside a hierarchy context:

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

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

See Also

- "Dimensions View" on page 22
- "Hierarchies View" on page 23
- "Members View" on page 26

Dimensions View

The Dimensions workspace includes these views:

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

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

The following options are available:

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

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

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

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

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

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

Delete
deletes the selected dimension.

Note: Deletion is possible only if the dimension is not referenced by another object.

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

Hierarchies
opens the Hierarchies view for the selected dimension.

Members
opens the Members view for the selected dimension.

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

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

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

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

Refresh
refreshes the view.

See Also

- "Using Views" on page 16
- "Working with Dimensions, Hierarchies, and Members" on page 21
- "Working in SAS Financial Management Studio" on page 9

Hierarchies View

Overview of the Hierarchies View

Hierarchies view displays the hierarchies in a dimension.

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

Hierarchy Options

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

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

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

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

You can 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.

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

Copy 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 with the following option: **Delete its members (that are not in another hierarchy)**. If you leave this check box empty (the default), the selected hierarchy is deleted from the dimension, but not its members.

If you select the check box, the selected hierarchy and all members that do not belong to another 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 (and in some cases edit) the properties of the selected hierarchy.

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

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

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

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

Refresh
refreshes the view.

Member Options

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

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

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

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

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

Collapse All
collapses all hierarchies completely.

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

Properties

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

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

See Also

- "Using Views" on page 16
- "Working with Dimensions, Hierarchies, and Members" on page 21

Members View

Overview of the Members View

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

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

Available Options

The following options are available:

New Member

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

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

Copy Here

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

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

Delete

deletes the selected member or members from the selected dimension.

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

Hierarchies

opens the Hierarchies view for the selected dimension.

Properties

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

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

Filter

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

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

Customize Columns

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

Refresh

refreshes the view.

See Also

- “Using Views” on page 16
- “Working with Dimensions, Hierarchies, and Members” on page 21

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 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 be assigned values for the custom property.

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

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

Ways to Define Custom Properties

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

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

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

Review or Define Custom Properties

To review existing custom properties or to define new ones:

1. Select **Tools** ⇨ **Custom Properties**.
2. Use the Custom Properties view.

Add Custom Property Values to Members

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

- Select one or more members, and then select **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 28

Custom Properties View

The Custom Properties view lists the existing custom properties.

The following options are available:

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

Delete
deletes the selected custom properties.

Properties
displays the properties of the selected custom property.

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

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

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

Refresh
refreshes the view.

See Also

- "Using Views" on page 16
- "Using Custom Properties" on page 27

Using Data Locales

How Data Locales Work

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

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

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

- language
- country or region

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

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

1. The language and the country/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 current data locale is the one that best matches the user’s regional settings. It is identified in the bottom margin of the main window of SAS Financial Management Studio.

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

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

Defining Data Locales

You can define data locales in two ways:

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

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

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

See Also

“Data Locales View” on page 30

Data Locales View

The Data Locales view lists the available data locales.

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

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

The following options are available:

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

Set As Default
makes the selected data locale the default data locale.

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

Delete
deletes the selected data locale.

Properties
displays the properties of the selected data locale.

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

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

Refresh
refreshes the view.

See Also

- "Using Data Locales" on page 29
- "Using Views" on page 16

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 SAS Financial Management staging area, then you do not need calendar templates.

Create a Calendar Template

To create a calendar template:

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

Add Time Periods to a Hierarchy with a Calendar Template

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

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

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

Add Time Periods to a Dimension with a Calendar Template

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

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

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

See Also

"Calendar Templates View" on page 32

Calendar Templates View

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

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

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

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

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

Delete
deletes the selected calendar templates.

Properties
displays the properties of the selected calendar template.

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

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

Refresh
refreshes the view.

See Also

- "Using Calendar Templates" on page 31

- “Using Views” on page 16

Chapter 4

Required Dimension Types

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 36
- Analysis on page 42
- Currency on page 42
- Organization on page 41
- Source on page 43
- Time on page 44
- Trader on page 45

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 on page 36
- Analysis on page 42
- Currency on page 42
- Organization on page 41
- Time on page 44

The Source and Trader dimension types do not apply to operational cycles.

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.

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

Introduction to the Account Dimension Type

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

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

- account type
- balance type
- exchange rate type

- intercompany property

There are some exceptions:

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

Typically, the members of an account hierarchy represent general ledger accounts and categories of general ledger accounts. A typical account hierarchy has several top-level members that represent the major categories of accounts, such as Assets, Liabilities, Revenue, and Expenses. Leaf members of an account hierarchy represent the accounts for which data is loaded from the SAS Financial Management staging area or entered through forms.

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

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

Introduction to Account Types

One of the properties of an account is the account type that it belongs to. The account type of a given crossing's account can affect 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 37
- Liability on page 38
- Equity on page 38
- Statistical Balance on page 40

The flow account types are the following:

- Revenue on page 39
- Expense on page 38
- Statistical Flow on page 40

The special account types are the following:

- Cumulative Translation Adjustment on page 38
- Retained Earnings on page 39
- Statistical on page 39

Asset Account Type

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

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

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

Cumulative Translation Adjustment Account Type

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

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

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

Equity Account Type

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

Expense Account Type

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

Liability Account Type

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

Retained Earnings Account Type

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

This account type has the following properties:

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

Roll-Forward Method
: specifies the period of time for which to calculate the retained earnings activity.

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

See Also

"Introduction to the Source Dimension Type" on page 43

Revenue Account Type

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

Statistical Account Type

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

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

See Also

"Overview of Formula Types and Formulas" on page 109

Statistical Balance Account Type

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

Statistical Flow Account Type

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

Balance Type Property of Accounts

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

Debit
indicates that the account normally has a debit balance.

Credit
indicates that the account normally has a credit balance.

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

Exchange Rate Type Property of Accounts

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

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

Intercompany Property of Accounts

The intercompany property of an account determines whether the account should be examined by intercompany balancing rules and participate in the elimination process. The elimination process depends on the Source and Trader hierarchies, which are required in all financial cycles but not relevant for operational cycles.

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

Yes

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

No

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

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

- Eliminations
- ICAccounts
- Intercompany
- Non Intercompany

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

Introduction to the Organization Dimension Type

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

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

See Also

- “Organization Details” on page 42
- “Ownership Rules: Organization Hierarchy Requirements” on page 98

Organization Details

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

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

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

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

- All facts that are entered into a form or form set template are stored in the functional currency that is assigned to the organization member for the crossing.
- All facts that are entered for manual adjustments and rules, with the exception of allocation rules, are stored in the functional currency that is assigned to the organization member for the crossing.

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

Introduction to the Analysis Dimension Type

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

Analysis members are typically used to identify different sets of data used for comparison purposes. Examples include Actual, Forecast, and Budget.

Introduction to the Currency Dimension Type

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

If you use more than one currency member, then you need to maintain exchange rates via 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.

See Also

"Organization Details" on page 42

Introduction to the Frequency Dimension Type

The Frequency dimension type contains a single Frequency dimension with a single predefined, flat Frequency hierarchy. Each member of the Frequency dimension is characterized by 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**. Although the Frequency hierarchy is flat, you can reorder its members.

All facts are stored as Periodic Activity, regardless of the frequency that is used when they are entered or loaded.

In a form, facts can be entered using the PA, PTD, or YTD frequency.

Facts can be loaded using the PA or YTD frequency.

For all rules, facts are queried using the PTD frequency.

Introduction to the Source Dimension Type

The Source dimension type contains a single Source dimension, 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 reference the Source dimension type.

Calculated members cannot be assigned to members of the Source dimension.

Each member of the Source dimension represents the source that the data was derived from.

The Source dimension members are as follows:

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

          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 values entered in the Base or BaseJourn Source dimension member..

          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 balances.
- **TotalBeforeElim**: the subtotal that combines all base data, all manual adjustments, and rules.
    - **Adj**: the subtotal of all manual adjustments and rules.
        - **CPO**: adjustments that are generated by ownership rules.
        - **Alloc**: adjustments that are generated by allocation rules.
        - **Reclass**: adjustments that are generated by reclassification rules.
        - **Bal**: adjustments that are generated by balancing rules of all types.
        - **Manual**: manual adjustments.
    - **TotalBeforeAdj**: the subtotal of BaseForm and TotalAfterImport values.
        - **BaseForm**: the values entered via Data Entry, loaded from Operational Planning, and loaded from models and the SAS Financial Management staging area.
        - **TotalAfterImport**: the subtotal of the subordinate BaseJourn and Base values.

          BaseJourn: represents the values loaded from the GL_JRNL_DETAILS table in the SAS Financial Management staging area.

          Base: the values loaded from the GL_TRANSACTION_SUM table in the SAS Financial Management staging area.

Introduction to the Time Dimension Type

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

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

Calculated members are not supported in the Time dimension.

See Also

"Using Calendar Templates" on page 31

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 reference the Trader dimension type.

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

- EXT: typically used for external trade values. This is the default write member assigned in the Trader dimension.
- ALL: the sum of all Trader dimension members, including EXT.

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

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

See Also

"Introduction to the Organization Dimension Type" on page 41

Chapter 5
Working with CTA Accounts

Creating a CTA Account: Best Practices

Overview

The Cumulative Translation Adjustment (CTA) account type references values from its source accounts and renders the translation differences between the source account’s exchange rates and the Period Close exchange rate of the CTA account.

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

When you create a Cumulative Translation Adjustment account, we recommend the following selections

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

Source Accounts

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

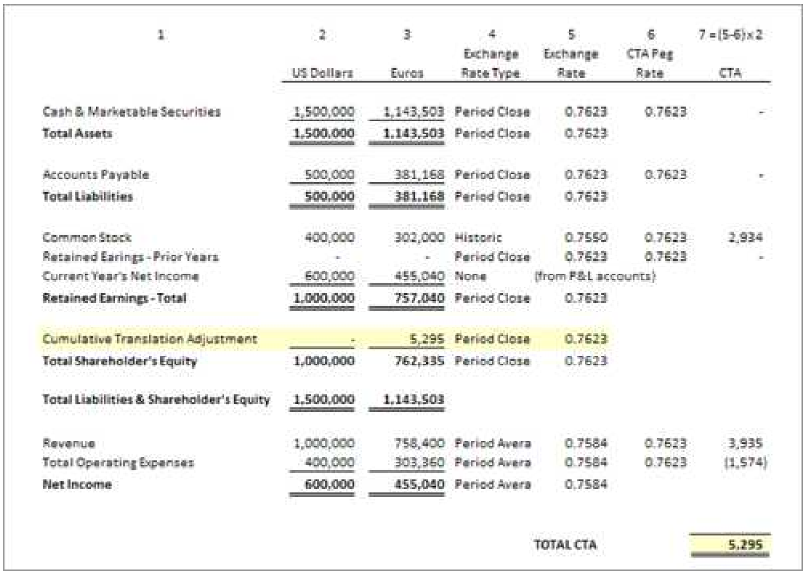
You cannot create a CTA account without at least one source account. Consequently, you should create CTA accounts after creating other accounts in the balance sheet. If necessary, you can modify source accounts selections later in the account properties window.

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

CTA Calculation

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

Figure 5.1 *Simple CTA Calculation*



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

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

The CTA calculation happens automatically and is not displayed to the user. However, the example above shows the data elements used in the CTA calculation to provide an understanding of how the calculation works. The columns in the report are explained in the following table.

Column	**Contents**
Column 1	Account descriptions.
Column 2	The balances for an entity with a functional currency of US Dollars. The balance sheet balances in the functional currency and there is no Cumulative Translation Adjustment.

Column	Contents
Column 3	The US Dollars balances of column 2, translated to euros (EUR). Note that there is a Cumulative Translation Adjustment amount and the balance sheet balances in euros.
Column 4	A CDAProperty query of each account's exchange rate type. The exchange rate type is used in column 5.
Column 5	Using the exchange rate type of each account (from column 4), a CDAXRate query is used to look up the exchange rate for each account.
Column 6	The exchange rate that the CTA account uses as the peg rate. The peg rate is considered to be the Period Close rate that is assigned for the period.
Column 7	Recalculates CTA using the elements surfaced on the report, to prove the value calculated by SAS Financial Management. It uses columns 2, 5, and 6 in a formula: `(5 - 6) x 2 = CTA Value`

Other Considerations for CTA Accounts

Balanced Balance Sheet

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

Writing Data to CTA Accounts

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

Multiple CTA Accounts on Balance Sheet

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

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

- Sales : -10.00 USD
- Cash & Cash Equivalents: 5.00 USD

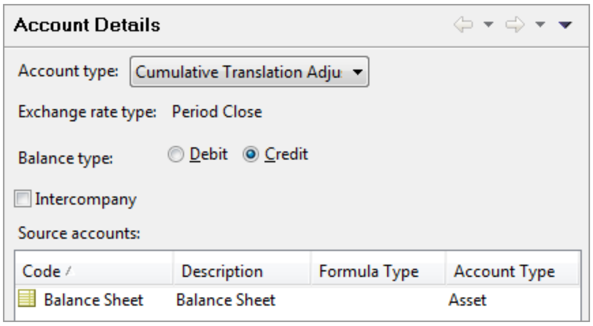
- Investment in Subs: 5.00 USD

The exchange rates for Jan 2011 are as follows:

- Period Average EUR/USD: 0.7584
- Period Close EUR/USD: 0.7623
- Historical Rate, Inv in Sub EUR/USD: 0.7600

The source account for the CTA account is as follows:

Figure 5.2 *CTA Account Details*



Account Details

Account type: Cumulative Translation Adju

Exchange rate type: Period Close

Balance type: ○ Debit ● Credit

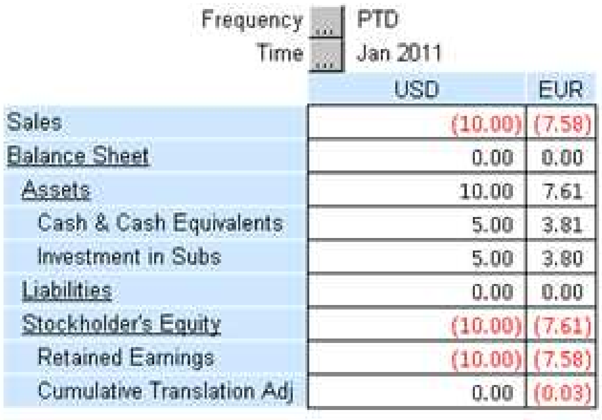
□ Intercompany

Source accounts:

Code	Description	Formula Type	Account Type
Balance Sheet	Balance Sheet		Asset

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

Figure 5.3 *Single CTA Account*

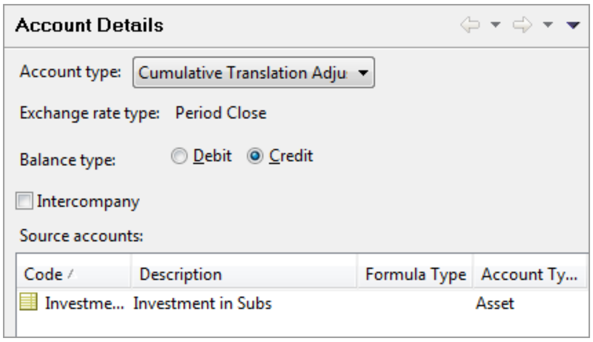


Frequency PTD
Time Jan 2011

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

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

Figure 5.4 CTA Account Details



Account Details

Account type: Cumulative Translation Adju

Exchange rate type: Period Close

Balance type: ○ Debit ● Credit

□ Intercompany

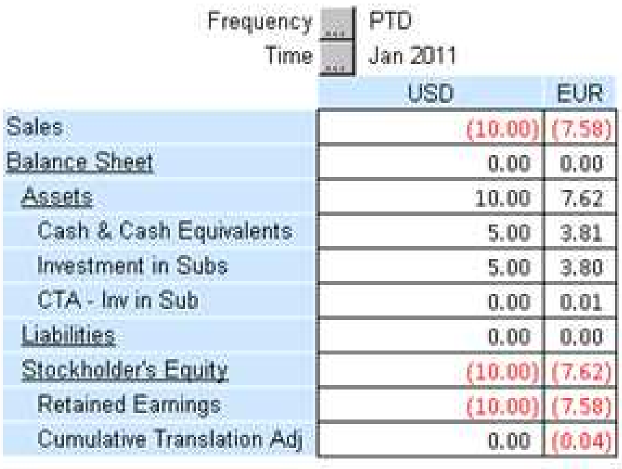
Source accounts:

Code	Description	Formula Type	Account Ty...
Investme...	Investment in Subs		Asset

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

Note: The Cumulative Translation Adj account that is designed to balance the balance sheet is now -0.04 euros and the CTA- Inv in Sub account is 0.01 euro.

Figure 5.5 Multiple CTA Accounts

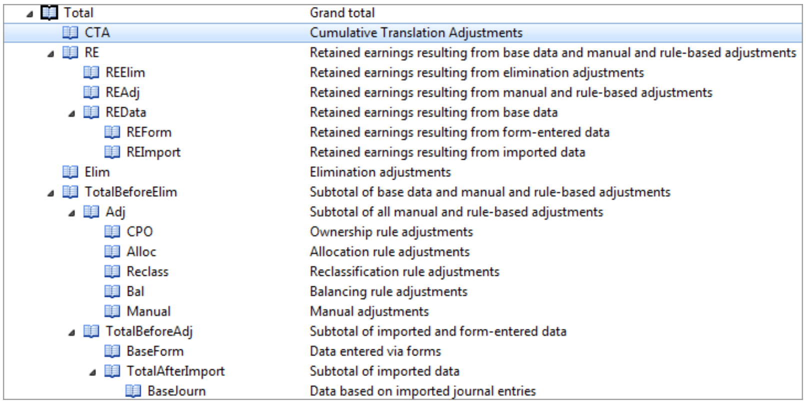


Frequency PTD
Time Jan 2011

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

Queries and the Source Dimension

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



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

If a user imports source system data to the Source member Base and creates a balance sheet that displays only Base data, the values that are generated by the CTA accounts are not included, because they appear at a higher point in the Source dimension hierarchy.

Chapter 6

Working with Retained Earnings Accounts

Retained Earnings: A Hybrid Account

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

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

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

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

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

Creating a Retained Earnings Account

Overview

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

The following selections are recommended:

- **Balance Type**: Credit

  This setting does not affect Retained Earnings accounts.
- **Intercompany**: No

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

Source Accounts

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

Note: For multiple hierarchies with varying members, ensure that the proper accounts are selected.

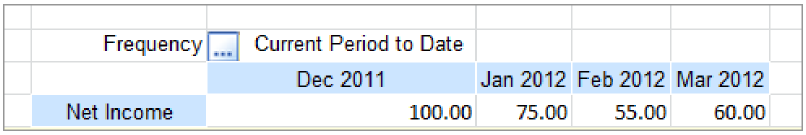
Roll-Forward Method

In the roll-forward method selection, you choose the time span over which the Retained Earnings account accumulates data. This selection enables Retained Earnings results to be segmented into current and prior year values, if desired. The following roll-forward methods are available:

Each Retained Earnings account can have only one roll-forward method, although you can create multiple Retained Earnings accounts with different roll-forward methods. Several options are available:

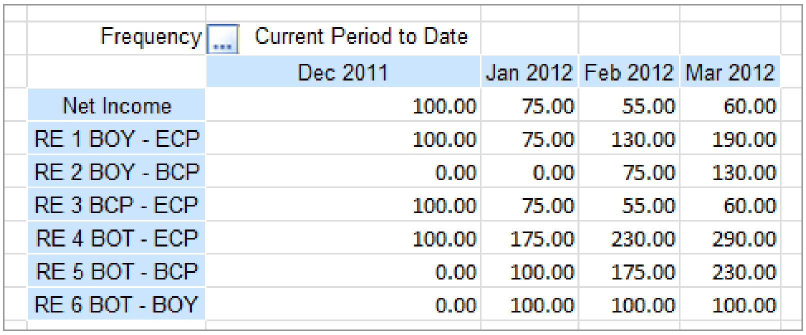
- You can choose a method that conforms to the way that retained earnings from the source system is rolled forward.
- You can group retained earnings by time (for example, into Current and Previous Years).
- You can tell SAS Financial Management to automatically calculate the value of Retained Earnings.

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



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

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



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

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

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

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

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

1. Rollforward method 4: **From beginning of time through end of current period**
2. Rollforward methods 6 and 1: **From beginning of time to beginning of year** and **From beginning of year through end of current period**
3. Rollforward methods 5 and 3: **From beginning of time to beginning of current period** and **From beginning of current period through end of current period**

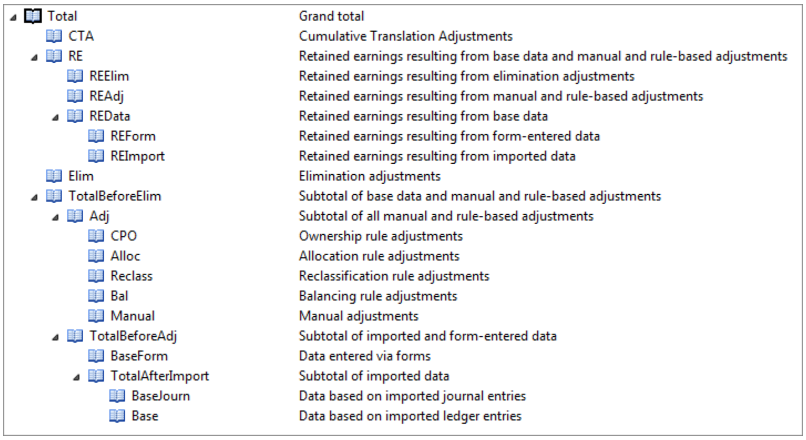
4. Rollforward methods 6, 2 and 3: **From beginning of time to beginning of year**, **From beginning of year to beginning of current period**, and **From beginning of current period through end of current period**

Basis Data

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

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

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



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

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

Retained Earnings Accounts and Currency Conversion

In multi-currency environments, the use of retained earnings accounts greatly simplifies the currency conversion process. The retained earnings account inherits the exchange rate type assigned to the source accounts. Consequently, the retained earnings account automatically retrieves the proper value and exchange rate for each period.

Loading Data to a Retained Earnings Account

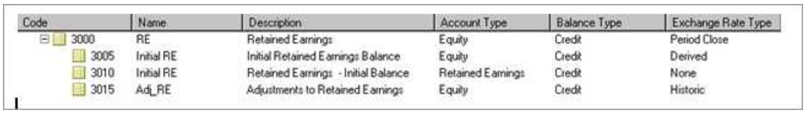
Any data that is loaded to a retained earnings account is ignored at query time. The retained earnings account instead returns the value as determined by the source accounts, roll-forward method, and basis data.

Sample Configurations

Single Retained Earnings Account

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

Figure 6.1 *Single Retained Earnings Account*



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

The purpose of each account is as follows:

- **Account 3000**: This account is the roll-up member in the hierarchy for presentation on a balance sheet.

  It has an account type of Equity.

  It uses a period close rate for any entries made to the virtual child.

- **Account 3005**: This account stores the opening balance of retained earnings.It has an account type of Equity.

  It uses a derived exchange rate.

- **Account 3010**: This account is the single retained earnings account.

  It uses Net Income as the source account.

  It uses the roll-forward method that covers all of time: **From beginning of time to end of current period**.

- **Account 3015**: This account captures any adjustments to retained earnings.

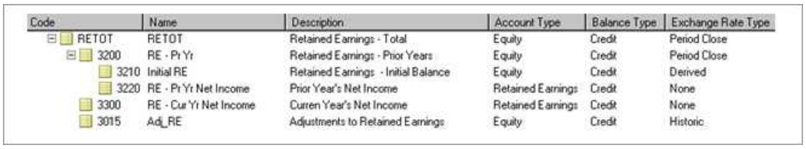
  It has an account type of Equity.

  It uses a historic exchange rate.

Current and Prior Years - Retained Earnings Accounts

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

Figure 6.2 *Two Retained Earnings Accounts*



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

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

The purpose of each account is as follows:

- **Account 3200**: This account is a roll-up member that allows reporting on Prior Years' Retained Earnings from a single account.

  It aggregates the initial balance of retained earnings and the retained earnings from Net Income that is related to prior years

- **Account 3210**: This account stores the opening balance of retained earnings.

  It has an account type of Equity.

  It uses a derived exchange rate.

- **Account 3220**: This account captures the retained earnings from Net Income related to prior years.

  It uses the roll-forward method **From beginning of time to beginning of year**.

- **Account 3300**: This account captures the retained earnings from Net Income related to the current year, using the roll-forward method **From beginning of year through end of current period**.

- **Account 3015**: This account captures any adjustments to retained earnings. It has an account type of Equity.

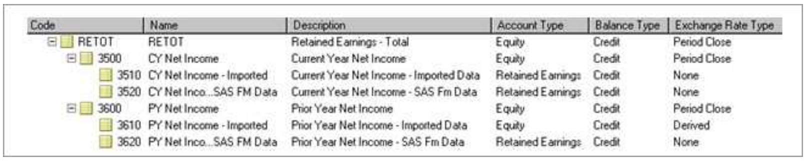
  It uses a historic exchange rate.

Varying Basis Data

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

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

Figure 6.3 *Different Accounts for Prior Years and Current Years*



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

The purpose of each prior year account is as follows:

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

  The balance that is rolled forward annually is captured here.

  It uses a derived exchange rate calculated externally and entered into SAS Financial Management Studio.
- **Account 3620**: This account is the retained earnings account type for prior years.

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

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

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

The purpose of each current year account is as follows:

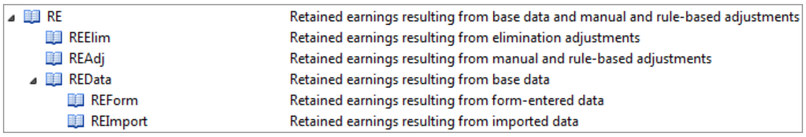
- **Account 3500**: This account is the roll-up member that can be used for simplified reporting.
- **Accounts 3510 and 3520**: These accounts are configured almost identically. The only difference is that account 3510 uses the basis data selection for Imported Data, and account 3520 includes the remaining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

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

Other Considerations

Querying the Source Dimension

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



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

Calculated Members as Source Accounts

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

Rolling Forward Balance Sheet Accounts

Overview and Objectives

Despite its name, the Retained Earnings account type can be used to roll forward balance sheet accounts, because source accounts can refer to both flow account types (for example, Revenue and Expense) and balance account types (for example, Asset, Liability, and Equity).

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

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

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

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

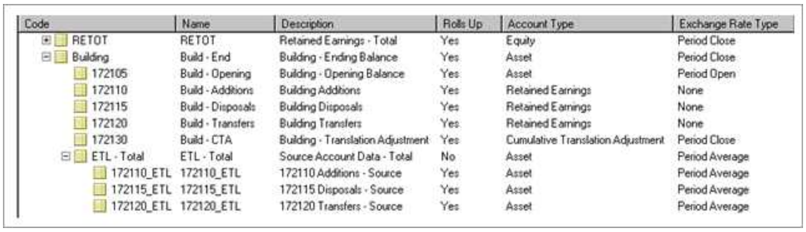
In this scenario, creating a roll-forward for balance accounts is achieved by means of the following:

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

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

Sample Balance Sheet Account Roll-Forward

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



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

The purpose of each account is as follows:

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

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

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

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

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

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

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

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

  Basis data: All four basis data selections are included.

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

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

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

  The data loaded to these accounts is the activity for each account since the last account balance roll-forward. For example, if the account balance for Building were rolled forward at December 2010, the values loaded to the Additions account in the 2011 periods would include all activity in 2011.

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

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

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

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

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

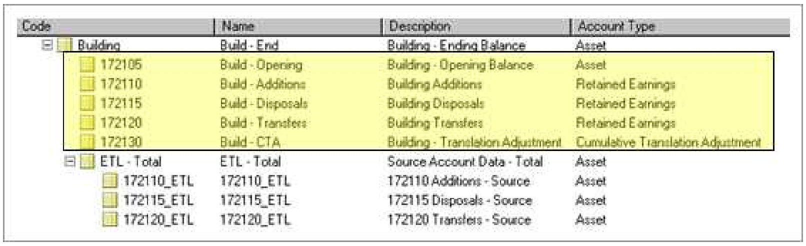
- **Building – Translation Adj. -172130**: This account is used to calculate cumulative translation adjustment (CTA) on a balance sheet.

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

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

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

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



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

Chapter 7
Cycles and Cycle Periods

Introduction to Cycles

Overview

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

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

Data in a Cycle

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

Only one cycle can be open at a time, and operations such as loading or deleting data apply to the open cycle.

See Also

- "Data Records in a Cycle" on page 69
- "Managing Data in a Cycle" on page 69

Objects That Belong to Cycles

Objects of the following types belong to a cycle:

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

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

Cycle Types

There are two types of cycles:

- financial reporting and planning cycles
- operational cycles

Consider the following capabilities when you are determining whether to create a financial or operational cycle:

Financial Cycles	Operational Cycles
	recommended for large volumes of data
data entry tables and read-only tables	data entry tables only
supplemental schedules	
CDA-based reporting	
Excel and Web-based data entry	Web-based data-entry only
accounting logic (for example, retained earnings, cumulative translation adjustment)	
intercompany logic	
driver, modeling, and reporting formulas	driver formulas only
manual adjustments and rules	manual adjustments and rules

Financial Cycles	Operational Cycles
multiple authors and reviewers	single author and reviewer

See Also

"Comparing Financial and Operational Planning" on page 5

Cycle Manager Window

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

The 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.

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

Open

makes the selected cycle the active cycle.

Copy

launches the Copy Cycle wizard.

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

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

Delete

deletes the selected cycle or cycles.

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

Lock

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

The icon for a locked cycle includes a padlock. Locking a cycle also locks all the time periods in the cycle and all the models, form sets, and forms that belong to the cycle. Data cannot be added to or deleted from a locked cycle.

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

See "Locked Objects" on page 13.

Unlock

unlocks the selected cycle or cycles that are currently locked.

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

See "Locked Objects" on page 13.

Attachments

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

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

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

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

Close
Closes the Cycle Manager window.

Create a 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.

Copy a Cycle

To copy a cycle:

1. From the **Tools** menu, select **Cycle Manager**.
2. Select a cycle and click **Copy**.

   For details, see the online Help for the wizard.

The following items are included in the copy:

- data
- periods and period locks
- form sets

The following items are not included in a cycle copy:

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

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

Open a Cycle

To open a cycle:

1. Select **File** ⇨ **Open Cycle**.

   The Open Cycle window 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**.

   Only one cycle is open at a time.

If no cycle is open when you select the Periods, Rates, Models, or Forms workspace, then the default view in the workspace displays an error message and 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, select a cycle from the displayed list, and click **OK**.

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

Data Records in a Cycle

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

See Also

“Introduction to Required Dimension Types” on page 35

Managing Data in a Cycle

Overview of Managing Data in a Cycle

SAS Financial Management supports the following tasks for loading and managing data:

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

- loading model data (financial cycles only)
- loading operational data (financial cycles only)

Loading Data from an External Database

To load data from an external database into a cycle:

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

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

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

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

See Also

"Additional Documentation" on page 6

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. 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 this 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

There are three ways to lock a cycle or part of a cycle:

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

If a cycle or part of a cycle is locked, then no one can add data to it or remove data from it.

Deleting Cell Comments from a Cycle

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

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

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

See Also

- “Loading Data to the SAS Financial Management Data Mart” on page 12
- “Locked Objects” on page 13

Chapter 8

Working with Exchange Rate and Driver Rate Sets

About Exchange Rates

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

In SAS Financial Management, exchange rates can be simple or complex. Simple exchange rate values are defined by exchange rate type and members of the Time and Currency dimensions. For example, a simple exchange rate might be defined for the following members:

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

Complex exchange rate values can vary by all of the same components as simple exchange rates. To be complex, the definition must also include a member of at least one of the following dimensions: Account, Organization, or any user-defined dimensions. For example, a complex exchange rate might be defined for the following members:

- Historic exchange rate type
- the JAN 2005 time period
- EUR currency
- Dividend account
- Italy organization

- a user-defined dimension such as Product

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

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

About Driver Rates

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

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

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

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

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

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

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

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

Best Practices for Exchange Rate and Driver Rate Sets

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

When analyzing data with different sets, users might want to view Actual results translated at Budget rates. They also might want to view Actual results across years using one year’s rates for all years, instead of using the actual rates for each individual year. If you create additional sets with the desired values, users can apply the alternative sets to the underlying data. There are several possible ways to achieve this type of analysis, based on the user’s preference and reporting needs:

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

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

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

- **Option 2: Create additional analysis members** within the existing model and assign the desired exchange rate set to the new analysis members. For example, a new analysis member called Actual at Budget contains actual data but is assigned the budget exchange rate set. With this approach, you must copy the Actual data, as well as any adjustments and rules on the Actual analysis member, to the new analysis member.

- **Option 3: Assign a different rate set.** Simply change the set that is assigned to a particular analysis member and re-post adjustments and rules. This approach is expedient, especially if it is done infrequently for a single reporting objective.

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

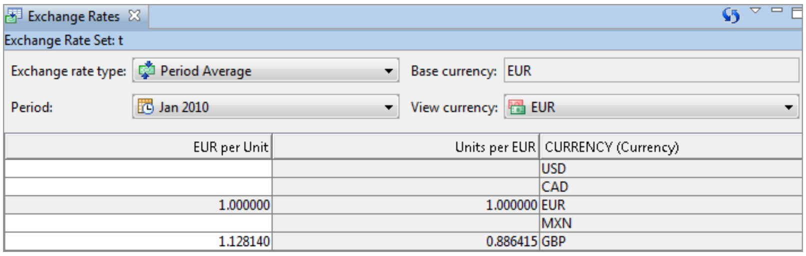
Base Currency for Exchange Rate Sets

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

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

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

Figure 8.1 *Exchange Rate and Reciprocal*



Exchange Rate Set: t

Exchange rate type: Period Average

Base currency: EUR

Period: Jan 2010

View currency: EUR

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

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

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

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

- 1.128140 EUR/GBP
- 0.692488 EUR/USD

The GBP/USD rate is calculated as the following:

```
( 0.692488 EUR/USD ) x ( 1 / 1.128140 EUR/GBP) = 0.613832 GBP/USD
```

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

Precision of Exchange Rates

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

SAS Financial Management always multiplies when performing conversion. It must calculate a reciprocal from the exchange rate that was entered and stored. In certain cases, the reciprocal calculation might introduce a minimal degree of inaccuracy. For example, the inverse of the value 3 is 0.333333333333333 (15 significant digits). If the value 0.333333333333333 is entered into the exchange rate editor, the related inverse is calculated as 3.000000000000003. The difference is approximately 0.000000000001%.

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

- The base currency of the table is USD.
- The exchange rate is 1,489,400 TRL per USD.
- Because the base is USD, the TRL must be expressed in USD per TRL.
- The rate is 0.000000671411306566403, expressed to 15 significant digits (21 digits to the right of the decimal point in this case.) Note that the SAS Financial Management rate editor allows up to 25 places to the right of the decimal, but only 15 significant digits.

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

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

Displaying and Copying Exchange Rates and Driver Rates

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

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

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

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

It is not possible to enter a value of zero.

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

Querying Exchange Rates and Driver Rates in a Table

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

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

For more information about calculated members and these functions, see Chapter 24, “Dictionary of Calculated-Member Functions,” on page 247 and the *SAS Financial Management: Formula Guide*.

Querying Exchange Rates and Driver Rates outside a Table

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

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

For more information about these functions, see Chapter 23, “Dictionary of CDA Functions,” on page 239.

Chapter 9
Overview of Models

Introduction to Models

Each model belongs to a single cycle. A model provides a view of the selected cycle. There are both financial and operational models.

- Financial models support financial form sets. They are also used to generate reports in the SAS Financial Management Add-In for Microsoft Excel. They can include manual adjustments and adjustment rules.

  Two or more financial models can be brought together in a composite model.

- Operational models are used exclusively to support operational form sets.

  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. Then use financial models to create the report.

See Also

- "Introduction to Composite Models" on page 105
- "Cycle Types" on page 66

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.

Here are the required features of a model:

- an association with a cycle

  The underlying cycle stores the data from data-entry tables that are based on the model 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

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

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

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

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

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

- a time span

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

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

- (optional) formulas that are associated with the model.

- the default read member for each hierarchy

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

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

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

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

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

  For any crossing whose value is calculated by means of a currency conversion, the 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 rates for different analysis members.

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

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

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

- formatting specifications for debit and credit accounts

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

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

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

Optional Features of a Model

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.

- cell protection rules

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.

- prior period adjustments

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

Chapter 10
Models: Rules and Adjustments

Overview of Adjustments and Adjustment Rules

Administrators can create manual adjustments and adjustment rules for a model. Manual adjustments and adjustment rules are used to enter and manage a variety of accounting entries, adjustments, and allocations that are not performed in an ERP system or entered via data entry forms.

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 SAS Financial Management staging area 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 dimension uses the following members to capture data that is associated with manual adjustments and adjustment rules:

- Manual
- Bal
- Reclass
- Alloc

- CPO

See Also

“Introduction to the Source Dimension Type” on page 43

Posting Status of Manual Adjustments and Adjustment Rules

Every manual adjustment and every adjustment rule has a posting status. There are three possible posting status values:

Not posted
The item has never been posted.

Posted
The item has been posted. It has not been modified since it was last posted.

Modified since last post
The item has been posted. However, it has been modified since it was last posted. Therefore, new reports do not reflect the latest modifications until you post the item again.

An item's current posting status is displayed in the **Posting Status** column of the view where the item is listed. You can also see it on the **General** tab of its Properties window.

In the **Models** view, if a 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 there are adjustments or rules that need to be posted.

See Also

“Warnings for Financial Models” on page 87

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. 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 85

What Happens When You Post Adjustments

Overview

The administrator can select whether to post a manual adjustment before or after adjustment rules. If a manual adjustment is posted first, its output can be referenced in adjustment rules.

For a specified set of time periods and analysis members, the sequence of events in the posting process is as follows:

1. All the currently posted adjustment values are deleted.
2. All the **Before rules** manual adjustment values are posted.
3. The adjustment rules run in the order that is specified in the Adjustment Rules view. For each rule, adjustment values are generated and posted. The input to a given rule can include values from any **Before rules** manual adjustment and any adjustment that was generated by a rule that has already run.

   By default, the adjustments from certain adjustment rules are subject to a threshold. See "Threshold for Adjustment Rules" on page 86.
4. All the **After rules** manual adjustment values are posted.

If you make changes to manual adjustments or adjustment rules, you must repost the adjustments, as well as any rules that rely on reposted values.

To delete posted adjustment values:

1. Select the **Put On Hold** option for the adjustments and rules whose posted values you want to delete.

   If you do not plan to use the adjustments or rules again, then you can delete them instead of placing them on hold.
2. Repost adjustments, including any rules that rely on deleted or reposted values.

Threshold for Adjustment Rules

Adjustments that are generated by certain adjustment rules are subject to a materiality threshold that ignores (does not store) very small values that are considered immaterial. This threshold affects the following:

- allocation rules
- reclassification rules
- intercompany balancing rules
- net intercompany balancing rules
- self-elimination rules
- reporting entity rules
- automatic allocation and explicit allocation in forms and form templates
- driver formulas, whether they are initiated from writeback or from selecting **Run driver formulas** in SAS Financial Management Studio

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

- `Account A`: value 5.0 (target)
- `Account B`: value 0.0008 (target)
- `Account C`: value –5.0 (offset)
- `Account D`: value –0.0008 (offset)

With a threshold value of 0.001, any entry whose absolute value is less than 0.001 is not stored in the database. In the previous example, there would be two line items posted rather than four:

- `Account 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 0.0008 (target)
- `Account F`: value –6.0008 (offset)

With a threshold value of zero, all three of the line items are posted. With a threshold value of 0.001, only two line items are posted:

- `Account 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 materiality threshold is 0.001. You can modify the threshold (or effectively eliminate it by setting its value to 0). See “Post-Configuration Steps” in the *SAS Financial Management: System Administrator’s Guide*. After modifying the threshold value (and restarting the managed servers), you can repost adjustment rules to use the new threshold.

Note: Materiality threshold is not applied to ownership rules or to facts that are entered directly by the user (not formula-generated) or ETL facts.

See Also

“Being on Hold” on page 84

Warnings for Financial Models

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

These warnings are present only to notify administrators that there have been changes that might affect data values. For information about the changes that are responsible for the warning, select **Properties** and examine the **Warnings** region on the Adjustments page of the model's Properties window. Here are some example warning messages:

- Data records have changed
- Exchange rates have changed
- Driver rates have changed
- Hierarchies have changed

See Also

- “What Happens When You Post Adjustments” on page 85
- “Posting Status of Manual Adjustments and Adjustment Rules” on page 84

Working with Manual Adjustments

A manual adjustment is an entry that is made at a model level for specified crossings. Values for a model are assumed to have a frequency of Period Activity.

SAS Financial Management supports both balanced and unbalanced manual adjustments. A cycle’s properties determine whether manual adjustments are available, the adjustment rules that are permitted, and whether unbalanced manual adjustments are permitted in models that are associated with the cycle.

Balanced manual adjustments require an Organization member that has been designated as a reporting entity.

For each manual adjustment, you must designate a currency. All the adjustment values that are part of the adjustment are expressed in the designated currency.

To create, edit, delete, or copy the manual adjustments that belong to a particular 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.

See Also

- "Manual Adjustments View" on page 88
- "User Roles and Capabilities" on page 11
- "Organization Details" on page 42

Manual Adjustments View

The Manual Adjustments view lists the manual adjustments that belong to the selected financial model. The adjustments are filtered based on the following selections:

- **Before rules**: manual adjustments that are posted before any adjustment rules are posted
- **After rules**: manual adjustments that are posted after all adjustment rules are posted
- **Analysis**: manual adjustments that apply to the selected analysis member
- **Time**: manual adjustments that apply to the selected time period

Depending on the types of adjustments allowed by the cycle's properties, the following options are available:

New Balanced Manual Adjustment
launches the New Balanced Manual Adjustment wizard. This wizard enables you to create a balanced manual adjustment.

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

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

This option is active if at least one selected manual adjustment is not on hold.

Take Off Hold

takes off hold all the selected manual adjustments that are currently on hold.

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

Move After

moves the selected manual adjustments from **Before rules** to **After rules**.

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

Move Before

moves the selected manual adjustments from **After rules** to **Before rules**.

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

Delete

deletes the selected manual adjustments.

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. This wizard enables you to make a copy of the selected manual adjustment for the same analysis member, time period, and relationship to adjustment rules.

This option is active only if a single manual adjustment is selected.

Copy To

launches the Copy Manual Adjustment To wizard. This wizard enables you to copy the selected manual adjustment to any available model, analysis member, and time period. The option to copy as a reversing entry is also available.

You can copy a manual adjustment from 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 a dimension type is in the source model but not in the target model, then all references to that dimension type are discarded in the copy.

If a dimension type is in the target model but not in the source model, then its default write member is added to the adjustment wherever appropriate.

If source and target members do not match up properly within a dimension type, then a warning is displayed.

This option is active only if a single manual adjustment is selected.

Find Errors

displays the Manual Adjustment Errors window, which displays a summary of any rules in error, as well as the analysis and period for each rule that is in error.

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 a single manual adjustment is selected.

Filter

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

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

Refresh
refreshes the view.

For details, see the online Help for the wizard or property pages.

See Also

- "Using Views" on page 16
- "Working with Manual Adjustments" on page 87
- "Being on Hold" on page 84

Working with Adjustment Rules

SAS Financial Management supports six different types of adjustment rules. Depending on the types of adjustments allowed by the cycle's properties, the following types of rules are available:

- Intercompany balancing rules on page 95
- Net intercompany balancing rules on page 96
- Reporting-entity balancing rules on page 96
- Reclassification rules on page 97
- Allocation rules on page 97
- Ownership rules on page 98

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

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

   In the Adjustment Rules view, all the adjustment rules for the selected model are displayed in a single list, in the order in which the rules are executed.
3. Use the Adjustment Rules view as explained in the online Help for that view.

See Also

- "Adjustment Rules View" on page 90
- "Promoting Objects to Another System" on page 16

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**.

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

New Intercompany Balancing Rule
launches the New Intercompany Balancing Rule wizard. This wizard enables you to create an intercompany balancing rule.

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

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

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

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

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

You can create only one ownership rule per 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. This wizard enables you to import an ownership rule or allocation rules from a SAS package file.

Export Adjustment Rules
launches the Export Adjustment Rules wizard. This wizard enables you to export the selected adjustment rules to a SAS package file.

This option is active if any adjustment rules are selected.

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

This option is active if at least one selected adjustment rule is not on hold.

Take Off Hold
takes off hold all the selected adjustment rules that are currently on hold.

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

Delete

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

Copy Here

launches the Copy Rule wizard. This wizard enables you to make a copy of the selected adjustment rule for the same model.

This option is active only if a single adjustment rule is selected.

Copy To

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

You can copy an adjustment rule from 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 a dimension type is in the source model but not in the target model, then all references to it are discarded.

If a dimension type is in the target model but not in the source model, then its default write member is added to the rule wherever appropriate.

If source and target members do not match up properly within a dimension type, then a warning is displayed.

This option is active only if a single adjustment rule is selected.

Reorder Adjustment Rules

displays the Reorder Adjustment Rules window. This window enables you to change the order of the adjustment rules for the selected model.

The adjustment rules run in the order in which they are listed in this view.

Posted Adjustment Data

displays a Web page that lists the currently posted values that were produced by the selected adjustment rules.

For a list of all the currently posted adjustment values for a 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.

Filter

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

Customize Columns

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

Refresh

refreshes the view.

See Also

- "Using Views" on page 16

- “Working with Adjustment Rules” on page 90
- “Being on Hold” on page 84

Chapter 11
Models: Adjustment Rule Types

Intercompany Balancing Rules

An intercompany balancing rule compares the selected intercompany debit and credit accounts for every possible Organization/Trader combination, using the PTD balances for the TotalBeforeElim source member. Any difference in the totals for the selected accounts is recorded in the adjustment account, with an offsetting entry recorded in the specified offset account.

Adjustments are recorded based on the **Intercompany Balance Rule** selection in the cycle properties. This selection is made when the cycle is created and cannot later be modified. If the **Intercompany Balance Rule** selection is **Write values to the originating organization**, the following rules apply:

- If the balancing account is a debit account, then the balancing account is adjusted for the organization that contributed the debit accounts to the unequal comparison.

- If the balancing account is a credit account, then the balancing account is adjusted for the organization that contributed the credit accounts to the unequal comparison.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

Note: The adjustments that are generated by an intercompany balancing rule are subject to a materiality threshold. See "Threshold for Adjustment Rules" on page 86.

See Also

- "Introduction to the Source Dimension Type" on page 43
- "What Happens When You Post Adjustments" on page 85

Net Intercompany Balancing Rules

A net intercompany balance rule compares the selected intercompany debit and credit accounts for every possible Organization/Trader combination, using the net sum PTD balances for the TotalBeforeElim source member. Any difference in the net totals for the selected accounts is recorded in the adjustment account, with an offsetting entry recorded in the specified offset account.

The adjustment is always made to the lowest common parent organization of the two compared organizations.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

Note: The adjustments that are generated by a net intercompany balancing rule are subject to a materiality threshold. See "Threshold for Adjustment Rules" on page 86.

See Also

- "Introduction to the Source Dimension Type" on page 43
- "What Happens When You Post Adjustments" on page 85

Reporting-Entity Balancing Rules

A reporting-entity balancing rule compares the selected non-intercompany debit and credit accounts for every reporting entity, using the PTD balances for the Total source member. Any difference in the total value for a selected account results in a single-sided entry to the specified adjustment account. The typical use of a reporting-entity balancing rule is to verify the standard balance-sheet equation, Total Assets = Total Liabilities + Total Equity.

For any non-required custom dimensions, you can specify the member to write to via the **Write Members** tab of the intercompany balancing rule's properties.

See Also

- “Introduction to the Source Dimension Type” on page 43
- “What Happens When You Post Adjustments” on page 85

Allocation Rules

An allocation rule allocates the PTD value of a specified source crossing to a set of target crossings. A single allocation rule can allocate values from multiple source crossings, provided that all the allocations have the same pattern. The allocation table displays a preview of the allocation weights based on the source and target crossings and the way the allocation should be performed (equal, weighted, or proportional).

You have the option of selecting an offset account for a balanced entry.

The selections that you make in the allocation wizard define the allocation table. The source crossings define the rows of the allocation table, and the target crossings define the columns of the allocation table. The allocation method then determines the weights per crossing.

It is possible to chain two or more allocation tables together in a single allocation rule. The outputs from the first allocation table become the inputs to the second allocation table, and so on. Each allocation table describes a distinct step of the allocation. For example:

- A one-step allocation rule could distribute certain sales and marketing costs among several 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.

By default, the allocation rule allocates the PTD balance of the Total member of the Source dimension. Only the last step of an allocation rule is recorded, to the Alloc member of the Source hierarchy.

Note: The adjustments that are generated by an allocation rule are subject to a materiality threshold. See “Threshold for Adjustment Rules” on page 86.

See Also

- “Introduction to the Source Dimension Type” on page 43
- “What Happens When You Post Adjustments” on page 85

Reclassification Rules

A reclassification rule reclassifies the values at selected source crossings to selected target crossings. You have the option of assigning an offset to create a balanced entry.

By default, the reclassification rule allocates the PTD balance of the Total Source dimension member.

The values generated by the reclassification rule are stored in the Reclass Source dimension member.

Note: The adjustments that are generated by a reclassification rule are subject to a materiality threshold. See "Threshold for Adjustment Rules" on page 86.

See Also

- "Introduction to the Source Dimension Type" on page 43
- "What Happens When You Post Adjustments" on page 85

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.

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, contains the real-world organization's complete consolidated results. The other member, called the standalone parent, contains 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 relationship between each consolidation parent and its corresponding standalone parent. The ownership rule uses the Direct Parent holding type to mark this relationship.

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 41
- "Ownership Rules: Holding Types" on page 99

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 12
Composite Models

Introduction to Composite Models

A composite model combines two or more financial models. The models can belong to the same cycle or different cycles, as long as the cycles use the same set of dimension types.

Composite models can be used to combine models that span different time periods or Analysis members. In the SAS Financial Management Add-In for Microsoft Excel, composite models can be used in read-only tables and CDA tables.

A composite model might reference more than one cycle with values for the same crossing. In that case, the displayed value for the crossing is the sum of those values.

A composite model uses all the posted adjustments of all its referenced models. You cannot create additional adjustments for the composite model itself.

To work with composite models:

1. Open a 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 79
- "Introduction to Cycles" on page 65

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.

Here are the required features of a composite model:

- a code, name, and description
- two or more financial models
- a hierarchy for each dimension type that is used in the models

  Although the selected models must use the same set of dimension types, they might not use the same set of hierarchies. Therefore, you need to select a hierarchy and as-of date for each dimension in the composite model.
- the default read member for each hierarchy

  For a composite model, the default read members are always the default members of the hierarchies.
- an association of an exchange rate set with each analysis member in the composite model's analysis hierarchy
- an association of a driver rate set with each analysis member in the composite model's analysis hierarchy
- formatting specifications for debit and credit accounts

For more information about these properties, see “Required Features of a Model” on page 79. Many of these properties can be changed in the Properties window for the composite model, just as they can for a financial model. However, the default read members and rank ordering of hierarchies cannot be changed in a composite model.

Optional Features of a Composite Model

Here 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 all composite models, regardless of the cycles the models belong to.

Options for Composite Models

The following options are available:

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

Copy Here
makes a copy of the selected composite model.

Delete
deletes the selected composite model.

Attachments
displays the **Attachments** view, which enables you to add, open, or delete attachments for the selected composite model.

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

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

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

Refresh
refreshes the view.

See Also

- “Using Views” on page 16
- “Introduction to Composite Models” on page 105

Chapter 13
Formulas

Overview of Formula Types and Formulas

A formula is a mathematical expression that returns a value when it runs. A formula can be created on a calculated member in the Dimensions workspace in SAS Financial Management Studio.

SAS Financial Management supports four types of formulas:

- reporting formulas

- modeling formulas
- driver formulas
- Excel-based calculated member formulas

The formula type names reflect when the formulas run and where they are most commonly used in financial reporting and planning.

Reporting formulas and Excel-based calculated member formulas run post-query, meaning that the formulas are typically based on data inputs from a query. The most common examples are ratios such as Gross Profit Percentage and Earnings per Share.

Modeling formulas and driver formulas can be thought of as data-creation formulas. They generate data that can be consumed by reporting formulas and Excel-based calculated member formulas. These formulas also generate accounting logic such as retained earnings and cumulative translation adjustment (CTA) accounts. For this reason, they are often termed pre-query formulas. Typical examples include Sales based on Price and Units, Estimated Benefits Expense based on Salaries, and Training Expense based on Headcount.

See Also

SAS Financial Management: Formula Guide

Order of Execution

The order of execution for formulas is as follows:

1. facts
2. intercompany eliminations
3. driver formulas
4. modeling formulas
5. retained earnings and cumulative translation adjustment accounts
6. reporting formulas

This execution order allows both modeling formulas and driver formulas to be indirectly or directly referenced as source accounts of retained earnings accounts and cumulative translation adjustment accounts.

Reporting formulas are calculated after retained earnings accounts and cumulative translation adjustment accounts. Therefore, reporting formulas do not contribute to the results of retained earnings accounts and cumulative translation adjustment accounts. Each level can consume its own level as well as the preceding levels.

Note: Currency conversion occurs at each level of execution.

Server-Side Calculated Members

Account Types for Calculated Members

The behavior of calculated members in the Account dimension is influenced by the member's account type. Here are the available account types, grouped by category:

Balance account types

- Asset
- Liability
- Equity
- Statistical Balance

Flow Account Types

- Revenue
- Expense
- Statistical Flow

Other Account Types

- Statistical

The Retained Earnings and Cumulative Translation Adjustment account types cannot be associated with formulas and are excluded from the preceding list.

Formula results are calculated by a distinct method for each account type category.

- Balance account types: aggregate and then convert results
- Flow account types: convert and then aggregate results

Statistical accounts do not participate in aggregation or conversion.

The calculation and currency conversion methods depend on the account type of the calculated member.

Resolving Conflicts between Dimensions

Formula conflicts are limited to reporting formulas. Due to execution order, driver formulas and modeling formulas always run before reporting formulas and are available only in the Account dimension. Because only the Account dimension permits driver formulas and modeling formulas, there are no conflicts with these formula types.

Virtual Child Members

In SAS Financial Management, a virtual child is automatically available to a member that is designated as a roll-up. A virtual child enables you to enter values at a roll-up point where less detail is required or where spreading or allocations are involved.

Virtual children are available for these dimension types:

- IntOrg
- Account

- Custom

A formula can be executed for a virtual child member, just as it can for an ordinary leaf member. However, you cannot assign a formula to a virtual child the way you can to an ordinary leaf member.

A formula on a roll-up member is always ignored, and the result is the sum of leaf values.

Viewing Formula Information in SAS Financial Management Studio

SAS Financial Management Studio provides formula information in the Dimensions and Models workspaces.

In the Dimensions workspace, formula information is available on the **Members** and **Hierarchies** tabs for all dimension types that support calculated members.

The Account dimension type offers optional informational headers and columns for both **Formula Type** and **Formula Count**.

Viewing Formula Information in Microsoft Excel

Formula information is available on a crossing-by-crossing basis in the SAS Financial Management Add-In for Microsoft Excel. For either a data-entry table or a read-only table, select **Tools** ⇨ **Cell Information** from the SAS Financial Management menu to view formula information. The following formula information is displayed:

- formula type
- dimension of calculated member
- formula expression
- formula name
- fixed members, if any

Writing the Formula Expression: Explicit Members and Fixed Members

In SAS Financial Management, a formula expression reads from the same crossings where it executes. There are two exceptions: explicit members in the formula expression text and fixed member references.

- **an explicit member.** If a member is explicitly named in the expression, that member is used. For example:

  ```
  ["ACCOUNT"="A100"]["PRODUCT"="P1100"]+["ACCOUNT"="A200"]["PRODUCT"="P1100"])
  ```

  The bracketed expression uses this syntax:

  ```
  [dimension-type=member-code]
  ```

- **a fixed member.** At the next level of precedence is a fixed member. Fixed members are useful for complex formulas. Rather than specifying the same member multiple times, you can define a fixed member.

  The fixed member is applied to all formula inputs that do not explicitly include a member of that dimension. For example, you select a fixed member of the Product dimension, “P1100”. Using the example above, but adding a fixed member of “P1100”, the formula can now be written as follows:

```
["ACCOUNT"="A100"]+["ACCOUNT"="A200"]
```

As long as the expression does not explicitly include a member of that dimension, the fixed member applies. In the following formula expression, a fixed member of “P1100” would apply only to the first operand:

```
["ACCOUNT"="A100"]+["ACCOUNT"="A200"]["PRODUCT"="P1200"]
```

- **the same member as in the target crossing for the formula.** In the absence of fixed member or explicit member references, the formula expression executes at the same crossings that it reads from (after considering formula scope). If a dimension is off the table, the formula reads from the default read member and executes at the default write member for that dimension.

Using a Function in a Bracketed Member Reference

With some exceptions, you can use a function within a bracketed member reference. Here is an example:

```
SUM(["ACCOUNT"="Accounts & Notes Receivable"]
["TIME"=OPENINGPERIOD(ANCESTOR("TIME","Year"))]:
["ACCOUNT"="Accounts & Notes Receivable"]
["TIME"=CURRENT("TIME")]) /(PROPERTY("TIME","Month_number"))
```

In this example, the OPENINGPERIOD function works with the SUM function and the PROPERTY function to calculate the year-to-date average balance of the Accounts and Notes Receivable account. The PROPERTY function retrieves the values of a custom property (Month_number) of the Time dimension.

The following string functions are supported using this syntax:

- ANCESTOR
- CLOSINGPERIOD
- COMPRESS
- CURRENT
- FIRSTCHILD
- FIRSTSIBLING
- LASTCHILD
- LEFT
- LOWCASE
- NEXT
- OPENINGPERIOD
- PARENT
- PREVIOUS
- PROPERTY

- REPEAT
- REVERSE
- RIGHT
- SUBSTR
- SCAN
- TRIM
- UPCASE
- VIRTUALCHILD

If a function is not supported using this syntax, you can still use it within an IF or NESTIF function.

Formula Scope

Formula scope is an optional means of restricting where a formula runs. It can be applied on an expression-by-expression basis and is available for all server-side formulas (reporting, driver, and modeling formulas). From a performance standpoint, formula scope is most effective when it is used with modeling formulas to limit the number of crossings where a formula runs.

Defining Multiple Formulas on One Member

You can define an unlimited number of formulas on a single calculated member. To improve formula performance, maintenance, and manageability, define multiple formulas on one member as an alternative to the following functions:

- SUBSTR
- IF
- NESTIF

Use fixed members and formula scope to customize the formula and where it is applied.

Each formula is assigned a rank when you create it. The initial rank order for a calculated member is based on the order in which the formulas were created. The rank of a formula determines which formula is executed in the event of formula scope overlap.

Formulas that are Ignored, Invalid, or Not Rendered

This section identifies formula expressions that are ignored, invalid, or not rendered in SAS Financial Management. A warning message is provided on the Formulas page for model properties and under **Details** when you publish an operational planning form set.

- **Ignored:** Calculated members in this category are not executed due to the priority of other computations such as roll-up logic. Since the formula expression is ignored, there is no cell information. Examples of this category:

  - formulas on a roll-up member
  - formulas that run out of bounds
  - referencing a dimension and/or member not in the model

- **Invalid:** Calculated members in this category are not executed at query time because they are incorrect in the context of the model. In most cases, these expressions pass validation in the formula editor. However, the expressions have warning messages as described above.

  In a table, an invalid driver formula is displayed as a gray, non-writable cell with a value of zero. Modeling and reporting formulas render as red cells. Select **Tools ⇨ Cell information** to view warnings for modeling and driver formulas. Examples of this category:

  - circular references
  - divide by zero

- **Not rendered:** This type of formula restriction applies to the use of constants in driver and modeling formulas. For both formula types, the result is a gray, non-writable cell with a value of zero. No warning message is available in **Cell Information**. An example of this category:

  - referencing only constant values

Working with Driver Formulas

Overview

Driver formulas provide a mechanism to create data using the context of a data-entry table. The following items define the range of execution for a driver formula:

- formula scope (defined on the member)
- writable analyses (defined in the form set)
- writable crossings on the form set template, including system filters

What Triggers Execution of a Driver Formula

Based on their design and limited range of execution, driver formulas provide an efficient way to calculate values. There are two ways to execute driver formulas:

- entering data into a form.

  Driver formula execution is triggered by the change in a value on a data-entry form. When the value is entered, the formula result is calculated and stored in the database. No additional action is required.

- selecting the **Run driver formulas for this form set** option in the Forms workspace of SAS Financial Management Studio.

  This option triggers the execution of driver formulas for the entire form set.

  Note: Selecting **Refresh** for the form set does not trigger the execution of driver formulas.

Driver Formula Execution at Data Entry

The following guidelines apply to driver formula execution at the time of data entry:

- Driver formulas read from and write to the BaseForm member of the Source dimension. A driver formula can read from other members of the Source dimension if you explicitly state the members in the formula expression or implicitly state them as default members. A driver formula can write only to BaseForm; this is not modifiable.
- Driver formula inputs that are referenced in the formula expression must be included on a data-entry table to trigger driver formula execution at writeback.
- Driver formula expressions with relative time references such as `["TIME"=CURRENT("TIME")-1]` can read from Time members that are not included in the form. However, the formula expressions write only to those Time members that are included in the form.
- To render a result at the time of data entry, driver formula input crossings must match driver formula result crossings for all dimensions. The only exceptions are the Time dimension (for relative time addressing) and the Account dimension.

  This means that an input crossing that is not identical to the output crossing does not trigger a driver formula at the time of data entry. This applies to formulas that either explicitly state dimensions other than Time and/or Account in the expression or reference them via Default Members.

The Run driver formulas Option

The **Run driver formulas for this form set** option is required for the following scenarios:

- changes to global values such as exchange rates, PRATE or DRATE rates, and formula expression inputs that are not in a given form
- changes to a driver formula expression after data input
- creation or deletion of a driver formula after data input
- loading data records that affect driver formulas
- driver formula expressions that reference dimensions other than Account and/or Time
- formula inputs that are not included on the table

Form Design and the Run driver formulas Option

Form design defines the scope for execution of driver formulas when you select the **Run driver formulas for this form set** option. This option examines member selection for all slicers, rows, and columns. Limiting the number of slicers and members in a form limits the number of drivers that need to be executed. This improves performance.

To optimize the performance of the **Run driver formulas for this form set** option, we recommend the following guidelines:

- Limit the dimension members in rows, columns, and slicers to the members that are required for data entry. Be sure to consider the Source and Trader dimensions.

- Use a separate read-only table for data that is needed in a form for information purposes only. For example, if you need to enter data for a budget, maintain the Actual data in a separate read-only table in the same form.
- Use System Filters to restrict the number of crossings required to run driver formulas.

  Note: All driver formulas in the hierarchy are executed, regardless of whether they are included on the form.

Formulas and Operational Planning

Formula Types for Operational Planning

Operational Planning supports two formula types:

- driver formulas

  For operational planning, driver formulas cannot contain references to any dimensions other than Account and/or Time. This applies to formula expressions that are either explicitly stated or assigned by using fixed members in the formula editor.

  Note: These restrictions apply only to the formula expression, not formula scope.
- calculated member functions in the SAS Financial Management Add-In for Microsoft Excel

Triggering Driver Formulas in Operational Planning

Two actions trigger the execution of driver formulas in operational planning:

- entering data into a form
- publishing an operational planning form set

Excel-Based Calculated Members

With the SAS Financial Management Add-In for Microsoft Excel, you can create Excel-based calculated members on a table-by-table basis. Excel-based calculated members look the same as members that are created in SAS Financial Management Studio. They are most similar in behavior to reporting formulas, sharing similar calculation methods and currency conversion methods. They differ primarily in reuse because formulas are created and saved locally, not on the server. Excel-based calculated members support any combination of the following in a formula expression:

- reference to any member in the same dimension
- absolute reference to any crossing in a table in the same workbook
- absolute reference to any cell in the same workbook
- any Excel function or valid Excel expression
- any of the SAS Financial Management calculated-member functions:

- **`fmValue`**
- **`fmCode`**
- **`fmProperty`**
- **`fmRate`**
- **`fmXRate`**
- **`fmCXRate`**

For a description of each of these functions, see the “Dictionary of Calculated-Member Functions” in the online Help for the SAS Financial Management Add-In for Microsoft Excel

Dictionary of Functions

The following functions can be used in SAS Financial Management Studio formulas. They are available for selection on the **Functions** tab of the Formula Editor window. For detailed examples that use many of these functions, see the *SAS Financial Management: Formula Guide*.

CAUTION:

Every complete formula must return a numeric value. Therefore, only functions that return numeric values can be used as complete formulas. Any function that returns a string value or a Boolean value must be used in an argument of another function.

ABS

returns the absolute value of its only argument. The argument must have a numeric value.

ACOS

returns the arc cosine, in radians, of its only argument. The argument must have a numeric value between –1 and +1, inclusive.

ANCESTOR

The ANCESTOR function has two signatures:

```
ANCESTOR("dimension type code", number_of_levels)
```

returns the member code of the ancestor a specified number of levels above the current member for a dimension.

```
ANCESTOR("dimension type code", "level name")
```

returns the member code of the ancestor at a named level (such as **`"Year"`**) above the current member for a dimension.

Use this function to navigate a hierarchy, typically the Time hierarchy. It enables a formula to use a member that is one or more levels above the current member, or to use a specified period type, such as Year.

The ANCESTOR function takes two arguments and supports an optional third argument:

- The first argument can be either a dimension type code or a function that returns a member code:

  - If the first argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
  - If the first argument is a function that returns a member code, then the returned code explicitly specifies a member.
- The second argument is an ancestor designator. The ancestor designator can take two forms: number_of_levels or "level name".
  - number_of_levels: An integer indicates a number of hierarchical levels above the member that is specified by the first argument. You can use an integer as the second argument no matter what the first argument is.
  - "level name": If the first argument is the Time dimension type code or a function that returns a time period code, you can use one of the following period type values to indicate a particular level of the time hierarchy: **`AllYears`**, **`Year`**, **`HalfYear`**, **`QuarterYear`**, **`Month`**, or **`Day`**.
- (Optional) Third Argument: The optional third argument specifies a number of time periods in the past or future. This argument is typically used to get a prior year value.

This formula might be applied to the Beginning Balance - Equipment account to return the prior year value of the Ending Balance, Equipment account:

```
["ACCOUNT"="Ending Balance, Equipment"]["TIME"=ANCESTOR("TIME","Year",-1)]
```

ASIN

returns the arc sine, in radians, of its only argument. The argument must have a numeric value between –1 and +1, inclusive.

ATAN

returns the arc tangent, in radians, of its only argument. The argument must have a numeric value.

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]
```

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()
```

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.

In this example formula, the DRATE function works on the Income Taxes account member:

```
=IF(["ACCOUNT"="Income Before Taxes"]<0,
["ACCOUNT"="Income Before Taxes"]* DRATE("TaxRate2012")*-1,0)
```

The desired outcome is to multiply Income before Taxes by a predefined rate that varies by Organization to return an estimated Income Tax value. If Income Before Taxes is less than zero, then Income Taxes is zero.

The DRATE for TaxRate2012 can be displayed in an Excel report using an Excel calculated member with the following syntax: **`=fmRate("TaxRate2012")`**.

EXP

returns the result of raising **`e`** to the power that is specified in its only argument. The argument must have a numeric value. **`e`** is the base of the natural logarithms, which is approximately 2.718.

EXP is the inverse of LOG . For example, **`EXP(LOG(2.65))`** returns **`2.65`**.

FIRSTCHILD

returns the code of the member that is the first-listed hierarchical child of the member that is specified either explicitly or implicitly by the argument of the function.

The FIRSTCHILD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"]["TIME"=FIRSTCHILD("TIME")]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the first quarter of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the first month of that quarter.

Instead of a dimension type code, the argument of FIRSTCHILD can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="Headcount"]["TIME"=FIRSTCHILD(NEXT(("TIME"))]
```

If the FIRSTCHILD function is applied to a member that has one child, then it returns the code of that one child.

If the FIRSTCHILD function is applied to a leaf member, then the operand that the function is part of receives a value of 0.

FIRSTSIBLING

returns the code of the member that is the first-listed hierarchical sibling of the member that is specified either explicitly or implicitly by the argument of the function.

The FIRSTSIBLING function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"]["TIME"=FIRSTSIBLING("TIME")]
```

When evaluated at a January, February, or March crossing, this reference retrieves the value of the Headcount account for January. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the first quarter of the same year.

Instead of a dimension type code, the argument of FIRSTSIBLING can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="Headcount"]["TIME"=FIRSTSIBLING(PARENT("TIME"))]
```

If the FIRSTSIBLING function is applied to an only child, then it returns the code of that one child.

FLOOR

returns the largest integer that is less than or equal to its only argument. For example, **`FLOOR(5.3)`** returns `5` and **`FLOOR(5.0)`** returns `5`. The argument must have a numeric value.

IF

returns a value that depends on the truth value of a Boolean expression. The IF function takes three arguments, as illustrated in the following example:

```
IF(["ACCOUNT"="Net Sales"]=0,0,["ACCOUNT"="Net Income"]/["ACCOUNT"="Net Sales"])
```

The first argument is the Boolean expression. The second and third arguments can be expressions of any kind. If the Boolean expression is true, then the IF function returns the value of the second argument. If the Boolean expression is false, then the IF function returns the value of the third argument.

The Boolean expression can compare two character values or two numeric values. Within the Boolean expression, you can use any of the Boolean operators and comparison operators that are available on the symbol toolbar of the Formula Editor window. You can also use the ISCURRENT, ISLEAF, and ISLEAFCROSSING functions, which return Boolean values.

The second and third arguments must be of the same data type. They must both yield numeric values, or they must both yield character-string values, or they must both yield Boolean values. (Recall that a character-string or Boolean value can be returned only as an argument of another function.)

INDEX

returns an integer that indicates the starting position of one string within another string. The INDEX function takes two arguments:

- The first argument is the string to search in.
- The second argument is the string to search for.

If the second string occurs more than once in the first string, then the INDEX function returns the starting position of its first occurrence. If the second string does not occur at all in the first string, then the INDEX function returns `0`. For example:

- **`INDEX("herewego","we")`** returns `5`
- **`INDEX("nono","no")`** returns `1`
- **`INDEX("yesyes","no")`** returns `0`

INDEXC

returns an integer that indicates the first position in an input string that contains any character in a specified set of characters. The INDEXC function takes two arguments:

- The first argument is the input string.

- The second argument is a string that represents the set of characters to search for.

If none of the specified characters is in the input string, then the INDEXC function returns `0`. For example:

- **`INDEXC("education","aeiou")`** returns **`1`**
- **`INDEXC("school","aeiou")`** returns **`4`**
- **`INDEXC("jklmn","aeiou")`** returns **`0`**

ISCURRENT

returns a Boolean value that indicates whether a specified member is in the crossing where the function is evaluated.

The ISCURRENT function takes two arguments: a dimension type code and the code of a member that belongs to the dimension type. The ISCURRENT function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF(ISCURRENT("TIME","JAN2007"),100,0)
```

ISLEAF

returns a Boolean value that indicates whether the member of a specified dimension type that is in the crossing where the function is evaluated is a leaf member.

The ISLEAF function takes one argument: a dimension type code. The ISLEAF function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF(ISLEAF("TIME"),100,0)
```

Note: A virtual child member counts as a leaf member.

ISLEAFCROSSING

returns a Boolean value that indicates whether the crossing where the function is evaluated consists entirely of leaf members.

The ISLEAFCROSSING function does not take an argument. The ISLEAFCROSSING function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF(ISLEAFCROSSING(),100,0)
```

Note: A virtual child member counts as a leaf member.

ISVIRTUALCHILD

returns a Boolean value that indicates whether the member of a specified dimension type that is in the crossing where the function is evaluated is a virtual child member.

The ISVIRTUALCHILD function takes one argument: a dimension type code. The ISVIRTUALCHILD function is typically used as the first argument of the IF function or the NESTIF function, as in the following example:

```
IF(ISVIRTUALCHILD("INTORG"),100,0)
```

LASTCHILD

returns the code of the member that is the last-listed hierarchical child of the member that is specified either explicitly or implicitly by the argument of the function.

The LASTCHILD function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.

- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"]["TIME"=LASTCHILD("TIME")]
```

When evaluated at a year crossing, this reference retrieves the value of the Headcount account for the last quarter of that year. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the last month of that quarter.

Instead of a dimension type code, the argument of LASTCHILD can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="Headcount"]["TIME"=LASTCHILD(NEXT(("TIME"))]
```

If the LASTCHILD function is applied to a member that has one child, then it returns the code of that one child.

If the LASTCHILD function is applied to a leaf member, then the operand that the function is part of receives a value of 0.

LASTSIBLING

returns the code of the member that is the last-listed hierarchical sibling of the member that is specified either explicitly or implicitly by the argument of the function.

The LASTSIBLING function takes one argument, which can be either a dimension type code or a function that returns a member code:

- If the argument is a dimension type code, then this code implicitly specifies the member of that dimension type that is in the crossing where the function is evaluated.
- If the argument is a function that returns a member code, then the returned code explicitly specifies a member.

Consider a year-quarter-month time hierarchy and the following reference:

```
["ACCOUNT"="Headcount"]["TIME"=LASTSIBLING("TIME")]
```

When evaluated at a January, February, or March crossing, this reference retrieves the value of the Headcount account for March. When evaluated at a quarter crossing, this reference retrieves the value of the Headcount account for the last quarter of the same year.

Instead of a dimension type code, the argument of LASTSIBLING can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="Headcount"]["TIME"=LASTSIBLING(PARENT(("TIME"))]
```

If the LASTSIBLING function is applied to an only child, then it returns the code of that one child.

LEFT

returns a string that is formed by removing all the leading spaces from an input string. It takes one argument, which must have a character value. For example, **`LEFT(“   abc”)`** returns the character string **`abc`**.

LENGTH

returns the length of an input string. It takes one argument, which must have a character value. For example, **`LENGTH(“January”)`** returns **`7`**.

LOG

computes the natural logarithm of a number. It takes one argument, which must have a numeric value greater than zero.

LOG is the inverse of EXP. For example, `LOG(EXP(2.65))` returns `2.65`.

LOWCASE

returns a character string that is formed from an input character string by converting each uppercase letter to the corresponding lowercase letter. This function takes one argument, which must have a character value. For example, `LOWCASE("HIGH5")` returns `high5`.

MAX

returns the value of the argument that has the largest value. The MAX function can have any number of arguments, 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`**

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(ANCESTOR("TIME","Year"))]
```

In this case, when the formula is evaluated at a year, quarter, or month, the value of the Headcount account is for the first month of that year.

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. The property code is case-sensitive.

For example, the following use of the PROPERTY function returns the account type of the account that is in the crossing where the function is evaluated:

```
PROPERTY("ACCOUNT","AccountType")
```

Here are some property codes that you can use in this function and the values that the function can return for each of them:

AccountBehavior

the category to which the account type belongs. This property is valid only if the member is an account. The following values can be returned: **`Balance`**, **`CTA`**, **`Flow`**, **`Hybrid`**, and **`NonFrequency`**. (**`NonFrequency`** is another name for the Statistical account type. **`Hybrid`** is another name for the RetainedEarnings account type.)

AccountType

the account type of an account. This property is valid only if the member is an account. The following values can be returned: **`Asset`**, **`Liability`**, **`Equity`**,

`Revenue`, **`Expense`**, **`RetainedEarnings`**, **`CTA`**, **`StatisticalBalance`**, **`StatisticalFlow`**, **`NonFrequency`** (another name for the Statistical account type)

BalanceType
: the balance type of an account. This property is valid only if the member is an account. The following values can be returned: **`Credit`**, **`Debit`**

ExchangeRateType
: the exchange rate type of an account. This property is valid only if the member is an account. The following values can be returned: **`PeriodAverage`**, **`PeriodClose`**, **`PeriodOpen`**, **`Custom1`**, **`Custom2`**, **`Derived`**, **`Historic`**

Functional Currency
: the functional currency of an organization. This property is valid only if the member is an organization. Any currency code can be returned.

Intercompany
: whether an account is an intercompany account. This property is valid only if the member is an account. The following values can be returned: **`True`**, **`False`**

Level
: the period type of a time period. This property is valid only if the member is a time period. The following values can be returned: **`AllYears`**, **`Year`**, **`HalfYear`**, **`QuarterYear`**, **`Month`**, **`Week`**, **`Day`**

ReportingEntity
: whether an organization is a reporting entity. This property is valid only if the member is an organization. The following values can be returned: **`True`**, **`False`**

REPEAT

returns a string that is formed from an input string by appending the input string to itself a specified number of times. The REPEAT function takes two arguments. The first argument is the input string. The second argument is the number of repetitions, beyond the first occurrence. For example:

- **`REPEAT(“no”,1)`** returns **`nono`**
- **`REPEAT(“ha”,2)`** returns **`hahaha`**

REVERSE

returns a string that is formed by reversing the character sequence of an input string. It takes one argument, which must have a character value. For example, **`REVERSE(“nuts”)`** returns **`stun`**.

RIGHT

returns a string that is formed by removing all the trailing spaces from an input string. It takes one argument, which must have a character value. For example, **`RIGHT(“abc   ”)`** returns the character string **`abc`**.

ROUND

returns the result of rounding its first argument to the number of decimal places given by its second argument. The first argument must have a numeric value. The value of the second argument must be a nonnegative integer. The second argument is optional. Omitting the second argument is equivalent to giving the second argument the value 0: the result is rounded to the nearest integer.

If the most significant digit that is dropped is 5, then the least significant digit that is not dropped is increased by 1. In other words, the absolute value increases.

For example:

- **`ROUND(2.425,2)`** returns **`2.43`**
- **`ROUND(2.425,1)`** returns **`2.4`**
- **`ROUND(2.425,0)`** returns **`2`**
- **`ROUND(2.425)`** returns **`2`**
- **`ROUND(-2.425,2)`** returns **`-2.43`**
- **`ROUND(-2.425,1)`** returns **`-2.4`**
- **`ROUND(-2.425,0)`** returns **`-2`**
- **`ROUND(-2.425)`** returns **`-2`**

SCAN

returns a requested segment of a segmented input string. The following characters are treated as delimiters that divide the input string into countable segments:

`<(+&!$*);^-/,%|`

The SCAN function takes two arguments:

- The first argument is the input string.
- The second argument is the numeric position of the requested segment. Positions are counted from the beginning of the input string if the second argument is positive, and from the end of the input string if the second argument is negative.

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
- Source

The VIRTUALCHILD function is equivalent to the *.vc* suffix. For example, each of the following refers to the virtual child of the CURRLIAB member of a model's account hierarchy:

```
["ACCOUNT"=VIRTUALCHILD("CURRLIAB")]
```

```
["ACCOUNT"="CURRLIAB.vc"]
```

The *.vc* suffix must be lowercase.

Instead of a literal member code, the argument of VIRTUALCHILD can be a function that returns a member code, as in the following example:

```
["ACCOUNT"="UnitsSold"]* -1 *["ACCOUNT"="Price"]
["PRODUCT"=VIRTUALCHILD(PARENT(CURRENT("PRODUCT")))]
```

XRATE
returns a specified exchange rate.

The XRATE function takes two arguments: an exchange rate type code and the code of the target currency. It gets the other information that it needs from the crossing where it is evaluated.

For any exchange rate type, the XRATE function uses the following members of the crossing where it is evaluated:

- The analysis member determines which exchange rate set to use.
- The time member determines the relevant time period.
- The currency member determines the source currency.

For a complex exchange rate that depends on the members of certain other dimension types, the XRATE function also uses the members of those other dimension types in the crossing where the function is evaluated.

The first argument must be 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")
```

See Also

“Overview of Formula Types and Formulas” on page 109

Chapter 14
Forms and the Forms Workspace

Forms and Form Sets

A form set is a group of forms that are typically used for data entry via Web data entry or the SAS Financial Management Add-In for Microsoft Excel. Each form set includes a workflow process for submitting, approving, rejecting, and recalling forms.

When you create a form set, you select a target hierarchy that determines the members to be included in the workflow.

Workflow Types

Overview

SAS Financial Management supports two types of workflows:

- bottom-up workflows
- top-down workflows

Each workflow type has a predefined set of states through which a form advances. You can customize a workflow by writing a stored process that executes before or after the form is advanced to the next state in the workflow. For details, see “Customizing a Workflow” in the *SAS Financial Management: Customization Guide*.

Bottom-Up Workflows

In a bottom-up workflow, users enter data for leaf members of the target hierarchy and then submit it to higher levels for review. Each review can result in either approval or rejection.

Forms that have been submitted but not yet approved can also be recalled by the author.

Note: Data that has not been approved is not visible in parent forms.

Top-Down Workflows

In a top-down workflow, an initial form author enters data at the highest level of the form set's target hierarchy, using the virtual child (VC) member. The form author allocates these amounts to subordinate members of the target hierarchy and pushes the form to the next level in the workflow. Subsequent authors determine the amount to allocate to their children and descendants, if applicable.

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.

Each form set 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

The following table summarizes form set capabilities.

Table 14.1 *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 each selected member of the target hierarchy in the workflow. However, a form can 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 selected member of the target hierarchy in the workflow.
Non-target hierarchies	Every form has access to all crossings as defined by the tables in the template, with consideration for security and the target hierarchy.	For non-target hierarchies, forms might differ in the display and access of crossings on a group basis as defined on the Form Details page, in addition to consideration for security and the target hierarchy.
Form authors and reviewers	A bottom-up form can have multiple authors and reviewers. A top-down form can have multiple authors.	A bottom-up form can have only one author and one reviewer. A top-down form can have only one author.
Formula types	Driver, modeling, reporting, and Excel-based calculated members are supported.	Driver and Excel-based calculated members are supported.
Form data entry	Users can edit forms in Microsoft Excel or via Web data entry, depending on the form set properties.	Users edit forms using Web data entry. (The form template is designed in Microsoft Excel.) Supplemental schedules are not available.

Chapter 15

Working with Forms and Form Sets

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 145)

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 135.
   - (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 143.
   - whether users can add comments to cell crossings in these forms; if so, the scope of those comments.
   - the writable members of the Analysis dimension.

     Note: At least one writable analysis member is required for data entry. If you plan to use the form set only to circulate data for approval or informational purposes, without any data entry, no writable analysis members are required.

- the date and time that the form set is due and the actions, if any, that will be taken regarding this deadline. You can apply the same deadline to all forms in the form set or set different deadlines for individual forms. See "Enforcing Form Set Deadlines" on page 147.

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. (CDA tables cannot be viewed in forms that are being edited on the Web.)

   (Operational 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 "Overview of Designing Data Entry Forms" on page 187.

2. As necessary, apply formatting, calculated members, and Visual Basic Application (VBA) code that operates on the data-entry form and calls methods in the SAS Financial Management API for Microsoft Excel.

   To protect cells from data entry, you can add cell protection rules to the template or to the model that is used by a data-entry table. To hide cells from display, you can create filters in the template.

3. Select **Save Template** to save the template on the server. (You do not need to save the template on your desktop.)

See Also

"Designing Cell Protection for Data-Entry Tables" on page 195

Edit the Form Set Properties

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 🔄.)

Edit the form set properties as needed. For example:

- Change the editor for financial forms.

  By default, financial forms open in Microsoft Excel. If you want the forms to be available for Web data entry, you can make that selection in the form set properties after you create the form set. (Operational forms can be edited only on the Web.)

- Change the way a form is identified.

  By default, a form is identified by the name of its target member. In the form set properties, you can change the identification to be the description of the target member.

- Select different members of the target hierarchy.

When you select the form set, the **Forms** tab displays the forms in the workflow. Each form has a status of Draft until the form set is published. In Draft state, you can open a financial form in Microsoft Excel if the form set properties permit it. However, the data-entry tables are read-only. (In a financial form template, data-entry tables are writable, even in Draft state.)

Note: You cannot open an operational form in Microsoft Excel, and the data-entry tables in an operational form template are not writable.

You can edit individual form properties as necessary. For example, you might want to assign additional authors or reviewers to a financial form, or you might want to stagger the deadlines for various forms.

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. From a Web browser, users can log on to SAS Financial Management and view their available forms. (Form administrators can access all forms in the Forms workspace.)

   In a bottom-up workflow, users open a form, enter data, and submit the form to the next state in the workflow.

   In a top-down workflow, users allocate data and push the form to the next state in the workflow.

4. When all the forms are submitted, or when the review period is over, the administrator marks the form set Complete. Changing the status to Complete removes the forms from display in SAS Financial Management.

For financial form sets, data that is loaded into the associated cycle is immediately visible in both the draft or published form set state. For operational form sets, data that is loaded into the associated cycle is immediately visible only in the draft state. Once the form set is published, any data that is loaded to the operational cycle is not visible in the forms.

Republishing a Form Set

If necessary, you can reset a form set and republish it.

If you republish 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.)

If you republish an operational form set, you are given the choice of deleting or preserving the data that was previously entered into the form set.

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 target hierarchy's as-of date and time, as specified in the form set properties.

Selecting a Hierarchy

To select a target hierarchy for a financial form set, you make these selections:

- **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.

- **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.

- **As-of date**: The default as-of date is **Current until form set is published**. With the default, when you publish a form set, the target hierarchy is based on the date and time of publish.

  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. It is always the set of users that is specified on the Authors page of the form's Properties window at the moment that the form set is published.

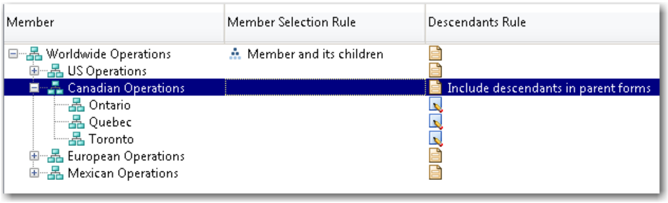
Note: If you are creating a form set that will include an HCM supplemental schedule, then you must select the organization dimension and an organization hierarchy that is used by SAS Human Capital Management at your site.

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 a single form that includes additional subordinate members.

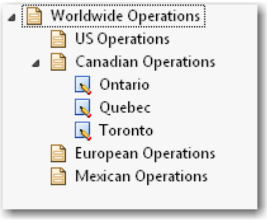
To include additional members in a parent form, select the **Include descendants in parent forms** descendant rule. Use additional rules to include or exclude specific members as writable descendants. For details, see the online Help.

For example, assume that you have a member, Canadian Operations, with three child members, Ontario, Quebec, and Toronto, that you want to include in the form that is created for the Canadian Operations member. You set the member selection rule for Worldwide Operations to **Member and its children**. As a result, forms are assigned to the Worldwide Operations, US Operations, Canadian Operations, European Operations, and Mexican Operations members. Then you set the descendant selection rule for Canadian Operations to **Include descendants in parent forms**.



Member	Member Selection Rule	Descendants Rule
Worldwide Operations	Member and its children	
US Operations		
Canadian Operations		Include descendants in parent forms
Ontario		
Quebec		
Toronto		
European Operations		
Mexican Operations		

As a result, the Ontario, Quebec, and Toronto members are included in the form for Canadian Operations. Because you did not apply a similar descendant selection rule to the other children of Worldwide Operations, none of their descendants are included on their forms, as shown in the preview region:



Worldwide Operations
US Operations
Canadian Operations
Ontario
Quebec
Toronto
European Operations
Mexican Operations

Modifying the Workflow After Publishing the Form Set

For financial forms only, you can update the form set's as-of date, member selection rules, or descendant selection rules after publishing a form set. You do not need to reset and republish the form set.

Only forms that are affected by changes in hierarchy member selection result in a change in workflow status. For example, if a parent form has additional child forms, the parent takes on a status of **Unedited**. You are given the opportunity to send notifications to form authors.

Note: In a bottom-up form set, if changes result in the removal of target members, you have the option of deleting or preserving the data that is associated with the target member. In a top-down form set, if changes result in the removal of target members, you have the option of moving the data to the parent's virtual child or preserving the data that is associated with the target member.

(Operational Form Sets) Selecting Target Members

Overview

Operational planning is designed to support different levels of detail for different target hierarchy members. Each target hierarchy member is identified by a member property value. These member property values facilitate grouping forms by property values.

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:

- The first step is to make broad selections of target members based on member property values.
- The second step is to refine the selection using member selection rules.

When you publish the form set, the rules are applied based on the hierarchy's as-of date and time.

The Target Hierarchy

For the target hierarchy, operational form sets support selecting from the organization hierarchy or from a custom dimension type that is 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. If the model's as-of-date is current, then when the form set is published, both the model and form set target hierarchy are set to that date and time.

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, 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.

The Form Detail selection window enables you to manage the level of detail (for non-target hierarchy dimensions) that are available to each form in a group. If you select a group, a separate Form Detail window is displayed for that group. For each available dimension, you can remove hierarchy levels to display in that group's forms.

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. There are no reviewers in top-down workflows.

An operational form can have only one author and one reviewer.

A financial form can have multiple authors and reviewers. If a form has more than one author, any author can edit the form. For forms with multiple reviewers, see "Assigning Multiple Reviewers to a Financial Form" on page 146.

Authors can be assigned to forms automatically and manually. For both options, users must have appropriate capabilities in order to be added as a form author or reviewer. (See "Assigning Groups and Roles" in the *SAS Financial Management System Administrator's Guide*.

Automatically Assigning Authors and Reviewers

Form authors and reviewers are automatically assigned as follows:

- **form authors:** the form's target hierarchy member is automatically included on the **Authors** tab. In the Dimensions workspace, you can assign users to this member. Those users are assigned as form authors.

  For operational forms, only one user is assigned.

- **form reviewers (bottom-up forms only):** the closest parent member in the target hierarchy is automatically included on the **Reviewers** tab. In the Dimensions workspace, you can assign users to this member. Those users are assigned as form reviewers.

  (For operational forms, only one user is assigned.)

Manually Assigning Authors and Reviewers

In the form properties or form set properties, you can manually add authors and reviewers to a form using one of these options:

- Open the form properties and select the **Authors** tab.
- Open the form properties and select the **Reviewers** tab.
- Alternatively, select **Authors** or **Reviewers** for a form set. In the window that appears, select a form.

You can add dimension members (from the target hierarchy or from another dimension) or users. For financial forms, you can also add groups.

Users are filtered to ensure that they have the appropriate capabilities. Groups are not filtered at this time. However, users' capabilities (from user or group properties) are checked when they open the Forms workspace of SAS Financial Management on the Web. Only users with the appropriate capabilities can edit or approve forms.

For more information, see the online Help for the appropriate tabs.

Assigning Multiple Reviewers to a Financial Form

There are two ways to assign multiple reviewers to a bottom-up financial form:

- Assign multiple reviewers at the same level (the default).

  Any reviewer can approve the form.

- Assign a reviewer workflow order within individual forms.

  In this case, a reviewer at one level must approve the form for it to advance to the next review level. (If there are multiple reviewers at the same level, only one must approve the form at that level.) The form state remains Partially Approved until the form has been approved at all levels.

Enforcing Form Set Deadlines

Setting Deadlines

On the Deadlines page of a form set, you can set the date and time that a form set is due and specify what action, if any, is taken regarding the deadline.

By default, the form set deadline applies to all forms in the form set. However, you can set different due dates for specific forms or for specific hierarchy levels. Deadlines that are set at the form level cannot be later than the form set deadline.

Select **Do not lock forms after the deadline** to set a date and time when all forms in a form set are expected to be submitted and approved. In the SAS Financial Management Web application, an overdue indicator is shown for forms that are past due. However, forms remain available for data entry and other workflow actions (subject to workflow rules) until they are explicitly locked or until the form set is completed.

Select **Lock all forms after the form set deadline** to enforce a due date and time when all forms in a form set must be submitted and approved. This autolock option is enforced at the form set deadline and applies only to the forms, not the form set template. This option does not apply to staggered form deadlines.

In either case, you have the option of sending automated reminders of upcoming form deadlines.

Note: After you publish a form set, you can send additional (non-automated) reminders by selecting the form set or one or more forms and then selecting **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 13

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. The model's default read can be overridden in the table properties, so that form sets can reference the same model with varying default read members. If you modify the default read member in the table properties, it is no longer affected by changes to the default read member in the model.

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. 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. You would not need to modify the form templates.

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.

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 15.1 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.

Prompt	Selection	Description
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, writable analysis members, and locks). If a crossing would otherwise be editable but contains a period that cannot be edited, it is displayed in gray and is protected against data entry.

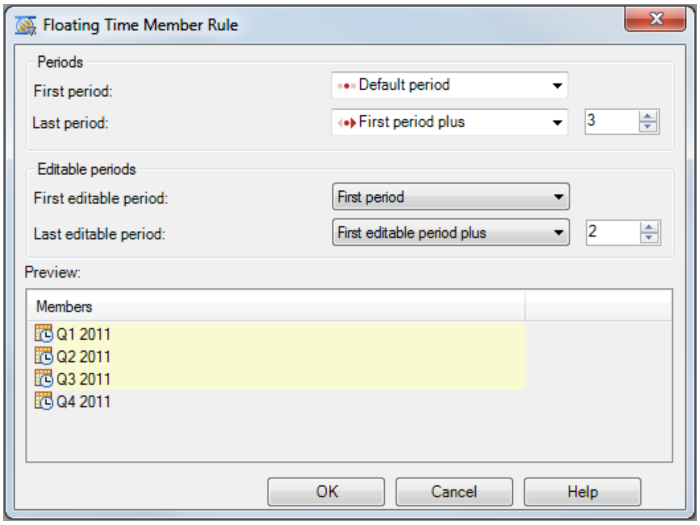
To define the editable time period, in the **Editable Periods** region, make selections for **First editable period** and **Last editable period**. The selections are as follows:

Table 15.2 *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.
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 gray. 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.



Floating Time Member Rule

Periods

First period: Default period

Last period: First period plus 3

Editable periods

First editable period: First period

Last editable period: First editable period plus 2

Preview:

Members

Q1 2011
Q2 2011
Q3 2011
Q4 2011

Managing Cell Comments

About Cell Comments

A cell comment is text that is associated with a crossing in a form or a read-only table.

Enabling Cell Comments

For forms, cell comments can be enabled in the form set wizard or in the form set properties. They can have the following scope:

- Form set scope: comments are available only to other forms in the same form set.
- Cycle scope: comments are available to any form or read-only table that uses the same cycle.

  Cycle scope applies only to financial forms.

For financial reports, cell comments can be enabled in the table properties. By default, cell comments are disabled. Comments can have the following scope, which applies to all tables in the workbook:

- Report scope: comments are available only within the current report.
- Cycle scope: comments are available to any form or read-only table that uses the same cycle.

Note: Modifying cell comment scope is not retroactive. After a comment has been added, changing the cell comment scope for a form set or a read-only table has no effect on existing comments.

Deleting Cell Comments

Within a form or report, users can select and delete cell comments, with some restrictions.

In SAS Financial Management Studio, administrators can delete comments for one or more periods in a cycle, via the Delete Cell Comments wizard in the Periods workspace. In this wizard, you can restrict the deletion to specific dimension members.

Cell comments are typically useful only for a limited period of time. We recommend regular deletion of cell comments from older cycles so that the system does not run out of comment storage space.

Note: If you delete a financial form set, you have the option of preserving cell comments that are visible outside the form set. Deleting a cycle deletes the cell comments for that cycle, regardless of scope.

Allow Data to Be Entered for Parent Members (Financial Forms Only)

On the **Data-Entry** tab of the table properties, an administrator can enable or disable options for entering data directly into a roll-up (parent) cell of a data-entry table. These options apply only to financial forms in a bottom-up workflow. By default, these options are disabled.

On the **Data-Entry** tab of the table properties, an administrator can enable or disable options for entering data directly into a roll-up (parent) cell of a data-entry table.

These options apply only to financial forms in a bottom-up workflow.

By default, these options are disabled.

Allow data to be entered for parent members other than time

If this option is enabled, 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 does not apply to the Time dimension.

Allocate from Parent members other than Time using predefined weights

This option enables automatic allocation. A value that is entered in a parent cell is automatically distributed among eligible leaf members. The allocation is based on weights that are defined in the Allocation Weights window.

This option does not apply to the Time dimension and cannot be selected if either of the other options is selected.

Note: If automatic allocation is enabled for any of the data-entry tables on a form, the form cannot be edited offline.

Allow data to be entered for Time Parent members

If this option is enabled, users can enter a value in a non-leaf member of the Time dimension. Because the Time dimension has no .VC members, you must specify how the new value is distributed to the parent’s leaf members. (The leaf members must be on the table. Otherwise, no allocation takes place.)

Under **Spread options**, select one of the following:

- **Proportionally**: the value is distributed across the time leaf members based on their weights for the selected Analysis member.

  If you select **Same as target**, the value is distributed in proportion to the cells’ current values.

If you select a different Analysis member, the weight comes from the corresponding cell crossing for that Analysis member. Users must have Read access to that member.

- **Proportionally using prior year**: the value is distributed across the time leaf members. The allocation is based on the members' weights in the corresponding cell crossing for the previous year.

  With this option, you can also select a different Analysis member. Users must have Read access to that member.

- **Evenly**: the value is distributed equally across the time leaf members.

- **Predefined pattern**: the pattern that you select is used as weights for distributing the value to the time leaf members.

  These predefined patterns serve the special purpose of spreading amounts over the three months of a quarter. They are meaningful only if you are spreading over months and you are using the accounting convention that each month consists of either four or five whole weeks. The pattern repeats as necessary.

If you select **Allow spread options to be changed in the form**, users can make a different selection during data entry, if they are editing the form in Microsoft Excel. In Web data entry, the default spread option applies.

If any of these options is enabled, be aware of the following behavior:

- 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

When auto allocation is enabled, form users can directly enter values into the cell for parent members in dimensions other than time and have that value allocated based on predefined weights.

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 151
- "How the Allocation Is Applied" on page 153

Eligible Cells

For automatic allocation to take place, the crossing must contain a roll-up point in at least one dimension other than Time. (If the crossing 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, a leaf descendant must meet the following requirements:

- It must be writable.
- 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 and filter member combinations 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.

There is a special case: when member selection rules for a hierarchy include a roll-up point but not its descendants, 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. See the list above.

See Also

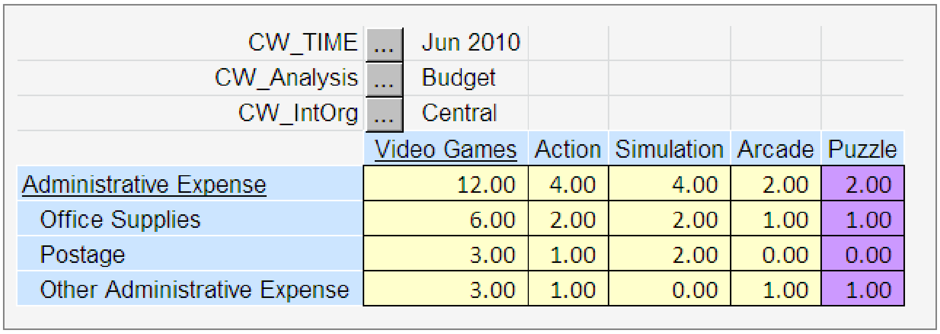
- "Allocations to Members of the Target Hierarchy" on page 155
- Chapter 16, "Security in SAS Financial Management," on page 159

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 the Source dimension, the default is Total.) For more information about the weight table, see "Editing Allocation Weights" on page 156.

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:

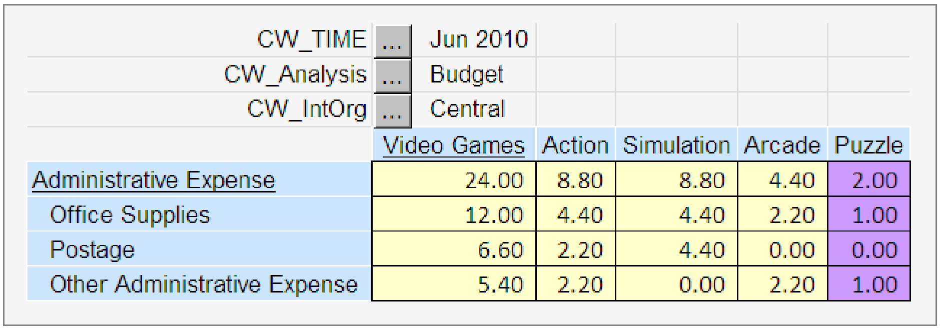


CW_TIME ...	Jun 2010				
CW_Analysis ...	Budget				
CW_IntOrg ...	Central				
	Video Games	Action	Simulation	Arcade	Puzzle
Administrative Expense	12.00	4.00	4.00	2.00	2.00
Office Supplies	6.00	2.00	2.00	1.00	1.00
Postage	3.00	1.00	2.00	0.00	0.00
Other Administrative Expense	3.00	1.00	0.00	1.00	1.00

The **`Puzzle`** member of the Product dimension is displayed in gray, 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.



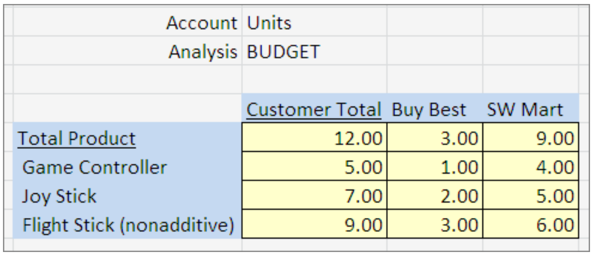
CW_TIME ...	Jun 2010				
CW_Analysis ...	Budget				
CW_IntOrg ...	Central				
	Video Games	Action	Simulation	Arcade	Puzzle
Administrative Expense	24.00	8.80	8.80	4.40	2.00
Office Supplies	12.00	4.40	4.40	2.20	1.00
Postage	6.60	2.20	4.40	0.00	0.00
Other Administrative Expense	5.40	2.20	0.00	2.20	1.00

The automatic allocation is to all eligible leaf members that roll up (directly or indirectly) to the cell where the value is entered.

Allocations to Nonadditive Members

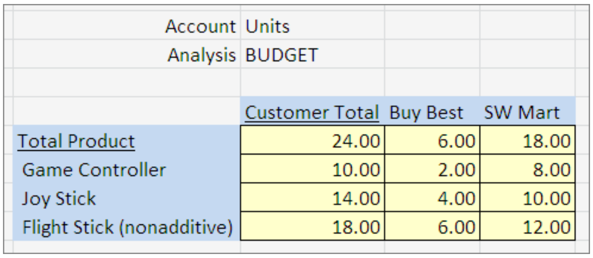
A hierarchy might contain members that have enabled the option not to roll up to the parent. If no cells in the hierarchy do roll up to the parent, then no allocation occurs at all. Otherwise, nonadditive cells are updated in proportion to the changes in the additive cells.

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 nonadditive 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 nonadditive 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).

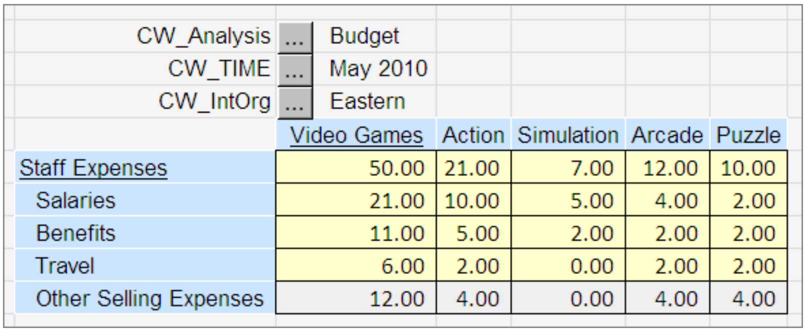
There are two situations in which allocation does occur for the target hierarchy:

- in a form template, which can display all members of the target hierarchy as writable
- in a form that includes writable target members

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.

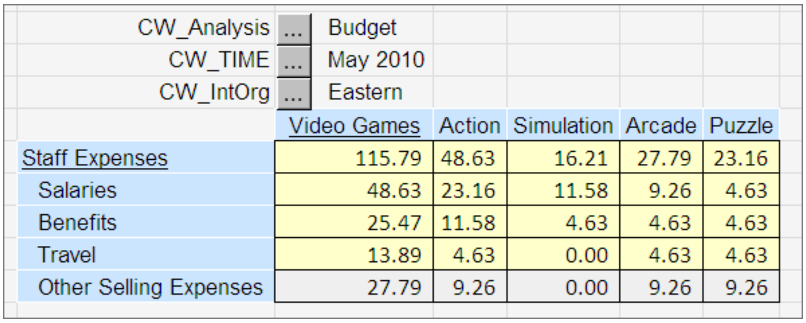
In this example, the value of Other Selling Expenses is computed by this formula: `["ACCOUNT"="Travel"]*2`. The cells for Other Selling Expenses are gray and cannot receive an allocation.



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.

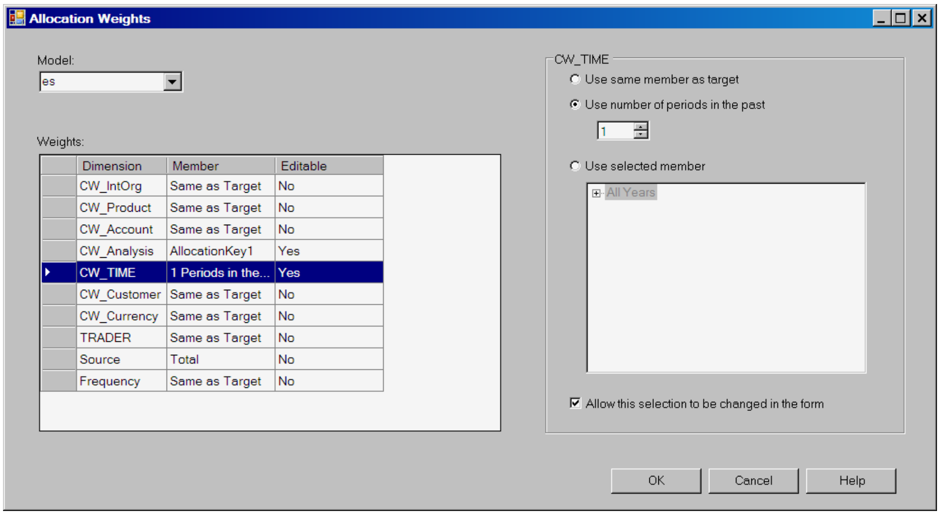
Reallocation

For a form template or form that is being edited in Microsoft Excel, the **Reallocate** option is available. This option is typically used when you modify the allocation weights but do not need to modify the amount entered at the parent. After modifying the weights, highlight the crossing to reallocate and select **Edit** ⇨ **Reallocate**.

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.



Allocation Weights

Model: es

Weights:

Dimension	Member	Editable
CW_IntOrg	Same as Target	No
CW_Product	Same as Target	No
CW_Account	Same as Target	No
CW_Analysis	AllocationKey1	Yes
CW_TIME	1 Periods in the...	Yes
CW_Customer	Same as Target	No
CW_Currency	Same as Target	No
TRADER	Same as Target	No
Source	Total	No
Frequency	Same as Target	No

CW_TIME

○ Use same member as target

● Use number of periods in the past

1

○ Use selected member

All Years

☑ Allow this selection to be changed in the form

2. For each editable dimension, select a member (or rule, for Time) to define the allocation weights. When an amount is entered at a parent for allocation, the values that are allocated to the eligible crossings are based on the crossings that are defined in the allocation weights table.

   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 158.

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 allocation weight member 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).

By default, weights cannot be changed by form users. In the form template, you can mark the weights for one or more dimensions as editable in the context of an Excel form.

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. Typically, the Analysis dimension is referenced. However, any dimension can be used other than the target hierarchy, Source, Currency, or Frequency dimension.
2. Create a form set and a data-entry table that include this member.

   Load allocation weight values via data entry or SAS Data Integration Studio.

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.

Supplemental Schedules (Financial Forms Only)

Supplemental schedules are designed to enable users to enter values at a more detailed level for a specific set of crossings. They apply only to financial form sets.

A supplemental schedule is associated with a data-entry table. It contains measures that correspond to members of the data-entry table, as well as custom measures that were designed for use in a supplemental schedule.

For bottom-up form sets, the sum of the facts for each measure for detail members is saved to the related crossing in the source data-entry table. The facts from a supplemental schedule are also stored in a separate database that is maintained by the supplemental data provider.

For top-down form sets, facts in a supplemental schedule are not saved to the source data-entry table. However, the facts from the supplemental schedule are stored in the supplemental data provider’s database. SAS Financial Management does not require or enforce that the values in the supplemental schedule remain synchronized with the source data-entry table.

See Also

“Creating a Supplemental Schedule” on page 192

Chapter 16

Security in SAS Financial Management

About Security in SAS Financial Management

Security features in SAS Financial Management include the following:

- **Metadata permissions and operating system permissions** can be used to define permission settings for folders and files.

  For information about metadata permissions, see “Authorization” in the *SAS Intelligence Platform: Security Administration Guide*. For information about operating system permissions, see “What to Do Next: Administration Tasks” in the *SAS Intelligence Platform: System Administration Guide*.

- **Role-based security** determines a user's ability to perform various tasks, such as creating a dimension or editing a form. Roles are mapped to capabilities, which determine the operations a user can perform.

  For information about groups and roles for SAS Financial Management, see “Assigning Groups and Roles” in the *SAS Financial Management: System Administrator's Guide*.

- **Object security** consists of permissions that you apply to cycles, dimension types, dimensions, hierarchies, models, composite models, and custom properties, in order to restrict users' access to those objects.

- **Data security** consists of permissions that you apply to members of one or more dimensions, to restrict users' access to data.

The remainder of this chapter discusses the object security and data security features of SAS Financial Management.

Object Security

Why Use Object Security?

By default, SAS Financial Management grants access to all objects, as long as a user has the necessary capabilities.

For cycles, models, and composite models, security can be used to restrict access to the object. For dimension types, dimensions, hierarchies, and custom properties, there are separate Read, Update, and Delete permissions.

Access Permissions for Cycles, Models, and Composite Models

For cycles, models, and composite models, permissions are defined simply in terms of access. If a user is denied access to any of these objects, they are also denied access to any objects that depend on it. For example, if a user is denied access to a specific model, the user is also denied access to any composite models, form sets, reports, and CDA formulas that reference that model.

Read, Update, and Delete Permissions

Dimension types, dimensions, hierarchies, and custom properties have separate Read, Update, and Delete permissions.

- **Read permission.** If Read permission is denied, it affects access to any objects that depend on it. For example, if a user is denied Read permission for a dimension type, the user cannot access any dimensions, hierarchies, cycles, models, or form sets that rely on that dimension type.

  If a user is denied Read permission for a custom property, the user cannot access that property in SAS Financial Management Studio. In a form or report, the custom property is omitted (without affecting other access to the form or report).

- **Update permission.** If Update permission is denied, users cannot modify the associated object.

  Update permission requires Read permission.

- **Delete permission.** If Delete permission is denied, users cannot delete the associated object.

  Delete permission requires both Read and Update permission.

A dimension inherits Read and Update permissions from its dimension type. A hierarchy inherits Read and Update permissions from its dimension. For example, if you have read-only access to a dimension type, you have read-only access to its dimensions and to the hierarchies that belong to those dimensions.

A dimension member also inherits Read and Update permissions from its dimension. These permissions affect the ability to view or modify dimension member properties in

SAS Financial Management Studio. They do not affect data security, which is set in the member properties.

Delete permissions are object specific and do not affect any objects that depend on that object. For example, a user might be denied Delete permission for a dimension but be granted Delete permission for a hierarchy within that dimension.

Note: Permissions that are applied to dimension types, dimensions, and hierarchies do not affect permissions that are assigned to custom properties.

Denying Access to an Object

To deny access to a cycle, dimension type, dimension, hierarchy, model, composite model, or custom property, do the following:

1. Log on to SAS Financial Management Studio. You must have the capability for the appropriate workspace and the Security Administration capability.
2. Open the object's properties and click the **Security** tab.

   You can modify security for multiple objects of the same type: select the objects, right-click, and select **Properties**. The Properties window appears, displaying only security properties.

   Note: You must modify security properties for each cycle and model separately.
3. Set the permissions for one or more identities.

   For details, see the online Help.

Superuser Access to Cycles and Models

There is an exception to object security: members of the Administrators group can access all objects, regardless of the security settings. (These users still require roles with specific capabilities in order to manipulate these objects.)

This superuser status does not apply to data security. You can apply member permissions that deny the Administrators group access to data.

Precedence Rules for Object Security

Precedence rules for object security are as follows:

- Permissions that are assigned to a user take precedence over permissions that are assigned to a group.
- If two groups with the same relationship to the user (identity precedence) apply conflicting permissions to the object, then the grant permission wins.
- When there is no specific permission, the default is to grant access.

In the following examples, assume that user Joe belongs to Group1 and Group2. Group2 is a member of Group3.

Scenario	Results
In SAS Financial Management Studio, you deny Group1 access to ModelA. You grant Joe access to ModelA.	Joe is granted access to ModelA. The permission that you assigned directly to the user takes precedence.

Scenario	Results
You grant Group1 access to ModelA. You deny Joe access to ModelA.	Joe is denied access to ModelA. Again, the permission that you assigned directly to the user takes precedence.
You deny Group1 access to ModelA. (You assign no permissions to Joe.)	Joe is denied access to ModelA.
You deny Group1 access to ModelA. You grant Group2 access to ModelA. (You assign no permissions to Joe or Group3.)	Joe is granted access to ModelA. Because both Group1 and Group2 are equally close to Joe, the grant permission takes precedence.
You deny Group1 access to ModelA. You grant Group3 access to ModelA. (You assign no permissions to Joe or Group2.)	Joe is denied access to ModelA. Because Group1 is closer to Joe than Group3 is, Group1’s permission takes precedence.

Data Security

Why Use Data Security?

Data security secures the data that is displayed in a report, in the output from a SAS program such as a stored process report, or in a planning form. The security settings can be broad or very specific.

By default, all users and groups have Read and Write permission for all members of each dimension. Regardless of the as-of date for a hierarchy, the permissions that are currently set on the member apply.

There are two ways to set permissions for data security:

- In SAS Financial Management Studio, you can set the permission on the **Security** tab of a selected member or members. This method directly populates the tables in the SAS Financial Management Data Mart.
- You can also create an external permissions file and use SAS Data Integration Studio jobs to populate the tables in the SAS Financial Management Data Mart.

TIP If you delete a member, it is archived along with its properties, including security settings. If you re-create the member, the security settings are also re-created.

Effects of Data Security

Denying Read or Write access to a dimension member has the following effects:

Task or Area	Effects
Editing a data-entry form or opening an Excel report	If a user is denied Read access to a member, all crossings that include that member are displayed as red cells. Their contents are not visible, and the cells are not writable. If a user is denied Write access (but not Read access), all crossings that include that member are protected. Their contents are visible but the cells are not writable. In Microsoft Excel, if the user right-clicks one of these cells and selects **Tools** ⇨ **Cell information**, a pop-up message explains the restriction. In Web data entry, the user right-clicks a cell and selects **Cell Information**.
Opening an Excel report	When a user opens a report that contains a read-only table, the slicer opens to the top-level member to which the user has access.
SAS programs	In a SAS program (such as a stored process), values are returned as `NaN` if data security prevents Read access.

Data security is inherited by subordinate members in a hierarchy. If there are no specific security permissions on a member, the permissions of its parent are applied. If a member does have security permissions, those permissions take precedence.

Precedence Rules for Data Security

By default, all users and groups have Read and Write permission for all members of each dimension. Authorization for member permissions takes the following path:

1. If permissions are directly assigned to the member and include the user or a group that the user belongs to:
    - Permissions that are assigned to a user take precedence over permissions that are assigned to a group. The closer the relationship between the user and the group, the higher the precedence.
    - If two groups with the same relationship to the user apply conflicting permissions to the member, the grant permission wins.
2. Otherwise, the authorization process checks to see whether any permissions are directly assigned to the parent member in the hierarchy. It not, the process works its way up the hierarchy.

Viewing Security Information

The Administration Workspace

To view security information, log on to SAS Financial Management from a Web browser and select the Administration workspace. For details, see the online Help for the Administration workspace.

The following information is available:

- object and data permissions
- registered users, groups and group membership, and the capabilities that are assigned to users and groups.

  Capabilities determine the actions a user can perform in an application, as well as the menus and links that are displayed. For SAS Financial Management, capabilities include administering the various workspaces in SAS Financial Management Studio, administering security, and submitting and approving forms.

You can save the currently displayed user, group, object, or data permissions in a comma-separated values (CSV) file.

Clear User Cache

On the **Users** or **Groups** tab of the Administration workspace, you can also clear the user cache. For performance reasons, user roles and capabilities are cached when the Web application server is started. This cache affects users' eligibility to edit or review forms, or to be assigned as form authors or reviewers.

Part 3

Using the SAS Financial Management Add-In for Microsoft Excel

Chapter 17

Getting Started with the Excel Add-In

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.

Accessing the Excel Add-In

The following describes how to access the Excel Add-in through typical tasks:

- **Design a financial report.** Open Microsoft Excel on your desktop and then use the **SAS Financial Management** ⇨ **Log On** option.
- **View a financial report.** From a Web browser, log on to SAS Financial Management and select **Reports**. From the navigation tree, select an Excel report. No additional logon is necessary.
- **Design a data-entry form template.** In the **Forms** workspace of SAS Financial Management Studio, select a form set and select **Show Template**. The form template opens in Microsoft Excel. No additional logon is necessary.
- **Enter data in a form.** From a Web browser, log on to SAS Financial Management and select **Forms**. From the list of available forms, select a form and click the Open in Excel button. The form opens in Microsoft Excel. No additional logon is necessary.

Chapter 18
Working with Financial Reports

About Financial Reports

Overview

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. It can contain read-only tables or cell data access (CDA) tables, or both.

Financial reports are based on financial models, not operational models.

Read-Only Tables

Read-only tables are multi-dimensional tables that provide interactive capabilities such as expanding and collapsing hierarchies, drill-down, filtering, pivoting, and commenting.

A dynamic report that is based on a read-only table displays current values from the SAS Financial Management database. A static report displays values at the time the report was published.

CDA Tables

Cell data access (CDA) tables are collections of cells that use CDA functions (such as CDAGet) to retrieve values from the SAS Financial Management database for the specified model. Reports can also contain single cells that use CDA functions to access database values.

A report that is based on a CDA table displays current values from the SAS Financial Management database. In a read-only table, users have options such as dynamic member selection and property selection rules. In contrast, the references in a CDA table are static.

Viewing a Financial Report

If you open a report from the Web, you are already logged on to SAS Financial Management and can view the report (subject to security provisions). If you open a dynamic report from a local directory, you must first log on.

If the report is dynamic and includes a read-only table, 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. A worksheet with multiple tables has the option of sharing common dimensions that are placed on the slicers.
- 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** to affect your view of the table.

Designing a Financial Report

Overview

With the appropriate capabilities, you can create a financial report in Microsoft Excel. There are three ways to retrieve numeric values:

- read-only tables
- CDA tables
- single-cell uses of a CDA function

Multiple read-only tables, CDA tables, and CDA functions can be included in a single worksheet. You can include ordinary Microsoft Excel structures and use the functionality of Microsoft Excel, with certain limitations. 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 SAS folders. To publish a report, select **Publish** and work through the Publish Report wizard. For details, see "Publishing a Financial Report" on page 172 as well as the online Help for the individual wizard pages.

Defining Read-Only Tables

A read-only table is based on a financial model, which retrieves data from a financial cycle.

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 a fixed time period, one or more floating time periods, or a combination. The floating time period provides a dynamic, flexible means of displaying time periods for reports that require frequent updates. It is defined relative to the table default read member, which is available in the table properties.

In addition to its row and column dimensions, a read-only table can include slicer dimensions. 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.

Multiple tables can share slicers for the same dimensions. Changes in the slicer for the source table updates any target tables that share slicers for that dimension.

See Also

"Coordinating Slicers between Tables" on page 210

Defining CDA Tables

Because of their flexibility, CDA tables are useful for designing formal financial reports. You can customize a CDA table using Microsoft Excel capabilities such as formatting, sorting, inserting rows and columns, or moving or hiding rows and columns. The formulas for CDA cells use absolute cell references, so that you can move a cell reference without affecting its value.

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 values.

Single-Cell Uses of CDA Functions

A single-cell use of a function such as CDAGet displays the return value in a single cell. 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 **Insert** ⇨ **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. If you want to use a Microsoft Excel reference to the cell that contains the label as an argument of a CDA function, then 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 report one cell at a time.

See Also

"Using CDA Functions in Microsoft Excel Formulas" on page 239

Publishing a Financial Report

Overview

To publish a financial report, select **Publish** from the **SAS Financial Management** ⇨ **File** menu.

Dynamic Reports

A dynamic Excel report is fully functional and contains all the worksheets in the workbook.

Publishing a dynamic Excel report is equivalent to using the Excel **Save as** to save the report as an Excel binary (.xlsb) file. The difference is that you publish a dynamic report to **`SAS Folders`** rather than to a local directory.

Static Reports

A static report cannot be modified and its data cannot be refreshed. You can publish the report as an Excel report, as a PDF, or as a SAS report that can be opened in SAS Web Report Studio. The report can include the entire workbook or only the active worksheet.

If you publish as an Excel report, you can choose to include public comments that are applied to any crossings that are visible in the report.

Selecting Slicer Members

If the tables contain slicers, you can select slicer members to be used in a static report. By default, a separate worksheet in the report is created for each combination of slicer members that you select. (In a PDF or SAS report, a separate section is created for each combination.) If the report contains multiple tables, only slicer dimensions and members that the tables have in common are eligible for selection.

For any dimensions where slicer selections are not specified, the currently displayed member on the table is used.

If you select **Create separate files**, a separate file is created for each member that you selected for one of the slicers. The other slicers become worksheets or sections in each report. This option is useful if you want to create separate reports for different groups of people.

Delivery Options

You can deliver a static report by saving it to a SAS folder, by saving it to a local directory, or by attaching it to an e-mail message that you send to one or more recipients.

If your table has slicers and you publish it to different files, you can give each file a different destination. For example, if your delivery method is e-mail, you can send each report to a different recipient.

Saving the Report Settings

When you publish a static report, you can choose to save the report settings as a batch file that you can execute later.

You can save the report settings as part of the workbook or in an external file. Later, you can reopen the workbook and use the saved settings as a basis for republishing the report. You can also use the settings to publish reports from a batch file.

Publishing Reports from a Batch File

When you save the report settings in the workbook or in an external file, you can later reference these settings from a Windows batch file.

This example uses report settings that are stored in the workbook:

```
"C:\Program Files\Microsoft Office\Office14\EXCEL.EXE"
/e/fm_operation=BatchPublish/fm_username=sasdemo/fm_password=DemoPasswd
/fm_batchpublish_settingsembedded=true "C:\test\MyReport.xlsb"
```

Note: Line breaks inserted for readability. The path to Microsoft Excel might be different in your installation.

- **`"C:\Program Files\Microsoft Office\Office14\EXCEL.EXE"`** is the path to the Microsoft Excel executable file. Replace this path as necessary.
- **`fm_operation`** is the operation to perform. It is always **`BatchPublish`**.
- **`fm_username`** and **`fm_password`** are the name and password of a user with the appropriate role and permissions for this operation.
- **`fm_batchpublish_settingsembedded`** should be **`true`** for report settings that are embedded in the workbook

- The last argument is the path to the workbook that is the source for the report. If it contains spaces, enclose it in quotation marks.

This example uses report settings that are stored in an external file:

```
C:\Program Files\Microsoft Office\Office14\EXCEL.EXE"
/e/fm_operation=BatchPublish/fm_username=sasdemo/fm_password=DemoPasswd
/fm_batchpublish_settingsfile="C:\test\MyReportSettings.xml"
"C:\test\MyReport.xlsb"
```

The **fm_batchpublish_settingsfile** is the path to the XML file that holds the report settings. If it contains spaces, insert **_FMsp_** in place of the spaces in the batch file. For example, with a report settings file called **My Report Settings.xml**, the example above would look like this:

```
"C:\Program Files\Microsoft Office\Office14\EXCEL.EXE"
/e/fm_operation=BatchPublish/fm_username=sasdemo/fm_password=DemoPasswd
/fm_batchpublish_settingsfile="C:\test\My_FMsp_Report_FMsp_Settings.xml"
"C:\test\MyReport.xlsb"
```

Note: When you publish from a batch file, the entire workbook is included. The report's scope (entire workbook or active worksheet) is not saved in the report settings.

Security for Financial Reports

When you create a SAS Financial Management table, security is applied based on your credentials. For example, you must have access permission for both the model that you select and the associated cycle. In addition, you must have Read permission for a dimension member in order to view data in a crossing that contains that member.

When a user accesses a published report, security is applied as follows:

- **Static reports.** If you publish a static report and make the report available to other users, they can view the report just as you created it. No additional object-level or member-level security is applied.
- **Dynamic reports.** To view a dynamic report, a user must be logged on to SAS Financial Management. Security applies just as it does for any dynamic content.

  Note: If the user does not have permission for a model, SAS Financial Management tries to substitute a compatible model (one with the same dimension types). The substitute model might be associated with a different cycle. SAS Financial Management displays a message explaining the substitution and encouraging the user to select an appropriate model (via the table properties).

If a report is published to the Web, an administrator can log on to SAS Management Console and apply permissions to restrict access to the folder or file. If it is published or saved to the file system, file system permissions can be used to restrict access to the report itself or to the folder that contains the report.

Chapter 19
Working with Data Entry Forms

About 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 SAS Financial Management Web application.

Each form typically contains one or more data-entry tables. Some forms also contain supplemental schedules, read-only tables, or CDA tables.

A form set is a collection of forms that can include data entry tables and are subject to a defined workflow process. The workflow for the form set is based on the selected target hierarchy and associated members within that hierarchy. Each form is associated with a member of the target hierarchy, although only certain members of the hierarchy might be selected to have forms.

To open a form, you log on to the Forms workspace of the SAS Financial Management Web application. Based on form set properties, you can open the form on the Web or in Microsoft Excel. From Microsoft Excel, you can also check out a form for offline editing, with some restrictions. For example, the form cannot include a supplemental schedule and cannot enable writing to parent members.

When you complete your edit or review, you use the Forms workspace to send the form to the next state in the workflow.

Note: You must have an appropriate role to enter or review data in a data-entry form.

See Also

- "Entering Data into a Supplemental Schedule" on page 180
- "Entering Data Offline" on page 179

Bottom-Up and Top-Down Form Sets

Bottom-Up Form Sets

The route that a form takes is determined by the form set's workflow. In a bottom-up form set, data is entered at the lowest hierarchical level that is defined in the form set. When the data entry for a form is complete, the author submits the form set to the reviewer for either approval or rejection. For each form level, data is entered, submitted to the next level up for approval, and aggregated if approved. If a form is rejected, it might be returned to the previous author for additional editing.

Top-Down Form Sets

In a top-down form set, data-entry proceeds down the target hierarchy. If you are the author for the top-level form, you enter all the amounts that will cascade down the target hierarchy for the workflow. To do this, you manually enter data into crossings that include the virtual child of the top member in the target hierarchy. Then you allocate those amounts to one or more lower levels of the target hierarchy. When the data entry is complete, you use the **Push** action to move the form to the next level down in the form set hierarchy.

Pushing a form does not change the amounts that were entered by the author. It only makes the subordinate forms accessible to the next set of form authors.

If you are the form author at the next level in the target hierarchy, you allocate the data that you received and push it down another level. Use the Allocate wizard to allocate amounts to one or more lower levels of the target hierarchy. Alternatively, you can manually enter an amount that is less than or equal to the allocated amount. Any difference is returned to the parent's virtual child member. You can also spread allocated amounts to siblings of the allocated crossing's dimension members. The total amount allocated must be the same or less than the original amount allocated.

At the lowest level of the target hierarchy, you cannot allocate, but you can use the **Spread** option to redistribute those amounts.

Note: If you are the author of a form that has descendants, you can allocate amounts to all lower levels and then use the **Push to All** action to bypass any further allocations.

Entering Data

By default, data cells are color-coded as follows:

Color	Meaning
Yellow	This cell is writable. You can enter data into it, and it can be the target of a spread, adjust values, or paste operation.
Red	This cell is not readable or writable. You might not be authorized to view its contents.
Gray	This cell is not writable. It might be a parent cell, whose value comes from its subordinate members. It might also be the target of a formula, one of its members might be read-only, or it might be protected, part of a non-writable floating time period, or locked. You can view its contents but you cannot enter data into it, and it cannot be the target of a spread, adjust values, or paste operation. *Note:* In some form sets, parent cells are writable. In that case, they are displayed in yellow. See “Entering Data into Parent Cells” on page 178.
Green	The cell receives data from a supplemental schedule. (See “Entering Data into a Supplemental Schedule” on page 180.)

For more information about a crossing, right-click the cell and select **Tools** ⇨ **Cell Information**.

You enter data in the yellow (writable) cells. These cells belong to the organization member or members 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.

Note: You can use Excel's **Copy** and **Paste** functionality to copy values to another writable location in the form. If the Excel clipboard is open, you can paste the same selection from the clipboard multiple times.

See Also

“Entering Data Offline” on page 179

Refreshing Values

By default, data is stored in the database as you enter it. If the new data affects additional cells, those cells are automatically updated.

If the **Delay writeback until refresh** option is enabled, data is not written to the database until a refresh operation takes place. (Until then, newly entered data is displayed in bold face type.) To explicitly refresh the display, click **Refresh** (to refresh the current worksheet) or **Refresh All** (to refresh all worksheets in the workbook). Other actions, such as changing a slicer member, also trigger a refresh. The refresh operation saves the data, formats recently entered numeric values correctly, and recomputes values that depend on the new data. For example, the data that you enter into a cell might trigger a calculation that affects other cells.

When you complete your data entry, close the form. If you have pending data records, you are asked if you want to save or discard them.

Unless you are entering data offline, there is no need to save the file on your local hard drive. However, if you made changes to the form (for example, if you removed an analysis member from the data-entry table or if you made formatting changes), select **Save Form Design** before closing the form.

Virtual Children

A virtual child (VC member) is automatically assigned to any member that has child members that roll up to it. In a data-entry table, the virtual child is a writable member whose values contribute to the parent member.

If the VC member is displayed in the table, you can use it to enter data for the parent without associating that data with a (real) child member. Virtual children are available in all hierarchical dimensions except the Time and Source dimensions.

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. 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 these ways:

- **Allow data to be entered for parent members other than time**

  The change in value is added to the parent's virtual child (VC) member.

- **Allocate from Parent members other than Time using predefined weights** (automatic allocation)

  A value that is entered in a parent cell is automatically distributed among eligible leaf members, based on weights that are defined in the Allocation Weights window.

- **Allow data to be entered for Time Parent members**

  Users can enter a value in a non-leaf member of the Time dimension. The distribution is defined in the table properties.

For more information, see the online Help for the table properties.

Note: These options 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 170

Entering Data Offline

Typically, you enter data into a form while the SAS Financial Management Add-In for Microsoft Excel is connected to the SAS Financial Management server.

You can also enter data into a form while it is offline. To make this possible, check out the form and save it as a local Excel file. Each time you edit the form, 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.

Note: You cannot check out a form that is already checked out, a form in which writing to parent members is enabled, or a form that contains a supplemental schedule.

To check out a form for offline data-entry:

1. Open the form in Microsoft Excel.
2. In Excel, 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

Overview

A form can include one or more supplemental schedules, which provide additional information to support data entry. Each supplemental schedule is associated with a data-entry table and can contain two types of measures:

- measures that correspond to members of the data-entry table.
- custom measures that were designed for use in a supplemental schedule. These measures might be numeric, or they might contain character strings, dates, or true or false values.

You can recognize a supplemental schedule by its last rows, which display column totals and averages in green cells. The corresponding crossings in the data-entry table are also displayed in green.

Enter Data in a Supplemental Schedule

Bottom-Up Forms

In a bottom-up form, entering data into a supplemental schedule is like entering data into a data-entry table. You enter data in yellow (writable) cells of detail records for a specific member of the data-entry table. For example, the detail records might be associated with the organization dimension. When you open the form, only the detail records for your organization member (or members) are displayed.

At the bottom of the supplemental schedule are two rows that display totals and averages for the detail records. If a measure is also a member of the data-entry table, its totals are saved in the data-entry table. (The **Detail averages** row is provided only for reference.)

Top-Down Forms

In a top-down form, the supplemental schedule contains a green **Non-allocated** row. This row displays the amount from the data-entry table that remains to be allocated in the supplemental schedule. You can distribute that amount among the detail records in the supplemental schedule.

No data entered in a supplemental schedule is saved to the data-entry table in a top-down form. The supplemental schedule for a top-down form enables you to create supplementary records that contain more detail than would be available in the data-entry form.

See Also

- "Entering Data" on page 177
- "Using Excel-Based Calculated Members" on page 207

Add a Detail Record

In addition to entering data into existing rows, you can add new detail records. For example, if the forms are being used for a budgeting process, each new detail record might represent a planned (but not yet hired) employee.

To add a row to a supplemental schedule, right-click a row heading and select **New Detail**. In the New Detail window, respond to the following prompts:

Code, **Name**, and **Description**

Enter a code, name, and description for the new detail record. The name appears in the row heading for the new record.

If you type a code that already exists, a subscript is added when you click **OK**. For example, “MyDetail” might become “MyDetail[2]”.

Select the scope for this detail

This prompt appears only if the form set designer did not already set the scope for all detail records. Select one of the following:

- **Make it available to other form sets.**

  The detail record is available to any forms that have the same detail dimension.

- **Limit its availability to this form set.**

  The detail record is available to any forms in this form set.

Prompts for measures

Depending on form set design, you might be prompted to enter or select initial values for some of the custom measures. These fields are optional.

Some custom measures are subject to validity checks. For example, a numeric value or a date might need to fit within a specific range. If the value that you enter does not pass a validity check, an error symbol ⊗ is displayed above the prompt. An error message is displayed in a tooltip for the error symbol.

Note: The **Reset group defaults** link resets all prompts to their default values.

The following functions are also available for working with detail records:

- To modify the name and description of a detail record, right-click its row heading and select **Edit Detail**. (You cannot edit the member code.)
- To delete a detail record, right-click its row heading and select **Delete Detail**.

Save the Supplemental Data

After you enter data into a supplemental schedule, click **Save All Supplemental Data**. The option affects all supplemental schedules in the form.

In a bottom-up form, **Save All Supplemental Data** saves the totals for numeric measures to the related crossing in the source data-entry table. (Custom measures are not saved to the data-entry table.)

For a top-down form, no data is saved to the data-entry table. However, for both bottom-up and top-down forms, the supplemental schedule data is saved to the supplemental data provider’s database. This data is available when you select **Contributing Data** for a cell in the data-entry table.

Adjusting Values for a Range of Cells

Overview of the Adjust Values Window

Use the Adjust Values window to change the values in selected cells:

1. Select a range of cells.

   Note: The range that you select cannot contain a read-only cell, a protected cell, or a parent cell.

2. Right-click and select **Adjust Values**.
3. In the Adjust Values window, select the type of adjustment (multiplier, fixed value, or proportional value) and enter an adjustment amount.

The **Total selected value** field displays the sum of the values in the selected cells. The **Total adjusted value** field gives a preview of the sum of those values after adjustment.

Adjust Values by Multiplier

To multiply the value in each selected cell by a specified number:

1. In the Adjust Values window, select **By multiplier**.
2. Enter the multiplier in the adjacent field. It can be positive or negative.

   The value of each cell is multiplied by the value that you enter.

Adjust Values by a Fixed Amount

To change the value in each selected cell by a fixed amount:

1. In the Adjust Values window, select **By value**.
2. Enter the amount in the adjacent field.

   The amount can be positive or negative. This value is added to each selected cell.

Adjust Values by a Proportional Amount

To allocate an amount to the selected cells in proportion to their original values:

1. In the Adjust Values window, select **By value**.
2. Enter the total amount to allocate in the adjacent field.
3. Select the **Modify each cell proportionally** check box.

Here are some examples of proportional adjustment, each example affecting two cells:

Original Values	Adjustment	Resulting Values
5 and 10	3	6 and 12

Original Values	Adjustment	Resulting Values
5 and 10	-3	4 and 8
(5) and (10)	3	(4) and (8)
(5) and (10)	-3	(6) and (12)

Note: Proportional adjustment is not possible if the selected range of cells contains both positive and negative values. In these cases, the **Modify each cell proportionally** check box is not available.

Spreading Values across Cells

Overview of the Spread Window

Use the Spread window to spread values over a selected range of cells at the same level of the hierarchy. The spread can be horizontal or vertical.

In a horizontal spread, the values are spread from left to right. The source cells are in the leftmost column of the range.

In a vertical spread, the values are spread from top to bottom. The source cells are in the topmost row of the range.

General Tab

Select a Spread Pattern

Use the drop-down list for the **Pattern** field to select a spread pattern.

The following spread patterns are always available:

Even
The value in each source cell is spread evenly over the associated set of target cells. For example, if a source cell has four associated target cells, then each target cell receives 25% of the value in the source cell.

Enter weights
The value in each source cell is spread over the associated set of target cells in a way that you specify in the **Weights** section.

predefined patterns
Predefined patterns: **4,4,5**; **4,5,4**; **5,4,4**.

These patterns are available only if you are spreading across time periods.

Enter Weights

If you select **Enter weights** in the **Pattern** field, select one of the following in the **Weights** section:

- **Relative values**. Enter a comma-separated list of numeric weights. Each relative weight represents a percentage of the whole (100%), and target cells receive that percentage of the amount.
- **Percentages**. Enter a comma-separated list of numeric percentages in the field below the radio buttons.

  This pattern is similar to the relative weight pattern. Instead of weights, you assign a percentage of the amount to target rows or columns. Percentages must total 100%.
- **Cell references** . Click the Select Cells button to select a range of cells.

  This pattern is similar to the relative weight pattern. In this case, the weights come from a range of cells that you select from a single row or column.

In each case, if the pattern that you specify is shorter than the range of target cells, the pattern is repeated.

Spread across Time

If you spread across time periods, then the following predefined spread patterns are also available:

- **4,4,5**
- **4,5,4**
- **5,4,4**

These predefined patterns are meaningful only if you are spreading over months and you are using the accounting convention that each month consists of either 4 or 5 whole weeks.

As with relative weights, if the pattern that you specify is shorter than the range of target cells, the pattern is repeated.

Change the Source or Target Selection

To modify the source or target selection, click the Select Cells button beside the **Source** or **Target** box.

Advanced Tab

On the **Advanced** tab, you can specify the following spread options:

Exclude virtual children

If this check box is selected, then a target cell that includes a virtual child member in any of its dimensions does not participate in the spread operation.

Note: This check box is disabled if a source cell includes a virtual child member in any of its dimensions.

Specify how to handle existing values

If any target cell that is not a source cell contains a preexisting nonzero value, then you must specify how to handle existing values. To do this, select this check box and one of the radio buttons below it.

- **Ignore existing values**: Overwrite existing values in the target cells.

  This option is not available in top-down forms.
- **Keep existing values**: If a target cell has an existing value, it does not participate in the spread operation. The source amount is spread over the remaining target cells.

- **Add existing values to spread result (but not to source amount)**: Instead of overwriting the target cell, the spread results are added to the existing value of the target cell.
- **Add existing values to source amount (but not to spread result)**: Any existing values in the target cells are added to the source amount. Then the spread results are written to the target cells (overwriting their existing values).

For example, suppose that a source value is spread evenly over three target cells, with values of 6, 0, 9. The first target cell (with a value of 6) is the source cell. The results are as follows:

Ignore existing values

```
2, 2, 2
```

Keep existing values

```
3, 3, 9
```

Preexisting nonzero values in target cells that are not source cells are left intact, and the entire source amount is spread over the other target cells.

Add existing values to spread result (but not to source amount)

```
2, 2, 11
```

Preexisting nonzero values in target cells that are not source cells are used in the final step of the computation. First, a set of target values is computed from the value in the source cell. Then, for any cell that had a preexisting nonzero value, the preexisting value is added to the value that was computed from the source value.

In the example, after the spread operation the first target cell contains 2, the second target cell contains 2, and the third target cell contains 2 + 9 = 11.

Add existing values to source amount (but not to spread result)

```
5, 5, 5
```

Preexisting nonzero values in target cells that are not source cells are used in the first step of the computation. First, all nonzero values in target cells that are not source cells are added to the source value. Then, the resulting sum is spread over the target cells.

In the example, after the spread operation each of the three target cells contains (6+9)/3 = 5.

Adding Cell Protection Directly to a Form

In a data-entry table, protected cells are displayed in gray 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 20
Designing Data Entry Forms

Overview of Designing Data Entry Forms

Designing the Form Template

When you create a form set using the New Form Set wizard in SAS Financial Management Studio, you are given the option of opening the form template in Microsoft Excel. After you design the form template, select **Save Template**.

Note: You must have an appropriate administrative role in order to create templates for data entry forms.

Most form templates contain at least one data-entry table, which enables users to enter data into the forms that are based on the template. It is possible to publish a financial form set without a data-entry table, if you want to use the workflow to circulate data for approval or informational purposes. An operational form set requires at least one data-entry table. In general, a data-entry table should include the target hierarchy of the form set.

If you are designing a template for an operational form set, only data-entry tables are permitted.

If you are designing a template for a financial form set, then you can also include read-only tables, CDA tables, and supplemental schedules. (CDA tables cannot be viewed in forms that are being edited on the Web.)

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.

Validating the Form Template

Some of the features that you can add to a form template are supported by the Excel add-in but not by Web data entry. If you are designing a template for forms only for Web data entry, do not include any of these Excel-only features. There are two cases:

- **Financial form sets.** A financial form set can be designed to permit Web data entry, Excel data entry, or both. The **Save Template** option does not automatically perform validation for Web data entry. If the form set permits Web data entry, select **Validate Web Form** before you select **Save Template**.
- **Operational form sets.** The forms of an operational form set are available only for Web data entry. If you are designing a template for an operational form set, Excel-only options are not available, and the **Save Template** option automatically checks that the template is valid for Web data entry.

Web data entry does not support the following:

- CDA tables
- CDA functions
- references to cells that are not in a SAS Financial Management table
- cell formatting
- use of a member property as a grouper
- supplemental schedules that use the SAS Human Capital Management data provider

Web data entry supports the following Microsoft Excel functions but no others:

- AVG
- DATE
- DATEVALUE
- DAY
- IF
- MAX
- MIN
- MONTH

- SUM
- TODAY
- WEEKDAY
- YEAR

Publishing the Form Set

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**.

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 SAS Financial Management and displays the form template that is associated with the selected form set. Again, select **Save Template** to save your work. You do not need to republish the form set.

See Also

"Validate Web Form" on page 237

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 receives 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.

Keep the following restrictions in mind:

- If a form template includes more than one table, you must stack the tables vertically. Do not place two tables side by side.
- If forms are being designed for Web data entry, do not put any extraneous content (for example, instructions or images) on the same rows as a data-entry table. When the form is opened on the Web, the extraneous content is not displayed.

See Also

“Coordinating Slicers between Tables” on page 210

Functional Currency

Data that is entered in a data-entry table is stored using the functional currency of the Organization member for the crossing. This is true regardless of the currency that is displayed or used for data entry.

The functional currency is defined in SAS Financial Management Studio, on the Organization Details page of the organization's Properties window.

Frequency Members for Entering Data

Data entry, the **Spread** option, and the **Adjust Values** option are available only with certain frequencies and under certain conditions:

- **Data entry:** available only for Period Activity (PA), Period To Date (PTD), and Year To Date (YTD) frequencies.
- **Spread and Adjust Values options:** available only for PA, PTD, and YTD frequencies. The following additional restrictions apply:
  - **Spread** and **Adjust Values** are not available across Frequency or Currency dimensions.
  - For flow accounts and the YTD frequency, **Spread** and **Adjust Values** are not available across multiple Time dimension members.
  - For balance accounts and the PTD or YTD frequency, **Spread** and **Adjust Values** are not available across multiple Time dimension members.

To design a data-entry table that receives numeric values for only one frequency, select one of the following approaches:

- Include Frequency in the table layout as a slicer. 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 208

Enabling Users to Enter Data at Parents

Note: These options apply 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.

In the table properties of a form template, an administrator can enable or disable options for entering data directly into a roll-up (parent) cell of a data-entry table.

There are three options (disabled by default):

- **Allow data to be entered for parent members other than time**

  The change in value is added to the parent's .VC member.

- **Allocate from Parent members other than Time using predefined weights** (automatic allocation)

  A value that is entered in a parent cell is automatically distributed among eligible leaf members, based on weights that are defined in the Allocation Weights window. (See "Designing Automatic Allocation for Data-Entry Tables" on page 191.)

- **Allow data to be entered for Time Parent members**

  Users can enter a value in a non-leaf member of the Time dimension. The distribution is defined in the table properties.

For more information, see the online Help for the table properties.

Designing Automatic Allocation for Data-Entry Tables

Note: The automatic allocation option applies only to financial forms in a bottom-up workflow.

When auto allocation is enabled, form users can directly enter values into the cell for parent members in dimensions other than time and have that value allocated based on pre-defined weights.

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.

For automatic allocation to take place, the crossing must contain a roll-up point in at least one dimension other than Time. (If the crossing 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, a leaf descendant must meet the following requirements:

- It must be writable.
- 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 and filter member combinations 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.

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.

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. (CDA tables are not permitted in forms on the Web.) You can also include a supplemental schedule that is associated with a data-entry table.

In addition to SAS Financial Management tables, you can include Microsoft Excel structures and use the capabilities of Microsoft Excel. However, a cell in a SAS Financial Management table cannot contain a Microsoft Excel formula.

Creating a Supplemental Schedule

Overview

With a supplemental data provider, you can add one or more supplemental schedules to a template for a financial form set.

A supplemental schedule is associated with a data-entry table. It contains measures that correspond to members of the data-entry table, as well as custom measures that were designed for use in a supplemental schedule.

Supplemental schedules can do the following:

- display more detailed data than is available in the SAS Financial Management data mart
- receive values that are entered by users
- roll up totals to the associated data-entry table

Via the **Contributing Data Records** option, users can view the data that was entered in a supplemental schedule.

SAS Financial Management includes a data provider. If your site also has SAS Human Capital Management installed, its data provider is available. In addition, it is possible to create a custom data provider; contact your SAS consultant for more information.

The SAS Financial Management Data Provider

The SAS Financial Management data provider is available by default. With this data provider, a supplemental schedule can include the following measures:

- **Data-entry measures:** any members of the supplemental schedule's measure dimension that are included in the data-entry table.
- **Custom measures:** any custom properties that are associated with any member of the hierarchy for that dimension (based on the model for the data-entry table). The member does not have to be included in the data-entry table, and it does not have to be numeric. For example, it might be used to record the justification for an entry.

  Custom properties can include validation. For example, a custom property with the character data type might be limited to specific character strings. A custom property with the numeric data type might be limited to a specific range.

The as-of date for the hierarchy determines eligibility for inclusion in the supplemental schedule.

Changes in the data-entry table are reflected in the supplemental schedule, with this exception: If you change the model for the data-entry table in the table properties, you must click **Refresh All** or save the template to synchronize the change with the data provider.

Changes outside the form template require a **Refresh All**. For example, if you remove members or custom properties from the hierarchy that is used for the measures, those changes are reflected in the supplemental schedule after a **Refresh All**.

The SAS Human Capital Management Data Provider

If SAS Human Capital Management is also installed at your site, then you can insert a SAS Human Capital Management supplemental schedule into a template for a financial form set. Be aware of the following restrictions:

- Supplemental schedules that use the SAS Human Capital Management data provider can be used with the SAS Financial Management Add-In for Microsoft Excel. However, they cannot be used in Web data entry.
- Any supplemental schedules that were created in a previous version of SAS Financial Management, using the SAS Human Capital Management data provider, cannot be used in SAS Financial Management 5.3. The supplemental schedules would need to be re-created.
- A form set cannot be copied or imported if the form template contains a supplemental schedule that uses the SAS Human Capital Management data provider.

For information about this data provider, including instructions about defining measures for supplemental schedules, see the *SAS Human Capital Management Administrator's Guide*.

Create a Supplemental Schedule

You can add one or more supplemental schedules to a financial form template. Supplemental schedules are not available for operational forms.

To insert a supplemental schedule:

1. Create the data-entry table that the supplemental schedule will be associated with.

   The data-entry table must include all the crossings that you want the supplemental schedule to roll up to.

2. On the same worksheet or a different worksheet, 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.

   Among other things, you select the following:

   - a detail dimension. This dimension appears in the rows of the supplemental schedule. New detail records become members of this dimension.
   - a dimension to be used for measures. Selected members of this dimension appear in the columns of the supplemental schedule.
   - members of the Account dimension to be affected, if that dimension is not used in the rows or columns.

   Any other dimensions in the data-entry table become slicers in the supplemental schedule.

When you finish the wizard, the supplemental schedule appears.

TIP Some slicers are shared with the data-entry table, so that a selection in the data-entry table also applies to the supplemental schedule. By default, those shared slicers are not displayed as part of the supplemental schedule. To display the shared slicers, select **Table Properties** and use the **Hide Slicers** drop-down menu on the **General** tab.

Saving Supplemental Schedule Data

Only the totals for numeric measures that are also in the data-entry table are saved to the data-entry table. In the data-entry table, the affected members are displayed in green cells. In the supplemental schedule, the totals and averages for these measures are also displayed in green.

- **For bottom-up form sets:** If the measure for a detail member has changed in value, its total is saved to the related crossing in the source data-entry table. (Custom measures are not saved to the data-entry table.)

  This action also refreshes the data-entry table.

  Note: If writing to parent members is enabled, take additional care in designing supplemental schedules. If you include both a leaf member and its parent as measures, be aware that the values from one member can overwrite the other.

  In addition, the data in the supplemental schedules is saved to a separate database that is maintained by the supplemental data provider.

- **For top-down form sets:**No totals are saved to the data-entry table. However, the data in the supplemental schedules is saved to the supplemental data provider's database.

At run time, a user enters data in the supplemental schedule and clicks **Save all supplemental data**. At that point, the totals (in green) from all the supplemental schedules in the form or form template are saved to their associated data-entry table. In addition, the data in the supplemental schedules is saved (outside the financial model). When a user selects **Contributing Data** for a cell in a data-entry table or read-only table, the data from related detail records in the supplemental schedule is also available.

Some points to be aware of:

- A supplemental schedule does not honor system filters or filter member combinations that are applied to its data-entry table.
- Currency conversion takes place in the data-entry table. However, there is no currency conversion in the supplemental schedule itself. Make sure that users are aware of the currency to use for entering data in a supplemental schedule.
- Data that is saved outside the financial model might not be qualified by all the members that qualify the supplemental schedule. What is saved depends on the data provider. The SAS Financial Management data provider does not save Currency or Frequency members. (It does save the Source and Trader members from the data-entry table.)

Deleting a Supplemental Schedule

To remove a supplemental schedule from a form set, delete its rows from the form template and save the template. Its data is not deleted from the SAS Financial Management data mart or data provider, and its detail members are not deleted.

If you delete data from a form set, data from any of its supplemental schedules is deleted from the SAS Financial Management data mart. Detail members are deleted if their scope is confined to the form set. However, supplemental schedule data is not deleted from the SAS Financial Management data provider.

If you delete a cycle, data from any of its supplemental schedules is deleted from the SAS Financial Management data mart and data provider. However, detail members from those supplemental schedules are not deleted.

Note: For the effects on other data providers, see their documentation.

See Also

- "Using Excel-Based Calculated Members" on page 207
- "Entering Data into a Supplemental Schedule" on page 180
- "Data from Supplemental Schedules" on page 223

Designing Cell Protection for Data-Entry Tables

About Cell Protection

You can protect cell crossings in a data-entry table by creating one or more rules that apply to the dimensions in the table.

Cells are protected against the following actions:

- manual data entry
- spread
- the effects of writing to parent members

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 gray. 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.

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 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 gray.
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 also 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.

Note: The measures in a supplemental schedule inherit cell protection from the associated data-entry table.

See Also

"Adding Cell Protection Directly to a Form" on page 185

Chapter 21
Working with Tables

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, for use in data-entry forms.

  This option is active only when you are editing a form template.

- **Insert ⇨ Read-only Table** creates a read-only table, which displays numeric values that are computed from data in the SAS Financial Management database.

  This option is active when you are editing a financial report or a financial form using the add-in.

- **Insert ⇨ CDA Table** creates a CDA table. This table displays numeric values that are computed from data in the SAS Financial Management database.

  This option is active when you are editing a financial report or a financial form using the add-in.

- **Insert ⇨ Supplemental Schedule** creates a supplemental schedule, which can be used to display and collect additional data in support of a data-entry table.

  This option is active only when you are editing a form template.

Keep the following restrictions in mind:

- If you create more than one SAS Financial Management table in the same worksheet, place the tables one below the other, not side by side.
- Do not put any other information in a row that is used by a SAS Financial Management table, including the header rows (even outside the table columns). For example, in a form template, place any data-entry instructions either above or below a data-entry table, not beside the table. Otherwise, the instructions are omitted from the published forms.

See Also

- "Overview of Designing Data Entry Forms" on page 187
- "Read-Only Tables" on page 169
- "CDA Tables" on page 170
- "Entering Data into a Supplemental Schedule" on page 180

Deleting Tables

If you use the SAS Financial Management Add-In for Microsoft Excel to design a report or a form template, 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 **Members** ⇨ **Pivot**. The Table Layout page and the Pivot window work in the same way, as described here.

See Also

"Pivoting a Table" on page 206

Defining a Table Layout

The **Available** section lists all the dimensions that have hierarchies in the selected model.

The three sections on the right—**Columns**, **Rows**, and **Slicers**—represent the table's column headings, row headings, and slicers. You must select at least one row heading and one column heading. Additional headings and slicers are optional.

The order in which you define the column headings, row headings, and slicers determines the order in which they appear in the table. The first-listed dimension or member property is on the left (or top). To change this order, use the arrows or drag a dimension or member property to a new position.

If member properties or custom properties are defined for a dimension, you can include them in the table as well. See "Using Member Properties in a Table" on page 202.

Row and Column Headings

If you define a single row or column heading, then the columns or rows have simple headings that identify those dimension members. If you define multiple row or column headings, then the rows or columns have nested headings.

Slicers

The dimensions 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. Typically, users can select different members or property values for a slicer. However, a slicer can also be defined with a fixed selection.

The table displays data that is associated with the currently selected slicer values.

Default Read Members

If a dimension in the model is not used in the table, then its table default read member is used. You can view the table default read members for a table in the table properties.

Using Member Properties in a Table

Overview

Member properties are associated with one or more members of a dimension. The Account, Organization, and Time dimension types have default member properties. In addition, custom properties might have been defined for members of those dimensions or other dimensions.

The values of member properties can be used to label, group, or filter the members in the table. The member property values are also available via the CDAProperty function. In a calculated-member formula, they are available via the fmProperty function.

Role of Member Properties in a Table

You add member properties to a table the same way you add dimensions to a table, in the Create Table wizard or the **Pivot** option. The table must already include the dimension to which the property applies.

A member property's role in a table depends on the role of the corresponding dimension, as follows:

Dimension Role	Permitted Roles for Member Property
Row heading	Row heading or Slicer
Column heading	Column heading or Slicer
Slicer	Slicer

If the member property is a slicer, it restricts the table to a subset of the members of the corresponding dimension. If both a member property and its dimension are slicers, then the available selections for the dimension are restricted by the member property selection.

If both the member property and its dimension are row or column headings, the member property serves as a label or grouper, depending on their relative positions:

- If the member property is nested inside the dimension, then the member property acts as a label.

  For example, assume that the Account dimension and the Account Type property are both row headings. The Account Type property is nested inside the Account dimension. Each account is labeled by the adjacent account type.

- If the dimension is nested inside the member property, then the dimension members are grouped by the values of the member property. The members are displayed without any hierarchical structure.

  For example, assume that the Account dimension and the Account Type property are both row headings. The Account dimension is nested inside the Account Type property. The accounts are then grouped by account type, rather than being listed under their parent accounts.

Member Property Filtering

Member properties have another important use in addition to their roles in slicers, rows, and columns. Via member property filtering, they can restrict the dimension members that are displayed in the table. You apply member property filtering to a dimension, via the table wizard or the Show Members window. Only the dimension members that match the filter can be displayed.

Notice that you cannot select member property values using member selection rules, the way that you can select dimension members. Instead, the available values for the member property are the same property values that you applied to the dimension. (If you did not enable member property filtering, then all the property values are available.) For details, see the online Help for the Show Members window.

CDA Tables

Contents of a CDA Table

Overview

A CDA table includes a table header, row and column headings, and data cells.

CDA Table Header

The header for a CDA table contains information about the table, including the model, the dimensions used for rows, columns, and slicers, and the currently selected value for each slicer.

Table 21.1 Contents of a CDA Table Header

Field	Description
Scale values by	Contains a scale factor for the numeric values displayed in the table. The actual computed values are divided by this scale factor to yield the displayed values. Computed percentages are not scaled. To display the actual computed values, leave the default scale factor of 1. To change the scale factor, enter another number in this field. For example, to display values in thousands, enter 1000.
Debits as positive	Determines how debit balances are displayed: • `1`: Display positive debit balances as positive numbers, no matter what the underlying model specifies. • `-1`: Display positive debit balances as negative numbers, no matter what the underlying model specifies.
Credits as negative	Determines how credit balances are displayed: • `1`: Display positive credit balances as negative numbers, no matter what the underlying model specifies. • `-1`: Display positive credit balances as positive numbers, no matter what the underlying model specifies.
slicers (d/m)	The dimension codes (**d**) and member codes (**m**) for the table slicers.
columns (d)	The dimension codes (**d**) for the table's column headings.
rows (d)	The dimension codes (**d)** for the table's row headings.

Row and Column Headings

Row and column headings are repeated in the table—first with the dimension member code (which is needed by the CDA functions), and again with the dimension member name or description. You can hide the cells containing the member codes, but do not delete them. Their values are used in CDA expressions that populate the data cells.

Slicers

In the CDA table wizard, when you define a slicer, you have the option of selecting **Show Slicer Members List**. If you select this check box, the resulting CDA table contains a combo box that allows the user to select a different slicer member. If you do not select this check box (the default), only the first member that you select in the wizard appears in the table.

Data Cells

The data in a cell comes from a CDA function that references the model code and one or more sets of dimension codes (from the table header) and corresponding member codes (from the column and row headings). If a cell is not readable (for example, if it is protected by member-level security), an error message is displayed.

Formatting a CDA Table

After you create a CDA table, you can use any of the Microsoft Excel capabilities to format and rearrange the table, including moving cells, adding rows or columns, or hiding rows or columns. Be careful not to delete table cells that are referenced by CDA functions.

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 by selecting one of its members. Then use the **Show Members** option to define a different subset of the members of that dimension 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 206

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

Note: In a data-entry form, you can drag a dimension from one position to another (for example, from rows to columns), but you cannot add or delete dimensions, and the arrows in the Pivot window are disabled.

Using the Pivot Option

To open the Pivot window: from the **Members** menu, select **Pivot**.

Drag-and-Drop Pivoting

You can do many types of pivoting by dragging a table cell onto a target cell, as follows:

1. Select the dimension that you want to drag by clicking a 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 a mouse button, move the cursor to the border of the selected cell.

   The directional arrow symbol appears. Make sure that this symbol is visible before you continue.
3. Press the left mouse button and drag the selected cell to a target cell.
4. When the selected cell coincides with the target cell, release the mouse button.

   A popup message appears, asking if you want to replace the contents of the destination cells.
5. 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 heading cell	row dimension immediately to the left of the row dimension that contains the target cell
data cell adjacent to a row heading cell	row dimension adjacent to the data cells
column heading cell	column dimension immediately above the column dimension that contains the target cell
data cell adjacent to a column heading cell	column dimension adjacent to the data cells
cell outside the table	none (dimension is removed from the table)

These drag-and-drop operations have the following limitations:

- The target table cell must occupy only a single Excel spreadsheet cell. A large row heading cell or column heading 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 drag a row or column dimension into a slicer dimension if the table does not already have a slicer dimension.
- You cannot drag a dimension into the last slicer position, after all existing slicers.

Removing Dimensions with the Delete Key

You can remove a dimension from a table in the following way:

1. Select a member of a row or column dimension, or the dimension label of a slicer dimension.
2. Press the **Delete** key.

Note: You cannot remove the last row dimension or the last column dimension of a table.

Using Excel-Based Calculated Members

About Calculated Members

You can add calculated members to a read-only table, a data-entry table, or a supplemental schedule. Each calculated member is associated with a formula that is used to calculate its values. The calculation is done after any server-side calculations.

Calculated-member formulas use the syntax of Microsoft Excel formulas. In addition to Excel functions and expressions, the formula can include the calculated-member functions that are provided by the SAS Financial Management Add-In for Microsoft Excel.

To add, delete, or edit a calculated member, click anywhere in the table and select **Members ⇨ Calculated Members**.

To view the formula for a calculated member, select its heading in the table or supplemental schedule. The formula is displayed in a tooltip.

For more information, see the online Help for the calculated-member wizard and the dictionary of calculated-member functions. In addition, the *SAS Financial Management: Formula Guide* contains detailed information and examples.

References to Members That Are Not on the Table

Occasionally the formula for a calculated member refers to a member that is not on the table. The referenced member might be excluded because of member selection rules or property selection rules, or it might be hidden by a filter. The results are as follows:

- In a read-only table or a data-entry table, the reference becomes a CDA function and continues to work correctly in Excel.
- On the Web, the result is displayed as an empty cell.

  Note: If the calculated member's position refers to an excluded member, the calculated member is not displayed at all.
- In a supplemental schedule, the formula is invalid. A calculated-member formula works correctly only if every member or measure that it refers to is navigable in the supplemental schedule.

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.

Formatting Tables

Modifying Formats for SAS Financial Management Tables

You can modify the formatting in a SAS Financial Management table in these different ways:

- **Format Cells** applies ad hoc formatting to a selected cell or range of data cells in a table.
- **Format Members** assigns formatting styles to members. The style applies to all the data cells in the selected table with crossings that contain those members.
- **Cell Styles** applies formatting to table components including headings. This option can apply to a table, or you can create a template that applies to new reports. Using style merging, you can also apply style changes to existing reports or form templates.

You can affect the table display in other ways such as hiding slicers, selecting the display methods for zero and invalid values, and selecting whether to wrap labels. For details, see the online Help for table properties.

Format Cells

Use the Microsoft **Format Cells** option to apply ad hoc formatting to a selected cell or range of cells.

Apply this option only to data cells, not to row or column headings. Row and column headings revert to their original formatting when the display is refreshed.

To include these changes when you save the file, you must take the following additional steps:

1. Select a range of cells that includes the formatted cells.
2. Select **Attach Style Changes**.

Format Members

Use the Format Members window to assign cell formatting styles to members. These styles are defined in the Microsoft Excel Cell Styles window. A style that is assigned to a member is applied to all the data cells whose crossings contain that member. This style affects only the selected table.

To use this option, right-click a cell in the table and select **Members ⇨ Format Members**.

Modify Cell Styles

Overview

Styles for headings and cells are defined in the Cell Styles window of Microsoft Excel. You can create new styles and apply these them to the table's components, or you can modify existing styles. For example, you might want to change the font size or background color for headings, or you might want to change the color of protected cells.

On the **Styles** tab of the Table Properties window, styles can applied to the functional components of a table such as row headings, column headings, and data cells. Notice that separate styles are assigned to row headings and drillable row headings, and to column headings and drillable column headings.

The cell styles are also available in the Format Members window.

Save Styles in a Template

You can save these cell styles in a template and apply them to new or existing reports and form templates. Follow these steps:

1. Open a SAS Financial Management report.
2. Open the cell styles properties (in Excel 2010, select **Home (Styles)** ⇨ **Cell Styles**).
3. Modify the SAS Financial Management cell styles. For example, you might modify font colors or numeric formats. You can see the effect of any changes in the report. Do not create new styles. Instead, modify the existing styles.
4. With the report still open, open a new Excel workbook.
5. In the new workbook, select **Home (Styles)** ⇨ **Cell Styles**.
6. At the bottom of the **Cell Styles** window, select **Merge styles**, and select the report with the modified styles.

   The SAS Financial Management styles are copied to your new (blank) workbook.
7. Save the workbook as a template.

When you create new reports, begin with the template, which will have your modified cell styles.

Apply Styles to an Existing File

To apply the cell styles to an existing report or to a form template:

1. Open the Excel template with the modified cell styles.
2. Open the report or form template.
3. Merge the cell styles from the Excel template with the styles in your report or form template.

   When you are asked if you want to merge styles with the same name, respond **Yes**.

Resolving Conflicts between Format Specifications

If there are conflicts in data cells between formatting that you set in these different ways, then the conflicts are resolved by the following precedence ordering:

1. formatting that is set with the Microsoft **Format Cells** option
2. formatting that is set with **Format Members**
3. formatting that is set with **Table Properties**

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 the same hierarchy plays the role of a slicer 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** section of the **Slicers** tab to define the connection between tables. For details, see the online Help for the Table Properties window.

Note: This option is available for read-only tables and for data-entry tables in form templates.

Adding Comments to a Cell

About Cell Comments

A cell comment consists of text that you attach to a single cell in a read-only table or a data-entry table. A cell comment is associated with the cell crossing.

Note: If comments are not enabled for a report or a form, you can view existing comments but you cannot add new ones.

Add a Comment

To add a comment to a cell:

1. Select a data cell.

   The cell does not have to be writable, but it must be readable. You cannot add cell comments to supplemental schedules.
2. Click Cell Comments in the **Tools** group on the SAS Financial Management tab.
3. Click **Add Comment**.

4. If the comment is only for your own use, select **This comment is private**.

   Otherwise, the comment might be available in other forms or reports.

   Note: You cannot go back later and change the comment's privacy setting. For example, to make a public comment private, you must delete the comment and re-create it. All comments, both public and private, are stored in the SAS Financial Management Data Mart.

View Comments

To view cell comments:

1. Select a cell with a red flag in its upper right corner.

   Any comments that are associated with that crossing appear in the Cell Comments window. You can leave this window open (or minimized). Its contents change as your cell selections change.

2. To view comments that are associated with subordinate members, select **Show contributing comments indicator**.

   In the table, select a cell with a blue flag in its upper left corner. All comments that are associated with any of the crossing's subordinate members appear in the Cell Comments window.

From the drop-down menu at the right of the comment heading, you can reply to a comment, edit the most recent comment or a reply, sort the replies, or delete comments or replies.

Note: If a reply from someone else is attached to a comment, you cannot delete it. Unless you are an administrator, you can delete only comments and replies that you made.

View the Crossing for a Comment

To view the crossing for a comment:

1. Click the Slice to contributing crossing button .

   The display changes to reflect the crossing that is associated with this comment.

   Note: The Frequency and Currency dimensions are not included.

2. To display the previous crossing, click **Return to original view**.

   However, be aware that if you have modified the display (for example, by rearranging columns, rows, or slicers), clicking this button might not take you to the original view.

Copying a Read-Only Table to a CDA Table

To copy 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 left-most 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. If you want it to begin in cell B3 of Sheet2, type **`Sheet2!B3`**.

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.

Generating a Forecast

Overview

In a form template, an administrator can generate a forecast that is based on the historical data in the model for the data-entry table. The Forecast wizard uses SAS High-Performance Forecasting to generate the forecast data.

The forecasting software uses sophisticated automatic model selection techniques to choose the best-fitting model for the data. After a successful forecast is complete, this model is stored on the server and can be reused for more efficient forecasting. The forecast parameters are saved with the form template.

If forecasting is enabled for data entry, a user who is editing a form can also generate a forecast. The user's forecast can reuse the model or request to have a new model generated. However, a forecast model that is generated from a form is not saved.

For details, see the online Help for the Forecast wizard.

Note: Forecasting is available only in financial forms without supplemental schedules.

Execution Time

The time it takes a forecast to execute depends primarily on these factors:

- the number of by-variable members that are selected
- the number of historical and forecast time periods
- whether the forecast model is reused

Because execution time can be lengthy, a forecast runs asynchronously and sends a notification when the forecast completes.

If you have not closed the form or template, a pop-up message appears when the forecast completes. To view the results, refresh the display.

Forecast Results

The forecast results are written to the analysis member that was selected for the forecast. If you selected **Use confidence values**, then two additional members, for the upper bound and lower bound of the confidence interval, also hold forecast results.

Note: Writing the forecast results, like other data entry operations, causes driver formulas to be executed.

After generating a forecast, you can use the **Create Chart** option to create a Microsoft Excel PivotChart to display the results in Microsoft Excel. See "Creating a Chart from a Table" on page 214. You can also use Excel's copy and paste functionality to copy forecast results to another part of the table (for example, you might copy the forecast data or the lower bound data to the Budget member).

The object type for forecast-generated data is **`forecast`** and can be seen if you view contributing data for a crossing.

In SAS Financial Management Studio, the cycle properties contain a history record of the forecast. If the forecast failed, the record contains a link to a detailed report.

Creating a Chart from a Table

With the **Create Chart** option, you can create an ad hoc Microsoft Excel PivotChart from a SAS Financial Management table. The chart includes the members that are currently displayed in the table.

To create a chart:

1. Click one of the table cells.
2. Select **Edit** ⇨ **Create Chart** from the pop-up menu.
3. Move the chart to an appropriate location in the worksheet.

As you make changes in the table (such as changing a value in a data-entry table or selecting a different slicer member), those changes are reflected in the chart.

For additional customizations, use the **PivotChart Tools** and the **PivotTable Field List** that are available when you click anywhere in the chart. For more information about working with a PivotChart, see the online Help for Microsoft Excel.

Note: Charts are not available in Web data entry.

Copying a CDA Table to a Read-Only Table

To copy 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**.

   A corresponding read-only table is created in a new worksheet. 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.

Chapter 22
Dictionary of Menu Options

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 writable cells are selected in a data-entry table.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Adjust Values**
- Pop-up menu: **Edit** ⇨ **Adjust Values**

Allocate

The **Allocate** option launches the Allocate wizard. This wizard 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.

This option is active only if a writable cell is selected in a data-entry table for a form set that has a top-down workflow.

For detailed information about the Allocate wizard, see the online Help for the individual wizard pages.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Allocate**
- Pop-up menu: **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:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Allocation Weights**
- Pop-up menu: **Edit** ⇨ **Allocation Weights**

Attach Style Changes

The **Attach Style Changes** option attaches format settings to a SAS Financial Management table. The format settings are specified using the Microsoft Excel **Format Cells** option. The settings for the currently selected cells are attached to the table that contains those cells.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Attach Style Changes**
- Pop-up menu: **Edit** ⇨ **Attach Style Changes**

You must attach 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 Style Changes**, use **Clear** ⇨ **Style Changes** or **Clear** ⇨ **Table's Style Changes**.

Calculated Members

The **Calculated Members** option displays the Calculated Members window. This window 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:

- **SAS Financial Management (Navigate)** ⇨ **Members** ⇨ **Calculated Members**
- Pop-up menu: **Members** ⇨ **Calculated Members**

CDA Table

The **CDA Table** option launches the Create CDA Table wizard. This wizard guides you through the process of creating a CDA table. Read-only tables and CDA tables are the central components of financial reports.

For detailed information about the Create CDA Table wizard, see the online Help for the individual wizard pages.

Use the following selection path: **SAS Financial Management (Tables)** ⇨ **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:

- **SAS Financial Management (Tools)** ⇨ **Cell Information**
- Pop-up menu: **Tools** ⇨ **Cell Information**

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 the following selection path: **SAS Financial Management (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.

You cannot check out a form that is currently checked out or a form that contains a supplemental schedule.

Use the following selection path: **SAS Financial Management (File)** ⇨ **Check Out Form**

Clear Style Changes

The **Clear Style Changes** option removes attached cell formats from all cells in the selected range. It undoes the effect of **Attach Style Changes** for the selected cells.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Clear** ⇨ **Style Changes**
- Pop-up menu: **Edit** ⇨ **Clear** ⇨ **Style Changes**

Clear Table's Style Changes

The **Clear** ⇨ **Table's Style Changes** option removes attached cell formats from all cells in the selected table. It undoes the effect of **Attach Style Changes** for the entire table.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Clear** ⇨ **Table's Style Changes**
- Pop-up menu: **Edit** ⇨ **Clear** ⇨ **Table's Style Changes**

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:

- **SAS Financial Management (Navigate)** ⇨ **Navigate** ⇨ **Collapse**
- Pop-up menu: **Collapse**

Comments

The **Comments** option opens the Cell Comments window. In this window, you can attach a comment to a cell crossing.

Use the following selection path: **SAS Financial Management (Tools)** ⇨ **Comments**

Contributing Data

Overview

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.

The display includes an **Author** column (the user ID of the user who entered the value) and a **Date** column (the date and time of the entry).

If facts were copied, then the author is the user who performed the copy. If the table was copied during a migration, the **Author** column is empty and the timestamp reflects when the migration was performed.

Note: Contributing data records cannot be retrieved for Excel-based calculated members.

Use one of the following selection paths:

- **SAS Financial Management (Tools)** ⇨ **Contributing Data**
- Pop-up menu: **Tools** ⇨ **Contributing Data**

Computed Values

SAS Financial Management computes a number of values on demand:

- adjustment rules
- intercompany eliminations
- frequencies other than PA
- modeling and reporting formula types
- parent values
- currency
- Retained Earnings account types
- Cumulative Translation Adjustment (CTA) account types

Those items are not stored in the database and so are not included in the contributing data display. As a result, the items in the display might not match the total for a particular crossing.

Data from Operational Cycles

Some displays contain summarized versions of data that was loaded from an operational cycle. In those cases, click a button in the **Details** column to display the fully detailed operational data.

Data from Supplemental Schedules

If the OBJECT_TYPE column contains the value “supplemental,” the source of that data is a supplemental schedule. In the **Details** column of the contributing data display, click the ... button next to the **Total** entry to view more information about data that came from detail records in a supplemental schedule.

In the supplemental data display, the OWNER column shows the code for the member that is associated with each detail record. The display includes any non-numeric data that was part of the detail record.

Note: Viewing supplemental data is available only with the SAS Financial Management data provider.

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. The selected cell must be outside the SAS Financial Management table.

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.

For example, a worksheet might contain a read-only table or data-entry table, and a cell outside that table might contain a formula that refers to one or more cells in the table (such as **`=SUM(B2:B9)`**). The **Convert to CDA** converts the Excel references to CDA references.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Convert to CDA**
- Pop-up menu: **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:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Copy as CDA**
- Pop-up menu: **Edit** ⇨ **Copy as CDA**

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:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Copy as Table**
- Pop-up menu: **Edit** ⇨ **Copy as Table**

Create Chart

The **Create Chart** option creates an Excel PivotChart from a read-only table or a data-entry table.

Click a cell in the table and use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Create Chart**
- Pop-up menu: **Edit** ⇨ **Create Chart**

See Also

"Creating a Chart from a Table" on page 214

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 the following selection path: **SAS Financial Management (Tables)** ⇨ **Insert** ⇨ **Data-Entry Table**

Delete Detail

The **Delete Detail** option deletes the currently selected detail row from a supplemental schedule.

Use one of the following selection paths:

- **SAS Financial Management (Supplemental Data Provider)** ⇨ **Delete Detail**
- Pop-up menu: **Delete Detail**

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:

- **SAS Financial Management (Navigate)** ⇨ **Navigate** ⇨ **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:

- **SAS Financial Management (Navigate)** ⇨ **Navigate** ⇨ **Drill Up**
- Pop-up menu: **Drill Up**

Edit Detail

The **Edit Detail** option opens the Change Description window, in which you can change the name and description of the currently selected record in a supplemental schedule.

Note: You cannot change the code for a detail record.

Use one of the following selection paths:

- **SAS Financial Management (Supplemental Data Provider)** ⇨ **Edit Detail**
- Pop-up menu: **Edit Detail**

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:

- **SAS Financial Management (Navigate)** ⇨ **Navigate** ⇨ **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:

- **SAS Financial Management (Navigate)** ⇨ **Navigate** ⇨ **Expand All**
- Pop-up menu: **Expand All**

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. It is available for read-only tables and for data-entry tables in form templates.

Use one of the following selection paths:

- **SAS Financial Management (Navigate)** ⇨ **Filter** ⇨ **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:

- **SAS Financial Management (Navigate)** ⇨ **Filter** ⇨ **Filters**
- Pop-up menu: **Members** ⇨ **Filters**

Forecast

The **Forecast** option launches the Forecast wizard. The wizard enables you to generate forecast data for a data-entry table, based on its historical data. This option applies only to bottom-up financial form sets without supplemental schedules.

Use one of the following selection paths:

- **SAS Financial Management (Tools)** ⇨ **Forecast**
- Pop-up menu: **Tools** ⇨ **Forecast**

Format Members

The **Format Members** option displays the Format Members window. This window 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:

- **SAS Financial Management (Navigate)** ⇨ **Members** ⇨ **Format Members**
- Pop-up menu: **Members** ⇨ **Format Members**

This is one of several ways in which you can format table cells. See “Formatting Tables” on page 208.

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:

- **SAS Financial Management (Navigate)** ⇨ **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 the following selection path: **SAS Financial Management (File)** ⇨ **Log Off**

Log On

The **Log On** option displays the SAS Log On window, which accepts your user name and password. After you enter your user name 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 the following selection path: **SAS Financial Management (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 the following selection path: **SAS Financial Management (Tables)** ⇨ **Insert** ⇨ **Member Labels**

New Detail

The **New Detail** option displays the Add New Detail window, in which you can add one or more detail rows to the selected supplemental schedule.

For information about completing the fields in this window, see "Add a Detail Record" on page 181.

Note: The **Reset group defaults** link resets all prompts to their default values.

Use one of the following selection paths:

- **SAS Financial Management (Supplemental Data Provider)** ⇨ **New Detail**
- Pop-up menu: **New Detail**

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.
- 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:

- **SAS Financial Management (Navigate)** ⇨ **Pivot**
- Pop-up menu: **Members** ⇨ **Pivot**

Properties

See "Table Properties" on page 235.

Protect Cell

To protect selected cells in a data-entry table, use one of the following selection paths:

- **SAS Financial Management (Navigate)** ⇨ **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 gray 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 236
- "Unprotect All Cells" on page 236

Protection Rules

Edit rules to protect crossings in data-entry tables.

In a form template, click anywhere in a data-entry table and use the following selection path: **SAS Financial Management (Navigate)** ⇨ **Protection Rules**.

For a model, use the same selection path after selecting the protection rules option for a model in SAS Financial Management Studio. A data-entry table is not required.

Publish Information Map

The **Publish** ⇨ **Information Map** option opens the New Information Map window, which enables you to save a table as an information map that can be opened in SAS Web Report Studio.

Use the following selection path: **SAS Financial Management (File)** ⇨ **Publish** ⇨ **Information Map**.

Publish Report

The **Publish** ⇨ **Report** option launches the Publish Report wizard, which enables you to create different types of reports and save them in SAS folders or on your desktop.

Use the following selection path: **SAS Financial Management (File)** ⇨ **Publish** ⇨ **Report**.

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 the following selection path: **SAS Financial Management (Tables)** ⇨ **Insert** ⇨ **Read-Only Table**

Reallocate

The **Reallocate** option is available if the following conditions are true:

- either automatic allocation or writing to parent members of the Time dimension is enabled for the current data-entry table; and
- you have selected a nonzero writable parent cell in the table.

It redistributes the current value of the selected cell, according to the allocation weights that you specified. This option is useful if the parent cell value did not change, but the allocation weights changed or (for automatic allocation) some selections in the allocation weight table changed.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Reallocate**
- Pop-up menu: **Edit** ⇨ **Reallocate**

Refresh

The **Refresh** option refreshes numbers but not members. This option recomputes and redisplays all numeric values. However, it does not retrieve any changes that have been made to hierarchies since you opened the file.

If **Refresh tables in the active worksheet only** is selected in the table properties, then selecting **Refresh** applies only to the active worksheet. Otherwise, it applies to all worksheets in the workbook.

Use the following selection path: **SAS Financial Management (Navigate)** ⇨ **Refresh**

Refresh All

The **Refresh All** option refreshes numbers and members. This option recomputes and redisplays all numeric values. It also retrieves any changes that have been made to hierarchies since you opened the file.

Selecting **Refresh All** applies to all worksheets in the workbook.

Use the following selection path: **SAS Financial Management (Navigate)** ⇨ **Refresh All**

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:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Repeat Spread**
- Pop-up menu: **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 the following selection path: **SAS Financial Management (Tools)** ⇨ **Help** ⇨ **SAS Financial Management Add-In Help**

Save All Supplemental Data

The **Save All Supplemental Data** option does the following for all supplemental schedules in a form or form template:

- It rolls up numeric values from the totals row of each supplemental schedule to the corresponding crossings in the associated data-entry table.
- It saves values from each supplemental schedule to the database that is served by the data provider.

This option is active only if you are designing a form template or using a data-entry form that includes a supplemental schedule. Selecting this option has an effect only if there is unsaved supplemental data.

Use one of the following selection paths:

- **SAS Financial Management (Supplemental Data Provider) ⇨ Save All Supplemental Data**
- Pop-up menu: **Save All Supplemental Data**

See Also

“Entering Data into a Supplemental Schedule” on page 180

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 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 are editing a form in Microsoft Excel.

Use the following selection path: **SAS Financial Management (File) ⇨ Save ⇨ 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 Web data entry. If you are working on a financial form template, then **Save Template** does not make this check automatically. If the financial forms can be edited using Web data entry, then select **Validate Web Form** to check that the template is valid for Web data entry.

Use the following selection path: **SAS Financial Management (File) ⇨ Save ⇨ Save Template**

See Also

“Validate Web Form” on page 237

Show Members

The **Show Members** option displays the Show Members window. In this window, you can modify the display of a read-only table or a data-entry table. For example, you can select members for display, reorder members, and select a labeling method. Specific features depend on the table and its use (in a report, form, or form template).

Use one of the following selection paths:

- **SAS Financial Management (Navigate)** ⇨ **Members** ⇨ **Show Members**
- Pop-up menu: **Members** ⇨ **Show Members**

Spread

The **Spread** option displays the Spread window, which enables you to spread numeric values across a selected range of cells.

Many spread patterns are available. To reuse the same spread pattern for several ranges of cells, use **Spread** the first time and then continue with **Repeat Spread**.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Spread**
- Pop-up menu: **Edit** ⇨ **Spread**

Supplemental Schedule

The **Supplemental Schedule** option launches the Supplemental Schedule wizard, which guides you through the process of creating a supplemental schedule.

This option is active only if you are designing a form template.

For detailed information about the Supplemental Schedule wizard, see the online Help for the individual wizard pages.

Use the following selection path: **SAS Financial Management (Tables)** ⇨ **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:

- **SAS Financial Management (Tables)** ⇨ **Properties**
- Pop-up menu: **Edit** ⇨ **Table Properties**

Unassign

The **Unassign** option computes the sum of the values in all the writable 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 writable cell.

Use one of the following selection paths:

- **SAS Financial Management (Tables)** ⇨ **Edit** ⇨ **Unassign**
- Pop-up menu: **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 the following option: **SAS Financial Management (Navigate)** ⇨ **Unprotect All Cells**

See Also

- "Protect Cell" on page 230
- "Unprotect Cell" on page 236

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:

- **SAS Financial Management (Navigate)** ⇨ **Unprotect Cell**
- Pop-up menu: **Edit** ⇨ **Unprotect Cell**

See Also

- “Protect Cell” on page 230
- “Unprotect All Cells” on page 236

Validate Web Form

The **Validate Web Form** option checks that all the features of the open form template are supported in Web data entry. 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 a financial form template, then menu options that add Excel-only features are available. The **Save Template** option does not make Web checks. If the forms based on the template can be edited on the Web, we recommend that you run **Validate Web Form** before saving the template.
- 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 performs a validation before it saves the template. For these form templates, the **Validate Web Form** option is not necessary.

For a list of features that are not supported in Web data entry, see “Validating the Form Template” on page 188.

This option is active only if you are working on a form template.

Use the following selection path: **SAS Financial Management (File)** ⇨ **Save** ⇨ **Validate Web Form**

Chapter 23

Dictionary of CDA Functions

Using CDA Functions in Microsoft Excel Formulas

You can use a cell data access (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.

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** section.
4. Click one of the following functions to select it:
   - CDACXRate on page 240
   - CDADesc on page 240
   - CDAGet on page 241
   - CDAName on page 241
   - CDAProperty on page 242

- CDARate on page 244
- CDAXRate on page 244

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.
- **time** is the code of the time period of the exchange rate.
- Each dimension and member pair (for example, *dim_1*, *mem_1*) contains the code for an additional dimension and member that are associated with the exchange rate.

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.

Here 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 dimension and member pair (for example, *dim_1*, *mem_1*) contains the code for an additional dimension and member that are associated with the driver rate.

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 24
Dictionary of Calculated-Member Functions

Using Calculated-Member Functions

Calculated-member functions can be used in the formulas that are associated with calculated members. They are used to access the SAS Financial Management database.

For Web data entry, all arguments must refer to members that are on the table. Calculated members that resolve to CDA formulas are retrieved at the time the form is opened. fmValue results that resolve to cell references are updated as a result of the following actions:

- expanding and collapsing
- pivoting

- slicing
- clicking **Refresh**

For a calculated member in a supplemental schedule, only the fmValue, fmCode, and fmProperty functions are valid.

See Also

"Using Excel-Based Calculated Members" on page 207

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 member code of the *dimensioncode* dimension 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:

```
IF(fmCode("ACCOUNT")="XYZ",1,0)
```

Otherwise, it returns zero.

Syntax Including Optional Arguments

The fmCode function can also be used 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, as follows:

```
fmCode("timedimensioncode","timedimensioncode",offset)
```

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.

offset 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 member for January 2010.

- `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 member for the third quarter of 2010.

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.

This function resolves to a CDA function. It is not valid for supplemental schedules. On the Web, the value is retrieved when the user opens a form.

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 lookup. 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 lookup.

For example, if accounts play no role in the lookup, then the following formula, associated with a calculated account member, can be used to display derived exchange rates from United States dollars to euros:

```
fmCXRate("Derived","USD","EUR")
```

Note: With this three-argument syntax, the fmCXRate function performs its lookup 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 lookup.

Syntax Including Optional Arguments

The fmCXRate function can also include dimension-member pairs, as follows:

```
fmCXRate("exchangeratetype","fromcurrency","tocurrency"
[, "dimensioncode","membercode", ...])
```

Each *dimensioncode*, *membercode* pair represents a dimension and member in the model.

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 Time dimension, 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 so if the function is associated with a calculated time member. For examples, see "fmCode" on page 248.

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 lookup 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:

```
fmCXRate("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 the code of a dimension member's standard or custom property.

Property codes are case sensitive. For a list of standard member property codes, see "CDAProperty" on page 242.

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.

This function resolves to a CDA function. It is not valid for supplemental schedules. On the Web, the value is retrieved when the user opens a form.

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 lookup. If driver rates are associated with organizations in the relevant driver rate set, then the organization member of the crossing is used in the lookup.

For example, if accounts play no role in the lookup, 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 lookup 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 lookup.

Syntax with Optional Arguments

The fmRate function can also be used with one or more dimension-member pairs, as follows:

```
fmRate("driverratetype"[,"dimensioncode","membercode",...])
```

Each *dimensioncode*, *membercode* pair represents a dimension and member in the model.

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 248.

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 lookup 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 dimension that the calculated member is associated with.

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 also include dimension-member pairs, as follows:

```
fmValue("membercode1"[,"dimensioncode","membercode", ... ])
```

membercode1 is the code of a member of the calculated-member dimension.

Each *dimensioncode*, *membercode* pair represents a dimension and member in the model.

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.

Instead of a *membercode* from the Time dimension type, you can use a positive or negative integer to specify a displacement relative to the crossing where the function is evaluated. For example, this function call would return the value of “TShirts” from the previous period:

```
fmValue(“TShirts”, “My_Time”, -1)
```

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.

This function resolves to a CDA function. It is not valid for supplemental schedules. On the Web, the value is retrieved when the user opens a form.

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 syntax, the fmXRate function performs its lookup 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 include dimension-member pairs from the model's time and analysis dimension, as follows:

```
fmXRate("exchangeratetype","fromcurrency","tocurrency"[,"timedimensioncode",
"timemembercode"][,"analysisdimensioncode","analysismembercode"][, offset])
```

timedimensioncode, *timemembercode* contain the dimension code and a member code for the model's time dimension..

analysisdimensioncode, *analysismembercode* contain the dimension code and a member code for 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, see "fmCode" on page 248.

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")
```

Index