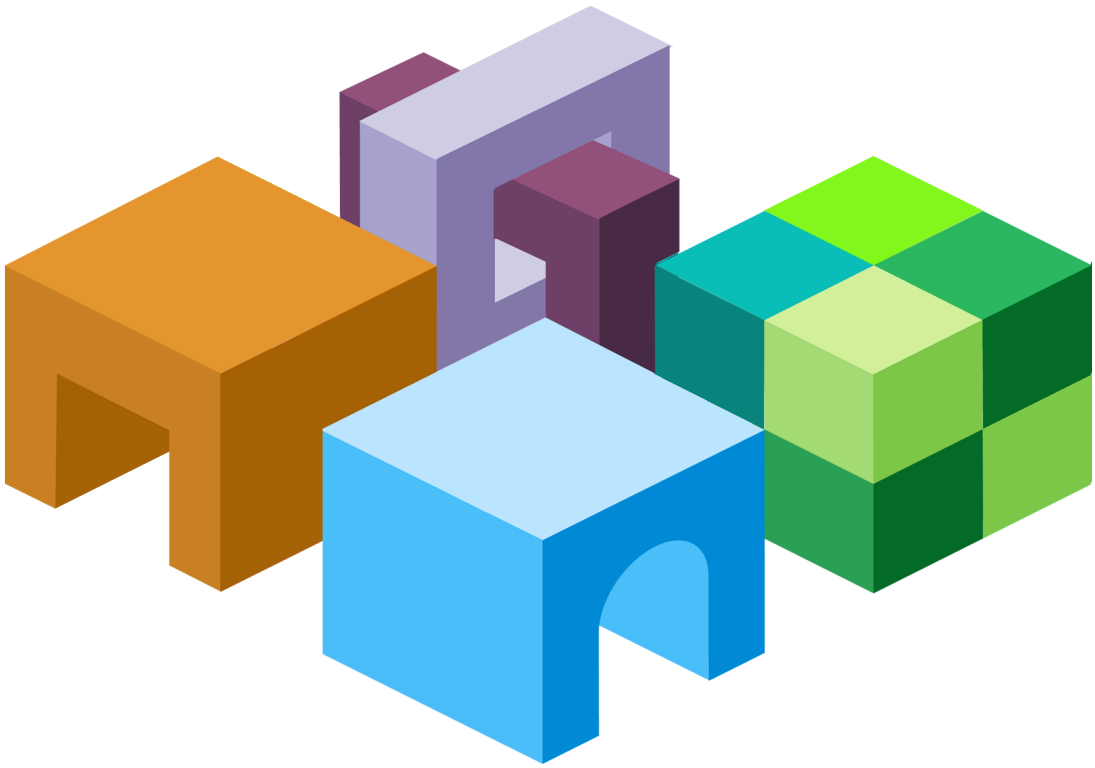


ORACLE® HYPERION ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

ORACLE® HYPERION FINANCIAL DATA QUALITY MANAGEMENT, FUSION EDITION

RELEASE 11.1.2

ADMINISTRATOR'S GUIDE

ORACLE®
ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

FDM Administrator's Guide, 11.1.2

Copyright © 1989, 2010, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited. The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS:
Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Basics of FDM

In This Chapter

Overview of FDM

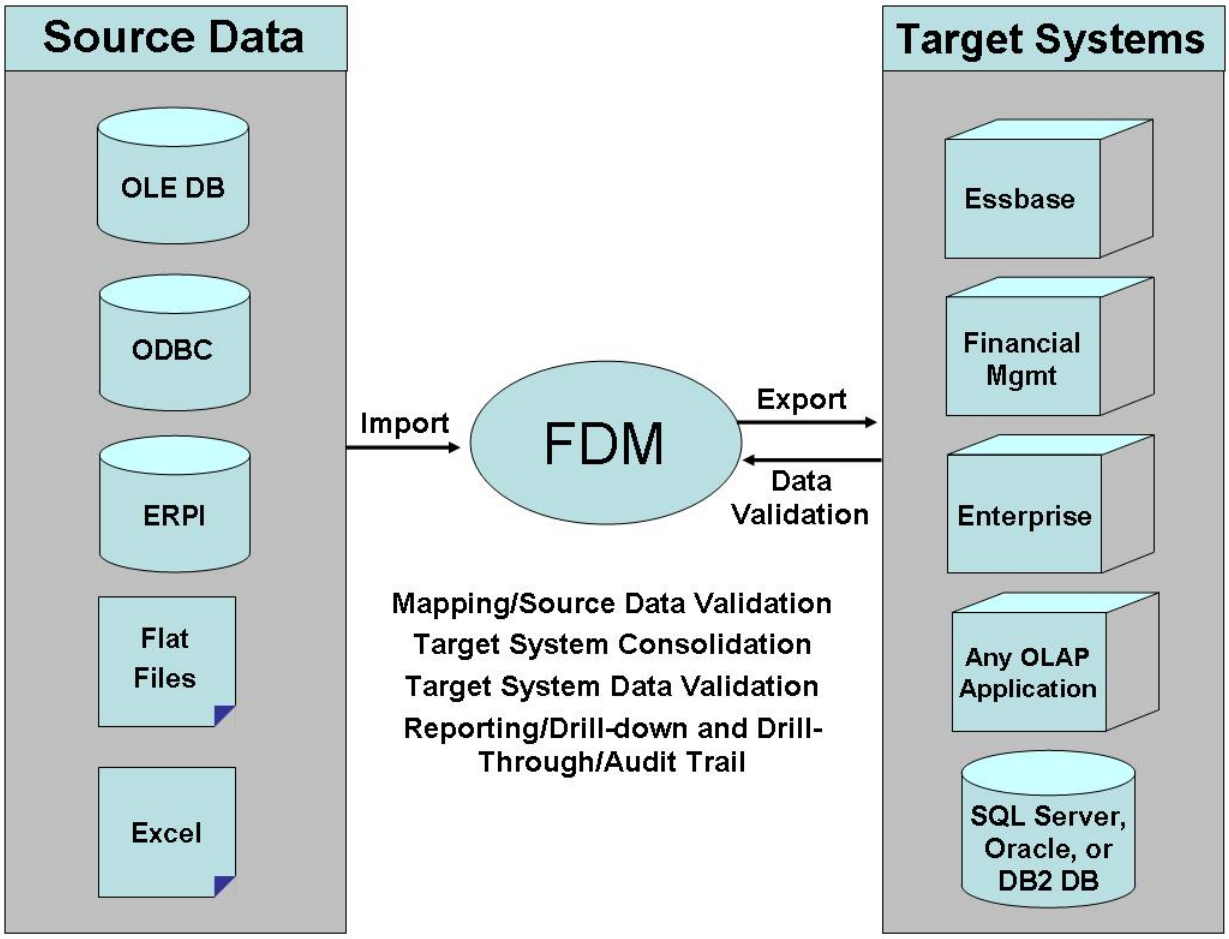
Oracle Hyperion Financial Data Quality Management, Fusion Edition (FDM) is an out-of-the-box data transformation tool that feeds source-level financial data to consolidation, reporting, planning, and analytical applications. More than a data translation tool, it provides these capabilities:

- Provides an audit trail to source financial data
- Ensures the integrity and mapping consistency of source financial data
- Enables easy reconciliation of financial data
- Reduces the time that users, administrators, and auditors must spend investigating, identifying, and correcting errors

FDM provides a uniform data collection process for all reporting units within an organization. FDM also contains financial control functionality that assists with internal financial control processes. FDM features adapters that enable it to integrate with Oracle's Hyperion® Enterprise® , Oracle Hyperion Financial Management, Fusion Edition (Financial Management), Oracle Essbase (Essbase), and Oracle Hyperion Planning, Fusion Edition (Planning). However, FDM can also be used to load data into other financial consolidation and analytical applications.

The basic FDM process includes six steps:

1. Import source data
2. Validate source data against mapping tables
3. Export source data to a target system
4. Consolidate target system data
5. Validate target system data
6. Review and validate internal financial control



Features and Benefits

FDM provides a systematic process for loading source data from disparate systems into target analytical applications. The process provides data visibility, integrity, and verification.

FDM features and benefits:

- A central repository that contains all source data
- Drill Down and Drill Through audit trails
- Archiving of source files, error logs, and load files
- An internal control assessment and certification feature that facilitates compliance with sections 302 and 404 of the Sarbanes-Oxley Act
- Corporate-wide process monitoring
- Import of source data from any formatted text file or data source
- Multiple dimension mapping and validating
- Data validation and quality evaluation
- Error identification and notification

- Consolidation of target system data
- Validation of and reporting on target system data
- Load adjustment through Excel journals
- Budget data loading for multiple periods
- Advanced reporting and audit functions
- "Lights out" batch loading
- Support for unlimited concurrent users
- Zero footprint Web deployment
- Support for SQL and Oracle databases

Product Set

FDM includes two primary products, a Windows client (Workbench) and a Web client.

Workbench

FDM Workbench is a Windows client that can be installed on a PC or on a terminal server. Workbench provides access to the application setup, integration, and development features of FDM and should be used only by administrators.

FDM Workbench basic functionality:

- Systems integration
- Script creation and maintenance
- Report creation and maintenance
- Application importing and exporting
- Batch processor setup and monitoring

Web Client

The FDM Web client consists of four components:

- Web server
- Application Server
- Load Balance Manager
- Task Manager

Web Server

FDM Web Server components can be installed on a Web server running IIS 6.0 or higher. The components enable users to access FDM applications from the Internet or from a corporate

intranet. The Web-based interface provides access to all end-user features and most administrator features of FDM.

Application Server

FDM Application Server enables FDM to execute resource-intensive tasks on one or more application servers, instead of on the Web server. To minimize network traffic and increase data transfer rates, installing the Application Server component on a server other than the Web or data (SQL or Oracle) server is recommended.

Load Balance Manager

FDM Load Balance Manager enables FDM to manage the application servers that execute resource-intensive tasks. Installing Load Balance Manager on the first application server is recommended.

Task Manager

FDM Task Manager runs scheduled tasks (FDM scripts) at specified intervals (daily, weekly, or monthly). You can also use Task Manager to add, modify, and delete tasks. Task Manager can be installed on any server that has access to the FDM application.

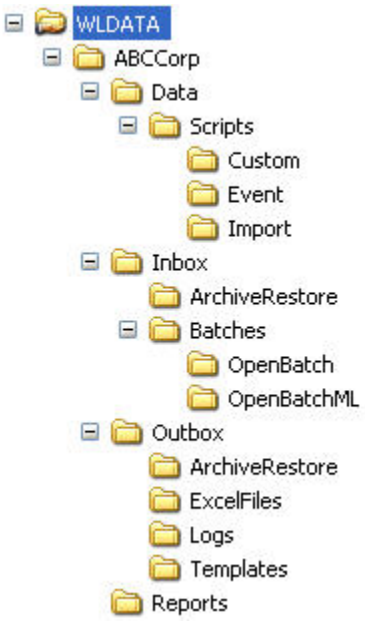
Overview of FDM Applications

A FDM application consists of a relational database management system (RDBMS) database and directories that contain the transactional data, metadata, reports, and other files that are used to integrate FDM with target applications.

One FDM application can load to multiple target applications of multiple systems. For example, one FDM application can load to two Hyperion Enterprise applications, a Financial Management application, and three Planning applications.

FDM Application Architecture

The following diagram shows the directories that are created when a FDM application is created. In this example, WLDATA is a user-created directory that is used to store all FDM applications. When the ABCCorp application was created and named, FDM created the `ABCCorp` directory and its subdirectories. Each application contains four main directories: `Data`, `Inbox`, `Outbox`, and `Reports`.



Data

In `Data`, FDM archives data files that are imported to and exported from FDM (imported source files, import logs, journal entries, multiload files, target system load files, and any attached memo documents). Source files, journals, multiload files, logs, output data files, and attachments are also archived here. Each file in `Data` is assigned a unique name, can be retrieved from the Import and Export screens, and provides an audit trail. See "Reviewing Import Audit Information" on page 138.

Scripts

`Data` contains a `Scripts` subdirectory, where FDM scripts are stored. `Scripts` contains subdirectories, each of which contains scripts of a specific type:

- **Custom**—Scripts that can be executed through a custom menu, which is created in the Menu Maker screen.
- **Event**—Scripts that are executed when a particular event runs. The scripts are named for the events with which they are associated.
- **Import**—Import scripts that are created when import formats are defined (Import scripts are executed during the source file import process).

Inbox

Inbox, the default directory from which to import source files, can be used as a central repository for all ledger extract files. Because source files can be retrieved from any accessible directory, you are not required to place import files in this directory. `Inbox` includes two subdirectories: `Archive Restore` and `Batches`.

Archive Restore

This directory is used to store restored import source files and logs that were previously archived. FDM stores the original archived source files and logs in the `Data\Archive` directory. See "Creating Validation Entity Groups" on page 134 and "Reviewing Import Audit Information" on page 138.

Batches

Files used for batch loading are stored in `Batches`, standard batch files in the `OpenBatch` subdirectory and multiload batch files in the `OpenBatchML` directory.

Outbox

`Outbox` provides a central location for all FDM export files. `Outbox` also contains four subdirectories: `Excel Files`, `Logs`, `Templates`, and `ArchiveRestore`.

Excel Files

When FDM exports the contents of a grid, the resulting Excel files are stored in `ExcelFiles`.

Logs

`Logs` contain the log files that are created when source files are imported. The logs contain the data lines that FDM did not import and an explanation of whey each line was not imported. `Logs` also contains error logs, which are named per the following convention (<*username*>.err), where <*username*> is the user that is logged on to FDM and `.err` is the extension used to identify error logs. `Logs` can be purged to reclaim disk space.

Templates

`Templates` is used by reporting locations to manage and distribute custom journal or multiload templates. The contents of `Templates` are displayed as links within the FDM Web client. You publish a template by placing it in `Templates`.

Archive Restore

`Archive Restore` stores restored data load files that were previously archived. FDM stores the original archived data load files in the `Data` directory. See "Reviewing Import Audit Information" on page 138. The contents of `Archive Restore` are deleted when a compact is performed.

Reports

`Reports` stores the Active Report files. Active Report files use the `*.rpx` extension.

Adapters

Adapters are software codes that communicate with source and target applications. Each adapter is designed to integrate with a specific target system (for example, Essbase or Financial Management) or source system (for example, Oracle Hyperion Financial Data Quality Management ERP Integration Adapter for Oracle Applications, or SQL). Workbench enables you to import and maintain the adapters used by FDM.

Target adapters are composed of an XML metadata component and a DLL or EXE. Other adapters are composed of an XML component only. The DLL or EXE components contain the system-specific instructions for interacting with the target system. The XML component contains all data that relates to the FDM application.

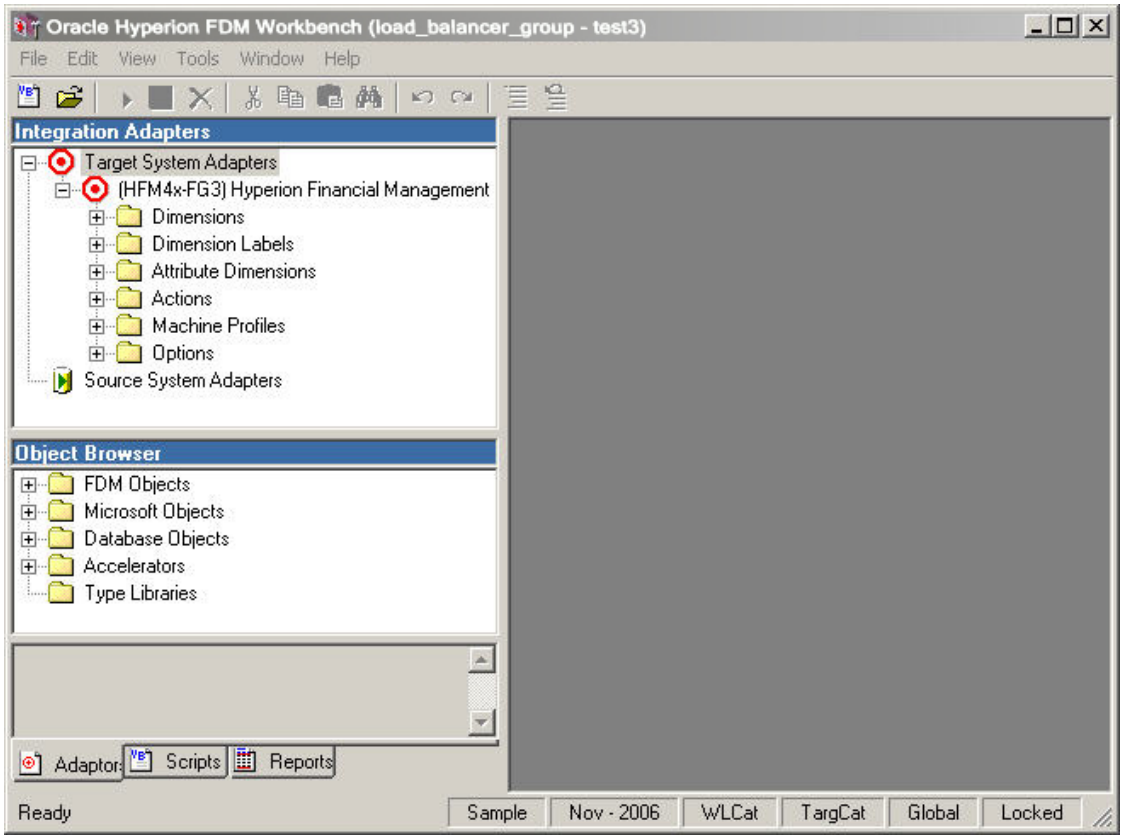
Adapter DLL and EXE

The target system adapter `DLL` or `EXE` acts as a buffer between FDM and a target application. The `DLL` contains the instructions that enable FDM to communicate with the specified target application (connecting to the database, loading data, extracting data, and so on). Each `DLL` contains the API calls for its application. Use of `DLL` enables FDM to integrate with many target systems without having to maintain large amounts of application-specific calls.

Adapter XML

The adapter `XML` acts as a second layer between FDM and the target application. The `XML` stores application setup parameters and options and scripts that call API functions and returns the results from the target application. The `XML` may also contain the information required to configure a FDM application (locations, security, import formats, reports, and so on).

You access the Adapters menu from the Workbench desktop by selecting the Adapters tab at the bottom of the left pane.



Each adapter listed in the Integration Adapters pane contains the following sections:

- Dimensions
- Dimension Labels
- Attribute Dimensions
- Actions
- Machine Profiles
- Options

Dimensions

The Dimensions section lists the dimensions for the target application. Each item under Dimension represents one available dimension and contains a script that retrieves, from the target application, a list of all members of the dimension from the target application. Only target adapters use the Dimensions section.

Actions

The Actions section contains scripts that interact with the target application and use the adapter DLL API calls to perform such functions as Connect, Load, ValueGet, Export, Drill Down, Drill Through, and so on.

Machine Profile

The Machine Profile contains information that enables FDM to determine on what computers or servers the source and target databases are located. The machine profile also stores global logon information for connecting to target applications. When the Global Login option is selected, the same user name and password is used to log on to FDM and the target application. Therefore, in this case, the user account must have access to the target application.

For FDM to load and retrieve values to and from the target application, the user account used to log on to the target application must have appropriate security privileges in the target application, regardless of what user account is logged on to FDM. You must configure a machine profile for every computer that has the FDM Application Server or Task Manager component installed.

Options

The Options section of the `XML` file contains the integration setting defaults and application settings for the target application with which FDM is integrating.

2 Managing FDM Applications

In This Chapter

Setting Up Load Balance Server Groups

Before accessing Workbench, you must assign a load balance server group. The server group defines the load balance servers that are used by FDM. When Workbench is launched for the first time, the Load Balance Server Group Setup screen is displayed.

➤ To set up load balance server groups:

1 **Open the Workbench Logon screen by selecting Start > Programs > Oracle EPM System> Financial Data Quality Management > Workbench > Workbench Client.**

The Load Balance Server Group dialog box is displayed (if this is the first time that Workbench was opened).

2 **Click Add.**

The Load Balance Server Group screen is displayed.

3 **Enter a server group name and description.**

4 **Enter the name of the primary load balance server.**

You can browse to find available servers by clicking ....

5 **Optional: Enter the name of the backup load balance server.**

You can browse to find available servers by clicking ....

6 **Click Connect.**

You can define multiple load balance server groups, thus enabling Workbench on multiple FDM applications on multiple servers.

Note: If you want to add or modify a server group, you can access the Load Balance Server Group form by logging on to Workbench and selecting File > Load Balance Server Group.

Creating, Removing, Modifying, and Adding Applications

Applications are managed (created, removed, modified, and added) from the New Application and Open Application dialog box boxes.

Creating Applications

➤ To create applications:

1 Access the FDM **Logon** page by launching the Web client or Workbench:

- **Web Client**—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Web Server Components** > **Web Logon**.
- **Workbench**—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Workbench** > **Workbench Client**.

2 From **Application**, select **<New Application>**.

3 Enter your FDM name and password.

4 Click **Logon** (Web client) or **OK** (Workbench).

The New Application dialog box is displayed. It features two tabs—General and Database .

5 On the **General** tab, enter application information.

You use the General tab to create the application name and description and to define where the application architecture is stored.

- **Name**—The name can contain up to twenty alphanumeric characters. Do not use spaces. Underscores (_) are acceptable.
- **Description**—Description of the application.
- **Path**—Path to the new application. If users access the application through multiple Web and application servers, it is recommended to use the UNC naming convention to avoid problems with inconsistent drive letter mapping.

When an application is created, a directory of the same name is created. The application name is added to the `HyperionFDMApplications.xml` file, which is stored in the FDM installation directory.

6 Select the **Database** tab, and enter database information.

You use the Database tab to define the properties of the RDBMS database that stores the FDM transaction data.

- **OLE DB Provider**—The database used by FDM. The default database is SQLOLEDB. SQLNCLI (SQL Server 2005 Native Client) and Oracle are also supported. If SQLNCLI is used, Microsoft SQL Server 2005 Native Client is required on all FDM application servers, load balance servers, and any computer where FDM Workbench is installed.

- **Database Server**—The location of the database used by FDM. This option is visible only when OLE DB Provider is set to SQLOLEDB or SQLNCLI.
- **Database Name**—Typically, the same as the application name. This option is available only when OLE DB Provider is set to SQLOLEDB.
- **Service**—The Oracle service used for connecting to the Oracle database. This option is available only when OLE DB Provider is set to ORAOLEDB.
- **Username**—The database administrator user name, for example, the SQLOLEDB administrator user name.
- **Password**—The database administrator password.
- **Options**—An override of the default table space where the application is created. This option is available only when OLE DB Provider is set to ORAOLEDB.

7 Click **OK**.

Note: If override table spaces is not selected, all tablespaces default to Users. Use of the default can severely degrade performance. See the *Financial Data Quality Management DBA Guide* for detailed tuning instructions prior to creating the FDM application.

Removing Applications

Removing an application does not delete or impact the data in the application. This task only deletes, from the FDM XML file, the current user's pointer to the application. The directory that contains application information remains intact.

➤ To remove applications:

1 Access the FDM **Logon** page by launching the Web client or Workbench:

- **Web Client**—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Web Server Components** > **Web Logon**.
- **Workbench**—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Workbench** > **Workbench Client**.

2 From **Applications**, select **<Add Application>**.

3 Enter your FDM user name and password.

4 Click **Logon** (Web client) or **OK** (Workbench).

The Add Application dialog box is displayed.

5 Select an application.

6 Click **Remove**.

7 Click **OK**.

Modifying Applications

➤ To modify applications:

1 Access the FDM **Logon** page by launching the Web client or Workbench:

- Web Client—Select **Start** > **Programs** > **Oracle EPM System**> **Financial Data Quality Management** > **Web Server Components**> **Web Logon**.
- Workbench—Select **Start** > **Programs** > **Oracle EPM System**> **Financial Data Quality Management** > **Workbench** > **Workbench Client**.

2 From **Application**, select **<Add Application>**.

3 Enter your FDM user name and password.

4 Click **OK**.

The Open Application dialog box is displayed.

5 Select an application.

6 Click **Modify**.

The Modify Application dialog box is displayed.

7 In **Modify Application**, change one or more settings.

You can change any application attribute except the application name. To change the name of an application, you must remove it and then use the Add Application feature to add it.

8 Click **OK**.

The procedure for creating applications is used when an application does not exist. The procedure for adding applications is used when the application exists but there is no pointer to the FDM XML file.

Adding Applications

➤ To add applications:

1 Access the FDM **Logon** page by launching the Web client or Workbench:

- Web Client—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Web Server Components** > **Web Logon**.
- Workbench—Select **Start** > **Programs** > **Oracle EPM System** > **Financial Data Quality Management** > **Workbench** > **Workbench Client**.

2 From **Application**, select **<Add Application>**.

3 Enter the FDM user name and password.

Note: The domain name may be required for certain configurations.

4 Click **Logon**.

The Applications dialog box is displayed; all applications defined on the computer are listed.

5 **Click Add.**

The Add Application dialog box is displayed. It provides two tabs—General and Database.

6 **On the General tab, enter application information.**

- **Name**—Do not use spaces. Underscores (_) are acceptable.
- **Description**—Application description.
- **Path**—Path to the application.

7 **Select the Database tab, and enter the database information.**

You use the Database tab to define the properties of the RDBMS database and stores the FDM transaction data.

- **OLE DB Provider**—The database used by FDM. The default database is SQLOLEDB (SQL). Oracle is also supported.
- **Database Server**—The location of the database used by FDM (only available when SQLOLEDB is selected in OLE DB Provider).
- **Database Name**—Typically the same as the application name. The database name you define will be the name of the RDBMS database (only available when SQLOLEDB is selected in OLE DB Provider).
- **Service**—The Oracle service used for connecting to the Oracle database. This option is available only when OLE DB Provider is set to ORAOLEDB.
- **Username**—The database administrator user name, for example, the SQLOLEDB administrator user name.
- **Password**—The database administrator password.
- **Options**—An override of the default table space where the application is created (available only when ORAOLEDB is selected in OLE DB Provider).

8 **Click OK.**

The `.ini` file and registry settings for the application that is referenced are created.

Logging on to Applications for the First Time

The FDM login page enables users to log on to applications, create applications, or add applications. The user name that you use to create an application is the only valid user name until the application is configured to accept other users.

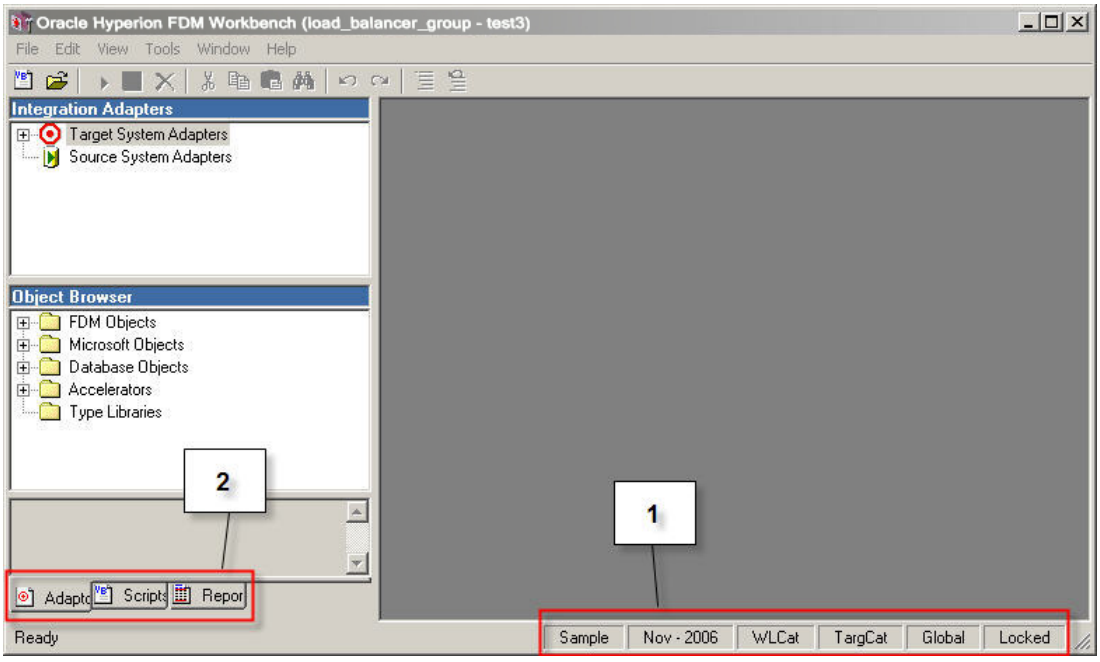
When logging on to a new application for the first time, you are prompted to set up a new FDM integration. See the *FDM Configuration Guide* for information regarding integration adapter setup, and Chapter 4, "Configuring System Settings" for information regarding configuration settings.

3 Navigating FDM

In This Chapter

Functions in Workbench

The Workbench desktop:



1. POV bar—Select the FDM location, FDM period, FDM category, and target category; global or local mode; and system lock status.
2. Screen selector tabs—Click to display the Adapters screen, Scripts screen, and Reports screen.

You can use keyboard shortcuts that perform various Workbench actions:

- Alt + F4: Close the program.

- Ctrl + O: Open the application login form.
- Ctrl + N: Open a script.

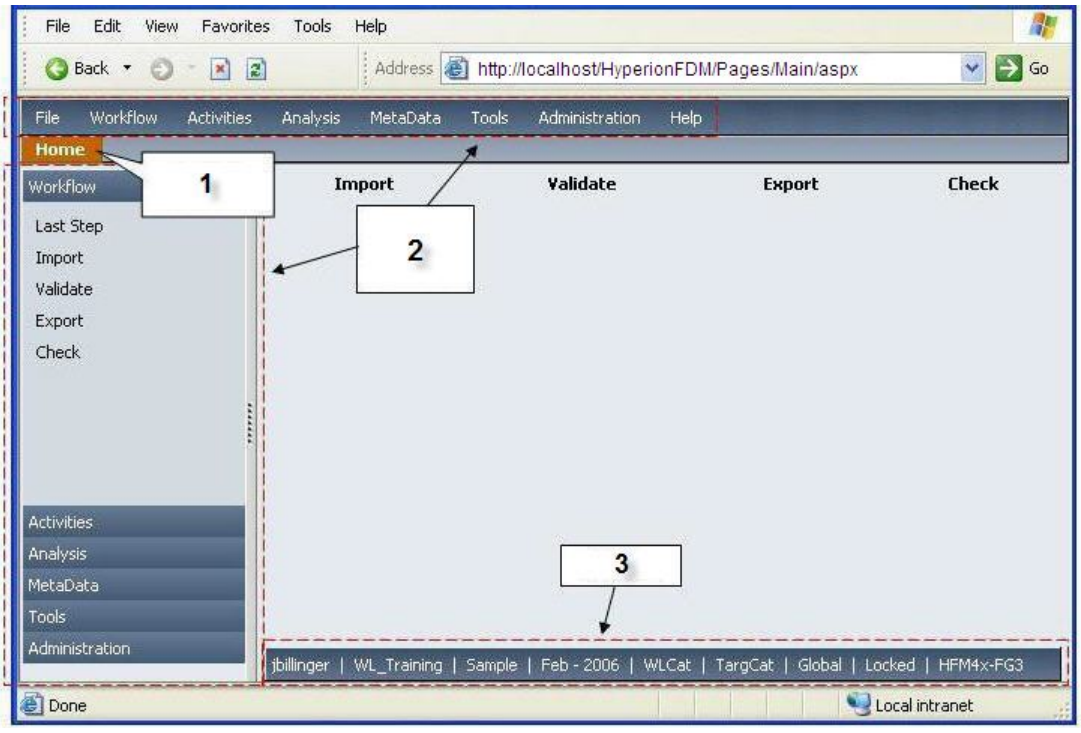
Functions in Web Client

Some common functions and features in Web client:

- Desktop
- Workflow menu
- Working with data in grids
- Sorting columns
- Searching control trees
- Customizing the interface

Desktop

The FDM Web client desktop is the control center where applications are built and administered. After you log on to an application, the desktop is displayed.



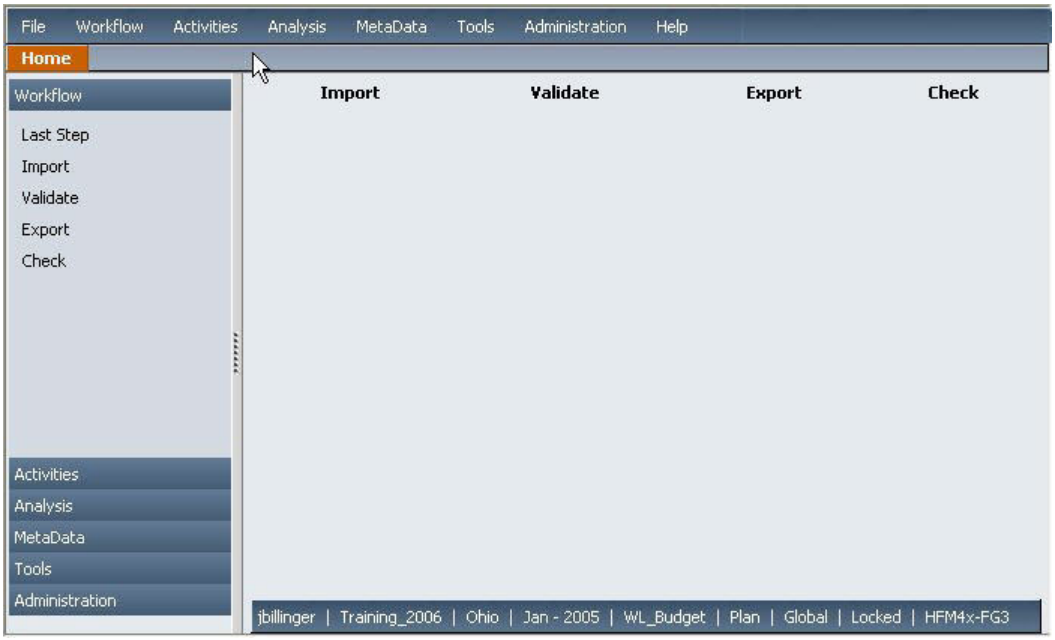
1. Screen indicator—Identifies the current FDM screen
2. FDM menu rollup and FDM menu—The rollup contains six tabs. and the menu contains seven menu items (six identical to the menu rollup plus the File menu). Select a tab or menu item to view its underlying menu. All FDM functions available in the Web client can be accessed from the menus. In the example figure, the Workflow tab is selected, and the

Workflow menu is displayed. You can access all functions from the menu bar at the top of the FDM desktop

3. POV bar—Displays the current user, FDM application, FDM location, FDM period, FDM category, target category, mode (global or local), system lock status, and adapter used by the current location

Workflow Menu

The Workflow menu enables users to open workflow tasks without reprocessing all workflow steps. Users can open a task and then decide whether to re-validate or export.



Five steps of the workflow process:

- **Last Step**—Opens the last Workflow Process screen that was run for the current POV (whether or not successful). If a user attempted to validate data and failed, click Last Step and your Validate screen is displayed
- **Import**—Displays the Import screen
- **Validate**—Displays the Validate screen (even if the Import process has not been run for the current POV) but does not validate the data
- **Export**—Displays the Export screen (even if the current POV has not validated its data) but does not initiate the Export process
- **Check**—Displays the Check report for the current POV (if there is no check report data for the current POV, a blank page is displayed)

Working with Data in Grids

Most forms (screens) in FDM display data in one or more grids. To manipulate grid data, perform one or more actions:

- To add a record, click **Add**.
- To delete, select a record and click **Delete**.
- To delete all records in a grid, click **Delete All**.

Note: If a form contains multiple grids, you cannot delete the contents of a grid.

- To edit a record, click within its cell, and start typing.
- To cancel all changes made to a row, click to the left of the row and select **Cancel Changes**.
- To save a record, click **Update Grid**.

  All changes on the current grid are sent to the database.
- To send updated rows (adds, deletes, and changes) to the database, click **Update Grid**.
- To output the contents of a grid to an Excel spreadsheet, click **Excel**.

  The Excel spreadsheet is saved to the `Excel` subdirectory inside the `Outbox` directory that is specified in the Connections Dialog form.

Note: Screens exported to Excel are configured for re-import (where available) into FDM. The named range is automatically defined, and the Import table name and column names are set.

Sorting Columns

➤ To sort grid columns:

1. Click a column heading (to activate the column).
2. Click the up-down arrow to the left of the column description.
3. Optional: To re-sort the column in the opposite, click the up-down arrow again to

Searching for Records

➤ To search for records:

1. In the grid, click a column heading (to activate it).
2. Click (search button).
3. In **Find**, enter the search criteria.
4. Click  (Find Next button) to cycle through all records that match the specified criteria.

Note: If a form contains multiple grids, you cannot sort any column of any grid.

Filtering Records

➤ To filter the records of a grid:

1 **In the grid, click a column heading (to activate it).**

2 **Click**  **(filter button).**

The Filter dialog box is displayed.

3 **From Filter Method, select an operator.**

4 **In Filter Value, enter a value by which to filter.**

5 **Optional: To remove filtering, select All.**

Note: If a form contains multiple grids, you cannot use the filter function.

Searching Control Trees

Screens that include control trees, such as the Locations screen, include a search function.

➤ To search control trees:

1 **Right-click a node, and select Search.**

The Search dialog box is displayed.

2 **Enter the search criteria.**

The search criteria can be an exact location name or part of a location name. The input is not case sensitive.

3 **Click OK.**

Customizing the Interface

The Web interface can be customized on a global application level or user level.

Web Client Settings

The Web Settings screen enables you (the administrator) to control status bar icons, themes, number of grid rows to display per page, and the operation of the information bar. You access the screen by selecting Administration > Web Settings. The screen features five tabs:

- General
- Grid

- Theme
- Info Bar
- Cache

General Tab

Allow Status Bar Icon Override—Enables users to override the default status bar icons. The status bar icons are the pass or fail icons that are displayed in the Workflow area under Import, Validate, Export, and Check.

Default Status Bar Icons—Uses the status bar icons that are displayed under the Workflow processes (Import > Check). The default icon group is used unless the user selects an override group. The available groups are stored on the Web server under`\\Hyperion\Products\FinancialDataQuality\WebServerComponent\WebSite\StatusBarIcons`.

➤ To create status bar icon groups:

1 **Create a directory under the `StatusBarIcons` directory.**

2 **Place additional icons in the directory that you created.**

Note: The new icons and the icons in the existing directories must have the same names.

Allow Default Report Publishing Type Override—Enables users to override the default report publishing type. The default report publishing type is the default format for displayed reports, including Check reports and Process Explorer reports.

Default Report Publishing Type—Uses the default report type for Check reports and Process Explorer reports. The Publish-type list on the Reports page is set to the default. Selections are PDF, Excel, Word, Rich Text, and HTML.

Generate Debugging Information—Enables the Info Bar tab to display stack trace information (if an error occurs). Stack trace information is helpful if you need to contact Customer Support for troubleshooting assistance.

Note: This option should be enabled only if an error occurs and Customer Support requires additional information.

Grid Tab

Allow Grid Override—Enables users to override the default number of grid rows displayed on a page. The default is 100. When this option is enabled, users can set any number up to the Maximum Grid Page Size setting, which is set by the administrator.

Default Grid Page Size—Used the default number of grid rows displayed on a page, unless the user specifies an override value.

Maximum Grid Page Size—Sets the maximum number of grid rows that users can specify when overriding the default number of grid rows per page. To avoid performance degradation, the maximum size for this field is 500.

Theme Tab

Allow Theme Override—Enables users to override the FDM default color scheme. The default theme contains all colors, styles, and general-use icons that are displayed on the Web interface. If this option is enabled, users can select from the list of available themes on the User Settings page.

Default Theme—Uses the default color scheme for the current application. The default theme is used unless the user selects an override group. The groups available are stored on the Web server under `\\Hyperion\Products\FinanicalDataQuality\WebServerComponent\WebSite\Themes`.

➤ To create themes:

1 **Add a directory under the `Themes` directory.**

2 **Place the additional icons and style files in the directory you created.**

Note: The icons that you create and the icons in the existing directory must have the same names.

Info Bar Tab

Only Display Errors and Script Messages—Disables the display of confirmation messages (update button clicks, records updated, rows deleted, and so on) in FDM Info Bar tab. If this option is enabled, FDM displays only script-generated messages in Info Bar.

Info Bar Timer—Sets the amount of time that the Info Bar tab is displayed. After the tab is hidden, it is accessed by clicking the Information icon ().

Cache Tab

Clear Web Cache—Clears the Web server memory of all XML values stored for the table editor. This option does not delete files from the Web server; rather, it deletes the cached views of files.

Note: For user interface changes to become effective, the FDM session must be restarted.

User Settings

User settings are accessed by selecting Tools > User Settings. These settings enable users to control themes, status bar icons, number of grid rows to display per page, and the default report publishing type. Four options are provided:

- Themes
- Status Bar

- Grid Page Size
- Default Report Publishing Type

Themes—The default theme contains all the colors, styles, and general-use icons that are displayed on the Web Interface. Users can select a theme from the Themes list.

Status Bar Icons—The status bar icons are the pass and fail icons that are displayed in the workflow area under Import, Validate, Export and Check. Users can select a status bar group from the Status Bar Groups list.

Grid Page Size—The maximum number of rows that a user can enter is set in Maximum Grid Page Size, which is set on the administrator Web Settings page.

Default Report Publishing Type—Users can select the default report type that is used when Check reports and Process Explorer reports are run. By default, the Publish Type field on the Reports page is set to the selected report-type value. Selections for this field are PDF, Excel, Word, Rich Text, and HTML.

Note: For changes to become effective, the FDM session must be restarted. An option can be set by a user only if the administrator enables an override for it.

Locking and Unlocking the POV

The POV Lock Settings consist of the following options:

- Lock Current Point-of-View
- Unlock Current Point-of-View
- Lock All Locations (Current Category/Period)
- Unlock All Locations (Current Category/Period)

Locking the POV prevents locations from modifying the data. When a location has been locked for a particular period or category, users cannot import, validate, export, or re-run the validation report. Run the Process Monitor report to view the lock status for all locations. When a location is locked, a lock symbol () is displayed in the POV bar.

The POV can be locked or unlocked for individual locations. This can be performed by administrators or by users who have access to the Lock Current Point-of-View/Unlock Current Point-of-View menu item. There is also an option, available only to administrators, to lock or unlock all locations simultaneously.

➤ To lock or unlock individual locations (all users):

1 **From the Web client, set the POV to the category and location you want to lock or unlock.**

2 **Select the appropriate menu item:**

- To lock—Select **Tools** > **Lock Current Point-of-View**
- To unlock—**Select Tools** > **Unlock Current Point-of-View**

➤ To lock or unlock all locations for the current category and period (administrators only):

1 From the Web client, set the POV to the category and period you want to lock or unlock.

2 Select the appropriate menu item:

- To lock all locations—Select **Administration** > **Lock All Locations** (**Current Category/Period**)
- To unlock — Select **Administration** > **Unlock All Locations** (**Current Category/Period**)

Locking and Unlocking the POV Mode

Use this option to control the availability of the Local POV mode. When this option is On, end users cannot change to the Local POV mode. When this option is off, Local POV mode is available to all users on the system. See "Switching POV Mode" on page 39

for information on the POV mode.

➤ To change the POV Mode lock:

1 From the Web client, select **Administration > Point-of-View Mode Lock**.

2 Select or clear **On/Off**.

3 Click **OK**.

Functions Common to Web Client and Workbench

You use the POV to set the FDM data focus. When the POV lock is enabled, the period and category values are globally controlled across the system, and only the system administrator can change them. This control mechanism ensures that end users can load data only to the proper period and category.

To increase flexibility for end users, you can disable the POV lock. This action enables end users to change to Local POV mode to process data for any category or period.

Switching POV Mode

The mode indicator in the POV bar displays the current POV mode—Global or Local. When the system is in Local mode, double-click Local to switch to Global mode. When the system is in Global mode, double-click Global to switch to Local mode.

- **Local POV Mode**—Allows any category or period to be selected for processing
- **Global POV Mode**—Restricts category and period processing to a global value that is active for all users on the system

Note: System administrators can change the POV mode to Local, but end users can access this option only if the POV lock is disabled.

Setting Location POV

Users can view only locations that they have permission to access. Location permission is determined and granted by system administrators. All users must be assigned one default location. When users log on, the POV is set to the user's default location.

➤ To set the location POV:

1 **In the POV bar , double-click the current location.**

The POV dialog box (Locations tab) is displayed.

2 **Select a location.**

3 **Click OK.**

Note: System administrators can change the POV mode to Local, but end users can access this option only if the POV lock is disabled.

Setting Category POV

The FDM administrator controls which data category that is active for all users. This feature prevents users from inadvertently loading data to incorrect categories.

➤ To set the category POV:

1 **In the POV bar, double-click the current category.**

The POV dialog box (Category tab) is displayed.

2 **Select a category.**

3 **Click OK.**

When the FDM category is changed, the target system category is changed. When a FDM category is defined on the Control Tables form, it is associated with a target system category.

Setting Period POV

The FDM administrator controls which accounting period is active for all users. This feature prevents users from inadvertently loading data into incorrect periods.

When a user logs on to FDM, the application identifies the global period value and automatically sets the POV to the current value.

➤ To set the period POV:

1 **In the POV bar, double-click the current period.**

The Period tab of the POV dialog box is displayed.

2 **Select a period.**

3 Click **OK**.

If the FDM period is changed, the target system period is changed.

Searching the POV

➤ To search from within the Web client POV:

1 Open the **POV** dialog box by double-clicking the current location, period, or category from the POV bar.

2 Click **Search**.

3 In the **Explorer User Prompt** dialog box, enter a full or partial string for which to search.

4 Click **OK**.

➤ To search from within the Workbench POV:

1 Open the **POV** dialog box by selecting the current location, period, or category from the POV bar.

2 In **Select Location**, enter a full or partial string for which to search.

3 Click **>**.

Locking and Unlocking the System

You use this option to terminate current user sessions and to prohibit new users from logging on. You can also enter a message. to be displayed to users when they are disconnected. You must terminate user sessions before performing database maintenance activities.

When the system is locked, users cannot log on. User sessions are not terminated while records are being written to the database.

➤ To lock and unlock the system:

1 In the POV bar (Web client or Workbench), double-click the system lock category.

The system lock category is the right-most option in the POV Bar and is labeled *Locked* or *Open*.

The System Lock dialog box is displayed.

2 Toggle the **On/Off** option to change the system lock status.

3 **Optional:** In **System Lock Message**, enter the message that you want displayed to users when they are disconnected.

4 Click **OK**.

The new system lock status will appear in the POV ("Locked" or "Open").

Note: System locking and unlocking privileges are restricted to administrators.

How User Default POVs Are Determined

When a user logs on to FDM, the following sequence of events executes to set the POV:

1. Location POV is set by retrieving the user's security profile.
2. POV mode is set to Global.
3. Category POV is set by retrieving the system global category (set by the administrator).
4. Period POV is set by retrieving the system global period (set by the administrator).

4 Configuring System Settings

In This Chapter

System Options

System options global values that control the behavior of FDM applications, are used as control and performance-tuning mechanisms. Three types of system options:

- Application settings
- Integration settings
- Configuration options

You can configure system options from both Workbench and the Web client.

Configuring Application Settings

➤ To configure application settings:

1 **Open the Application Settings dialog box:**

- **From Workbench:** Select **Tools** > **Application Settings**.
- **From the Web client:** Select **Administration** > **Application Settings**.

2 **From Options, select an application setting.**

3 **Select the option settings.**

The application setting that is selected in Options determines which options are displayed. The fifteen application-setting options are described in the following table:

Table 1 Application Setting Options

Application Setting Option	Description
System Code	Specifies the target system that you integrate with FDM. Each target-system integration adapter requires a target-system cache. One FDM application can be associated to an unlimited number of target applications.
Log Map Changes	When selected, specifies that all changes made to the mapping table, for any location, are written to the system log. Thus, users can maintain an audit trail of mapping changes. The default value is On.
Batch Loader	Controls access to Batch Loader.
Archive Method	Specifies whether archived files are copied or moved to the archive location.
Controls Lock	Used with the Financial Controls functionality of FDM. See Chapter 12, "Setting Up Financial Controls" for details.
Controls w/o Data	Used with the Financial Controls functionality of FDM. See Chapter 12, "Setting Up Financial Controls" for details.
Controls Unsubmit	Used with the Financial Controls functionality of FDM. See Chapter 12, "Setting Up Financial Controls" for details.
Fiscal Start Period	Relates to the Financial Controls functionality of FDM. See Chapter 12, "Setting Up Financial Controls" for details.
Period Key Date Format Mask	Used to set the date format of the FDM Periods description field. All subsequent FDM periods created adhere to the format specified for this option. The default value is *MMM-YYYY*.
Create Location Folder	Determines, when FDM locations are created, whether FDM creates a location directory in the Inbox. Options are Data Load, All, or None.
Delete Location Folder	Determines, when FDM locations are deleted, whether FDM deletes the location directories from the `Inbox`.
Memo Item Filter With Category and Period	Select 0–None to allow memo and audit attachments added in any category or period to be displayed in all categories and periods in Process Explorer for that location. Select 1–Category_Period so that memos and audit attachments will only be displayed in the category and period POV in which they were added.
Excluded File Upload Wild Cards	Use this option to specify file extensions that cannot be uploaded. Enter "*.*" to disallow all file uploads.

Configuring Integration Settings

Integration settings determine the levels of integration between FDM and the target system and control which integration points are active, the type of integration processes to run, and the default values that relate to integration actions.

➤ To configure integration settings:

1 **Open the Integration Settings dialog box.**

- **From Workbench**—Select **Tools** > **Integration Settings**.
- **From the Web client**—Select **Administration** > **Integration Settings**.

2 **From Options, select an integration setting.**

3 **Select the option settings.**

The option settings are located on the bottom half of the Integration Settings dialog box. The integration setting that is selected in Options determines which options are displayed.

Three categories of integration options:

- Connection
- Integration Point
- Default Load

Connection Options

- **Application Name**—Specifies the name of the target application to integrate with FDM
- **Logon Method**—Controls the method (Unified or Global) that FDM uses to log on to the target system.
  - **1–Unified**—Uses the FDM username and password.
  - **2–Global**—Uses the username and password specified in the Global Logon Information option. Thus, all users use the same user name and password.

    Note: A machine-specific user name and password, specified in Workbench, overrides the method specified in Logon Method.
- **Global Logon Information**—Used in conjunction with the Logon Method option. When Logon Method is set to Global, the global user name and password are used to log on to the target system. Separate the user name and password with a semicolon (*UserID;Password*).
- **Use SSO**—When set to On, enables Hyperion Single-Sign-On (for passing a token to FDM).
- **Sticky Server**—When set to On, forces use of one server for Load, Consolidate, and Check (required for Financial Management application server cache delay). When enabled, each individual Financial Management application server in the Financial Management cluster must be registered within the Financial Management Win32 client on the FDMapplication servers.

Integration Point Options

- **Enable Load**—When selected, forces FDM, after export, to load the export file (`*.dat` file) into the target system. If this option is cleared, FDM creates, but does not load, the export file.
- **Enable Consolidation**—After a data load, runs a consolidation in the target system.
- **Enable Validation Report**—When selected, forces FDM, after the data load and consolidate process, to run a validation report (if one is assigned to the location).
- **All Periods**—Controls the periods that can be viewed during integration with the target system. Select this option to show all periods. Clear this option to show only base periods.

Default Load Options

- **Consolidation Type**
- **Load Method**
- **Load Operation**
- **List2**—Not used
- **List3**—Not used
- **Calc**
- **Accumulate in File**
- **File as Ownership Data**
- **Status 3**—Not used.
- **Data Protection Switch**—Enables FDM to protect target-system data from being overwritten during data imports; is based on a specified protection value. Use this option when data is entered into the target system through a method other than FDM.
  - **Protection Value 1**—Stores the value used with Data Protection Switch (the value that is not to be overwritten).
  - **Protection Operator**—Stores an operator (= or <>) that determines what the items equal to the value stored in Protection Value 1 are to be protected.
- **Enable Line Item Detail Loading**—Enables loading of line-item detail to Financial Management.
- **Line Item Detail Load Type**—Sets the line-item detail loading type (Summarized or Detail).
- **Force Calculate**—Runs the default calculation call prior to a consolidation run.

Setting Configuration Options

Configuration options set database performance and tuning parameters. Configuration options should be used when network infrastructure requires tuning of the database engine I/O activity.

➤ To configure configuration options:

1 **Open the Configuration Options dialog box:**

- **From Workbench:** Select **Tools** > **Configuration Settings**.
- **From the Web client:** Select **Administration** > **Configuration Options**.

2 **From Options, select a configuration option.**

Details of each of the thirty-one configuration options are listed in the following topics:

- General Options
- SQL Tuning Configuration Options
- Oracle Tuning Configuration Options
- Other General Configuration Options

3 **Select the option settings.**

The option settings are located on the bottom half of the Configuration Options dialog box. The configuration option that is selected in Options determines which option settings are displayed.

General Options

- **DB Version**—Shows the version number of the installed FDM software.
- **SQL Query Date Format Mask**—The SQL format used for querying dates. Default is YYYY-MM-DD. Set this to match the regional format used by your SQL installation.
- **Decimal Replacement**—Enables FDM to specify the delimiter that is used when `BCP` files are loaded into the RDBMS server. Oracle recommends that servers that contain FDM component installs are all run under the same language.
- **Insert Batch Size**—Enables FDM to specify, for file import, the number of rows to insert into the FDM database. Select this option only for FDM locations that use the SQL Insert load type.
- **Total No. Data Segments**—Used to define the default number of segments to be created in the RDBMS database.
- **Number of Locations per Page**—Select as few as two locations up to ten locations that are displayed simultaneously in the Timeline Viewer.

SQL Tuning Configuration Options (SQL Server databases only)

- **Data Seg Table File Group**—Enables FDM to specify that the RDBMS server use a data segment-table-file group other than the RDBMS primary group.
- **Data Map Seg Table File Group**—Enables FDM to specify that the RDBMS server use an alternate file group other than the primary group for storing the Data Map Segment tables.
- **Work Table File Group**—Enables FDM to specify that the RDBMS server use a worktable file group other than the RDBMS primary group.
- **Work Table Index File Group**—Enables FDM to specify that the RDBMS server use a worktable index file group other than the RDBMS primary group.

Oracle Tuning Configuration Options (Oracle databases only)

- **Oracle Seg Insert Hint**—Defines how SQL loader loads data to the tDataSeg tables. The default setting, Append, enables FDM to append data to segment tables without finding available space on the table, as required by the Insert setting.
- **Oracle Segment Table Logging Value**—Used, for segment tables, to turn logging on or off.
- **Oracle Work TableSpaceName**—Specifies the tablespace name. The default is Users.

- **Oracle Work Storage Clause**—Specifies storage defaults for the Work tablespace. These defaults override the system defaults and become the defaults for objects created in the specified tablespace.
- **Oracle Work Pct Free**—In regard to the Work tablespace, specifies the portion of the data block that is reserved for later updates to the rows of the block. The default for PCTFREE is 10 percent. You can use any integer between 0 and 99, inclusive, provided that the sum of PCTFREE and PCTUSED does not exceed 100.
- **Oracle Work Init Trans**—Specifies the initial number of transaction entries that are allocated within each data block that is allocated to the table. The value can range from 1 to 255. Each transaction that updates a block requires a transaction entry. The size of a transaction entry depends on the operating system. This parameter ensures that a minimum number of concurrent transactions can update the block and helps to avoid the overhead of dynamically allocating a transaction entry.
- **Oracle Work Max Trans**—Specifies the maximum number of concurrent transactions (1–255) that can update a data block that is allocated to the work table.
- **Oracle Work Table Index TableSpaceName**—Defines the FDM work table index tablespace name. The default is Users.
- **Oracle Work Table Bitmap Index Switch**—On/Off switch used to toggle the work table bitmap index. This option should be ON for Oracle 9i and OFF Oracle 10g and 11g..
- **Oracle Data Map Seg TableSpaceName**—Defines the FDM Data Map Seg table tablespace name. The default is Users.
- **Oracle Data Map Seg Storage Clause**—Specifies storage defaults for the Data Map Seg tablespace. These defaults override the system defaults and become the defaults for objects created in the specified tablespace.
- **Oracle Data Map Seg Pct Free**—In regard to the Data Map Seg table, specifies the portion of the data block that is reserved for later updates to the rows of the block. The default for PCTFREE is 10 percent. You can use any integer between 0 and 99, inclusive, provided that the sum of PCTFREE and PCTUSED does not exceed 100.
- **Oracle Data Map Seg Init Trans**—Specifies the initial number of transaction entries (1–255) that are allocated within each data block that is allocated to the data map tablespaces. Each transaction that updates a block requires a transaction entry in the block. The size of a transaction entry depends on the operating system. This parameter ensures that a minimum number of concurrent transactions can update the block, and helps to avoid the overhead of dynamically allocating a transaction entry.
- **Oracle Data Map Seg Max Trans**—Specifies the maximum number of concurrent transactions (1–255) that can update a data block that is allocated to the Data Map Seg table.
- **Oracle Data Seg TableSpaceName**—Specifies the tablespace name for FDM Data Seg tables. The default is Users.
- **Oracle Data Seg Storage Clause**—Specifies storage defaults for the Data Seg tablespace. These defaults override the system defaults and become the defaults for objects that are created in the specified tablespace.

- **Oracle Data Seg Pct Free**—In regards to Data Seg tables, specifies the portion of the data block that is reserved for later updates to the rows of the block. The default for PCTFREE is 10 percent. You can use any integer between 0 and 99, inclusive, provided that the sum of PCTFREE and PCTUSED does not exceed 100.
- **Oracle Data Seg Init Trans**—Specifies the initial number of transaction entries (1–255) that are allocated within each data block that is allocated to the Data Seg tables. Each transaction that updates a block requires a transaction entry in the block. The size of a transaction entry depends on the operating system. This parameter ensures that a minimum number of concurrent transactions can update the block and helps to avoid the overhead of dynamically allocating a transaction entry.
- **Oracle Data Seg Max Trans**—Specifies the maximum number of concurrent transactions (1–255) that can update a data block that is allocated to the Data Seg table.

Other General Configuration Options

- **Allows custom description in period**—Enables the administrator to create custom period descriptions.
- **Lookup Indicator: Target System Data**—Specifies the terminating character for use by validation rules that pull data from the target systems. The default value is the pipe character (|).
- **Lookup Indicator: FDM Converted Data**—Specifies the terminating character for use by validation rules that pull data from FDM converted data. The default value is the grave accent character (`).
- **Lookup Indicator: FDM Source Data**—Specifies the terminating character for use by validation rules that pull data from FDM source data. The default value is the tilde character (~).

5 Assigning FDM Security and Managing Users

In This Chapter

Security

Within FDM, you can secure almost every menu item, button, screen, and report. Security features are restricted to administrators. FDM supports two levels of security:

- **Application object security**—Governs access to forms and form controls
- **Location security**—Governs access to locations

Security levels are applied to users and application objects, and the levels assigned to users and objects are compared at runtime. If a user is assigned a level that equals or exceeds the level assigned to the object that the user is trying to access, the object is available to the user. Object-maintenance tasks are performed in the Web client.

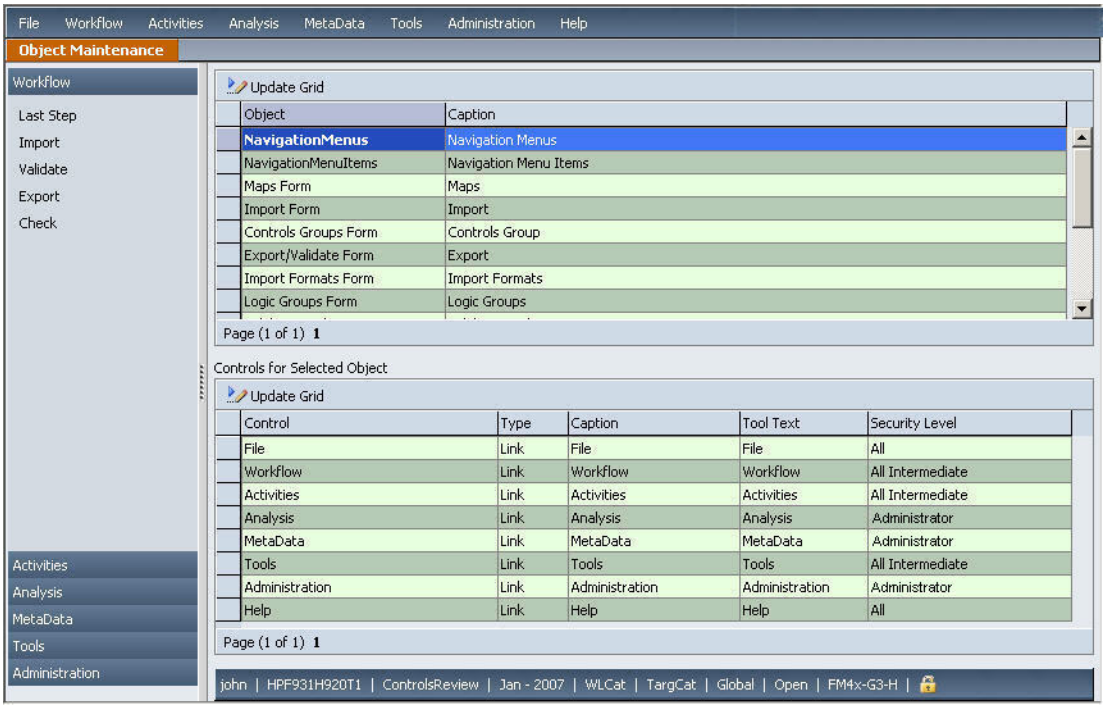
Assigning Application Object Security

Use the Object Maintenance screen in the FDM Web client to assign minimum security rights for FDM application objects. The Object Maintenance form lists the application objects and their dependant controls.

➤ To assign or change application object security:

1 **From the FDM Web client, select Administration > Object Maintenance.**

The Object Maintenance screen is displayed.



2 From the top grid, select an application object.

3 From the bottom grid, select a control for the object.

4 From **Security Level**, select a security level.

Location Security

Location security for FDM is configured by using the User Maintenance options.

Managing Users

Access the User Maintenance screen from within the Web client. The User Maintenance form enables you to perform the following tasks:

- Add users
- Set user rights
- Delete users

Adding Users

➤ To add users:

1 From within the Web client, select **Administration > User Maintenance**.

The User Maintenance form is displayed.

2 Click **New User**.

The New User dialog box is displayed.

3 Select a provisioned user from the drop-down list.

4 Click **OK**.

The User Rights dialog box is displayed. See "Setting User Application Rights" on page 53 for the procedure to set options in this dialog box.

Setting User Application Rights

Application rights govern access to forms and form controls. Administrators have access to every location, form, and control. Application security associates each user with a security level that is evaluated against each application object security level. This process determines which components of the user interface each user can access.

You set application rights when provisioning users in Oracle's Hyperion® Shared Services. See Chapter 10 "Migration Considerations" in the *FDM Configuration Guide* for the detailed information on provisioning users.

Setting User Location Rights

Every user must be assigned at least one location. To users assigned a user level of Administrator, all locations are available. To users assigned user-level application rights, only locations assigned to the users' profiles are available.

➤ To assign locations to users:

1 From the Web client, select **Administration > User Maintenance**.

The User Rights screen is displayed.

2 Double-click a user.

3 Click **Add**.

4 From the new row, double-click the **Location** column, and select a location.

5 **Optional:** To set the selected location as the user's default, select **Default**. .

6 Click **Update Grid**.

Note: Each user can be assigned multiple locations, but every user must be assigned one default location. Select Access to All Locations to allow non-administrator users to access all locations.

Location Override Security

The Security Level field enables administrators to configure alternate application rights for assigned locations. For a location, the level specified in the Security Level field, rather than the level specified for a user, is used.

For example, a user with default application rights of Intermediate-7 is assigned three locations. If one of the locations has a security level of Intermediate-5, when the POV is set to the Intermediate-5 location, the user's application rights are set to Intermediate-5.

Email Address

The Email Address field stores the user's e-mail address. You can create a script to e-mail users on the loading status of FDM locations.

Deleting Users

➤ To delete users:

1 From within the Web client, select **Administration > User Maintenance**.

The User Maintenance screen is displayed.

2 Next to a user name, click **Delete.**

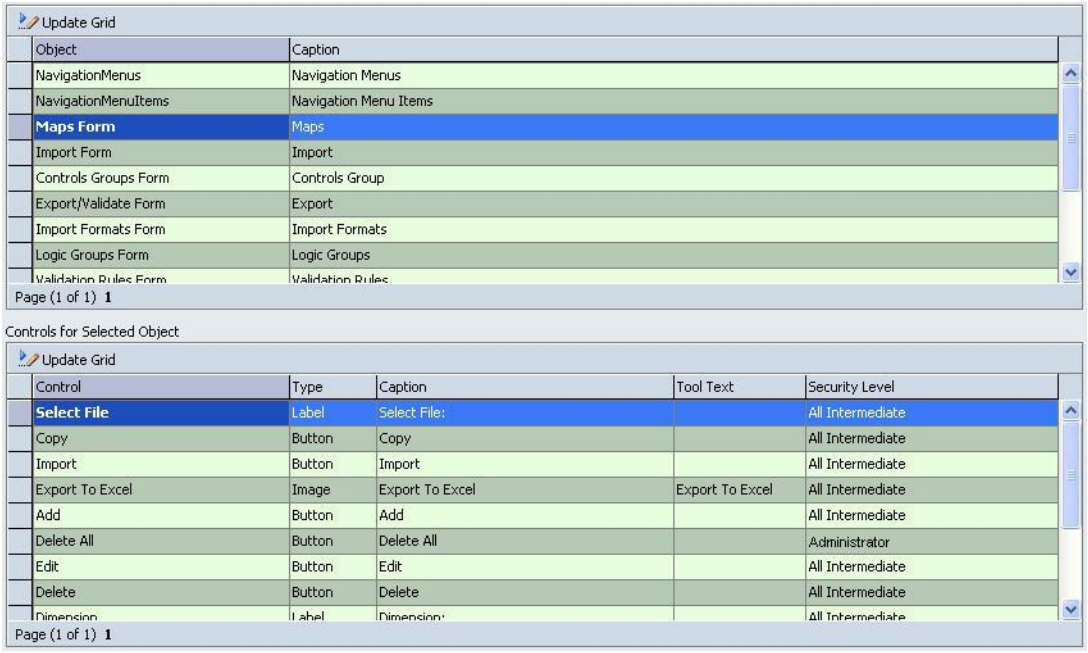
3 Click **Update Grid**.

Base Entities when Using Hyperion Enterprise

If a user connecting to Hyperion Enterprise does not have access to the base name selected when setting up the target system integration, the integration will not work. This issue is more likely to occur when using unified security in Hyperion Enterprise, where, by design, users won't have access to all base entities. In this case, Oracle recommends creating a base entity solely for FDM access and not for data entry.

Object Maintenance

Thirteen user levels are available: Administrator, Intermediate-2 through Intermediate-9, All Intermediate, Basic Reviewer and Submitter, Basic Reviewer, and All



Application security includes the application object, which controls the form that is being secured, and the controls for each object (button, selection). In the previous figure, the main FDM screen is being secured. The Locations link is set to Administrator users.

The administrator determines security levels for each object. Typically, different types of users access FDM, including the following:

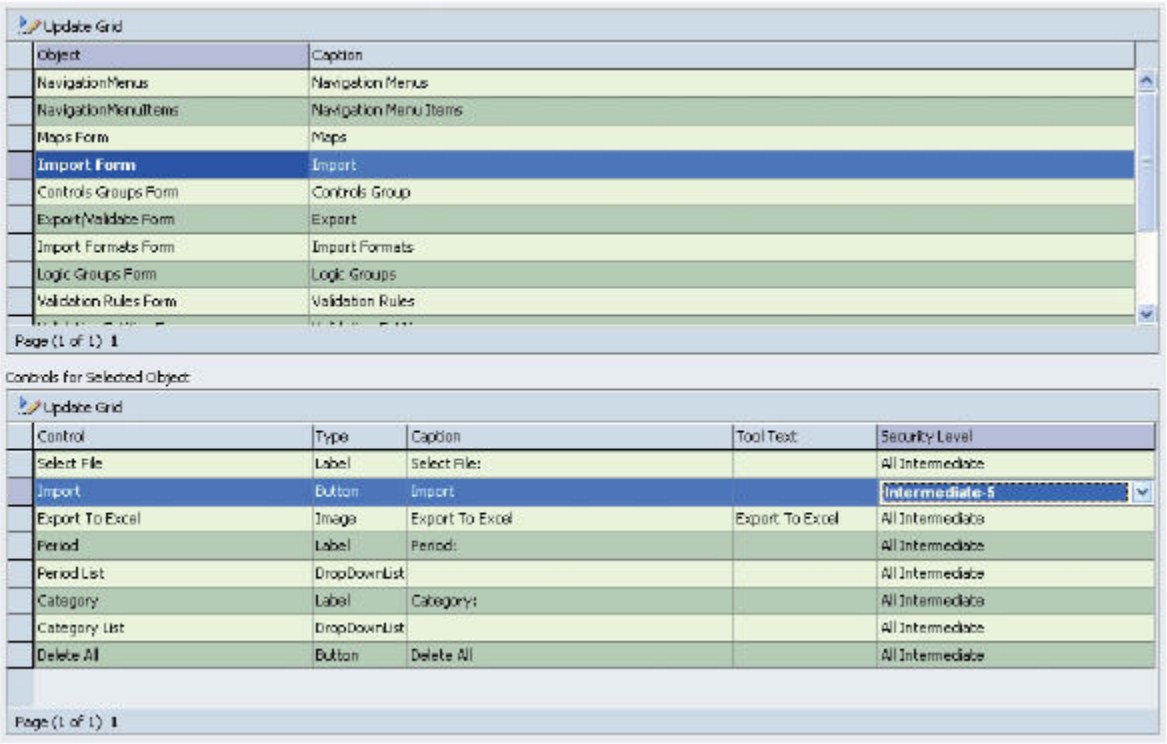
- The administrator controls the locations and import formats and other items.
- The intermediate user is responsible for loading data to the target system.
- The view user, such as an auditor, is one that should not change data.
- The controls review users are those who need only to access the Process Explorer screen.

When users are assigned a user level in the User Maintenance portion of security, they have access to every object assigned that level and higher. For example, if a user is assigned Intermediate-7, the following conditions apply:

- The user can access each object that has Intermediate-7 through Intermediate-9, and All.
- The user cannot access objects assigned to Administrator or to Intermediate-2 through Intermediate-7 levels.

Example

In the figure that follows, the Import Form Application object is shown. The import button access level was changed to Intermediate-5 which enables the user “Texas” with Intermediate-5 security level to import the trial balance but restricts user “auditor” to only viewing data.


Update Grid
Object
Caption
NavigationMenus
Navigation Menus
NavigationMenuItems
Navigation Menu Items
Maps Form
Maps
Import Form
Import
Controls Groups Form
Controls Group
Export/Validate Form
Export
Import Formats Form
Import Formats
Logic Groups Form
Logic Groups
Validation Rules Form
Validation Rules
Page (1 of 1) 1
Controls for Selected Object:
Update Grid
Control
Type
Caption
Tool Text
Security Level
Select File
Label
Select File:
All Intermediate
Import
Button
Import
Intermediate-5
Export To Excel
Image
Export To Excel
Export To Excel
All Intermediate
Period
Label
Period:
All Intermediate
Period List
DropDownList
All Intermediate
Category
Label
Category:
All Intermediate
Category List
DropDownList
All Intermediate
Delete All
Button
Delete All
All Intermediate
Page (1 of 1) 1

6 Activating and Labeling Dimensions

In This Chapter

Activating Mappable Dimensions

You activate dimensions by using the Web client. You select MetaData > Dimensions to configure the target-system mappable dimensions that FDM loads. For Hyperion Enterprise, standard practice is to activate only the Account and Entity dimensions. For other target systems, every FDM dimension can be activated.

The Dimensions screen contains five columns:

- Dimension
- Alias
- Calc Sequence
- Enabled
- Use as Lookup

Dimension

The Dimension column contains all predefined FDM dimensions. The dimension names cannot be modified.

Alias

In the Alias column, you can enter an alias (alternate name) for each dimension. FDM displays aliases in the list boxes within the mapping tables and also in the Validate and Export screens.

Calc Sequence

You uses the Calc Sequence column to override the default dimension calculation order. You override default order when you perform conditional mapping. The default order is the order in which the accounts are displayed on the Dimensions screen.

Enabled

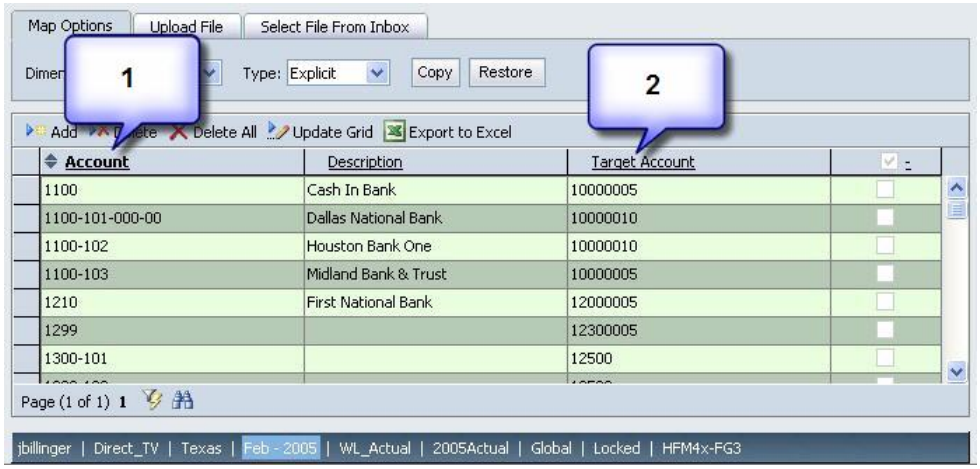
You select Enabled to activate the selected dimension. You can import, map, and load data into activated dimensions.

Use As Lookup

You use the Use as Lookup column to select a dimension for a custom lookup table. When Use as Lookup is selected, the Enabled field cannot be used. The custom lookup dimension is used only for custom scripting.

Labeling Dimensions

You select MetaData > Dimension Labels to assign source and target labels to FDM dimensions. You cannot modify dimension names, only dimension labels. Dimension labels are displayed in the mapping tables and also in the Import Formats screen.



1. Source label
2. Target label

7 Working with Control Tables

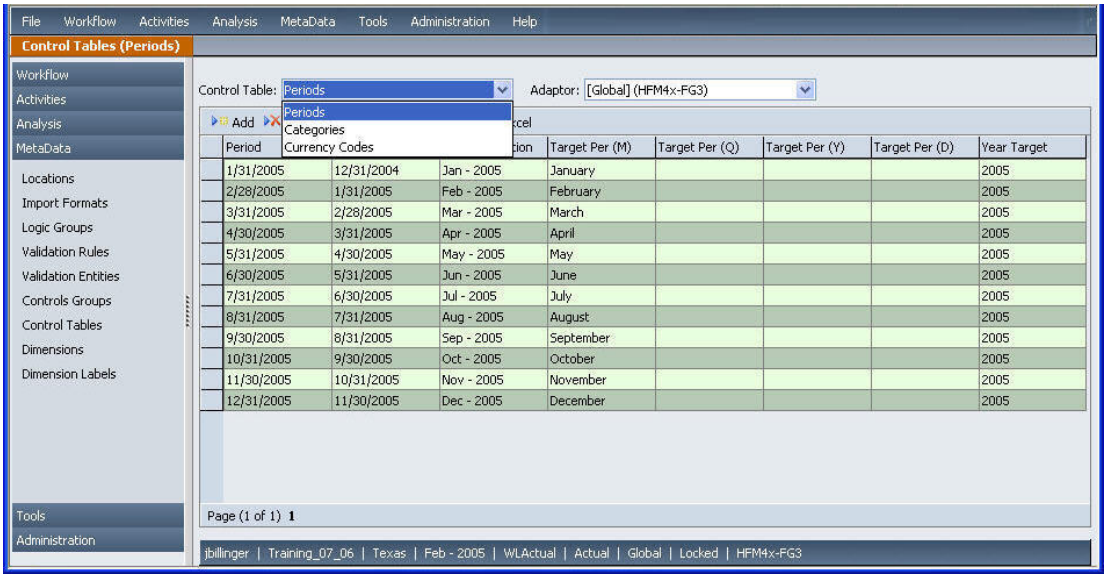
In This Chapter

Overview of Control Tables

Control tables enable system administrators to control the system options and POV values that users can select. For each target system, FDM uses two control tables (Periods and Categories). A third table, Currencies, is global for all target systems. You use the Web client to configure and maintain control tables.

Setting Up Periods

You use the Periods control table to define the fiscal periods that can be used by FDM.



The values in the Periods table identify the periods that can be selected from the period POV. Each FDM period is associated with a target system period. If the FDM period changes, where data is loaded in the target system changes.

- **Period**—Date value that is stored in the database during the trial-balance load process and that becomes part of a key that identifies a set of trial-balance records
- **Prior Date Key**—Prior fiscal period key that is used during export to determine whether a $0.00 entry must be made (to prevent ghosting) in Hyperion Enterprise versions 4.3 or earlier (the entry must be made if YTD values were loaded into a periodic category)
- **Text Description**—Text-based description of the date key (the format of the text description is controlled by the Date Format Mask system configuration. The format defaults to months but can be changed to enable loading of more than twelve periods)

The "Allows custom description in period" option enables administrators to create custom period descriptions that override the default values. Text override option restrictions apply to Text Description Override (value cannot resolve to a date) and Period Key Date Format Mask (option must be set to *MM-DD-YYYY*).

The options identify the target period to which data is loaded and represent months, quarters, years, or days, respectively:

- **Target (M)**
- **Target (Q)**
- **Target (Y)**
- **Target (D)**

➤ To insert periods into tables:

1 From within the Web client, select **Metadata > Control Tables**.

The Control Tables screen is displayed.

2 From **Control Table**, select **Periods**.

3 Select **Add**.

4 Select the cell to which to add or update information.

5 In the **Period** column, select a target period from the drop-down list.

To indicate that the change is pending,  is displayed left of the row.

6 Click **Update Grid**.

The change-pending icon is removed.

➤ To delete periods from tables:

1 Select a row.

2 Click **Delete**.

 is displayed left of the row, indicating that a deletion is pending.

3 Click **Update Grid**.

➤ To cancel pending changes:

1 Left of the row, click  or  .

2 Select **Cancel Row Changes**.

Setting Up Periods Control Tables for Override Adapters

The procedure "Setting Up Periods" on page 59 illustrates a setup that uses the global integration adapter (default). A Periods control table must be built for each override adapter used in the application.

➤ To set up control tables for override adapters:

1 Select **Metadata > Control Tables.**

The Control Tables screen is displayed.

2 From **Control Table**, select **Periods**.

3 From **Adapters**, select an integration adapter.

A new Periods control table is displayed. Only the Period column is populated.

4 Configure the table to integrate with the override target application.

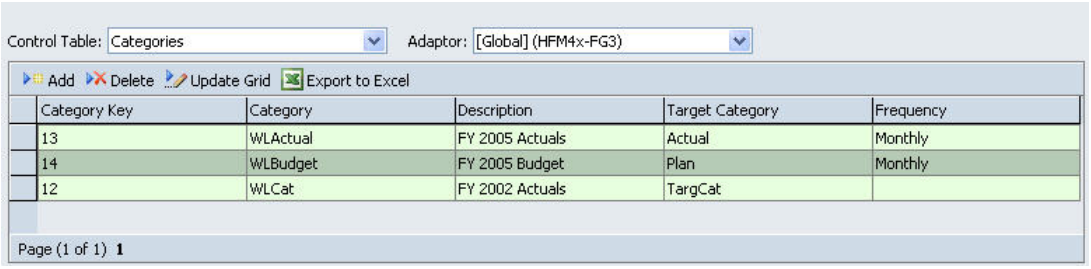
Note: Override adapters that are not in the Adapter list, have not been imported to and configured in the FDM application. See the *FDM Configuration Guide* for information about adding adapters to applications.

Data Maintenance Considerations

Deleted periods are removed from all categories in all locations. Before deletion, the system must be locked. After deletion, the database should be compacted. Database compaction is performed by the DB administrator.

Setting Up Categories

The Categories table contains definitions of data categories (containers into which data can be loaded).



The values in the table identify the categories that can be selected from the category POV link. Each FDM category is associated with a target system category. Changing the FDM category changes where data is loaded in the target system.

- **Category Key**—Numeric keys that are stored during the trial-balance load process and generated by FDM (each value identifies a set of trial-balance records)
- **Category**—The FDM category name
- **Description**—Description of the FDM category
- **Target Category**—The target system category to where the data is loaded
- **Frequency**—The category frequency

➤ To insert categories into tables:

1 From within the Web client, select **Metadata > Control Tables.**

The Control Tables screen is displayed.

2 From **Control Table**, select **Categories**.

3 Select **Add**.

4 Select the cell to which to add or update information.

5 In the **Category** column, select a target category from the drop-down list.

 is displayed left of the row to indicate a pending change.

6 Click **Update Grid**.

The change-pending icon is removed.

➤ To delete categories from tables:

1 Select a row.

2 Click **Delete**.

 is displayed left of the row, indicating that a deletion is pending.

3 Click **Update Grid**.

The deleted rows are removed.

➤ To cancel pending changes:

1 Left of the row, click  or  .

2 Select **Cancel Row Changes**.

Setting Up Category Control Tables for Override Adapters

➤ To set up control tables for override adapters:

1 Select **Metadata > Control Tables.**

The Control Tables screen is displayed.

2 From **Control Table**, select **Categories.**

3 From **Adapters**, select an integration adapter.

A new Categories control table is displayed. Only the Periods column is populated.

4 Configure the table to integrate with the override target application. For information about setting up categories, see "Setting Up Categories" on page 61.

Note: If the override adapter you wish to select is not in the Adapter list, then the adapter has not been imported and configured in the FDM application. See the *FDM Configuration Guide* for information about adding adapters to the application.

Data Maintenance Considerations

Deleted periods are removed from all categories in all locations. Before deletion, the system must be locked. After deletion, the database should be compacted. Database compaction is performed by the DB administrator.

Setting up Currency Codes

The Currency Codes table contains a list of currency codes. Currency codes are assigned to locations and displayed on reports. Because currency codes are used only for notation, they do not impact calculations.

- **Currency Code**—The short description of the currency
- **Description**—A detailed description of the currency

➤ To insert currencies into the Currency Codes table:

1 From within the Web client, select **Metadata > Control Tables.**

The Control Tables screen is displayed.

2 From **Control Table**, select **Currencies.**

3 Click **Add.**

4 In the new row, provide the currency information.

5 Click **Update Grid**.

8 Locations

In This Chapter

Overview

Note: This document details only FDM Data Load locations. See Chapter 12, "Setting Up Financial Controls" for information on Controls Review locations. Whenever a reference is made to a FDM location assume it is pertaining to a Data Load location.

A FDM location is a reporting unit responsible for submitting source data into the target system. Typically, there is one FDM data load location for each source file loaded to the target system.

System administrators assign end users access to one or more data load locations; one location is designated as the default location . When a user logs on to FDM, the POV is set to the user's default location.

FDM Location Types

FDM contains two types of locations, data load and controls review:

- **Data load**—Locations to where source system data is loaded. Data load locations can be assigned financial controls. Within the controls structure, data load locations can be children, but not parents. Data load locations are designated by the  icon.
- **Controls review**—Locations assigned only to financial controls. No data is loaded to or from a controls review location. Within the controls structure, controls review locations are typically parents. However, a controls review locations can be a child of another controls review location. Controls review locations are designated by the  icon.

Creating Data Load Locations

When a FDM application is created, one sample financial controls location named *ControlsReview*, and one sample data load location, named *Sample,* is created.

➤ To create data load locations :

1 From within the Web client, select **Metadata > Locations**.

The Locations screen is displayed.

2 Right-click a **Controls Review** location, and select **Add Child**.

3 From **Location Type**, select **Data Load**.

4 For the new data load location, enter a name.

5 Click **OK**.

Location names are restricted to alpha and numeric characters with no spaces and cannot be changed. Therefore, it is prudent to develop a naming convention before you begin.

Creating Location Directories

FDM creates a directory in the `Inbox` for each data load location. If FDM creates directories for all locations (data load and controls review), the Create Location Folder option in Application Settings is selected.

Location Attributes

The Location screen contains four tabs—General, Workflow Behaviors, Financial Controls, and Integration Options.

General Tab

- **Description**—Information about the location useful for tracking GL systems and highlighting system-related information. The default description is name of the location.

- **Currency**—The currency that is loaded into the location; used only for identification purposes in some FDM reports and in some forms that display data. Because FDM does not perform currency translation, the currency code does not impact calculations.
- **Parent Location**—The parent assigned to the location. The location uses the mapping table of its parent. Multiple locations can share a parent. This feature is useful when multiple locations use one chart of accounts. Changes to a child or parent mapping table are applied to all child and parent locations.
- **Load Type**—SQL Insert or Bulk Insert. SQL Insert is the default load method and should be used for locations that load smaller files (6000 records or less). Bulk Insert is more efficient for loading larger files.
- **Seq Map**—Option that enables wildcard maps to be sorted and processed by numeric values rather than alphabetically (the default). Use this option for locations that have complicated conversion rules that are predicated upon processing order (For example, a source account is included in multiple wildcard maps; therefore, the sort order of the map is important to determine by which map the account must be processed). Use this option only for locations that use complicated translation rules. When you select Seq Map, a new sequencing field is displayed in the Maps form.
- **Group Tag**—A field in which, by entering values, you can group multiple locations. In some process monitor reports, grouped locations are displayed in their grouping order rather than alphabetically.
- **Data Value**—An extra dimension that is used only for integration with multi-dimension target systems. This dimension is associated with a data load location. When FDM creates the load file, the dimension value is entered for every data line that is loaded by the location. For example, the Data Value dimension is associated with the Value dimension in Financial Management. By default, if no value is entered in this field, for integration with Financial Management, the Data Value <Entity Currency> is the default value.
- **Target Adapter (Adapter by Location)**—The option that, when left at [None], directs the location to use the global adapter defined during the adapter setup process. Any selection here overrides the global adapter for the location. This setting enables FDM to integrate with multiple target applications. Each location can load data to a unique target application.

Note: When you are working in FDM screens that browse for target categories, FDM browses by using the adapter selected for the current location. Unless the target systems contain common member values for category and period, control tables (Categories and Periods) are not partitioned by adapter and require an entry for each adapter.

Workflow Behaviors Tab

- **Import Format**—To enable loading of a source data file to a data load location, the location must be assigned an import format. The import format defines the structure of the source file. When a location is created, you do not need to assign an import format. You can set up all reporting locations without knowing all source file structures.

- **Logic Group**—Logic groups contain one or more logic accounts that are generated after a source file is loaded. Logic accounts are calculated accounts that are derived from the source data.
- **Validation Rules**—A validation rule group contains a set of validation rules that are displayed on the validation report. After completing a data load,FDM uses the rules to retrieve target system data from the target application.
- **Validation Entities**—A validation entity group contains a set of target system Validation entities that can be consolidated and displayed in the validation report (optional).

Financial Controls Tab

See Chapter 12, "Setting Up Financial Controls" for information on Controls Review locations.

Integration Options Tab

This screen shows available custom adapter options for the current location.

Organizational Changes

You can add, delete, and move locations. To record the organization structure at the time of submission, FDM logs the organization structure for each period in the tDataSubmitHierarchy table.

Best Practices with Financial Management Phased Submissions

Using the Phased Submissions feature in Financial Management when loading data from FDM can cause issues with audit trails and also result in failed loads. This is because after phase one is submitted, any subsequent loads of the same data will fail because of locked intersections. In addition, reloading of data in FDM to the same location will erase the audit trail of the previous export to Financial Management.

In order to preserve the audit trail and to avoid failed exports, Oracle recommends that you create a separate FDM location for each HFM phase. For example, a location named Texas would have locations named Texas1, Texas2, and Texas3 to load Texas for Phase 1, Phase 2, and Phase 3 respectively. Oracle also recommends separating the source data and the FDM maps by these phased locations.

9 Creating Logic Accounts

In This Chapter

Overview of Logic Accounts

Logic accounts are dynamically generated accounts that are used to calculate supplemental values that are not provided in source files. Like all other source accounts, logic accounts can be mapped and loaded into target systems. Logic accounts are used for various functions:

- **Statistical loading**—Map one source account to multiple target accounts
- **Conditional mapping**—Map a source account based on its value
- **Arithmetic mapping**—Perform arithmetic operations on source values

Logic accounts are created in FDM; therefore, detailed audit trails on logic account values are not available.

Creating Logic Groups

The first step in creating logic accounts is to create a logic group. The logic group is then assigned to one or more locations. When a source file is loaded to a location, logic accounts are generated.

Logic groups must be defined as simple or complex. Simple logic groups allow logic items to be derived only from the source account dimension. Complex logic groups allow logic items to be derived from any combination of dimensions.

➤ To create logic groups:

1 From the Web client, select **MetaData > Logic Groups.**

The Logic Groups screen is displayed.

2 Select the **Add** button above the top grid.

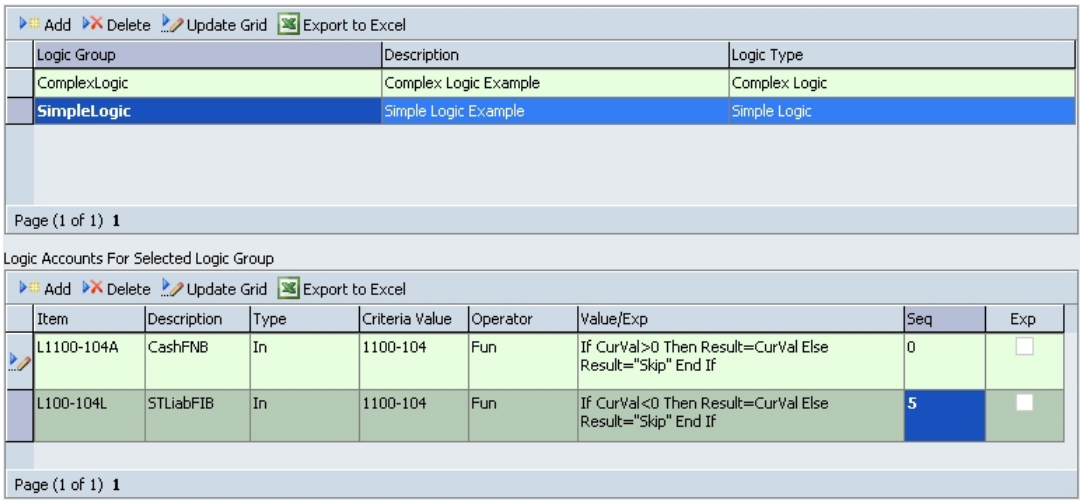
A new row is added to the grid.

3 Under **Logic Group**, enter a unique name.

4 **Optional:** Under **Description,** enter descriptive text.

5 Under **Logic Type,** select **Simple Logic** or **Complex Logic.**

6 Click **Update Grid.**



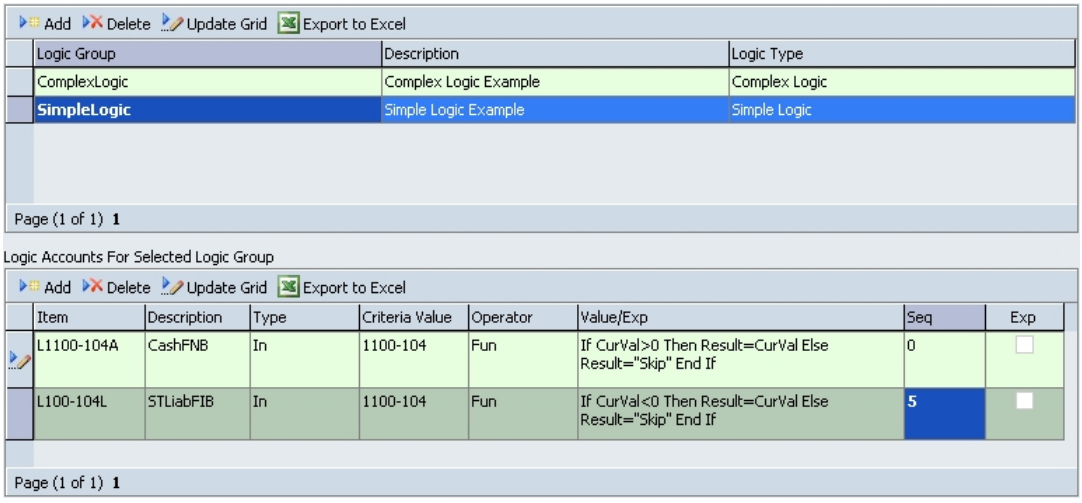
Creating Accounts Within Simple Logic Groups

Within simple logic groups, you can create individual logic accounts.

➤ To create accounts within simple logic groups:

1 **In the Web client, select the group within which you want to create an account.**

The logic accounts currently contained in with the selected logic group are listed.



2 Click **Add.**

3 **Within the fields, provide the requested information.**

For information about the fields, see the following topics:

Logic Group Fields

Logic accounts consist of the following fields:

- Item
- Description
- Type
- Criteria Value
- Operator
- Value/Exp
- Seq
- Exp

Item

In the Item field, you name the logic account. The name is displayed in the Account field on the Import screen. It is recommended to precede the names of logic accounts with an *L*, to distinguish logic accounts from standard source accounts. Logic accounts that are loaded to target systems must be mapped to target accounts.

Description

The description that you enter in the Description field is displayed in the Account Description field on the Import form.

Type and Criteria Value

The operator in the Type field works with the source account that is specified in the Criteria Value field to determine from which source accounts the logic account is derived. One logic account can be derived from multiple source accounts.

Operator

Specifying Source Accounts

Between operator—Used when a range of source accounts are specified in the Criteria Value field. Separate the accounts that specify the range by a comma.

Type Field	Criteria Value Field
Between	1000,1999

Like operator—Used when the source accounts in the Criteria Value field contain wildcard characters. Use question marks (?) as placeholders and asterisks (*) to signify indeterminate numbers of characters.

Type Field	Criteria Value Field
Like	1??0
Like	10*

In operator—Used to include one source account or a list of non-sequential source accounts.

Type Field	Criteria Value Field
In	1000
In	1000,1005,2001

Creating Summarized Logic Accounts

By default, a logic account is created for each center in the trial balance. For example, if the Criteria Value field is 12300, the result would be a logic account created for each source center that is associated with account 12300.

You can create a logic account that summarizes multiple source centers by, in the Criteria Value field, placing a semicolon after the account name and entering the number that identifies the number of characters that you want to group by.

For example, if the value in the Criteria Value field is 12300;4, the result would be a summarized account that includes all source centers that have the same first four characters in the source center name. The source center assigned to account 12300 is the four characters that start at position 1. In addition, if the value in the Criteria Value field is 12300;3;4, the result would be a summarized account that includes all source centers that have the same three characters in the source center, starting at position 4. The source center assigned to account 12300 is the three characters that start at position 4.

You can create a logic account that summarizes all source centers by, in the Criteria Value field, placing a semicolon after the account name and entering a text value. This hard-coded text value becomes the center for the summarized logic account. For example, if the value in the Criteria Value field is 12300;Dept100, the result would be a summarized account that includes all source centers. The source center assigned to account 12300 is Dept100.

Value/Exp

To perform calculations and, thereby, to derive values for a logic account, you select an operator, from the Operator field, to work with the Value/Exp value.

Seq

This field specifies the order in which the logic accounts are processed. Order specification enables one logic account to be used by another logic account, provided that the dependant account is processed first.

Exp

A Yes-No switch determines whether a logic account is considered an export account and subjected to the conversion table validation process. If the switch is set to Yes, the logic account must be mapped in the Account Conversion table.

Operator and Value/Exp

Numeric Operators

Use a numeric operator to perform simple mathematical calculations.

* Multiplication

- Subtraction

+ Addition

/ Division

NA (no operator)

If a numeric operator other than NA is specified in the Operator field, a numeric value must be entered in the Value/Exp field. If NA is specified, the source accounts that are specified in the Criteria Value field are summed.

Function Operators and Custom Logic Functions

Function operators are used to execute custom logic functions, which are defined in the Value/Exp field. To write a function, double-click in the Value/Exp field to display the logic function editor.

Logic functions are normally used for conditional mapping and other complex operations that involve multiple source accounts. Logic functions permit the use of variable and If statements and thus enable conditional mapping. The FDM Lookup function can also be used within a logic function.

The following function parameters can be used in a function:

- **CurVal**—Value of the logic account operation
- **StrLocation**—Active location name
- **strCenter**—Logic account center
- **strCatKey**—Active FDM category key (not name)
- **strPerKey**—Active FDM period

Function Return Values

The result of a logic function must be assigned to the keyword `RESULT`.

```
RESULT = CURVAL + (|810| * .5)
```

If no return value is assigned to the RESULT keyword, the logic engine sets the value of RESULT to zero. Thus, the calculation is skipped, and the logic account is not created.

The following function uses the CURVAL parameter to assign the result of the logic account calculation to RESULT, if the logic account calculation returns a value greater than zero.

```
If CURVAL > 0 Then
    RESULT = CURVAL
Else
    Result=”Skip”
End If
```

The greater-than-zero example illustrates conditional mapping. That is, if the source account specified in the Criteria Value column is less than zero, the logic account is not created (because the keyword “Skip” is used).

The following function uses the CURVAL parameter to assign the result of the logic account calculation to RESULT, if the logic account calculation returns a value less than zero.

```
If CURVAL < 0 Then
    RESULT = CURVAL
Else
    Result=”Skip”
End If
```

The following function assigns the result of the logic account calculation to RESULT only if 10 is the active FDM category key.

```
If strCatKey = “10” then
    RESULT = CURVAL
Else
    Result=”Skip”
End If
```

The following function assigns the result of the logic account calculation to RESULT only if the criteria account center is 000.

```
If strCenter = “000” then
    RESULT = CURVAL * 100
Else
    Result=”Skip”
End If
```

The following function uses the FDM Lookup function to add a source account (810) to the value of the logic account, if the current FDM period is December, 2003.

```
If strPerKey = “12/31/2003” then
    RESULT = CURVAL + |810|
Else
    Result=”Skip”
End If
```

The following function uses the FDM Lookup function to add a source account (810) and a source account from a specified source center, FDM category, and FDM period to the value of the logic account, if the active location is Texas.

```
If strLocation = "Texas" then =
    RESULT = CURVAL + |000,10,09/30/01,810|
Else
    Result="Skip"
End If
```

Expression Operators

Expression operators are used to execute custom logic expressions, which are defined in the Value/Exp field. Logic expressions, which cannot use variables or `If` statements, are simpler than logic functions. Except for `|CURVAL|`, expressions do not have built-in parameters. For expressions, you do not need to assign a value to `RESULT`.

Expressions execute faster than logic functions. The FDM Lookup function can be used within expressions, as it is used within logic functions. To write a custom expression, double-click the Value/Exp field to open the expression editor.

```
|CURVAL| + |810| + |238|
```

The function above uses the FDM Lookup function to add two source accounts to the value of the logic account. Notice that the `CURVAL` parameter can be used within expressions, as it can within logic functions, except that, with expressions, `CURVAL` must be enclosed in pipes.

```
(|CURVAL| + |000,10,09/30/01,810|) * 100
```

The function above uses the FDM Lookup function to add a source account (810) and a source account from a specified center, FDM category, and FDM period to the value of the logic account and then multiplies the resulting sum by 100.

Lookup Function

The FDM Lookup function is used to return the value of any source account that resides in FDM. The function can be used within a logic function or a logic expression.

To return the value of a source account for the active logic account source center, FDM category, and FDM period, use the following format:

```
Syntax: |GLAcct|
Example: |810|
```

To return the value of a GL account for a specified GL center, FDM category key, and FDM period, use the following format:

```
Syntax: |GLDept , CategoryKey , PeriodKey , GLAcct|
Example: |000, 10, 12/31/01, 810|
```

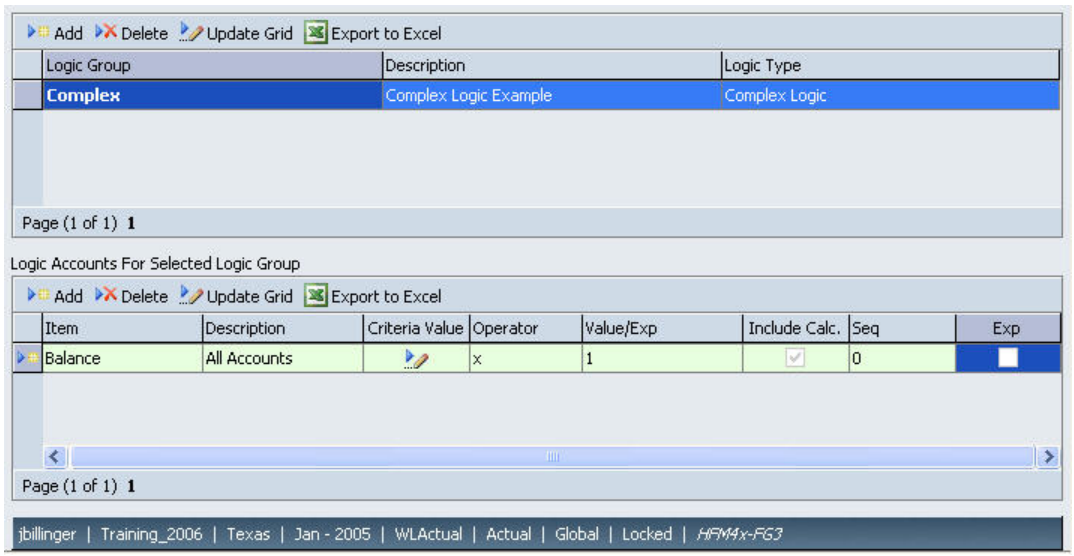
You can also use a combination of formats. The following example uses the default source center and FDM period but specifies a FDM category key.

```
|, 10 , , 810|
```

Creating Logic Accounts Within Complex Logic Groups

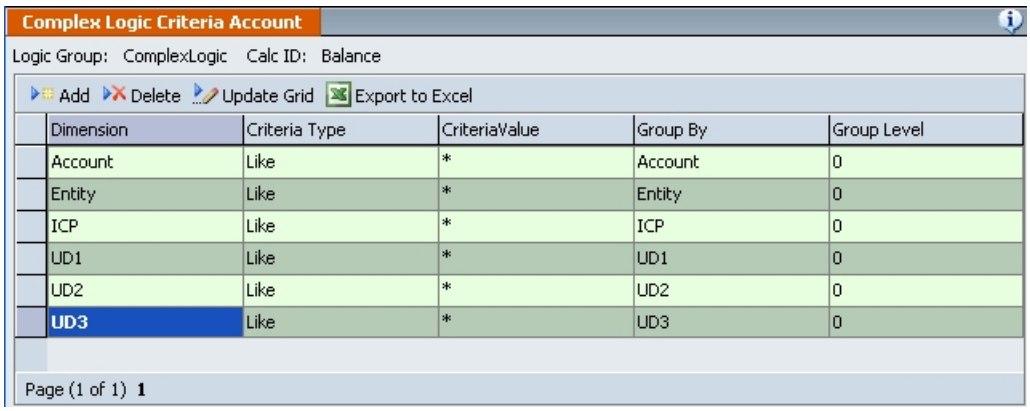
Within complex logic groups, you can create individual logic accounts. The fields for creating logic accounts within complex logic groups and the fields for creating logic accounts within simple logic groups are similar, varying only in the following ways:

- The Criteria Value fields function differently.
- For complex groups, an Include Calc field is provided.
- For complex groups, the Value/Exp field is not provided.



Criteria Value

To enter criteria for the dimensions, click the Criteria Value field to open a form, and, in the fields of the form, define the preferred criteria. The logic account is created only from the source records that meet the criteria.



Dimension

You can select any enabled source dimension. Each dimension can be selected only once.

Criteria Type

This field works with the Dimension and CriteriaValue fields to determine from which source values the logic accounts are derived. Criteria types available are In, Between, and Like. Criteria types determine how criteria values are interpreted.

CriteriaValue

This field provides the values that the criteria types use to determine what members are included in the logic calculations for the specified dimensions.

Group By

In the Import screen, in the dimension fields, the Group By specifications allow the display of logic accounts to override the display of members. Overriding displayed members allows dimensions to be grouped based on the values entered in the Group By field. The Group By field can be used to hard-code returned members or to append hard-coded values to original members (by entering hard-coded members and asterisks (*) in the Group By field).

For example, if you place the word *Cash* in the Group By field and select the Account dimension, the import form displays (for the logic account, in the Account field) the word *Cash*, rather than the original name of the Account member. If you place *L-** in the Group By field, the import form displays *L-1100,* the original name of the Account member.

If no value is entered in the Group By field, no grouping occurs for the relevant dimension, and a logic account is created for each unique dimension member.

Group Level

In the Import screen, the Group Level specifications work with the Group By specifications to override the display of members. The Group Level field accepts only numeric values.

For example, if you enter *3* in a Group Level cell, the left three characters of the Group By field are returned. If a cell of the Group By field does not contain a value and, in the associated Group Level cell, you enter *3*, the first three characters of the original name of the source member are returned.

You can use the Group By and Group Level fields to group the logic accounts that are displayed on import forms. For example, if you enter *L-** in the Group By cell, *L-1100*, the original name of the Account member that passed the logic criteria, is displayed in the import form. If, in addition, if you enter *2* in the associated Group Level cell, *L-11* is displayed; or, if, in addition, you enter *1* in the Group Level field, *L-1* is displayed.

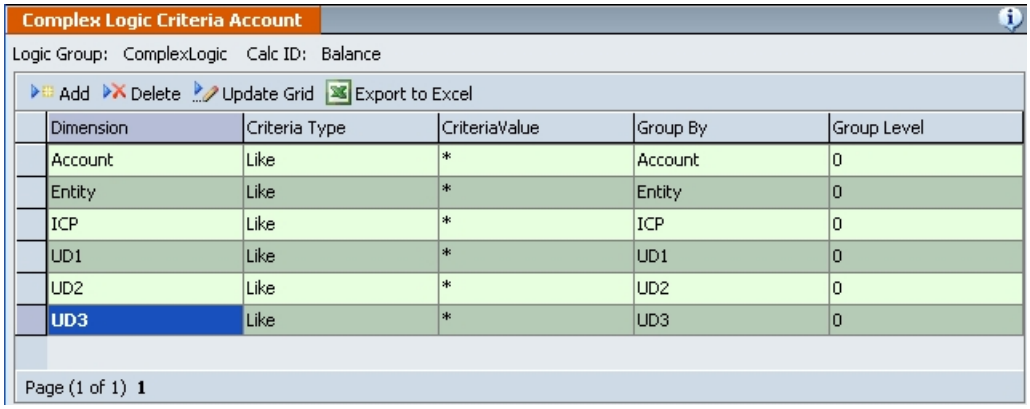
Note: The word "field" refers to the column; a specific entry is made in a cell.

Include Calc

If the entry in the Include Calc field meets the logic account criteria, the logic account can include previously calculated FDM values in its calculations.

Note: A sequence is attached to each logic account. Logic accounts are calculated in the specified sequence. If the Include Calc field is enabled for the second, or later, account, previously calculated logic accounts are included, provided that they meet the logic criteria.

Complex Logic Example 1



The first row specifies that source-record Account members must begin with *11*. The second row specifies that the source-record Entity member must be Tx. The third row specifies that source-record ICP members must be between 00 and 99. The last row specifies that source-record UD1 (Custom1) members must be 00, 01 or 02. Imported source records that do not meet all of the listed criteria are excluded from the calculated results.

As illustrated in the following tables, for Example 1, FDM derives one logic account from multiple source records. The first table lists examples of member combinations and, applying the logic criteria of the current example, identifies combinations (and thus data values) as included or excluded. The second table identifies the member combinations that are included —with the original member names replaced by the hard-coded values that are defined in the Group By field. The third table illustrates the final result.

Sample Imported Account Numbers

Account	Entity	ICP	UD1	Amount	Include or Exclude
1150	Tx	07	01	50,401.07	Include
1176	Tx	04	02	10,996.00	Include
1201	Tx	01	00	500.00	Exclude

Sample Imported Account Names

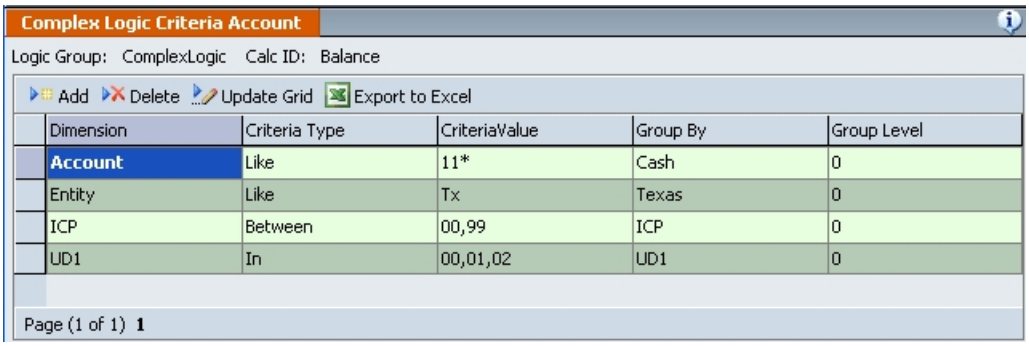
Account	Entity	ICP	UD1	Amount
Cash	Texas	ICP	UD1	50,401.07
Cash	Texas	ICP	UD1	10,996.00

FDM groups and summarizes the rows that include identical member combinations and thus creates the following result:

Final Result

Account	Entity	ICP	UD1	Amount
Cash	Texas	ICP	UD1	61,397.07

Complex Logic Example 2



The first row specifies that Account members must begin with *11*. The second row specifies that the Entity member must be Tx. The third row specifies that the ICP members must be between 00 and 99. The last row specifies that the UD1 (Custom1) member must be 00, 01. or 02. Imported source records that do not meet all of the listed criteria are excluded from the calculated results.

As illustrated in the following tables, for example 2, FDM derives two logic accounts from multiple source records. Two logic accounts are derived because two Group By cells include hard-coded values and two Group By cells include asterisk values. Therefore, within every source record that meets the specified criteria, the original Account and Entity members are replaced with the members listed in the Group By field. The other dimensions return all or some of the original members, as determined by the entries in the Group Level field.

Sample Imported Values

Account	Entity	ICP	UD1	Amount	Include or Exclude
1150	Tx	070	01	50,401.07	Include

Account	Entity	ICP	UD1	Amount	Include or Exclude
1176	Tx	040	02	10,996.00	Include
1121	Tx	045	02	9,050.41	Include
1201	Tx	100	00	500.00	Exclude

Logic Members

Account	Entity	ICP	UD1	Amount
Cash	Texas	07	UD1-01	50,401.07
Cash	Texas	04	UD1-02	10,996.00
Cash	Texas	04	UD1-02	9,050.41

FDM groups and summarizes the rows that include identical member combinations and thus creates the following result.

Final Result

Account	Entity	ICP	UD1	Amount
Cash	Texas	07	UD1-01	50,401.07
Cash	Texas	04	UD1-02	20,046.41

10 Working with Import Formats

In This Chapter

Overview of Import Groups

When importing data files, data load locations use import groups. Therefore, each data load location must be assigned an import group. Import groups instruct FDM how to interpret data files and how to transform files into data that can be loaded into the FDM data structure.

The Import Formats screen is arranged in a summary-and-detail format. The upper grid lists import groups, and the lower grid enables you to define import-group fields.

Note: If the format of a source file changes, you must update the import format that is associated with the source file.

Creating Import Groups

➤ To create import groups:

1 **From the FDM Web client, select Metadata > Import Formats.**

The Import Formats screen is displayed.

2 **From the upper grid, click Add.**

In the upper grid of the Import Formats screen, a row is added.

3 **For the import group, specify a name and description. .**

4 **From File Type, select Fixed or Delimited.**

5 **If the file type is delimited, in the delimiter column, select a type of delimiter.**

6 Click **Update Grid**.

Note: Selecting Script or Adapter in the File Type column enables you to build an integration script or use a target system adapter to import ledger data from an ODBC-compliant data source, rather than from a text file.

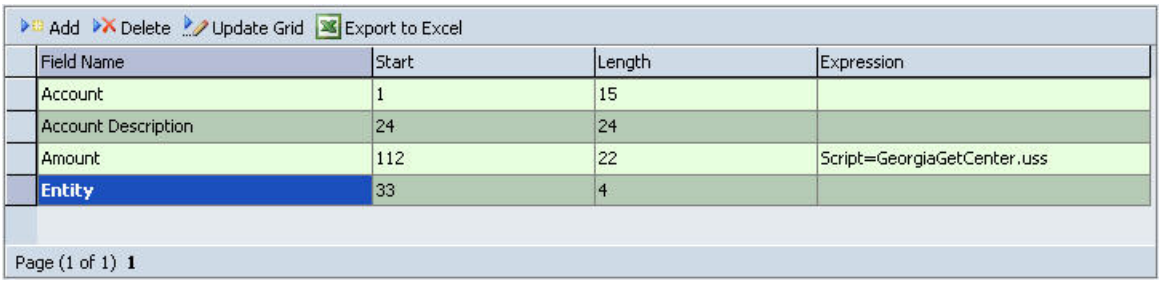
Defining Fields for Import Groups

Each row of the lower grid of the Import Formats screen represents one import field. If you assign multiple source fields to one FDM field name, FDM concatenates the string values during the file load process. If you want to skip records that contain a certain value, from the Field Name list, select Skip, and, in the Expression column, enter a text value.

Note: The Skip function for import formats is intended to work only with fixed file formats. To skip data in delimited files, refer to the section titled "Skip Function (Conditional Skip)" on page 194.

Fields for Fixed Import Groups

- **Field Name**—The field into which to import
- **Start**—The start location of the import field
- **Length**—The length of the import field
- **Expression**—An expression that overwrites the contents of the field, unless the field is defined as a Skip field



Fields for Delimited Import Groups

- **Field Name**—The field into which to import
- **Field No**—The field to be imported
- **# of Fields**—The number of fields that are included within each row of the text file
- **Expression**—An expression that overwrites the contents of the field location, unless the field is defined as a Skip field

Field Name	Field Number	Number of Fields	Expression
Source Entity	1	3	
Source Account	2	3	
Amount	3	3	Factor=1000

Using Drag and Drop to Define Fields

Instead of counting the start location and length of fields, then inputting them into the Import Format form, you can define fields by using the drag and drop feature from Import Format Builder.

➤ To use drag and drop to define fields in import formats:

1 In the lower grid of the **Import Formats** screen, select a field.

2 Click **Build**.

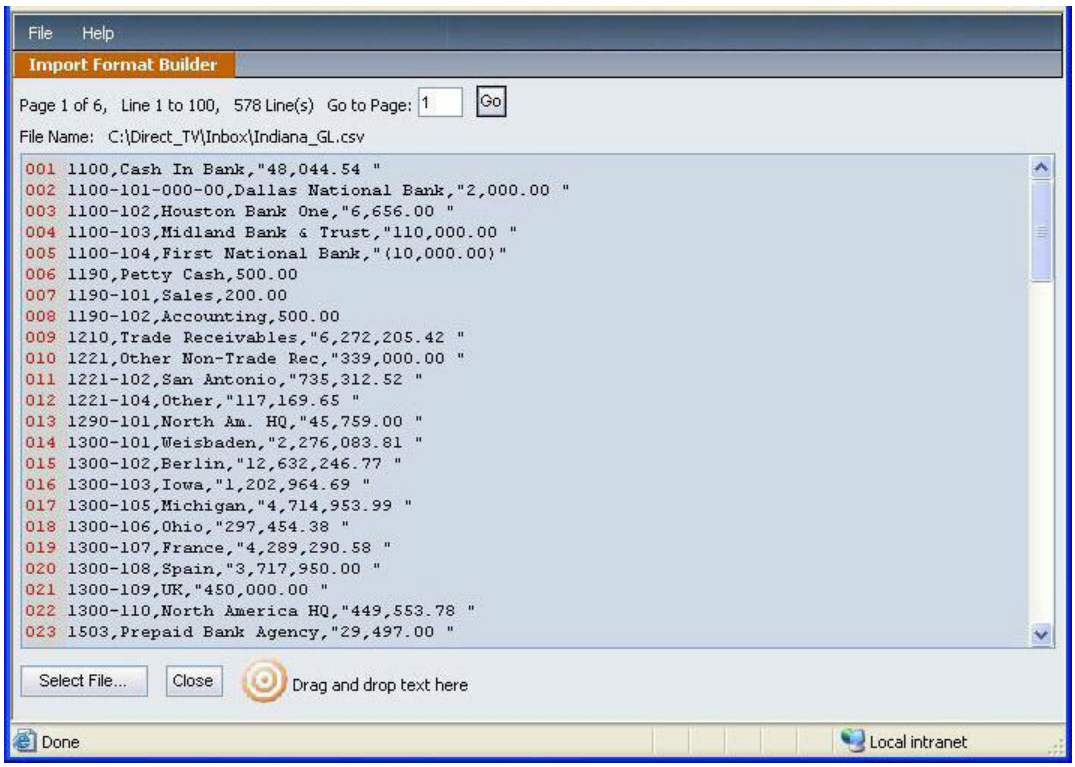
The Import Format Builder screen is displayed.

3 Click **Locate File**.

4 Locate and select the file to import.

5 Click **OK**.

After the import is complete, the file contents are shown in the Import Format Builder screen.



6 Select the text to import for the field.

7 Drag the selected text to the **Drag and drop text here** circle.

The field number and number of fields information is displayed in the Import Format form.

8 **Repeat steps 5 and 6 for each field name.**

A field name can be used more than once. During the load process, FDM concatenates multiple field names.

Assigning Import Formats

Import formats are not used until they are assigned to locations.

➤ To assign import formats to locations:

1 **From the FDM Web client, Select Metadata > Locations.**

The Locations screen is displayed.

2 **Select a location.**

3 **Select the Workflow Behaviors tab.**

4 **From Import Format, select an import format.**

5 **Click Save.**

The selected format is used every time a trial balance is imported for the location. One format can be used for many locations. If your company has a standard trial balance file format, all locations can share one import format.

Using Import Expressions

FDM provides a set of powerful import expressions that enable it to read and parse virtually any trial balance file into the FDM database. Advanced expressions are entered into the Expression column of the field grid of the Import Formats screen. Import expressions operate on the value that is read from the import file.

Adding Import Expressions

➤ To add import expressions or import scripts:

1 **In the lower grid of the Import Formats screen, in the preferred field, double-click the Expression column.**

2 **Select Add Expression.**

The Add Expression dialog box is displayed.

3 **From Expression Type, select a type of expression or script.**

The number and types of expressions available depend on the field that is being modified (for example, Account or Account Description).

4 **Enter the value to accompany the expression.**

5 Click **OK**.

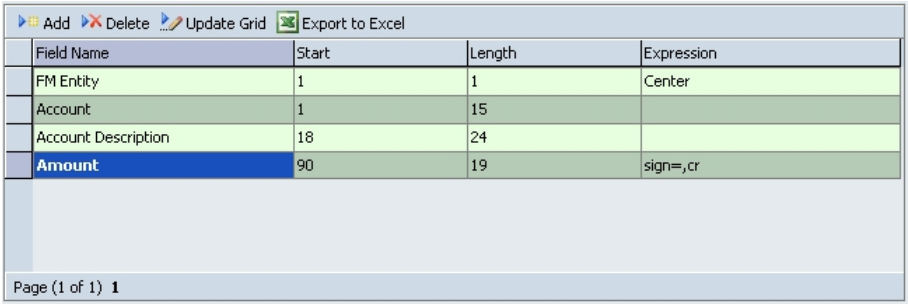
For the preferred field, the expression is entered into the Expression column.

Managing Nonstandard Numeric Sign Conventions

The `Sign` expression is used to mange nonstandard numeric sign conventions. FDM interprets numbers with leading and trailing minus signs and numbers within parentheses as negative numbers. However, you can use other trailing characters to indicate negative numbers. To define custom signs, use expressions that follow this form: `Sign=[Positive String],[Negative String]`.

For example, if positive numbers are followed by DR (`1,000.00DR`), and negative numbers are followed by CR (`1,000.00CR`), the expression is `Sign=DR,CR`

For example, if positive numbers are unsigned (`1,000.00`), and negative numbers are followed by CR (`1,000.00CR`). the expression is `Sign=,CR`.



Displaying Debit and Credit Columns

The `DRCRSplit` expression is used to parse split numeric columns. By default, FDM assumes that numeric values in Amount fields are debits. However, you can position debit values on the left and credit values on the right.

ACCOUNT NUMBER	DESCRIPTION	DEBIT	CREDIT
1000-000-00	Cash-Operating Account	68,603.91	
1010-000-00	Cash-FANB-AP		177,216.16

`DRCRSplit`, which enables FDM to interpret left-right positioning and to assign the correct sign, follows the format `DRCRSplit=Mid Point of the DR and CR columns`.

In the preceding example, the Amount field of the source file begins at column 46 of the fixed-column import file and contains thirty-one characters. Everything left of the midpoint (16) is a debit and everything right of the midpoint (16) is a credit. Therefore, the expression in the import format for Amount is `DRCRSplit=16`.

Add | Delete | Update Grid | Export to Excel

Field Name	Start	Length	Expression
FM Entity	1	1	Nevada
Account	1	11	
Account Description	14	32	
Amount	46	31	DRCRSplit=16
Skip	43	1	

Page (1 of 1) **1**

When the file is imported, credit amounts are assigned negative signs (and thus interpreted as positive), and debit amounts are unchanged (and thus interpreted as negative).

Converting from European to U.S. Notation

The `Fill=EuroToUS` expression is used with the Amount field to trigger a number format conversion from `(.,)` to `(,.)` format.

Padding Fields with Leading Fills

The `FillL=`*`LeadingFill`* expression is used to fill fields with leading characters. Text values that are shorter than the specified fill expression are padded, as directed by the fill expression. The following data file lists account numbers of varying lengths:

```
gltbrp.p                                25.15.4 Trial Balance Summary                                Date: 02/2/05
Page: 1                                 Texas Automotive Systems, Inc.                                Time: 15:42:

Texas xxxxxxxx                                Reporting Currency:   USD
                                               Exchange Rate:


                                          Beginning Balance     Period Activity       Ending Balance
Account          Description                    12/31/04                            1/31/05          Adjustments  Balance
---------------  -------------------------  ------------------  ------------------  ------------------  -----------  -------
1100             Cash In Bank                      283,767.29          135,722.75cr         48,044.54
1100-101-000-00  Dallas National Bank                 2,000.00                .00          2,000.00
1100-102          Houston Bank One                   9,986.39             3,330.39cr         6,656.00
1100-103          Midland Bank & Trust                                  110,000.00          110,000.00
1100-104          First National Bank              50,000.00            60,000.00          10,000.00cr
1190             Petty Cash                          500.00                .00            500.00
1190-101          Sales                              200.00                .00            200.00
1190-102          Accounting                         500.00                .00            500.00
1210             Trade Receivables                158,857.30         6,113,348.12       6,272,205.42
1220             Tooling & Prototype Rec           71,087.28            71,087.28cr            .00
1221             Other Non-Trade Rec                   .00             339,000.00          339,000.00
1221-101           N/T Rvbl Dallas                 93,145.54            93,145.54cr            .00
1221-102           San Antonio                    712,693.72            22,618.80          735,312.52
1221-103           United Parts                    50,817.31            50,817.31cr            .00
1221-104           Other                          117,169.65                .00          117,169.65
1290             Interest Receivable                91,250.00            91,250.00cr            .00
1290-101           North Am. HQ                    30,815.65            14,943.35          45,759.00
1299             Bad Debt Reserve                239,602.03cr         239,602.03               .00
```

The `FillL` expression is used as shown in the following example:

Field Name	Start	Length	Expression
Source Account	1	15	FillL=0000000
Account Description	18	24	
Amount	90	19	sign=,cr
Source Entity	1	1	Center
Source ICP	1	15	
Source Department	10	3	
Source Prod/Salary	14	2	

After import, account numbers are displayed as shown in the following example:

	Source Ent	Source Account	Account Description	Source ICP	Source Departmen	Source Prod/Salary	Amount
--	Center	0001100	Cash In Bank	1100			48,044.54
--	Center	0001190	Petty Cash	1190			500.00
--	Center	0001210	Trade Receivables	1210			6,272,205.42
--	Center	0001221	Other Non-Trade Rec	1221			339,000.00
--	Center	0001503	Prepaid Bank Agency	1503			29,497.00
--	Center	0001527	Prepaid Managment Fee	1527			27,861.05
--	Center	0001542	Prepaid Other	1542			148,359.67
--	Center	0001543	Prepaid Maintenance	1543			173,468.07
--	Center	0001548	Prepaid Rent	1548			5,339.94

Padding Fields with Trailing Fills

The `Fill=TrailingFill` expression is used to fill fields with trailing characters. Text values that are shorter than the specified fill expression are padded, as directed by the fill expression. For example, if the account number is 103950- and the expression `Fill=000000000` is used, the account number after import is 103950-000. Another example is an account number of 243150 with the expression Fill=111111111, the account number after import is 243150111.

Multiplying by Whole-Number and Decimal Factors

The `Factor=Value` expression is used to factor source-file amounts by user-defined values.

For example, the following `Factor` expression converts the amount 12500 to 12500000.

Field Name	Field Number	Number of Fields	Expression
Source Entity	1	3	
Source Account	2	3	
Amount	3	3	Factor=1000

Amounts can be multiplied by whole numbers or decimals; thus, for example, amounts can be doubled or halved.

Disabling Zero Suppression

The NZP expression is used to disable zero suppression during the data load process. By default, FDM bypasses accounts in the trial balance that have zero balances. In certain circumstances, you may want to load all accounts, to ensure that values that should be zero are replaced in the target consolidation system. You enter `NZP` in the Expression column of the Amount field to disable zero suppression.

Passing Values to Excel Files

The `Excel` expression is used to pass field values to Excel files. FDM uses the expression to automate Excel, to pass a field value into cell A1 and to return the value found in cell A2. This feature enables Excel worksheets to function as custom evaluation tools. The value in cell A2 can contain a standard Excel function or a custom VBA function.

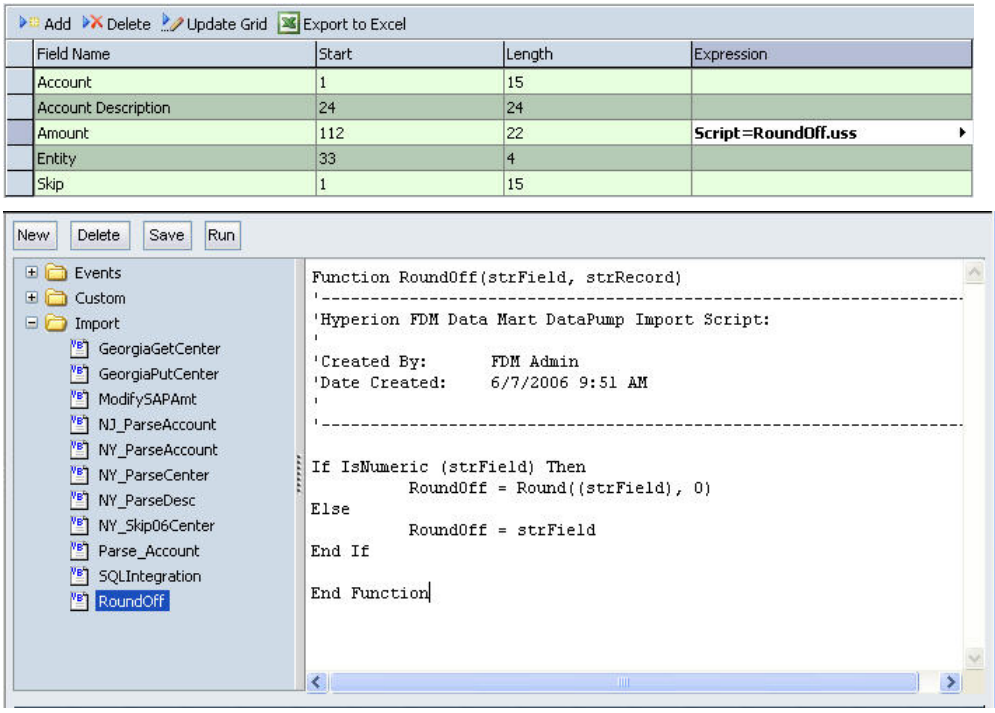
To define Excel parsing expressions, follow this form: `Excel=YourFileName.xls`. Set the Excel file to append Excel to imported descriptions by using `=A1&"Excel"` for the value of cell A2.

- The Excel file must be located in the `<application name>\Data\Scripts\Import\` directory.
- The `Excel` expression may consume a large amount of system resources, because Excel must run during the load process. In most cases, the `Script` expression, rather than the `Excel` expression, should be used.
- Excel must be installed on the application server.

Passing Values to FDM Script Files

The `Script` expression is used to pass field values to FDM script files. FDM uses `Script` to load and interpret a custom import script that runs against each line of the import file. Script files, which are based on the VB scripting language, enable end users to extend the power and flexibility of the FDM import process.

The following example shows the use of `Script` to round and remove decimals from values in the Amount field.



Note: The script file must be located in the `<application name>\Data\Scripts\Import\` subdirectory. The name of the script file and the name of the function that the file contains must be the same. This `Script` expression points to a script file that contains a custom function that is run against each line of the import file. The function accepts one argument, the value read for the field from the trial balance file, and returns the value of the argument after the function operates on it.

Creating Import Scripts and Integration Scripts

To perform actions that cannot be performed by predefined import expressions, you create import scripts. Import scripts are entered in the Expression column of the field to which they apply.

Rather than importing text files, you can build integration scripts to import ledger data directly from ODBC-compliant data sources. Integration scripts can be run only by selecting Script from the File Type column of the Import Group grid (top grid) of the Import Formats screen

➤ To create import scripts or integration scripts:

1 In lower grid of the **Import Formats** screen, in the preferred field, double-click the **Expression** column.

2 Select **Build Script**.

The New Script dialog box is displayed.

3 Click **New**.

4 From **Script Type**, select **Import (Data Pump)** (to create an import script) or **Import (Integration)** (to create an integration script).

5 Enter a name for the script.

Note: Custom scripts do not support multibyte or extended ASCII characters in the script name or in the script parameters.

6 Click **OK**.

Script Editor is displayed.

7 Create the script.

8 Click **Save**.

9 Return to the **Import Formats** screen, and follow the instructions in "Adding Import Expressions" on page 84 to add the script to the **Expression** column.

Integration Script Example

The following integration script uses an ADO connection to log on to a RDBMS database and appends the ledger data to the trial balance table of the FDM location.

```
Function SQLIntegration(strLoc, lngCatKey, dblPerKey, strWorkTableName)
'-----------------------------------------------------------------
'FDM Integration Import Script:
'
'Created By:      wladmin
'Date Created:     04/19/2004 2:18:39 PM
'
'Purpose:     Pull data directly from SQL DB
'-----------------------------------------------------------------
Dim objSS        'ADODB.Connection
Dim strSQL         'SQL string
Dim rs           'Recordset
```

```
Dim rsAppend     'tTB table append rs object
'Initialize objects
Set cnSS = CreateObject("ADODB.Connection")
Set rs = CreateObject("ADODB.Recordset")
Set rsAppend = DW.DataAccess.farsTable(strWorkTableName)
'Connect to SQL Server database
cnss.open "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=WLDemo;Data Source=DBServerName;"
    'Create query string
    strSQL = "Select * "
    strSQL = strSQL & "FROM tGL "
    'Get data
    rs.Open strSQL, cnSS

       'Check for data
    If rs.bof And rs.eof Then
        RES.PlngActionType = 2
            RES.PstrActionValue = "No Records to load!"
        Exit Function
    End If
  'Loop through records and append to tTB table in location's DB
    If Not rs.bof And Not rs.eof Then
        Do While Not rs.eof
        rsAppend.AddNew
            rsAppend.Fields("PartitionKey") = RES.PlngLocKey
            rsAppend.Fields("CatKey") = RES.PlngCatKey
            rsAppend.Fields("PeriodKey") = RES.PdtePerKey
            rsAppend.Fields("DataView") = "YTD"
            rsAppend.Fields("CalcAcctType") = 9
            rsAppend.Fields("Amount") = rs.fields("dblAmt").Value
            rsAppend.Fields("Desc1") = rs.fields("txtAcctDes").Value
            rsAppend.Fields("Account") = rs.fields("txtAcct").Value
            rsAppend.Fields("Entity") = rs.fields("txtCenter").Value
        rsAppend.Update
        rs.movenext
        Loop
    End If
    'Records loaded
    RES.PlngActionType = 6
      RES.PstrActionValue = "SQL Import successful!"
'Assign Return value
SQLIntegration = True
End Function
```

Using Adapter Import Groups

Rather than importing text files, you can use integration adapters to import ledger data directly from ODBC-compliant data sources. If you want to use an adapter for a group, in the upper grid of the Import Formats screen, in the File Type cell of the row that defines the group, select Adapter.

You can import source integration adapters into the FDM application by using the Import XML function (File > Import XML).

Stacking Import Expressions

One field can use multiple import expressions. To stack expressions, separate the expressions with a semicolon. Consider the order in which the layered expressions are processed.

Example: Expression Stacking Syntax

Field: GL Account

```
Expression List: Script=ParseAcct.txt;Fill=0000000
```

Processing Order

For all fields except the Amount field, FDM processes stacked expressions in the following order. .

1. `Script`
2. `Excel`
3. `Fill` or `FillL`

For the Amount field, FDM processes stacked expressions in the following order:

1. `DRCRSplit`
2. `Fill=EuroToUS`
3. `Script`
4. `Excel`
5. `Sign`
6. `Factor`
7. `Scale`
8. `NZP`

11 Mapping Tables

In This Chapter

Overview of Mapping Tables

Mapping tables map source members to target members.

➤ To open the **Maps** screen, select **Activities** > **Maps**.

On the Maps screen, from a list, you select a dimension to map. Because, in the target dimension list, FDM displays only valid values, you can be assured that you are mapping source members to valid target values.

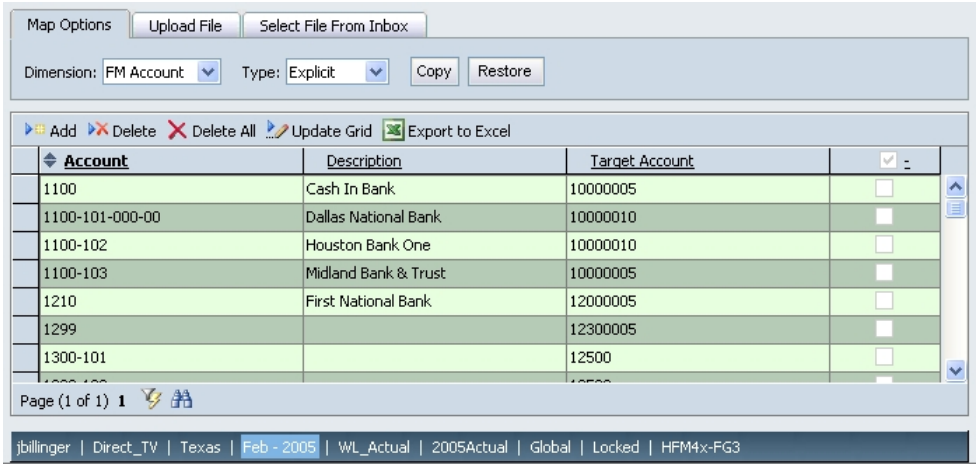
Mapping tables for the Account dimension contain a column that controls sign conversion. This column, which is titled with a minus (–), provides an option that reverses the sign of the source account value.

Map Types

General ledger accounts can be converted with one-to-one mapping (Explicit) or with wildcard mapping (In, Between, or Like). From the Type list, you select one of the four types.

Explicit Mappings

Explicit mapping assigns a target account to every source account.



The process of mapping source accounts to target accounts begins with identification of the structure of the source general ledger account. Therefore, each account segment must be identified and documented. You decide the level at which the source account is translated into the target account.

Example Account Structure

TT = Account Type = Division

PP = Primary Account Code = Location

SS = Sub Account = Department

Example Accounts

001100 Cash - Disbursement

001101 Cash - Lock Box

223500 Accounts Payable Trade

223501 Accounts Payable Intercompany

Primary Level Map (TTPP)

The example account structure can be mapped at two levels. If the target account structure is aggregated, so that no distinction is made between Cash - Disbursement and Cash - Lock Box, the source account can be truncated after the primary account segment. On the other hand, if the level of detail of the target account equals the level of detail of the source general ledger account, the source account must be mapped at the more detailed, sub-account level. The following maps illustrate this point.

```
GL Primary Account, Consolidation Account
0011, Cash
2235, AP
Sub Account Level Map (TTPPSS)
GL Primary Account / Sub-Account, Consolidation Account
001101, Cash.01
```

```
001102, Cash.02
223501, AP.01
223502, AP.02
```

Wildcard Mappings

For mapping source records to target records, wildcard conversion rules provide two advantages: increased flexibility and reduced table maintenance.

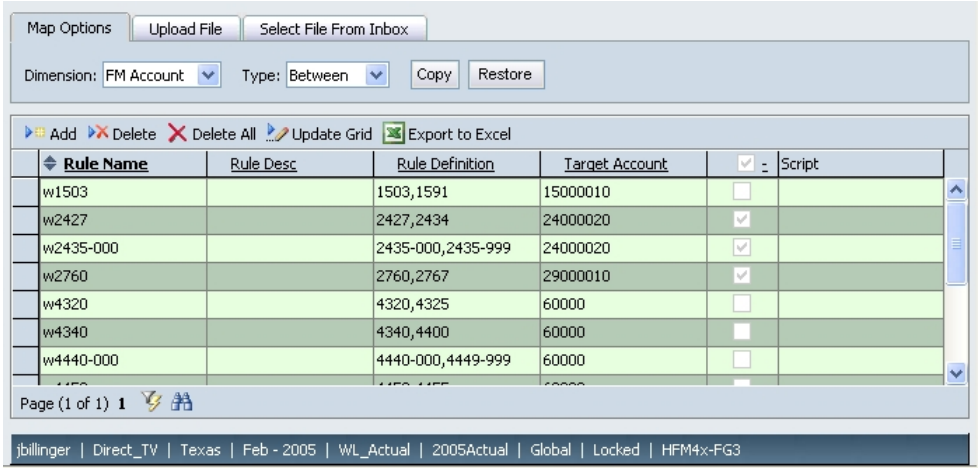
All source records are stamped with the conversion rule that was used to provide the record with its target values. This process creates a static map for each location, category, and period combination and thus ensures a static audit trail.

Multiple general ledger accounts can be grouped into one calculated account, and the calculated account can be mapped to one target account.

The processes for creating wildcard account mapping and for creating standard account mapping are the same. However, for wildcard account mapping, a wildcard rule name and rule definition must be created, and a target account must be assigned to the rule.

For both wildcard and calculated-account mapping, three types (Between, In, and Like) are available:

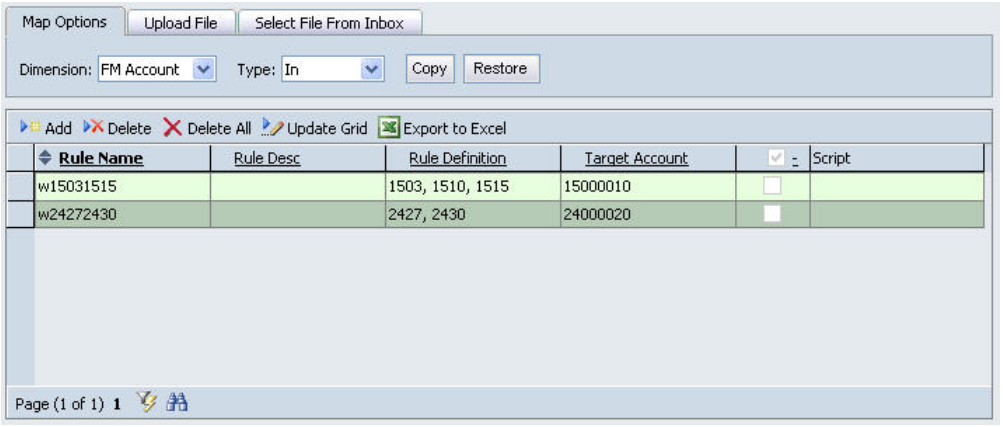
Between Mappings



When creating Between mappings, ensure that the sides of the mapping range are the same length. For example: `2000.000, 2991.999` is a valid mapping, and `2000.000, 2991` is not a valid mapping. This caution is especially important if a mapping excludes accounts. For example, `2000.000, 2991` excludes 2991.000 through 2991.999.

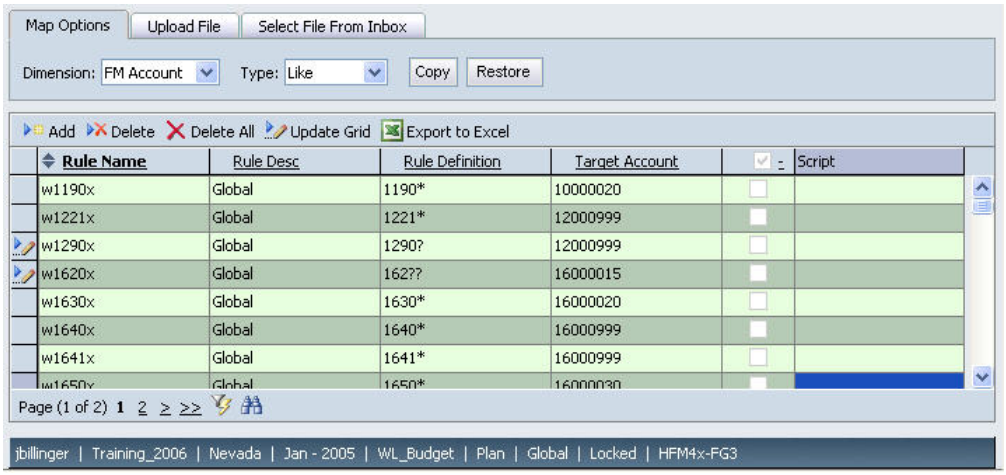
Additionally, Between mappings must be the same length to ensure that accounts use the correct map. For example: If the mapping `1530, 1540` and the mapping `1530000, 1540000` both exist, the account 1540 is imported by the second mapping (`1530000, 1540000`).

In Mappings



In mappings enable a list of non-sequential source accounts to be mapped to one target account. In this case, multiple accounts are mapped to one account within one rule, eliminating the need to create multiple rules (as is required for an Explicit map). The preceding example shows that source accounts 1503, 1510, and 1515 are mapped to the target account 15000010.

Like Mapping



Like mappings use wildcard characters: asterisks (*) and question marks (?). Asterisks are placeholders for any number of characters. For example, `1190*` maps the accounts 1190, 1190100, and 1190-200 to the Cash target account.

Question marks are placeholders for one character. For example, the source account of `119?` maps only to source accounts that contain four characters and that begin with 119.

Restricted Characters for Mapping

Using the following characters in source or target FDM mapping is not allowed and may cause mapping problems:

- Ampersand (&)

- Asterisk (*)
- Question Mark (?)
- Greater Than (>)
- Less Than (<)
- Apostrophe (')
- Quote (“)
- Semicolon (;)
- Underscore (_) (can be used for Explicit mapping only)

Automap Wildcarding

FDM enables target-account or target-entity derivation by permitting the placement of wildcard characters (* and ?) in both source and target members. Mapping-table records that have wildcard characters in both the source and target column are considered to be Automap.

Example Automap

Rule Name	Rule Description	Rule Value	Target Account
w0011--	Cash Accts	`0011??`	Cash.??

Example General Ledger Trial Balance Records

GL Account	Center	Description	Amount
001100	0160000	Cash In Bank	1000.00
001101	0000000	Cash Corp LB	2000.00
001116	0001000	Petty Cash	1000.00
223500	0160000	AP	5000.00

Resulting Record Conversion

GL Account Description	Hyperion Account
001100 0160000	Cash.00
001101 0000000	Cash.01
001116 0160000	Cash.16

Explanation of the Conversion Process

The criteria for the Automap entry (Like `0011??`) retrieves the first three records from the general ledger trial balance. Because the Automap entry contains wildcard characters in the target account column, FDM must search the source account to replace the wildcard characters within the target account with actual characters from the source account.

The source-account characters represented by the two hyphens (under Rule Name) are used to replace the two question marks that follow `Cash` (under Target Account). Next, FDM verifies that the new target accounts (Cash.00, Cash.01, Cash.16) are valid target accounts. If the accounts are valid, the derived values are assigned to the trial balance as the target account values.

Conditional Mapping

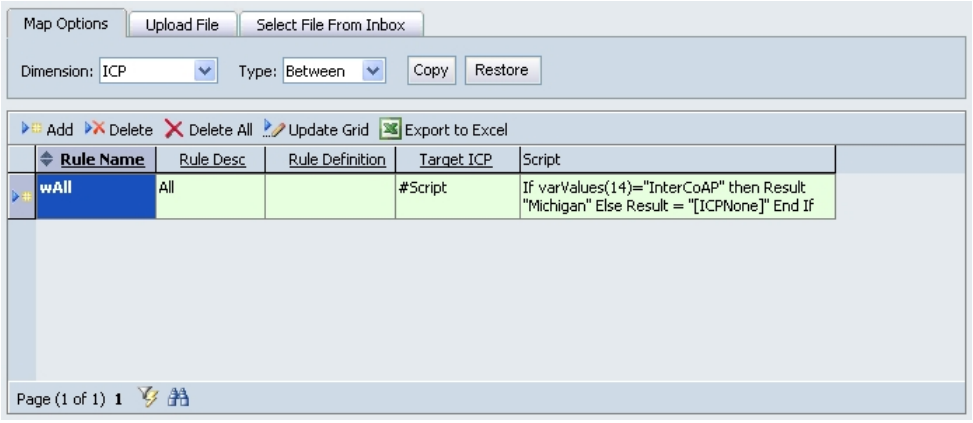
With conditional mapping, source members are mapped to script expressions, rather than to hard-coded target members.

Conditional mapping is valid only for rule-based mapping (Between, In, and Like). You activate conditional mapping by placing `#Script` in the target-member column. This entry triggers execution of the script listed in the Script column.

Because scripts are written in VBScript, all functionality of VBScript is provided and the running instance of the API can be accessed. Conditional mapping, in conjunction with dimension processing order, enables mapping that is based on the results of dimension mappings that were processed previously, prior to the processing of the current mappings.

Conditional Mapping Example

Assume that the Account and Entity dimensions process prior to the ICP dimension and that the ICP dimension uses a conditional mapping rule. In this case, when the ICP dimension is processed, the `varValues` array contains the mapped values for Account and Entity, and the mapped values for all other dimensions are null.



The ICP conditional script evaluates the mapped value of the Account dimension and determines how to map ICP.

```
'Check element 14 of the varValue array to get converted/mapped account
If varValues(14) = "InterCoAP" Then
```

```
    'Mapped value = "InterCoAP" map to Michigan
    Result = "[Michigan]"
Else
    'All others, map to none
    Result = "[ICP None]"
End If
```

Values

The following table defines the preset variables that are available for use in conditional script expressions:

API—Object reference to the running API

strLoc—String value that contains the name of the location being processed

VarValues—A variant array that contains the record values that are being mapped and the record values that were previously mapped

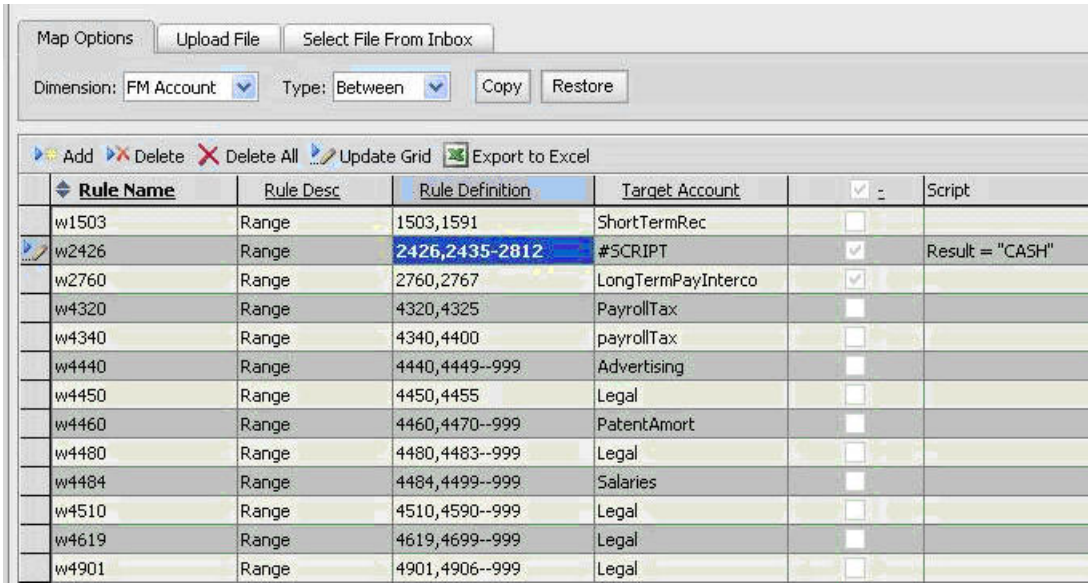
Variable	Record Value Mapped	Variable Type
varValues(1) =	Current Location ID number	Long Integer
varValues(2) =	Current Category ID number	Long Integer
varValues(3) =	Current Period Key value	Date
varValues(4) =	Current Data View	String (default = YTD)
varValues(5) =	Currency key assigned to the location	String
varValues(6) =	Calculation type for the current record	Long Integer: 9 = Base, 5 = Logic Export, -1 = Logic Non-Export
Logic	(No Export)	
varValues(7) =	Sign Change flag for the current record	Boolean: 0 = False, -1 = True
varValues(8) =	Journal ID for the current record	String (NULL, if not a journal)
varValues(9) =	Amount for the current record	Double
varValues(10) =	Supplemental Amount for the current record	Double (0, unless populated by a custom script)
varValues(11) =	Account description for the current record	String
varValues(12) =	Supplemental Account description for the current record	String (NULL, unless populated by a custom script)
varValues(13) =	Source Account	String
varValues(14) =	Target Account	String (NULL if the current dimension has not been processed)
varValues(15) =	Source Entity	String
varValues(16) =	Target Entity	String (NULL, if the current dimension has not been processed)
varValues(17) =	Source ICP	String

Variable	Record Value Mapped	Variable Type
varValues(18) =	Target ICP	String (NULL, if the current dimension has not been processed)
varValues(19) =	Source UD1	String
varValues(20) =	Target UD1	String (NULL, if the current dimension has not been processed)
varValues(21) =	Source UD2	String
varValues(22) =	Target UD2	String (NULL, if the current dimension has not been processed)
varValues(23) =	Source UD3	String
varValues(24) =	Target UD3	String (NULL, if the current dimension has not been processed)
varValues(25) =	Source UD4	String
varValues(26) =	Target UD4	String (NULL, if the current dimension has not been processed)
varValues(27) =	Source UD5	String
varValues(28) =	Target UD5	String (NULL, if the current dimension has not been processed)
varValues(29) =	Source UD6	String
varValues(30) =	Target UD6	String (NULL, if the current dimension has not been processed)
varValues(31) =	Source UD7	String)
varValues(32) =	Target UD7	String (NULL, if the current dimension has not been processed)
varValues(33) =	Source UD8	String
varValues(34) =	Target UD8	String (NULL, if the current dimension has not been processed)
varValues(35) =	Source UD9	String
varValues(36) =	Target UD9	String (NULL, if the current dimension has not been processed)
varValues(37) =	Source UD10	String
varValues(38) =	Target UD10	String (NULL, if the current dimension has not been processed)
varValues(39) =	Source UD11	String
varValues(40) =	Target UD11	String (NULL, if the current dimension has not been processed)
varValues(41) =	Source UD12	String
varValues(42) =	Target UD12	String (NULL, if the current dimension has not been processed)
varValues(43) =	Source UD13	String
varValues(44) =	Target UD13	String (NULL, if the current dimension has not been processed)
varValues(45) =	Source UD14	String

Variable	Record Value Mapped	Variable Type
varValues(46) =	Target UD14	String, Null if this dimension has not processed yet
varValues(47) =	Source UD15	String
varValues(48) =	Target UD15	String, Null if this dimension has not processed yet
varValues(49) =	Source UD16	String
varValues(50) =	Target UD16	String, Null if this dimension has not processed yet
varValues(51) =	Source UD17	String
varValues(52) =	Target UD17	String, Null if this dimension has not processed yet
varValues(53) =	Source UD18	String
varValues(54) =	Target UD18	String, Null if this dimension has not processed yet
varValues(55) =	Source UD19	String
varValues(56) =	Target UD19	String, Null if this dimension has not processed yet
varValues(57) =	Source UD20	String
varValues(58) =	Target UD20	String (NULL, if the current dimension has not been processed)
varValues(59) =	Archive Document ID number for the current record	Long Integer

Required Conditional Map when Using Oracle RAC

When using Oracle RAC, Between map rules that include non-numeric characters in the rule definition must use conditional mapping. The following example shows that rule w2426 contains "2426, 2435-2812" in the rule definition field. The hyphen is non-numeric, therefore, this rule is made conditional by entering #script in the Target Account field and `result="<target account"` in the Script field.



Rule Name	Rule Desc	Rule Definition	Target Account	-	Script
w1503	Range	1503,1591	ShortTermRec		
w2426	Range	2426,2435-2812	#SCRIPT		Result = "CASH"
w2760	Range	2760,2767	LongTermPayInterco		
w4320	Range	4320,4325	PayrollTax		
w4340	Range	4340,4400	payrollTax		
w4440	Range	4440,4449--999	Advertising		
w4450	Range	4450,4455	Legal		
w4460	Range	4460,4470--999	PatentAmort		
w4480	Range	4480,4483--999	Legal		
w4484	Range	4484,4499--999	Salaries		
w4510	Range	4510,4590--999	Legal		
w4619	Range	4619,4699--999	Legal		
w4901	Range	4901,4906--999	Legal		

If you are upgrading from an older version of FDM, or have recently enabled RAC in Oracle and are using existing maps, then you must update any "Between" mapping rules that contain nonnumeric characters in the rule definition. In this example, rules w4440, and w4460 through w4901 must be updated to be conditional mappings.

Map Processing Order

FDM processes maps within a specific order. Mapping order is significant if one source value is used with multiple map types (for example, Explicit and Between) and if one source value is used multiple times within one mapping type.

Processing Order Across Multiple Mapping Types

If a source value is used with multiple map types, FDM processes the maps in the following order:

1. Explicit
2. Between
3. In
4. Like

Explicit maps override all other map types. Between maps override In and Like maps. In maps override Like maps.

Processing Order Within One Mapping Type

By default, FDM wildcard maps (Like, In, and Between) are sorted alphabetically, from within the Rule Name column. To enable FDM to sort and process maps by numeric value, you enable map sequencing.

Between Maps

If a source account is valid within multiple Between maps, the source account is assigned to the last Between map in the mapping table.

In Maps

If a source account is valid within multiple In maps, the source account is assigned to the first In map in the mapping table.

Like Maps

If a source account is valid within multiple Like maps, the source account is assigned to the first Like map in the mapping table.

Map Processing Cost

Because of the methods used for processing, computer resources used to complete tasks varies among mapping types. Using the most appropriate mapping type for your GL files minimize the time required to process GL accounts importing from source files.

The table that follows shows the impact of each mapping type on processing time.

Map Explicit

Processing Cost—Low

Process Cost Explanation—Simple update pass-through to the database

Server Resources Used—Data server

Restrictions—None

Good Example—Source —> Target

Map Range

Processing Cost—Low

Process Cost Explanation—Simple update pass-through to data server

Server Resources Used—Data server

Restrictions—Source values in range must be same number of characters

Good Example—Between xxxx,yyyy —> Target

Poor Example—Between xxx,yyyyy —> Target

Map Range (Conditional)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the application server; results in high amount of data transfer and memory utilization

Server Resources Used—Application server

Recommendation—Restrict conditional ranges. Rather than using one large range (0000 to 9999), divide the range into multiple, smaller rule blocks (0000 to 1000, 1001 to 2000, for example); optimizes memory utilization

Restrictions—Source values in range must be same number of characters

Good Example—Between xxxx,yyyy —> #Script

Poor Example—Between xxxx,yyyyyy —> #Script

Map In

Processing Cost—Low

Process Cost Explanation—Simple update pass-through to the database

Server Resources Used—Data server

Restrictions—None

Good Example—IN(xx,yy,zz) —> Target

Poor Example—NA

Map In (Conditional)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the application server; results in high amount of data transfer and memory utilization

Server Resources Used—Application server

Recommendation—Restrict IN lists. Rather than using one large list, divide the list into multiple, smaller lists; optimizes memory utilization

Restrictions—None

Good Example—IN(xx,yy,zz) —> #Script

Poor Example—NA

Map Like (one-sided *)

Processing Cost—Low to Medium

Process Cost Explanation—Simple update pass-through to the database

Server Resources Used—Data server

Recommendation—Minimize the number of place holders (?) in statements; reduces time for database engine to process masks

Restrictions—None

Good Example—Like 12??45 —> Target

Poor Example—Like ??????? —> Target

Map Like (two-sided source values used to derive target values)

Processing Cost— High

Process Cost Explanation—Target value derivation process returns recordsets with the specified dimension's fields to the app server; results in high amount of data transfer and memory utilization

Server Resources Used— Application server

Recommendation— Restrict Like criteria. Don't use a * without other criteria in the rule definition, which causes a large volume of records to be written to the app server. Use criteria such as 1*, 2*, 3* or A*, B*, C*, for example, to limit each query to a subset of what you need to map; optimizes memory utilization

Restrictions—None

Good Example—Like 12* —> Target*

Poor Example—Like * —> Target*

Map Like (Conditional)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the app server; results in high amount of data transfer and memory utilization

Server Resources Used—Application server

Recommendation—Restrict Like criteria. Don't use a * without other criteria in the rule definition, which causes a large volume of records to be written to the application server. Use criteria such as 1*, 2*, 3* or A*, B*, C*, for example, to limit each query to a subset of what you need to map; optimizes memory utilization

Restrictions—None

Good Example—Like 12* —> #Script

Poor Example—Like * —> #Script

Map Like (* to *)

Processing Cost—Low

Process Cost Explanation—Simple update pass-through to the database. This rule type is optimized by FDM and processes quickly

Server Resources Used—Data server

Good Example—Like * —> *

Poor Example—NA

Logic (In)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the application server; results in high amount of data transfer and memory utilization

Server Resources Used—Application server

Restrictions—None

Logic (Between)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the app server; results in high amount of data transfer and memory utilization

Server Resources Used— Application server

Restrictions— None

Logic (Like)

Processing Cost—Very High

Process Cost Explanation—Conditional mapping process returns recordsets with all dimension fields to the app server; results in high amount of data transfer and memory utilization

Server Resources Used—Application server

Restrictions—None

Copying Maps

➤ To copy mapping tables:

1 **Set the POV to the location to be copied.**

2 Select **Activities > Maps**.

3 Select **Copy**.

4 **When prompted, select the target location.**

5 Click **OK**.

FDM copies the active conversion table from the current location to the target location.

Note: Only the active dimension conversion table is copied.

Note: Copy maps only from parent locations. Maps contained within child locations will not copy.

Deleting Maps

The feature to delete all maps can only be used from the parent location. Using Delete All from a child location will display a "success" message, but only the map from that location is deleted. The parent location will retain its map. Using Delete All from a parent location will delete the map from that location and all child locations.

Restoring Maps

When a file is imported for a category or period, FDM saves a copy of the mapping file that is associated with the import file.

➤ To restore maps that were used in previous periods or categories:

1 Click **Restore**.

The Map Restore dialog box is displayed. The dialog box lists mappings for all periods and categories that have been used for the current location.

2 **Select a map.**

The map is restored to the POV.

3 Click **OK**.

Recalculation of Logic Accounts and Maps

When a mapping or a logic account is modified, FDM must recalculate. The recalculation process first reprocesses all logic accounts and then, against the current mapping tables, remaps all imported source data. When FDM detects that recalculation is required, an orange calculate icon is displayed in the lower left corner of the POV bar.

At this point, you cannot load data into the target system until you re-validate the ledger data (by clicking the Validate link). You can click the calculate icon to force recalculation. However, even after a forced recalculation, FDM requires you to re-validate. If the location being recalculated is a parent location, when you click the calculate icon, you are asked whether you want to force recalculation of all child locations, in addition to the parent location.

Recalculation occurs only for the category and period set in the POV. This restriction ensures that the mapping audit trail remains intact for prior periods and other categories. If you want to apply a map or logic-account modification to other periods or categories, the POV must be changed, and a manual recalculation must be performed (by selecting Activities > Process Logic/ Maps). FDM reprocesses all logic accounts and remaps the imported source data against the current map.

Importing Mapping Tables

Mapping (conversion tables) can be imported from these sources:

- LedgerLink Conversion Tables
- Enterprise Conversion Tables
- Excel Maps

Importing LedgerLink Conversion Tables

FDM can read and import Hyperion LedgerLink Account and Name conversion tables. LedgerLink conversion tables must have the file extension `.tra` or the file extension `.trn`. FDM interprets ranges used in LedgerLink conversion tables as wildcards and generates the appropriate conversion table entries. Acceptable field delimiters are pipes (|), semicolons (;), exclamation points (!), and commas (,).

Note: Because LedgerLink extracts do not include location or dimension IDs, the POV location and the dimension to which the mapping table is to be loaded must be selected before import begins. FDM imports `.tra` and `.trn` files to the selected POV and dimension.

➤ To import LedgerLink conversion tables:

1 From **Dimension**, select the dimension into which to load the mapping table.

2 Select the **Upload File** tab.

3 Select the mapping table in one of two ways:

- If the preferred table is not listed in Inbox, browse for and select it.
- If the preferred table is listed in Inbox, from the **Inbox** tab, select it, and click **Select File**.

4 Click **Import** to complete the import of the mapping table.

5 Repeat steps 1–4 as many times as needed to load all mapping tables required for the application.

Conversion Between LedgerLink TRA and FDM

- This line creates a one-to-one map: `0011|CASH.01`
- During import into Hyperion Enterprise, the minus (–) sign in this line is used to convert the sign: `0100|-AP.01`
- In this line, the greater than sign (>) specifies a LedgerLink range; in FDM, the range is converted to Between: `0011>0099|CASH.01`
- In this line, the asterisk (*) specifies a LedgerLink like statement; in FDM, the asterisk is converted to Like: `0011*|CASH.01`
- This line is an Automap account entry; in FDM, it is converted to Like: `00110?|CASH.0?`

Conversion Between LedgerLink TRN and FDM

- This line is a one-to-one mapping: `200|DET_INP`
- In this line, the greater than sign (>) specifies a LedgerLink range; in FDM, the range is converted to Between: `220>250|CHI_INP`
- In this line, the asterisk (*) specifies a LedgerLink like statement; in FDM, the sign is converted to Like: `30*|MIL_INP`
- This line is an Automap Name entry; in FDM, it is converted to Like: `40?|DET40?`

Importing Descriptions in LedgerLink format

When importing a map file in the LedgerLink format (`.tra` or `.trn`), you can import account and center descriptions. Descriptions must be placed in the last field of the map file; for example, `0011,CASH.01,Bank America Acct` or `0011|CASH.01|Bank America Acct`.

Importing Hyperion Enterprise Conversion Tables

FDM can read and import Hyperion Enterprise account and entity conversion tables. The imported files must have the file extension `.asc`. Acceptable field delimiters are pipes (|), semicolons (;), exclamation points (!), and commas (,).

Conversion Tables and Sign Changes

The word *sub* designates that an account should have its sign changed; the sign change is indicated by a check being placed in the minus column of the account conversion table grid. Other values that indicate sign change are the minus sign (–) and the words *True* and *Yes*.

The word *add* designates that an account should not have its sign changed; the sign-change prohibition is indicated by no check being placed in the minus column of the account conversion table grid. Other values that indicate no sign change are the plus sign (+) and the words *False* and *No*.

```
54300.018!54300.018.USL!SUB 002!002.INP
54300.033!54300.033.US!ADD 005!005.INP
```

Importing Excel maps

You can import Excel maps by, on the Maps form, clicking the Import button and selecting an Excel map. Whether you use Import XLS functionality or use the Maps form, the same formatting rules apply within the Excel spreadsheet. Excel map templates with correct formatting are included in the `Outbox/Templates` directory.

12 Setting Up Financial Controls

In This Chapter

Certification Process

The certification process consists of reviewing and submitting user-defined questions that are associated with various locations at various levels within an organization. You can assign each location within a control structure a unique group of questions or assign a standard group of questions to all locations.

Overview of Financial Controls Groups

You can use financial controls groups to maintain and organize certification and assessment information. Therefore, controls groups can help you meet Sarbanes-Oxley requirements.

In the Controls Groups screen, each control-group type—Certification (302) and Assessment (404)—is represented by a folder and associated with a question recycle bin.

Creating Controls Groups

Certification (302) and Assessment (404) contain controls groups. Controls groups contain sections. Sections contain questions.

➤ To create controls groups:

1 Select **Metadata > Controls Groups**.

The Controls Groups screen is displayed.

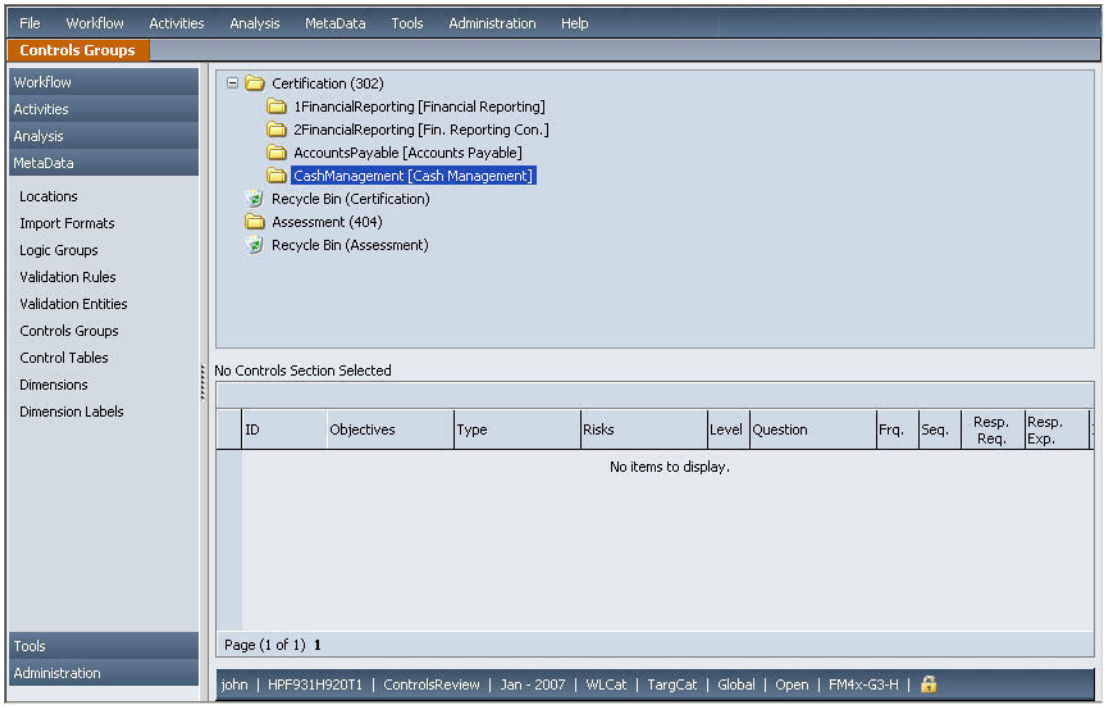
2 Right-click **Certification (302)** or **Assessment (404)**, and select **Add Controls Group**.

The Add Certification (302) Group or Add Assessment (404) Group dialog box is displayed.

3 For the new controls group, enter a name and description.

4 Click **OK**.

In the following Controls Groups example, Certification (302) contains four controls groups.



Creating Sections

➤ To create sections within controls groups:

1 In **Controls Groups**, under **Certification (302)** or **Assessment (404)**, right-click a controls group, and select **Add Controls Section**.

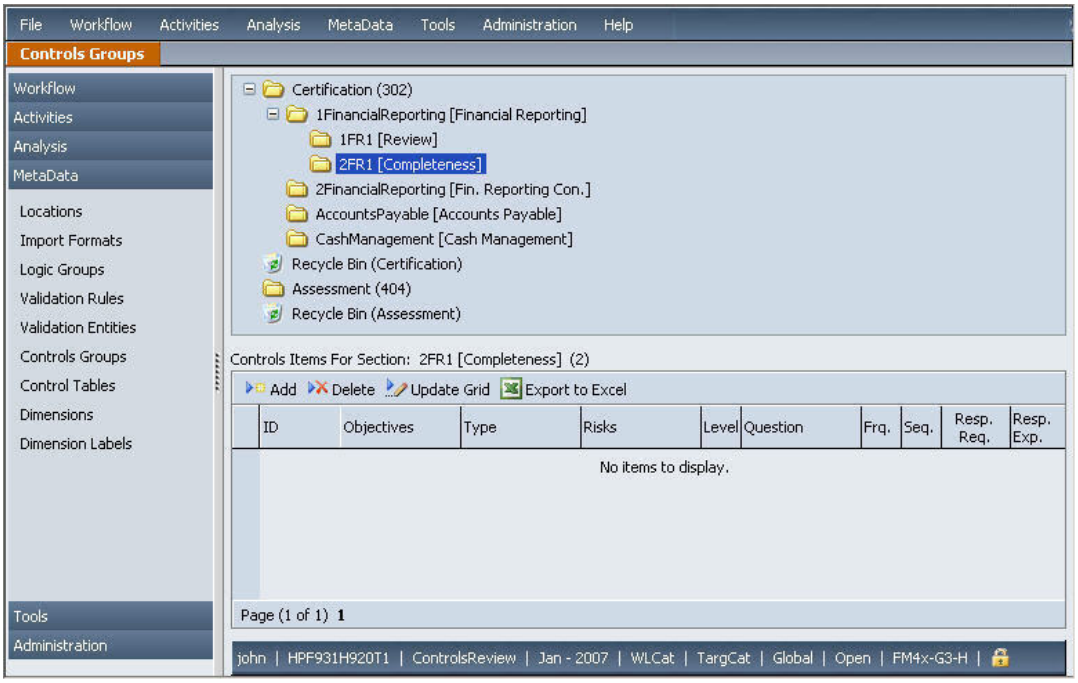
The Controls Section Properties dialog box is displayed.

2 For the new section, enter a name and description.

3 Click **OK**.

Note: Section names must be unique.

In the following Controls Groups example, the controls group 1Financial Reporting contains two sections.



Creating Questions

➤ To create questions within sections:

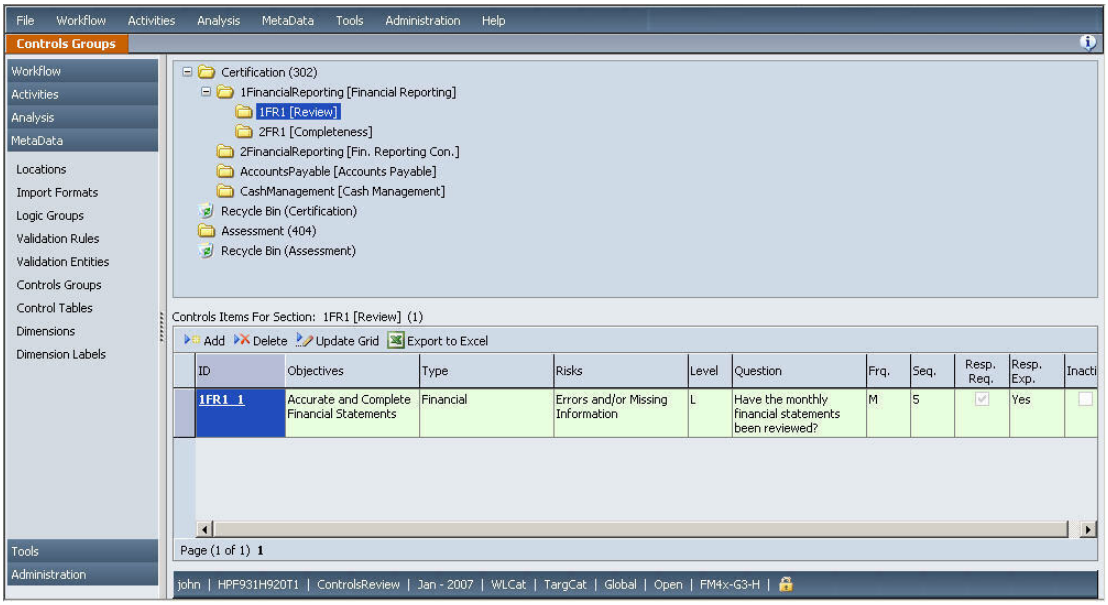
1 **In Controls Groups, under Certification (302) or Assessment (404), under a controls group, select a section.**

For the selected section, a question table is displayed.

2 **Click Add, and, as appropriate, within the fields, perform the following actions:**

- **ID**—Provide a unique ID that does not cannot contain periods.
- **Objectives**—State the purpose of the question.
- **Type**—Select one or more types: **Financial, Operations** or **Compliance**.
- **Risks**—Identify the factors that may prevent achievement of the objective.
- **Level**—Specify **L, L-M, M, M-H, H,** or **NA** (Low, Low-Medium, Medium, Medium-High, High, or Not Applicable, respectively) the likelihood of not receiving the expected response.
- **Question**—Compose the question.
- **Frq**—Specify **M, Q,** or **Y** (monthly, quarterly, or yearly)—the frequency with which you want the question answered.
- **Seq**—Specify the order in which you want the question presented.
- **Resp. Req.**—Select if a response to the question is required.
- **Resp. Exp.**—Select **Y** or **N** (yes or no) to identify the response that you expect.
- **Inactive**—Select if you do not want the question to be displayed or answered.

In the following Controls Groups example, a question within the 1FR1 [Review] section is displayed.



Importing Questions from Excel

You can import questions developed in Excel into FDM.

➤ To import questions from Excel:

1 **Log on to a FDM application using the Web client.**

2 **Select Metadata > Controls Groups.**

The Controls Groups screen is displayed.

3 **Select Tools > Import XLS.**

4 **If you want to import a file that is located in the application inbox:**

a. On the **Inbox** tab, click **Select File**.

b. Select a file.

c. Click **OK**.

d. Click **Import**.

5 **If you want to import a file that is located in a location other than the application inbox:**

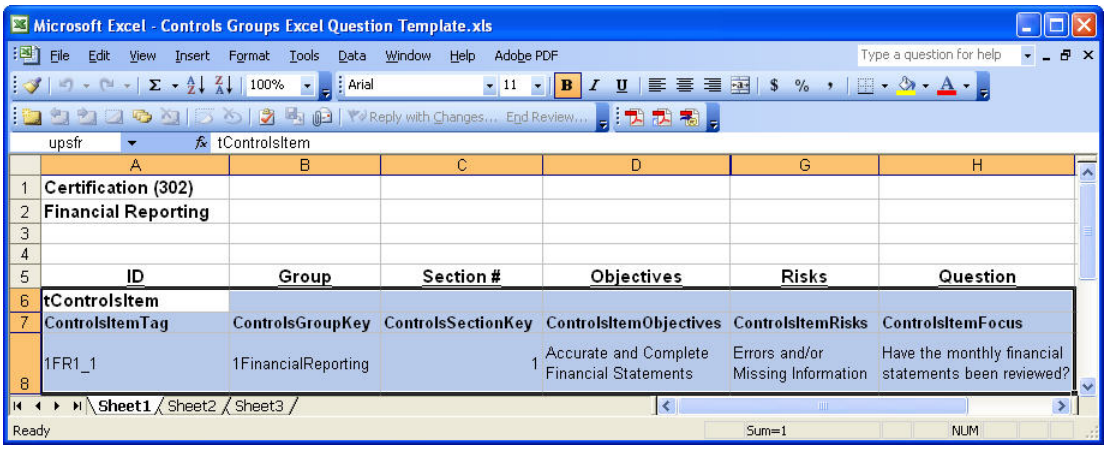
a. On the **Upload File** tab, click **Browse**.

b. Locate and select the file.

c. Click **OK**.

Creating Questions in Excel

The following question import template uses the import range name `upsfr` and contains a table named `tControlsItem`. The table includes only some of the available fields.



Tables must include the following fields that are marked *required* and may include any or all of the other fields.

- **ControlsItemKey**—Integer field key (required)
- **ControlsItemTag**—Question ID (required)
- **ControlsGroupKey**—Group (required)
- **ControlsSectionKey**—Section # (required)
- **ControlsItemObjective**—Text that identities the purpose of the question
- **ControlsItemType**—F (Financial), O (Operations), C (Compliance)
- **ControlsItemRisks**—Risks associated with the question
- **ControlsItemRiskLikelihood**—1 through 5
- **ControlsItemFocus**—Text of the question
- **ControlsItemCommentRequired**—True or False
- **ControlsItemSequence**—Order in which the question is presented
- **ControlsItemHasLinks**—True or False (required)
- **ControlsItemInactive**—True or False (required)
- **ControlsItemExpectedRespones**—Yes or No
- **ControlsItemActivationDate**—Date that the question starts beings used (required)
- **ControlsItemDeactivationDate**—Date that the question stops being used (required)

Each field that is to be imported must be included in a named range (expressed in the format `upsxxx`). You need not import all fields. You can import the general framework of a question and update in FDM.

Note: In Excel files, format dates as text.

Setting Effective Dates for Questions

Questions are first presented on their activation dates. Different questions can have different activation dates. For questions that are to be presented for all periods, activation dates are not required.

➤ To set activation dates for questions:

1 In **Controls Groups Excel Question Template**, click the **ID** link for a question.

2 Enter an effective date.

3 Click **OK**.

Deleting and Restoring Groups, Sections, and Questions

Questions deleted from certification sections are placed in Recycle Bin (Certification). Questions deleted from assessment sections are placed in Recycle Bin (Assessment).

To delete questions, you can delete groups, sections, or individual questions.

Items in recycle bins can be restored. Items deleted from recycle bins cannot be restored.

Deleting Groups, Sections, or Questions

➤ To send groups or sections to recycle bins:

1 In **Controls Groups**, right-click a group or section, and select **Delete Controls Group** or **Delete Controls Section**.

2 Click **OK**.

➤ To send individual questions to recycle bins:

1 In **Controls Groups**, select a section.

2 In the **Controls Items for Sections** table, select a question.

3 Click **Delete**.

The delete icon () is displayed at the left of the row to indicate that a deletion is pending.

4 Click **Update Grid**.

Restoring Groups, Sections, or Questions

➤ To restore groups from the recycle bin:

1 In **Controls Groups**, select **Recycle Bin (Certification)** or **Recycle Bin (Assessment)**.

2 Right-click a group, and select **Restore Controls Group**.

3 Click **OK**.

➤ To restore sections from the recycle bin:

1 In **Controls Groups**, select **Recycle Bin (Certification)** or **Recycle Bin (Assessment)**.

2 Select the folder named ***Deleted Sections**.

3 Right-click a section, and select **Restore Controls Section**.

4 Click **OK**.

The Select Controls Group dialog box is displayed.

5 From **Select Controls Group**, select the controls group in which to place the restored section.

6 Click **OK**.

➤ To restore questions from the recycle bin:

1 In **Controls Groups**, select **Recycle Bin (Certification)** or **Recycle Bin (Assessment)**.

2 Select the folder named ***Deleted Sections**.

3 Select the folder named ***Delete Questions**.

Deleted questions are displayed in the lower grid of the Controls Groups screen.

4 From the **Restore** column, click ( ).

5 Click **OK**.

The Select Controls Section dialog box is displayed.

6 From **Select Controls Section**, select the section to which to restore the question.

Creating Question Profiles

Controls groups can be organized into question profiles. Question profiles enable you to assign one set of questions to one location or to multiple locations. A profile can contain one or more controls groups and be designated as a certification or assessment profile.

➤ To create profiles:

1 Select **Metadata > Locations**.

The Locations screen is displayed.

2 Select the **Financial Controls** tab.

3 Select a location.

Note: Repeat this procedure to assign questions to additional locations.

4 In the **Certification (302)** or **Assessment (404)** menu, click **Edit**.

The Certification Profile ior Assessment Profile dialog box is displayed.

5 Click **Add**.

The Enter Profile Name dialog box is displayed.

6 For the new profile, enter a name.

7 Click **OK**.

8 From the **Certification Profile** dialog box, add controls groups to the profile:

a. From **Available Groups**, select a group.

b. Click  to add the group to the **Group Profile** table.

c. As needed, repeat steps a and c.

d. Click **OK**.

Assigning Submitters and Proxies to Locations

Each location is assigned a submitter and a proxy submitter (backup submitter). After all sections are reviewed, the submitter or proxy submitter submits the certification or assessment.

For certifications that are contingent on data loading, prior to submission, all data must be loaded and all sections reviewed.

➤ To assign submitters and proxies to locations:

1 On the **Financial Controls** tab of the **Locations** screen, select the **Submitter** or **Proxy** menu.

2 Select a user.

3 Click **Save**.

Upon logging on to a FDM application, users that are configured in FDM security as submitters are routed directly to Process Explorer.

Assigning Reviewers and Proxies to Sections

A reviewer and a proxy reviewer (backup reviewer) are assigned to each section of a certification or assessment—to answer the questions presented in the assigned section.

Each section can have its own reviewer, or one reviewer can be assigned to multiple sections.

➤ To assign reviewers and proxy reviewers to sections:

1 In the **Certification Reviewers** or **Assessment Reviewers** grid, double-click the **Reviewer** or **Proxy** column of a section.

2 Select a user.

3 Click **Update Grid**.

Upon logging on to a FDM application, users that are configured in FDM security as reviewers are routed directly to Process Explorer.

Overview of Risk Flag Levels

The Risk Flag Level field defines when red flags are displayed for unexpected responses. Questions are assigned a risk level of low (L), Low-Medium (L-H), Medium (M), Medium-High (M-H), High (H), and Not Applicable (NA). You set the risk flag level to indicate when you want an unexpected response to produce a red flag.

For example, a location with a risk flag level of medium (M) generates a red flag when a question with a risk level of medium or higher is answered with an unexpected response and does not generate a red flag when a question with a risk level of low to medium is answered with an unexpected response.

Working Within Process Explorer

➤ To access Process Explorer, perform an action:

- From the Web client, select **Analysis** > **Process Explorer**.
- After a location successfully loads data, click the Notepad icon (located to the right of the Check fish icon).

Within Process Explorer, locations are displayed in the left pane, and questions for the selected control are displayed in the right pane. The menu above the left frame controls what is displayed on the Process Explorer screen.

Items available in the Process Explorer navigation menu

- Certification Questions (Review Certification)
- Assessment Questions (Review Assessment)
- Certification Reports
- Assessment Reports
- Pull Assessment Answers from the Prior Period to the Current Period
- Audit Evaluations, Gap Analysis and Action Plans
- Status Reports
- View Journal Entries
- View Map Changes

Assigning Memos and Attaching Documentation to Memos

Supporting documentation can be attached to any question or focus point. FDM organizes documentation into two groups:

- **General documentation**—Any user with the appropriate security can attach general documents.

- **Audit documentation**—Only users who are granted Auditor status can attach audit documents. Auditor status is configured within FDM security configuration. Audit documentation is divided into the following sections:
  - Evaluation
  - Gap analysis
  - Action plans

  An unlimited number of memos can be assigned to one question or focus point. Ten supporting documents can be attached to each memo.

➤ To assign memos and attach documentation:

1. From the Web client, select **Analysis > Process Explorer**.

   The Process Explorer screen is displayed.
2. In the left pane of Process Explorer, select a **Controls Review** or **Data Load** location.
3. From **Section** (right pane), select a section.
4. From the memo column (in the column heading), click the memo link (— —) .

   The Memo Item dialog box is displayed.
5. Click **Add** (bottom-left button).
6. Enter a description of the memo item, and click **OK**.
7. In **Memo**, enter the text of the memo.
8. Click **Add** (to the right of an **Attachment** field), and browse to locate an attachment.
9. Select the attachment, and click **OK**.
10. Click **Update**.
11. Click **Close**.
12. **Optional:** To close Process Explorer, select **File > Home**.

Audit documentation includes a responsibility area for tracking and reporting audit memo items. You can designate Assigned To and Assigned By users. You can designate target dates for completion and verify when items are completed. Gap analysis and action plan memo items can be linked to the evaluation memo item from which they were derived. A gap analysis item can be assigned a value for ranking the significance or materiality of the identified gap.

Changing Memo and Audit Attachment Preferences

By default, memo and audit attachments are attached to the question, and not the answers. Therefore, memo and audit attachments added in any category or period will be displayed in all categories and periods in Process Explorer for that location. The default can be changed so that memos and audit attachments will only be displayed in the category and period POV in which they were added.

➤ To change Memo and Audit Attachment Preferences:

1 In the FDM Web client, select **Administration > Application Settings**.

The Application Settings screen is displayed.

2 From Options, select **Memo Item Filter with Category and Period**.

3 Select **0–None** or **1–Category_Period**.

4 Click **Save**.

Note: Any memo or audit attachments created prior to turning the filter setting on will not be affected.

13 Building Validation Rules

In This Chapter

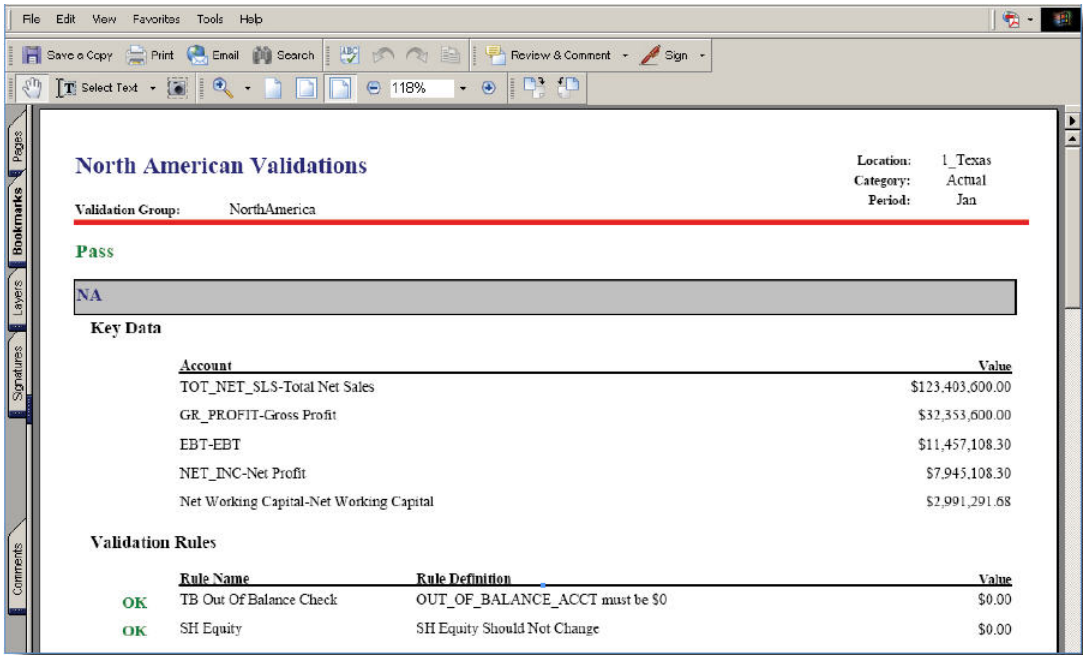
Overview of Validation Reports

System administrators use validation rules to enforce data integrity. A set of validation rules is created within a validation rule group, and the validation rule group is assigned to a location. Then, after data is loaded to the target system, a validation report is generated.

If a validation entities group is assigned to the location, the validation report runs for all entities that are defined in the group. If no validation entities group is assigned to the location, the validation report runs for each entity that was loaded to the target system. FDM validation reports retrieve values directly from the target system, FDM source data, or FDM converted data.

FDM analyzes the validation report and inserts a status entry in the process monitoring table. The location associated with the report shows a status of True only if all rules within the validation report pass. For rules used only for warning, no rule logic is assigned.

Validation reports run as data is loaded but can also be run manually. You can display validation reports, such as the following example, from the FDM toolbar or the Workflow menu.



Creating Validation Rule Groups

➤ To create validation rule groups:

1 From the Web client, select **Metadata > Validation Rules**.

The Validation Rules screen is displayed.

2 Click **Add** (located above the top grid).

A row is added to the top grid.

3 In **Validation Group**, double-click, and enter a name for the group.

4 **Optional:** In **Description**, double-click, and enter a description for the group.

5 Click **Update Grid**.

Note: A date is added in the Date Created column of the new validation group.

Creating Validation Rules

Each line of a validation report represents a validation rule.

➤ To create validation rules:

1 From the Web client, select **Metadata > Validation Rules**.

The Validation Rules screen is displayed.

2 In the top grid, select the validation rule group to which to add validation rules.

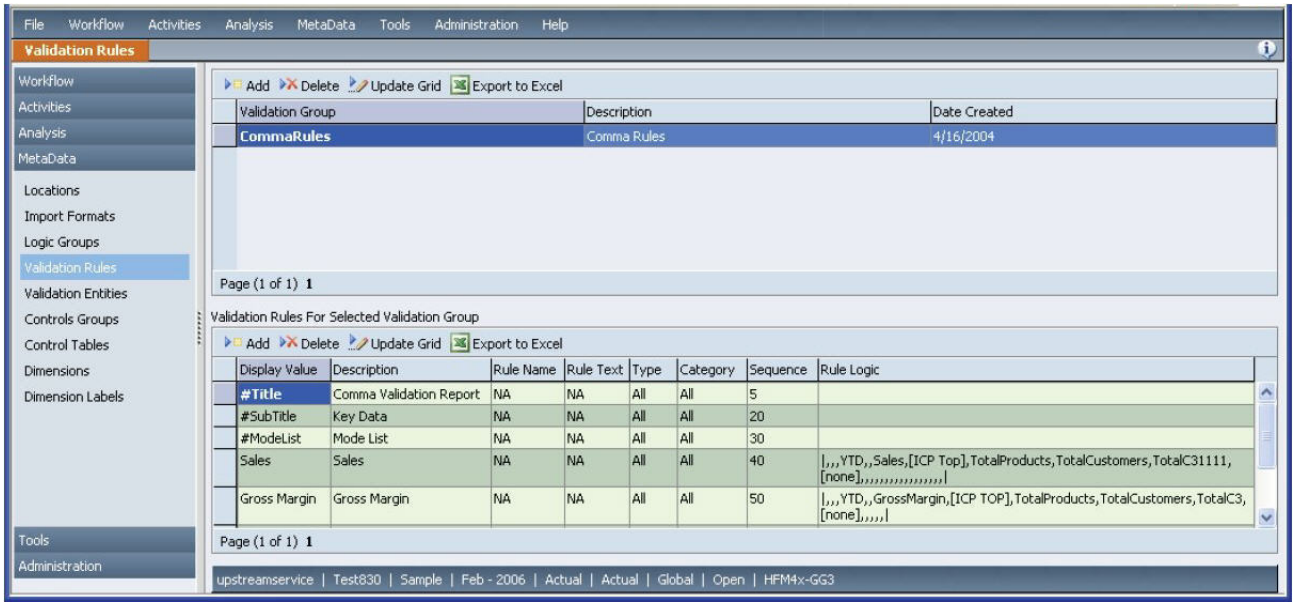
3 Click **Add** (located above the bottom grid).

A row is added to the bottom grid.

4 In each field, enter validation rule information:

- **Display Value**
- **Description** (optional)
- **Rule Name**
- **Rule Text**
- **Type**
- **Category**
- **Sequence**
- **Logic statement for the rule** (optional)

5 Click **Update Grid**.



Display Value Column

The Display Value column, which controls how FDM formats the data rows of validation reports, is used to select target accounts or report format codes or to create custom expressions.

Rules used to process Display Value columns:

- For fields that contain report format codes, no value lookup is attempted.
- For fields that contain other than report format codes, the result of the custom expression (rule logic) is displayed in the Value column. The Value column is limited to 75 characters.

For a rule in Edit mode, you can double-click within the Display Value field to display a menu that contains three active items: Zoom, Browse for Target Account, and Select Format Code.

Option—Zoom

This option, which opens field text in a text editor, is helpful for entering or editing large amounts of text.

Option—Browse for Target Account

This option, which displays the Validation Rule Target Account Lookup screen, enables you to insert an account (from a list of target-system application accounts) into the validation rules form.

Option—Select Format Code

This option, which displays the Select Format Code screen, enables you to enter format codes into the Target Account column.

Format codes are used to control the presentation of validation reports.

Format Code	Action Performed on Validation Reports
`#ModeList`	Sets the report to display the Display Value, Description, and Amount column values.
`#ModeRule`	(Default) Sets the report to display the Rule Name, Rule Text, and Amount column values The report evaluates each expression of the Rule Logic column and tests the true or false condition of each rule. The status of each rule (`OK` or `Error`) is displayed in the report.
`#Title`	Inserts the text of the associated Description field as a title line on the validation report
`#Subtitle`	Inserts the text of the associated Description field as a subtitle line on the validation report

Description Column

The Description column, which is displayed only for validation reports in `#ModeList` mode, displays account descriptions (which may be designated as titles or subtitles).

Example—Description

```
Out-of-Balance Account
```

Rule Name Column

The Rule Name column, which is displayed only for validation reports in `#ModeRule` mode, stores identifiers for validation rules. Rule Name values should be unique and easy to identify.

Example—Rule Name

```
Out-of-Balance Check
```

Rule Text Column

The Rule Text column, which is displayed only for reports in `#ModeRule` mode, defines the logic behind rules. In validation reports, the primary statement for a rule is the text of the Rule Text field associated with the rule.

Example—Rule Text

```
This account must be between [+10 and -10].
```

Type Column

For each validation rule, you select, in the Type column, to display the rule for base entities, parent entities, or both base and parent entities:

- If Parent is selected, the validation rule is displayed in the validation report only for entities for which, in the Validation Entity screen, the Type field is set to Parent or All.
- If Base is selected, the associated rule is displayed in the validation report only for entities for which, in the Validation Entity screen, the Type field is set to Base or All.
- If All is selected, the associated validation rule is displayed in the validation report for all entities, regardless of whether, in the Validation Entity screen, an entity is set to Parent or Base.

Category Column

In the Category column, you select an FDM category to restrict a validation rule to one FDM category. The rule is displayed in the validation report only if the FDM category that is selected in the Category field associated with the rule is the FDM category set in the POV. To display the validation rule in the validation report regardless of the category set in the POV, you must select All.

Sequence Column

Sequence column values (numbers) determine the order in which format codes and rules are processed. It is good practice to increment sequence numbers by 10—to provide a range for insertion of format codes and rules.

Rule Logic Column

The Rule Logic column is used to create multi-dimension lookups and validation rule expressions and functions. Rule Logic columns are processed only for reports in `#ModeRule` or `#ModeList` mode. After rule logic is processed for a rule, in the validation report, FDM flags the rule as passing or failing.

Validation Rule Expressions

Validation rule expressions are used primarily to validate target-system account balances. The expressions, which use standard expression capabilities of Microsoft VB Script, return a True or False result.

For example, the following expression returns True if the value of 1000.100 (a target account) plus $100,000 equals the value of 1000.400 (another target account) and false if it does not:

```
|1000.100| + 100000 = |1000.400|
```

Validation rule expressions can incorporate validation rule functions: the Hyperion Enterprise lookup function (when integrating with Hyperion Enterprise) or the multi-dimension lookup function (when integrating with multi-dimension target systems).

Enterprise Lookup Function

The following validation rule expression uses the Hyperion Enterprise lookup function. If the target account balance is greater than –$10 and less than $10, in the validation report, the rule is flagged `OK`. If the account balance is outside the specified range, in the validation report, the rule is flagged `Error`:

```
|Balance| >= -10.00 AND |Balance| <= 10.00
```

Multi-Dimension Lookup Function

The multi-dimension lookup function retrieves account values from the target system, FDM source data, or FDM converted data. Multi-dimension lookups can be used in Rule Logic columns and in the Display columns of validation rules forms.

Rule Data Sources

FDM can retrieve data from three sources:

- Target-system data
- FDM source data
- FDM converted data

Target System Data

The following format, which begins and ends the rule with the pipe character (|), enables FDM to retrieve target-system values for any dimension.

Unless otherwise specified, parameters are optional.

```
|Scenario, Period, Year, View, Value, Entity, Account (Required), ICP, Custom1,
Custom2,Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10,
Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18,
Custom19, Custom20|
```

The following examples illustrate ways that target-system values can be retrieved. In each example, Balance is a target account. For dimensions that are not referenced, commas must be used as placeholders.

Example 1

Look up the value of Balance for the target period and scenario (category) set in the POV and for each entity of the FDM validation entity group that is assigned to the location. The example rule passes validation if the target account is less than $10.

```
|,,,,,Balance,,,,,,,,,,,,,,,,,,,,,| > -10.00 AND |,,,,,Balance,,,,,,,,,,,,,,,,,,,,,| < 10.00
```

Example 2

Look up the value of Balance for the specified dimensions.

```
|Actual,March,2002,YTD,Ohio,Balance,Michigan,Engines,Ford,Trucks,
[None],,,,,,,,,,,,,,,,USD| > 0
```

Example 3

Look up the value of Balance for the specified dimensions and the prior period.

```
|Actual,-1,2002,YTD,Ohio,Balance,Michigan,Engines,Ford,Trucks,
[None],,,,,,,,,,,,,,,,USD| > 0
```

Example 4

Look up the value of Balance for the target scenario (category) set in the FDM POV, the prior target period, and each entity of the FDM validation entity group that is assigned to the location.

All missing custom dimensions default to [None]. The ICP dimension defaults to [ICP-None]. The Year dimension defaults to the year set in the POV. The Currency dimension defaults to 0. The View dimension defaults to YTD.

```
|,-1,,,,Balance,,,,,,,,,,,,,,,,,,,,| > 0
```

FDM Source Data

The following format, which begins and ends the rule with the tilde character (~), retrieves values from data that was mapped to a target member and then loaded into FDM.

Unless otherwise specified, parameters are optional. Parameters designated `UD#` are user-defined.

```
~FDM Category, FDM Period, Year (Field Not Applicable), FDM View, FDM Location, Source
Entity(Required), Source Account(Required), Source ICP, Source UD1,Source UD2, Source
UD3, Source UD4, Source UD5, Source UD6, Source UD7,Source UD8, Source UD9, Source UD10,
Source UD11, Source UD12, Source UD13,Source UD14, Source UD15, Source UD16, Source
UD17, Source UD18, Source UD19, Source UD20~
```

FDM Converted Data

The following format, which begins and ends the rule with the grave accent character (`), retrieves pull values from data that was loaded into FDM. Unless otherwise specified, the parameters are optional.

```
`FDM Category, FDM Period, Year (Field Not Applicable), FDM View, FDM Location,
Entity(Required), Account(Required), ICP, Custom1, Custom2, Custom3, Custom4, Custom5,
Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14,
Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`
```

Note: Beginning and ending characters for rules can be set to any character. You change the characters in the Configuration Settings screen.

Validation Rule Functions

In Rule Logic columns, you can use validation rule functions, in addition to validation rule expressions. FDM validation rules can use VB Script procedures. Therefore, within rules, you can set, for evaluation by a script or custom report, two custom string message fields and two numeric switch fields. Also, within rules, you can perform validations against data sources other than the target system (for example, a FDM logic account value).

Keywords Used in Validation Rule Functions

You must begin rules that use functions with the keyword `Fun`. The keyword triggers the API to interpret the rule as a function, rather than as an expression.

You must use the keyword `Result` to set the return values of functions. In validation reports, rules are flagged `OK` (if `Result` returns True) or `Error` (if `Result` returns False).

Parameters Used in Validation Rule Functions

The parameters of validation rule functions can be evaluated to determine whether, in regard to the current POV, validation rules pass or fail:

- **API**—Allows the use of all FDM data values and internal API functions
- **strEntity**—Target Entity
- **strTargCat**—Target Category
- **strTargPer**—Target Period
- **strTargYear**— Target Year
- **strTargFreq**—Target Frequency

Properties Used in Validation Rule Functions

As shown in the following script, you can set properties for RES objects which, in turn, set values for the 14 custom fields of the `tDataCheck` table. Custom-field values can be used within reports as warning messages and flags.

```
Fun:
If strEntity = "TEXAS.GL" Then
        'Rule not required
        Result = True
        'Set Messages and Switches
        RES.PstrCheckMessage1 = "Rule does not apply to " & strEntity
```

```
        RES.PstrCheckMessage2 = "Do not fret!"
        RES.PblnCheckWarning = True
        RES.PblnCheckClearData = True
Else
        'Check Cash
        IF |1000| <= 0 Then
            Result = True
            'Set Messages and Switches
            RES.PstrCheckMessage1 = strEntity & " is Good!"
            RES.PstrCheckMessage2 = "2"
        Else
            Result = False
        'Set Messages and Switches
        RES.PstrCheckMessage1 = strEntity & " is Bad!"
```

Using Validation Editor to Create Validation Rules

Validation Editor facilitates creation of validation rules. It helps you develop rule logic and enables you to create rules from Validation Editor, rather than from the Validation Rules screen. You can also use Validation Editor to modify validation rules.

➤ To open Validation Editor:

1 From the Web client, on the validation rules form, within the **Rule Logic** column, double-click.

2 Select **Validation Editor**.

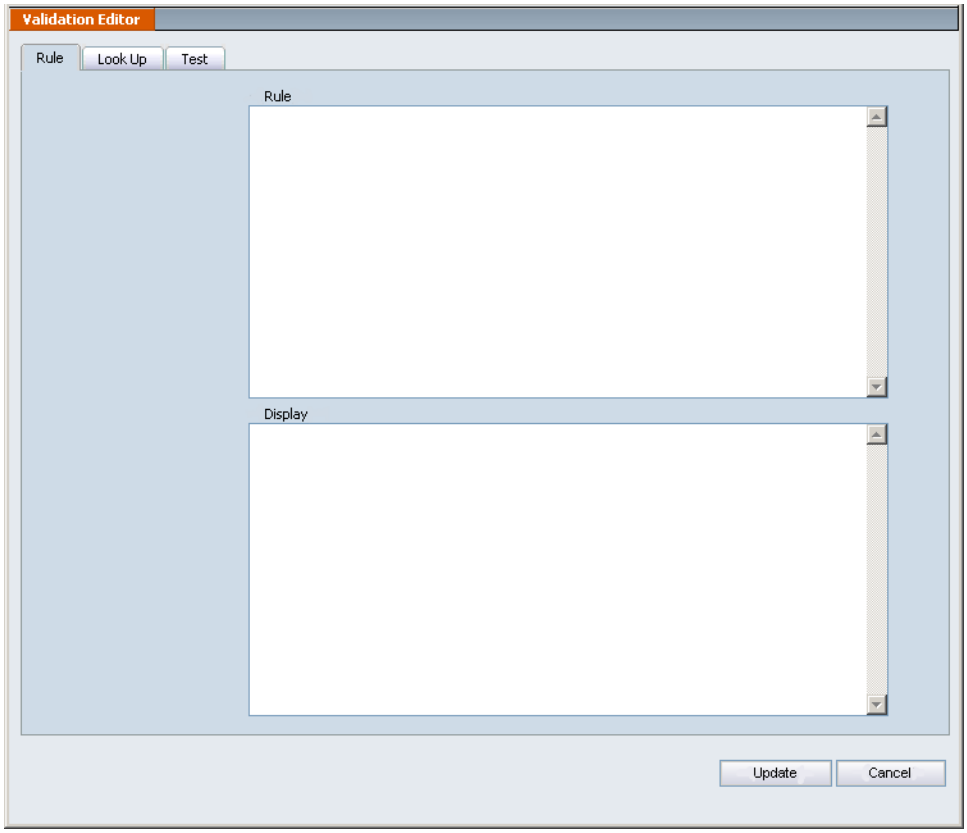
Validation Editor includes three tabs:

- Rule
- Look Up
- Test

Rule Tab

The Rule tab, which you use to create or modify validation rules, contains the following elements:

- **Rule**—Area in which the rule to be tested is created
- **Display**—Area in which the value to be returned to the validation report is created or modified (may differ from the value to be tested in the rule)
- **Update**—Button to save the contents of the Rule and Display areas to the validation rule being created or modified
- **Cancel**—Button to close Validation Editor without saving changes



Lookup Tab

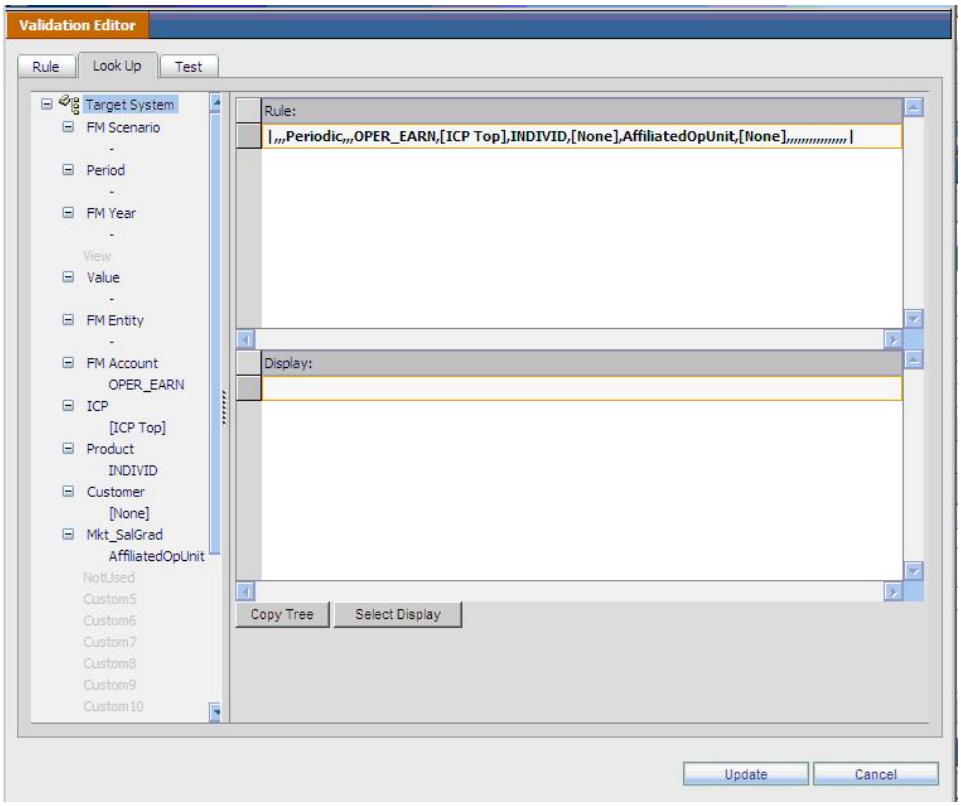
You use the Lookup tab, from which you can select retrieve dimensions directly from the target system, to ensure that required dimensions are entered and ordered correctly.

The Lookup tab contained the following elements:

- **Rule** and **Display**—Areas that, respectively, display the rule and the display rule that are being created. The rules are composed of Retrieve functions (one per line) and VBA script calls (displayed apart from the Retrieve functions). Therefore, in both areas, you can select individual Retrieve functions.
- **Rule Intersection**—A tree that becomes available when a Retrieve function is selected from the Rule or Display area. You use the tree to enter dimension values for the selected Retrieve function. The text for active dimensions is blue, and the text for inactive dimensions is gray. You double-click a dimension to display a list of valid selections.
- **Copy Tree** or **Paste to Editor**—button used to copy and paste dimension values. The copy or paste action is applied to the rule selected when the button is clicked. Therefore, to copy from one rule and paste to another rule, you must change the rule selection between the copy and paste actions. After copying, the action of the button changes to paste (caption becomes **Paste to Editor**). After clicking Paste to Editor, the action of the button toggles back to copy (caption returns to **Copy Tree**).

- **Select Display** or **Select Rule**—Toggle button to change the focus to the Rule or Display areas. When the Rule area is active the button caption is **Select Display**. When the Display area is active, the button caption is **Select Rule**.

  When the Display area is active, all interaction takes place in the Display area. For example, pasting a tree will copy all the tree members to the Display area. When the Rule area is active, all interactions take place in the Rules area. Any changes that have been made on the Lookup tab will be reflected on Rule tab in related editors.



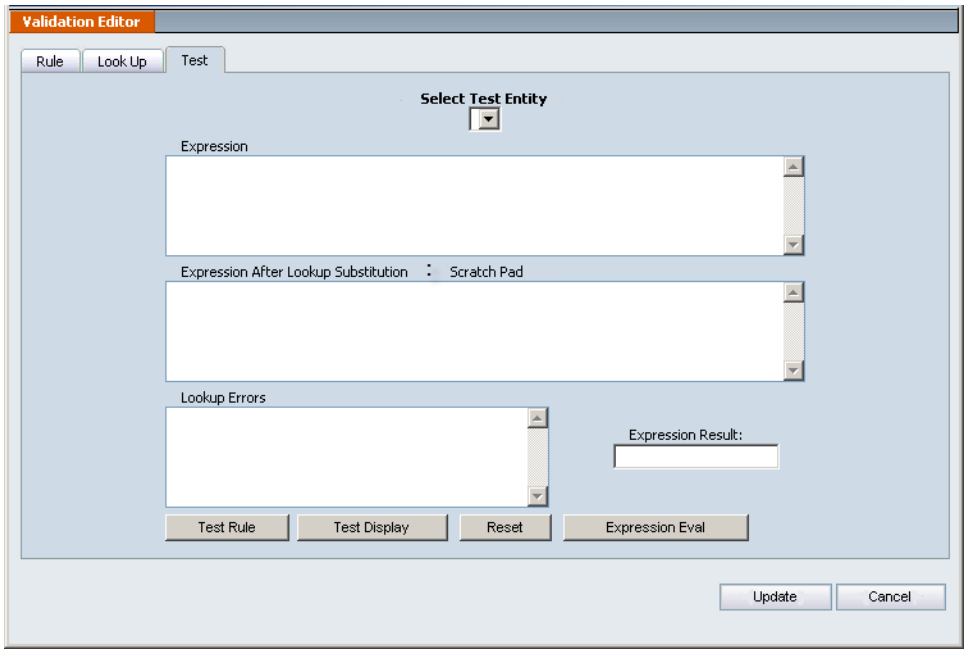
Test Tab

The Test tab, which you use to test the rule in the Rule or Display area of the Rules tab, contains the following elements:

- **Select Test Entity**—List from which you select the test entity
- **Expression**—Area that displays the rule being tested
- **Expression After Lookup Substitution (Scratch Pad)**—Area that displays the results of the rule being tested (You can use the area for temporary storage of text and expressions, by selecting text and right-clicking.)
- **Lookup Errors**—Area that displays errors that were found in the rule being tested
- **Expression Result**—Area that displays the results of the rule test (True or False)
- **Test Rule** and **Test Display**—Buttons that are used to run, respectively, the rule in the Rule or Display area on the Rule tab

When a button is clicked, a list of available target locations is displayed. You select the location from which values are pulled and, respectively, rule or display results are returned.

- **Reset**—Button used to return all areas of the Rules tab to default values.
- **Expression Eval**—Button used to calculate the expressions in the Expression After Lookup Substitution (Scratch Pad) area



Restricted Characters

Do not use a semicolon (;) in validation rules. The semicolon is a reserved name as the separator between the rule value and the display value.

Creating Validation Entity Groups

A validation entity group consists of one or more target-system entities. When a validation report is generated, the entities of the entity groups that are assigned to the report location are consolidated and displayed in the report. You activate validation entity groups by assigning them to locations. You define the validation entities of a validation entity group by entering values in the columns of the validation entities form of the Validation Entities screen.

➤ To open the **Validation Entities** screen, from the FDM Web client, select **MetaData** > **Validation Entities**.

Organization or Parent Column

For Hyperion Enterprise target systems, you select the organization in which the entity is consolidated. For other target systems, you select the parent of the entity. If the Consolidate option is not selected, the selection is irrelevant.

Entity Column

The Entity column identifies the target entity to be consolidated and displayed in the validation report. If the Consolidate option is selected, the entity is consolidated before it is displayed in the validation report.

Consolidate Column

This option selected in the Consolidate column determines whether an entity is consolidated prior to being displayed in the validation report.

Start Period Column

The Start Period column controls how FDM executes the consolidation process within the target system. If the column entry is 0 or a number greater than the number of the current period, only the current period is consolidated. If the entry is a number less than the number of the current period, the periods between and including the period of the specified number and the current period are calculated. For example, a start period of 1 forces FDM to consolidate from the beginning period to the current period.

Type Column

The Type column determines which entries are displayed for which validation rules:

- Base—Displayed only for rules for which the Type field entry (in the validation rules form) is Base
- Parent—Displayed only for rules for which the Type field entry (in the validation rules form) is Parent
- All—Displayed for all rules

On Report Column

The option selected in the On Report column determines whether an entity is displayed in the validation report. If On Report is not selected and Consolidate is selected, the entity is consolidated but not displayed.

Sequence Column

The Sequence column controls the order in which entities are consolidated and displayed in the validation report.

It is good practice to increment the sequence number by 10, to provide a range for insertion of entities.

14 Processing Data

In This Chapter

The FDM process flow consists of four main operations:

1. Import—the source data (GL) is imported into FDM.
2. Validate—ensures that all data in the imported GL is has a corresponding mapping. Unmapped items must be assigned to a target account before proceeding to the Export step.
3. Export—loads the mapped GL data to the target application.
4. Check—validates the data loaded to the target application using the validation rules (if applicable).

Step 1: Importing Source Data

After an import format is created and assigned to a location, you can import source data.

When directed to import a source file, FDM uses the current POV to determine location, category, and period and conducts the following process:

1. If the Replace option is selected, deletes the existing file
2. Loads the source file
3. If applicable, processes logic accounts
4. Maps source dimensions to target dimensions

➤ To import source data:

1 **From the FDM Web client, select Workflow > Import.**

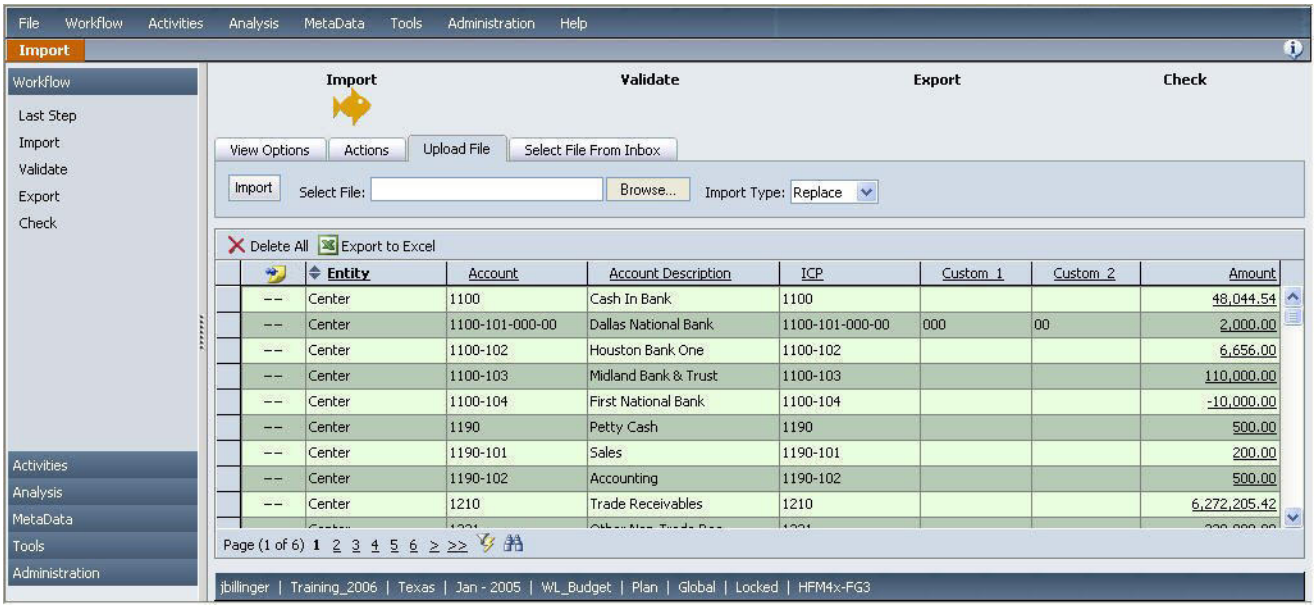
The Import screen is displayed.

2 **Depending upon whether you want to upload the data file from the application `Inbox` or from another location, select the Upload File tab or the Select File from Inbox tab.**

3 **Perform an action:**

- If you selected **Upload File**, click **Browse,** locate the file, click **Open**, then click **Import**.
- If you selected **Select File from Inbox**, click **Select File**, select a file, then click **OK**.

  FDM uses the import format assigned to the location, imports the source file, and displays an import form similar to the following example:



4 From **Import Type:**

- To delete existing data from the current location, category, and period, select **Replace**.
- To append data to (not merge data within) the current location, category, and period, select **Append**.

Viewing Import-Form Mappings

➤ To view the mappings of the detail lines of import forms:

1 **In an import form, within a detail line, click the link in the Amount field.**

2 **Select Show Conversion Rules.**

Reviewing Import Audit Information

FDM archives all source files, import logs, journal entries, target-system load files, and attached memos and documents. The items are stored in the `Data` directory of the application. Each item is assigned a unique name. The items, which are retrievable from the Import screen, provide an audit trail.

Viewing Archived Items from Import Forms

➤ To view archived items:

1 **In an import form, within a detail line, click the link in the Amount field.**

2 **Perform one or more actions:**

- To view source-file information, select **Show Archive Information**.
- To view the source file, select **Open Source Document**..
- To view the import log that was created when the source file was loaded, select **Open Processing Log**.

Restoring Source Files and Logs

➤ To restore source files to the archive directory:

1 **On an import form, on a detail line, right-click.**

2 **Select Restore Source Document.**

The source file and its associated import log is restored to the `Inbox\ArchiveRestore` directory.

Creating Memos to Import-Form Detail Lines

➤ To create memos to the detail lines of import forms:

1 **On an import form, click within the first column of a detail line.**

The Memo dialog box is displayed.

2 **Click Add.**

3 **Enter a description of the memo, and click OK.**

4 **In Memo, enter the memo.**

5 **Optional: To attach documents to the memo:**

a. Click **Add**.

b. Browse for and select a document.

c. In the text box to the left of the **Add** button, enter a description of the document.

d. As needed, repeat steps a–c (You can attach up to 10 documents per memo).

6 **Click Update.**

On the import form, a memo icon is displayed in the first column of the detail line to which the memo was attached. All memos and documents are stored `Data` directory of the application.

7 **Optional: Select a document, and perform one or more actions:**

- To display information about the document, select **Show Archive Information**.
- To display a document in its original format, click **Show Document**.
- To restore a document to the `Inbox\ArchiveRestore` directory, select **Restore Document**.

You can display document information or restore documents at a later time by clicking the memo icon with which a document is associated, and, in the Memo dialog box, performing the preferred actions.

➤ To delete memos and all attachments to the memos:

1 **From the Import form, select a memo (memo icon in the first column of a detail line).**

The Memo form is displayed.

2 **Click Delete.**

3 **Click Close.**

➤ To delete attachments:

1 **From the import form, select a memo (memo icon in the first column of a detail line).**

The Memo form is displayed.

2 **Next to an attachment, click Remove.**

Accessing Attribute -Up Forms from Import Forms

In addition to the custom dimensions UD1 through UD20, FDM features 14 custom attribute dimensions. The 14 dimensions cannot be mapped but can be used for custom data-warehousing tasks.

Each detail line of an import form can have attributes that were loaded during the import process. Data loaded to attribute dimensions can be viewed from the import form only by opening an attribute Drill Up form.

➤ To view attribute Drill Up forms:

1 **In an import form, click the link in the Amount field of a detail line.**

2 **Select Show Attributes.**

Step 2: Validating Source Data

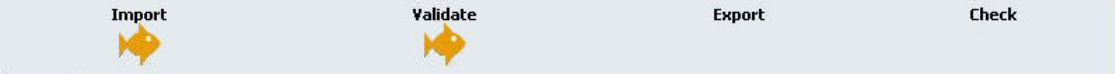
FDM forces the validation of the source data against the mapping tables. Validation compares the dimension mapping to the source file and identifies unmapped dimensions. The process flow cannot continue until all dimensions are properly mapped.

➤ To run the validation process, at the top of the main FDM screen, select **Validation**.

➤ To display the Validate screen without running the validation process, select **Workflow** > **Validation**.

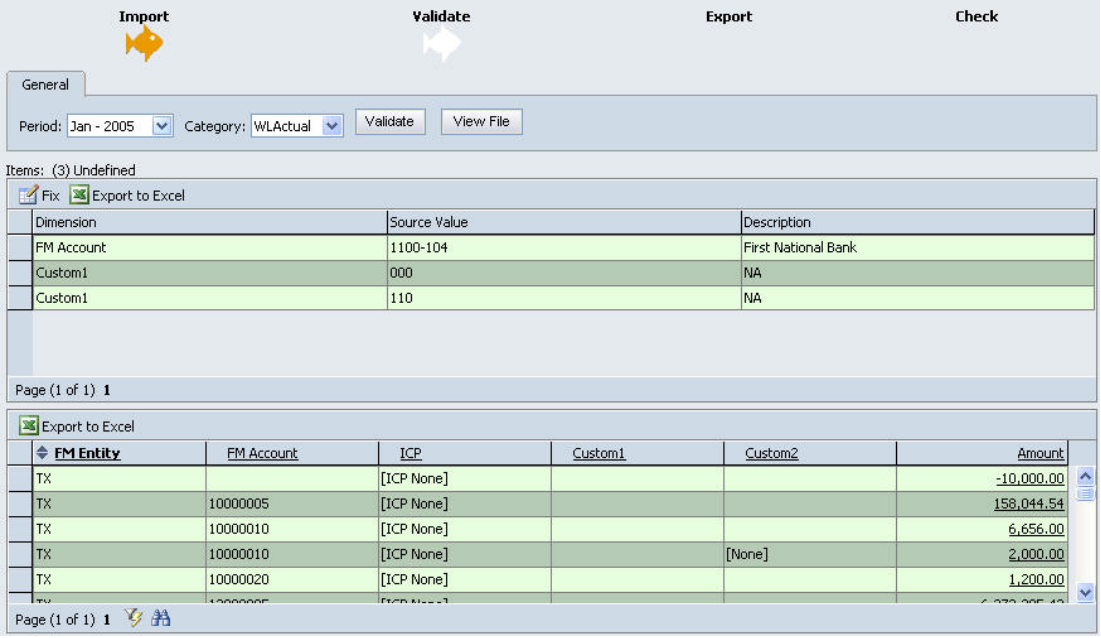
Validation Forms with No Mapping Errors

When validation is successful, the orange Validate fish is displayed in the header of the FDM desktop.



Validation Forms with Mapping Errors

Because newly added accounts may be unmapped, the addition of accounts to GL systems can produce validation errors. If an account is unmapped, the Validate fish is white. The Validation screen shows the number of accounts that are not mapped (that are, therefore, undefined).



The validation process displays the validation form, which lists all dimension values that are missing from the mapping tables.

➤ To correct conversion-table errors:

1 **In a validation form, highlight a row that requires correction.**

2 **Click Fix.**

The Map form is displayed, and the unmapped item is inserted into the Source column.

3 **Click Browse, and locate and select a target mapping.**

4 Repeat steps 2 and 3 until all incorrect items are corrected.

5 Click **Validate** to refresh the validation form.

Source data that passes the validation process can be loaded to the target system.

Step 3: Exporting Data to Target Systems

After source data passes the validation process, a load file is created.

➤ To export to target systems:

1 From the header of the FDM desktop, , click **Export**.

FDM creates the export file (in the format required by the target system) and places it in the `Outbox` directory of the application.

The Target System Load dialog box is displayed. Options that are displayed in the Target System Load Dialog box are adapter-specific. Refer to the target application documentation for information about the options.

Note: To display the Export screen without running the export process, select Workflow > Export.

2 Click **OK**.

Viewing Target-Line Source Accounts

➤ To display the source accounts that compose a target-line item, in an export form, select an account.

Attaching and Viewing Memos and Documents from Drill Down Forms

➤ To attach memos or documents:

1 On the Export form, click an amount.

The detail lines for the amount are displayed.

2 Click the Memo column of a detail line (first column).

The Memo Item dialog box is displayed

3 Select **Add** (the one at the bottom of the left pane) to create a new memo.

The Explorer User Prompt dialog box is displayed.

4 In the Explorer User Prompt dialog box, enter a description for the memo item, then click **OK**.

5 In the Memo field of the Memo Item dialog box, enter the memo.

6 **To attach a memo, click Add (one of the Add buttons in the right pane of the Memo Item dialog box).**

The File Browser dialog box is displayed.

7 **Browse for the file to attach and click OK.**

8 **Click Upload.**

9 **Click OK.**

FDM returns to the Memo Item dialog box.

10 **Click Update**

11 **Click Close.**

➤ To view attached memos or documents:

1 **On the Export form, click an amount.**

The detail lines for the amount are displayed.

2 **Click the Memo column of a detail line.**

The Memo Item dialog box is displayed

3 **From the left pane of the Memo Item dialog box, select the memo to view.**

In the right pane of the Memo Item dialog box, the memo ID, Memo text, and any attachments for the selected memo are displayed.

4 **To view the attachment, click the attachment link and select Show Document.**

Drilling Through from Hyperion Enterprise

Hyperion Enterprise 6.5.1 users can use Audit Intersection to view the sources used to load the Hyperion Enterprise intersections.

➤ To view the sources used to load Hyperion Enterprise intersections:

1 **In Hyperion Enterprise, right-click an amount.**

2 **Select Audit Intersection.**

FDM launches and the Drill Through from Target System screen is displayed . The top pane of the Drill Through window lists all sources that were loaded to the Hyperion Enterprise. The bottom pane (Summary tab) provides information about the item that is selected in the top pane. From the Drill Through window, you can access the FDM Drill Back window by clicking an amount in one of the rows in the top form. See "Viewing Target-Line Source Accounts" on page 142 for instructions.

See the Hyperion Enterprise documentation for information regarding enabling Drill Through functionality in from within Hyperion Enterprise.

Note: FDM allows loading to members or aliases. FDM only stores the information that is loaded into the target system. Drill Through will fail for information loaded to aliases. Oracle does not recommend alias mapping.

Drilling Through from Financial Management

Financial Management users can use Drill Through to FDM to view the sources used to load Financial Management intersections. Intersections that have been loaded by FDM or ERP Integrator are flagged to indicate they contain a drillable region.

➤ To view the sources used to load Financial Management intersections:

1 In Financial Management, right-click a cell.

2 Select **Drill Through to FDM**.

FDM launches and the Drill Through from Target System screen is displayed.

The top pane of the Drill Through window lists all sources that were loaded to the Financial Management intersection. The bottom pane (Summary tab) provides information about the item that is selected in the top pane.

From the Drill Through window, you can access the FDM Drill Down window by clicking an amount in one of the rows in the top form. See "Viewing Target-Line Source Accounts" on page 142 for instructions.

Note: FDM allows loading to members or aliases. FDM only stores the information that is loaded into the target system. Drill Through will fail for information loaded to aliases. Oracle does not recommend alias mapping.

Drilling Through from Essbase Studio

Oracle Essbase Studio users can use Drill Through to FDM to view the sources used to load the Essbase intersections.

Note: This feature is only available for Essbase data viewed in Oracle Hyperion Smart View for Office, Fusion Edition11.1.1 or higher for cubes built with Essbase Studio. Refer to the Essbase Studio User's Guide for detailed instructions on this feature.

Note: Essbase must be configured to enable this feature. Refer to the Oracle Essbase Studio Users's Guide to detailed configuration information.

FDM allows loading to members or aliases. FDM only stores the information that is loaded into the target system. Drill Through will fail for information loaded to aliases. Oracle does not recommend alias mapping.

Drilling Through from Planning

Planning users can use Drill Through to Source to view the sources used to load the Planning intersections. Intersections that have been loaded by FDM or Oracle Hyperion Financial Data

Quality Management ERP Integration Adapter for Oracle Applications are flagged to indicate they contain a drillable region.

➤ To view the sources used to load Planning intersections:

1 **In Planning, right-click a base-level amount that was loaded from FDM.**

2 Select **Drill Through to Source**.

FDM launches and the Drill Through from Target System screen is displayed.

The top pane of the Drill Through window lists all sources that were loaded to the Planning intersection. The bottom pane (Summary tab) provides information about the item that is selected in the top pane.

From the Drill Through window, you can access the FDM Drill Down window by clicking an amount in one of the rows in the top form. See "Viewing Target-Line Source Accounts" on page 142 for instructions.

Note: In order to use this function, it must be enabled from within Planning. Refer to the Oracle Hyperion Planning, Fusion Edition documentation for information regarding Drill Through to FDM.

FDM allows loading to members or aliases. FDM only stores the information that is loaded into the target system. Drill Through will fail for information loaded to aliases. Oracle does not recommend alias mapping.

Drilling Through from Strategic Finance

Oracle Hyperion Strategic Finance, Fusion Edition users can use Audit Intersection FDM to view the sources used to load Strategic Finance intersections.

➤ To view the sources used to load Strategic Finance intersections:

1 **On the Accounts tab of a Strategic Finance scenario (client application), right-click an amount that was loaded from FDM.**

Intersections loaded by FDM are designated by an orange arrow in the lower-right corner of the cell.

2 Select **Audit Intersection**.

FDM launches and the Drill Through from Target System screen is displayed.

The top pane of the Drill Through window lists all sources that were loaded to the Strategic Finance intersection. The bottom pane (Summary tab) provides information about the item that is selected in the top pane.

From the Drill Through window, you can access the FDM Drill Down window by clicking an amount in one of the rows in the top form. See "Viewing Target-Line Source Accounts" on page 142 for instructions.

Note: FDM allows loading to members or aliases. FDM only stores the information that is loaded into the target system. Drill Through will fail for information loaded to aliases. Oracle does not recommend alias mapping.

Drilling Through from Smart View

Products that support both FDM and Smart View allow Drill Through to FDM. Refer to the Oracle Hyperion Smart View for Office, Fusion Edition documentation for Drill Through processes and procedures.

Drilling Through from Financial Reporting

Products that support both FDM and Oracle Hyperion Financial Reporting, Fusion Edition allow Drill Through to FDM. Refer to the Oracle Hyperion Financial Reporting, Fusion Edition documentation for Drill Through processes and procedures.

Consolidating Target Systems

Target systems are consolidated in two ways: By you (selecting Activities > Consolidate) or by FDM (after target systems are loaded). The consolidation process requires that a validation entity group be assigned to the active location. Consolidated entities are specified in the validation entity group.

Enabling Target-System Consolidations

➤ To enable consolidations, select **Administration > Integration Settings > Enable Consolidation option.**

The validation entity group that is assigned to the FDM location is used to determine which entities are consolidated for a location. The current POV determines which target system category or period is consolidated.

Target-System Consolidation Errors

Only the current period is consolidated. If prior periods are impacted for the entity and category that is consolidated or if another user is accessing one of the entities in the consolidation path, an error is returned.

Validating Target-System Data

FDM validation reports retrieve values directly from target systems, FDM source data, or FDM converted data. Validation reporting is driven by two components—a group of validation rules

that compose the rows of the validation report and a group of validation entities, which determine the target entities to which the validation rules apply.

Validation reports are produced during the data-load process but can also be run manually. Reports are used primarily to review the quality of the data that was loaded.

FDM analyzes validation reports and enters status entries in the process monitoring table. If all rules for a location pass validation, a status of True is shown for the location. If a rule is used only for warning purposes, no rule logic is assigned to the row.

15 Importing Data in Various Ways

In This Chapter

Using Journal Templates to Import Data

Journal templates are Excel spreadsheets that are formatted as journal entry input screens.

FDM journal templates are typically used for the following types of adjustments:

- GAAP adjustments to general ledger files
- Transformations of gross balance accounts into roll-forward accounts (Beg, Add, Del, End)
- Supplemental data entries (Head Count, Ratios, and so on)

The journal template (`journal.xls`) is located on the Tools > Templates menu. A sample journal template is included with the templates downloaded from E-Delivery, and should be saved in `\\<FDMAppName>\Outbox\Templates` directory, where `FDMAppName` is the directory on the machine where the application was created.

Defining Journal Templates

You define an FDM journal template by defining the metadata header that FDM uses to interpret the data contained in the template. The header, which, typically, occupies the first five rows of the template, consists of a series of tags.

Metadata tags are presented in a prescribed tabular format. Row order is important, but column order is not.

The following template contains a metadata header (rows 1–5) and two lines of imported data (rows 6–7).

	A	B	C	D
1	ID-Texas100		1_Texas	
2			ACTUAL05	
3			1/31/2005	
4			A	
5	A	C	V	D
6	1100	Texas	500.00	Reclass Cash
7	1210	Texas	(500.00)	Reclass Cash

Row 1 (Journal ID and Location Tag)

The journal ID and the location tag are placed, respectively, in the Amount column and the Account column of row 1. Thus, row 1 sets the journal ID and the FDM location to which data is loaded.

Note: The journal ID is limited to ten characters.

Row 2 (FDM Category Tag)

The FDM category that the journal loads is identified in the Amount column of row 2.

Row 3 (FDM Period Tag)

The FDM period to which data is loaded is identified in the Amount column of row 3.

Row 4 (Load Method Tag)

The journal load method tag is placed in the Amount column of row 4.

Append and Replace tags are relevant only if a journal with the journal ID of the current journal exists within the current POV.

Journal Load Method	Tag in Row 4
Append	A or Append
Replace (default)	R or Replace
Append journal-zero suppress	AZ
Replace journal-zero suppress	RZ

Row 5 (Dimension Tags)

The tags that define the FDM dimensions to which data are loaded are placed in row 5.

Journal templates must contain tags for the following dimensions and for any user-defined dimension used by the target system:

- **Center**—From the source or target system

- **Account**—From the source or target system
- **Amount**
- **Intercompany** (required only if used by Financial Management)

FDM Dimension	Tag in Row 5
Account	A, Account, SrcAcctKey
Center	C, Center, SrcCenterKey
Description	D, Description, SrcAcctDesc
IC Counter Party	I, IC, ICCoParty
User Defined 1 – User Defined 20	1, UD1 – UD20, UserDefined1 –UserDefined20
Amount	V, Amount, SrcAmount

Creating Range Names Within Journal Templates

To define a journal template, you create a range name that includes all metadata and data cells and that begins with the prefix *ups*. For example, for a standard template, you might create the range name `[upsStandardJV (A1 to D7)]`.

The following template references UPSRange, which starts at row 16. Therefore, rows 16–20 are the first five rows (the metadata header) of UPSRange. And, therefore, the metadata of the template is placed in rows 16–20.

Rows 4–14 provide an interface that facilitates creation of the metadata header. Metadata information is input in rows 4–14 and referenced by the metadata header.

	A	B	C	D	E	F	G	H	I	J
1		HFM JOURNAL ENTRY								24-Jan-07
2										
3		Journal Voucher								
4										
5		Journal ID:		Limited to 10 Characters						
6										
7		Location:		Must be a valid FDM Location						
8										
9		Category:		Must be same as current FDM Category						
10										
11		Period:		Must be same as current FDM Period						
12										
13		Load Method:	R	A = Append, R = Replace						
14										
15		Account	Center	Amount	Description	IC	UD1	UD2	UD3	UD4
16		0		0						
17				0						
18				1/0/1900						
19				R						
20		A	C	V	D	I	1	2	3	4
22										
23										
24										
25										
26										
27										
28										
29										
30										
34			Balance:	$0.00						

Validating Journal Data

You can validate journal data in two ways: by using the `UpCheck` function and by executing a MultiLoadAction event before and after journal posting.

Using the UpCheck Function

Before loading a journal template, FDM searches for a custom VBA function named *UpCheck*. The function forces custom data validations (for example, restricting users to particular accounts or centers).

The `UpCheck` function is placed within a VBA module that is placed within the template. If custom conditions are met, `UpCheck=True`. Consider the following example:

```
-----------------------------------------
Public Function UpCheck() as Boolean
'Place your validation code here

If Validate = True Then
    UpCheck = True
Else
    UpCheck = False
End If

End Function
-----------------------------------------
```

Executing MultiLoadAction Events Before and After Journal Posting

In the FDM script editor, you can write a validation script within a MultiLoadAction event. The event is executed before and after the journal posting process. You can evaluate the `strEventName` parameter to determine which Multiload event is executing and write a validation script to stop the journal load process. See Chapter 19, "Creating and Using Scripts"for information about multiload events.

Processing Journal Templates

➤ To select journals to process:

1 From the FDM Web client, select **Activities > Journal**.

2 Click **Browse**, and select a journal.

3 Click **Open**.

Checking In Journals

Before journals can be posted, they must be checked in.

➤ To check in journals, from the Journals screen, **Check-in**.

When a journal is checked in, FDM examines the template for all ranges with names beginning with *ups*. It then examines and validates the metadata tags found in each `ups` range. FDM does not check in metadata segments that include an invalid range. Error information is added to the Processing Errors grid.

Posting Journals

After a journal is checked in, you click Post to post the journal. Posting a journal appends or replaces the data displayed in the Import screen (as determined by the load method specified in the journal template).

Journal Security

If the POV Lock option is turned off, FDM administrators and end users are restricted to posting journals to the FDM global POV.

Using Excel Trial Balance Files to Import Data

An Excel trial-balance file is an Excel spreadsheet that, through the import screen, is formatted to one period, category, and location.

Text Trial Balance Files Versus Excel Trial Balance Files

Text trial-balance files and Excel trial-balance files are similar in two ways: They are both loaded to the current POV (category and period), and, on the import form, they use the same Append and Replace options.

Text trial-balance files and Excel trial-balance files differ in one way: Text files can use only the standard import format, but Excel data files do not use any import formats.

Defining Excel Trial Balance Templates

To define an Excel trial-balance template, you define a metadata header. The header contains tags that FDM uses to interpret the data contained in the template.

Excel trial-balance templates use a tabular format. The first row contains the metadata tags.

The following template contains one line of metadata (row 1) and three lines of imported data (rows 5–7).

	A	B	C	D	E	F	G	H	I
1	A	C	D	IC	UD1	UD2	M1	M2	Y
2									
3									
4									
5	1100_001	TX	Cash In Bank	[icp none]	[none]	[none]	Memo	Memo Details	48,045
6	1190	TX	Petty Cash	[icp none]	[none]	[none]	Memo	Memo Details	500
7	4200	TX	Sales	[icp none]	[none]	[none]	Memo	Memo Details	(48,545)
8									
9									
10									
14	Must Balance to Zero:								-

Using Multiload Files to Import Data

A multiload file is an Excel or text file that is formatted to load multiple periods, categories, and locations.

FDM multiload templates are typically used for the following types of data:

- Planning (budget, forecast, and so on)
- Historical
- Administrator maintenance

Note: Use only Global mode when importing Multiload files. Do no use Local mode.

Multiload Text Files Versus Multiload Excel Files

When processing a multiload text file, FDM creates one load file and consolidates multiple periods simultaneously. When processing a multiload Excel file, FDM creates a load file for each period and consolidates one period at a time. Therefore, FDM processes a multiload text file at a much faster rate than it processes a multiload Excel file.

Defining Multiload Templates

To define a FDM multiload template, you define the metadata tags that specify the columns that FDM will import.

Metadata tags are presented in a prescribed tabular format. Row order is important, but column order is not. Metadata tags are contained in the first five rows (for Excel templates) or the first six rows (for text templates).

When defining an Excel multiload template, you must create a range name that includes all metadata and data cells and that begins with the prefix *ups*. The following Excel multiload template contains five rows of metadata tags (rows 1–5) and four rows of imported data (rows 8–11) and is assigned the range name `[upsBudget (A1 to K11)]`.

upsBudget =

	A	B	C	D	E	F	G	H	I	J	K
1										1_Texas	1_Texas
2										Actual04	Actual04
3										1/31/2004	2/28/2004
4										R,M	R,M
5	**Account**	**Center**	**Description**	**IC**	**UD1**	**UD2**	**UD3**	**UD4**	**DV**	V	V
8	1100	TX	Cash In Bank		MES	GIO		OP	Per	47,564	48,045
9	5000-101	TX	I/C Sales	MI		IOE	TV	OP	Per	1,980	2,000
10	2100-102	TX	I/C Rec	WI		GIO		OP	Per	6,589	6,656
11	3100-103	TX	I/C Pay	NY		TVE		OP	Per	108,900	110,000

The following text file contains six rows of metadata tags (rows 1–6) and four rows of imported data (rows 7–10). The delimiter that separates the dimensions of row 6 must be used as the delimiter that separates the data values. Valid delimiters include commas, semicolons, pipes (|), tabs, and exclamation points.

Row	Row Content
1	Texas
2	Budget
3	01/31/2004
4	12
5	R,N,Y
6	A,C,D,V,V,V,V,V,V,V,V,V,V,V
7	1100,TX,Cash In Bank,” 43,016 “,” 43,451 “,” 43,890 “,” 44,333 “,” 44,781
8	1100-101-000-00,TX,Dallas National Bank,” 1,791 “,” 1,809 “,” 1,827 “,” 1
9	1100-102,TX,Houston Bank One,” 5,959 “,” 6,020 “,” 6,080 “,” 6,142 “,” 6,
10	1100-103,TX,Midland Bank & Trust,” 98,487 “,” 99,482 “,” 100,487 “,” 101,
11	1100-104,TX,First National Bank,” (8,953)”,” (9,044)”,” (9,135)”,” (9,227
12	1190,TX,Petty Cash, 448 , 452 , 457 , 461 , 466 , 471 , 475 , 480 , 485 ,

Metadata Tag Structure

The metadata header of a multiload template (rows 1-5 for Excel templates and rows 1–6 for text templates) tells FDM how to locate relevant data segments within the template.

Row 1 (Location Tags)—Valid Only for Amount Columns

A location tag defines the FDM location to which data is loaded.

Row 2 (Category Tags)—Valid Only for Amount Columns

A category tag defines the FDM category to which data is loaded.

Row 3 (Period Tags)—Valid Only for Amount Columns

A period tag defines the FDM period to which data is loaded.

Row 4 (Number of Periods Tags)—Text Templates Only

A number of periods tag defines the number of periods (data columns) to be loaded. The tag is used with the Starting Period tag to derive the period key used in a data load.

Row 4 or Row 5 (Load Control Tags)

Load control tags, which are dependent upon the target system that FDM is loading, control the load methods for FDM and the target system.

Load control tags are placed in row 4 (for Excel templates) or in row 5 (for text templates) and are separated by commas.

Tag Number	Tag Name	Description
1	FDM Load Method	Method of loading FDM data
2	Target System Load Method	Method of loading target-system data
3	List Box 1	Varies by integration adapter
4	List Box 2	Varies by integration adapter
5	List Box 3	Varies by integration adapter
6	Check Box Calc	Enables calculation (Y or N)
7	Check Box 1	Varies by integration adapter (Y or N)
8	Check Box 2	Varies by integration adapter (Y or N)
9	Check Box 3	Varies by integration adapter (Y or N)
10	Data View	Specifies, fox example, whether the data being loaded is YTD data or periodic data

Row 5 or Row 6 (Dimension Tags)

Dimension tags (placed in row 5 for Excel templates and in row 6 for text templates) define the dimension data that is contained in the columns of multiload templates.

A valid delimiter must separate the dimension tags. The delimiter that separates the dimensions of the dimension list must be used as the delimiter that separates the dimension tags. Valid delimiters include commas, semicolons, pipes (|), tabs, and exclamation points (!).

Note: When memos are loaded from a Multiload file, each period receives the memo specified for each row of data in the Multiload file.

Load Control Tags—Application-Specific

Financial Management, Hyperion Enterprise, Essbase, and Strategic Finance use specific sets of load control tags in specific ways.

Financial Management Load Control Tags (Multiload Excel Files)

Financial Management does not use tags 4, 5, 6, and 9. For the unused tags, you must insert commas as placeholders.

If a Financial Management load control tag (2, 3, 7, 8, or 10) is missing, FDM defaults to the data-load options defined in Integration Settings. The default value for tag 1 is Replace, and the default value for tag 10 is YTD.

Tag 1: FDM Load Method	Tag Value
Append	A
Replace (default)	R
Append—zero suppress	AZ
Replace—zero suppress	RZ

Tag 2: Financial Management Load Method	Tag Value
Merge	M
Replace	R
Replace by security	RS
Accumulate	A

Tag 3: Financial Management Load Process	Tag Value
Load	0
Scan	1

Tag 7: Accumulate in File	Tag Value
Yes	Y or T
No	N or F

Tag 8: File Has Share Data	Tag Value
Yes	Y or T
No	N or F

Tag 10: Data View	Tag Value
Year-To-Date (default)	Y or YTD
Periodic	P or Per
Qtr year-to-date	Q, QYTD, or Qtr
Half-year	H or HYTD

Note: In row 5 (Excel template) or row 6 (text template), you can specify creation of a Data View column so that each data row can have a Data View tag. If no Data View tag is defined in the header and no Data View column exists, the default Data View (YTD) is used.

Financial Management Load Control Tag Examples (Multiload Excel Files)

Example 1—A value for each tag. Commas are required as placeholders for unused tags (4, 5, 6, 9):

```
R,M,0,,,,Y,N,,YTD
```

Example 2—A value for only the FDM Load Method tag. The other tags default to the values specified in Integrations Settings. The Data View tag defaults to YTD. Commas (placeholders) are not required after the last (in this case, only) value.

```
A
```

Example 3—A value for only the FDM Load Method and HFM Load Method tags. The other tags default to the values specified in Integration Settings.

```
R,M
```

Hyperion Enterprise Load Control Tags (Multiload Excel and Text Files)

Hyperion Enterprise does not use load tags 5, 6, 8, 9, and 10. For the unused tags, commas are used as placeholders. If a Hyperion Enterprise tag (2, 3, 4, 7) is missing, FDM defaults to the data-load options defined in Integration Settings. The default value for tag 1 is Replace.

Tag 1: FDM Load Method	Tag Value
Append	A
Replace (default)	R
Append–zero suppress	AZ
Replace–zero suppress	RZ

Tag 2: Enterprise Load Method	Tag Value
Merge	M

Tag 2: Enterprise Load Method	Tag Value
Replace	R
Accumulate	A

Tag 3: Data View	Tag Value
Category	C or Cat
Periodic	P or Per
Year-to-date	Y or YTD

Tag 4: Scale	Tag Value
None	N, 1, or “None”
Units	U, 0, Units
Tens	T, 1, Tens
Hundreds	H, 2, Hundreds
Thousands	TH, 3, Thousands
Ten thousands	T-TH, 4, Ten Thousands
Hundred thousands	H-TH, 5, Hundred Thousands
Millions	ML, 6, Millions
Ten millions	T-ML, 7, Ten Millions
Hundred millions	H-ML, 8, Hundred Millions
Billions	B, 9, Billions

Tag 7: Zero No Data	Tag Value
Yes	Y, Yes, True, T
No	N, No, False, F

Hyperion Enterprise Load Control Tag Examples (Multiload Excel and Text Files)

Example 1—A value for each tag. Commas are required as placeholders for tags that are not used (5 and 6). A comma is not required after the last tag value:

```
R,M,C,N,,,Y
```

Example 2—A value for only the FDM Load Method tag. The other tags default to the values specified in Integration Settings.

```
A
```

Example 3—Values for only the FDM Load Method and Oracle's Hyperion® Enterprise® Load Method tags. The other tags default to the values specified in Integration Settings.

```
R,M
```

Essbase Load Control Tags (Multiload Excel and Text Files)

Essbase does not use load tags 3, 4, 5, 6, 8, 9, and 10. For the unused tags, commas are used as placeholders. If an Essbase tag (1, 2, or 7) is missing, FDM defaults to the data-load options defined in Integration Settings. The default value for tag 1 is Replace.

Tag 1: FDM Load Method	Tag Value
Append	A
Replace (default)	R
Append–zero suppress	AZ
Replace–zero suppress	RZ

Tag 2: Essbase Load Method	Tag Value
Merge	M
Replace	R

Tag 7: Zero No Data	Tag Value
Yes	Y, Yes, True, T
No	N, No, False, F

Essbase Load Control Tag Examples (Multiload Excel and Text Files)

Example 1—A value for each tag. Commas are required as placeholders for tags that are not used (3, 4, 5, 6). A comma is not required after the last tag value:

```
R,M,,,,,Y
```

Example 2—A value for only the FDM Load Method tag. The other tags default to the values specified in Integration Settings.

```
A
```

Example 3—Values for only the FDM Load Method and EssbaseLoad Method tags. The other tags default to the values specified in Integration Settings.

```
R,M
```

Strategic Finance Load Control Tags (Multiload Excel and Text Files)

Strategic Finance does not use load tags 4, 5, 6, 8, and 9. For the unused tags, commas are used as placeholders. If a Strategic Finance tag (1, 2, 3, or 7) is missing, FDM defaults to the data-load options defined in Integration Settings. The default value for tag 1 is Replace.

Tag 1: FDM Load Method	Tag Value
Append	A
Replace (default)	R
Append—zero suppress	AZ
Replace—zero suppress	RZ

Tag 2: Strategic Finance Load Method	Tag Value
Replace	R

Tag 3: Data View	Tag Value
Category	C or Cat
Periodic	P or Per
Year-to-date	Y or YTD

Tag 7: Zero No Data	Tag Value
Yes	Y, Yes, True, T
No	N, No, False, F

Strategic Finance Load Control Tag Examples (Multiload Excel and Text Files)

Example 1—A value for each tag. Commas are required as placeholders for tags that are not used (4, 5, 6). A comma is not required after the last tag value:

```
R,M,C,,,,Y
```

Example 2—A value for only the FDM Load Method tag. The other tags default to the values specified in Integration Settings.

```
A
```

Example 3—Values for only the FDM Load Method and Oracle Hyperion Strategic Finance, Fusion Edition Load Method tags. The other tags default to the values specified in Integration Settings.

```
R,M
```

Forcing Custom Validations Before Loading Excel Templates

Before loading a multiload Excel template, FDM searches the template for a custom VBA function named *UpCheck*. The function forces custom data validations (restricting which categories or periods that users can load).

The `UpCheck` function is placed within a VBA module that is placed within the template. If custom conditions are met, `UpCheck=True`. Consider the following example:

```
----------------------------------------
Public Function UpCheck() as Boolean

'Place your validation code here

If Validate = True Then
UpCheck = True
Else
UpCheck = False
End If
End Function
----------------------------------------
```

Executing Validation Scripts During Multiload Processes

You can write a validation script within a MultiLoadAction event. During a multiload process, the script is executed before and after each event. You can evaluate the `strEventName` parameter to determine which multiload event is executing and write a validation script to stop the Multiload process. See Chapter 19, "Creating and Using Scripts" for information about the MultiLoadAction event.

Running the Multiload Process

The Multiload process follows much of the same workflow as a standard file load, but with extra actions required during some of the steps. The following are the steps required for running a Multiload:

1. Check-in
2. Import
3. Validate
4. Load (Export)
5. Consolidate (optional)
6. Run Validation Report (optional)

Checking In Multiload Files

Checking in a Multiload template prompts FDM to process the metadata tags of the template.

➤ To check in Multiload files:

1 **From the Web client, select Activities > Multiload.**

The Multiload screen is displayed.

2 **Click the Upload File tab.**

3 **Click Browse.**

The Chose File dialogue box is displayed.

4 **Locate and select a Multiload file (`.txt` or `.xls`) and click Open.**

The Choose File dialogue box closes and the Check-in button appears next to the Browse button on the Multiload screen.

5 **Click Check-in.**

The Multiload file is processed and the File Contents pane and Processing Details pane are populated.

Note: For multiload files selected from the Inbox screen, the check-in step is not required.

The File Contents window displays all the processed data sets.

The Processing Details area (on the right) displays validation errors and validation report errors.

After a multiload file is checked in, the FDM POV is changed to Local mode. Local mode enables the user to correct validation errors and view validation report errors. When a multiload screen is closed, the FDM POV returns to Global mode.

FDM does not check in invalid metadata segments but does add error information, in regard to invalid segments, to the processing errors grid.

Importing Multiload Files

You import a checked-in multiload template by clicking the Import button. Importing a template writes the data to the FDM location table. If an error occurs, the  icon is displayed in the directory tree on the left of the window.

Note: Use only Global mode when importing Multiload files. Do not use Local mode.

Validating Multiload Files

You validate an imported multiload template by clicking the Validate button. Validation verifies that all members of each dimension exist in their respective map tables.

Validation errors related to mapping are viewed by clicking the  icon. Unmapped members are listed in the Processing Details grid (on the right).

➤ To add unmapped members to their mapping tables:

1 Under **Activities**, click the **Maps** link.

2 Map invalid members to target members.

Note: The mapping table that is displayed is for the current POV location, which may not be the location being loaded. Therefore, you should identify the POV location before you modify mappings.

Loading Multiload Files

You export and load a validated multiload template by clicking the Export button. When a multiload text file is processed, one load file is created and loaded into the target system. When a multiload Excel file is processed, a load file is created and loaded for each period. If load errors occur, the  icon is displayed in the tree on the left.

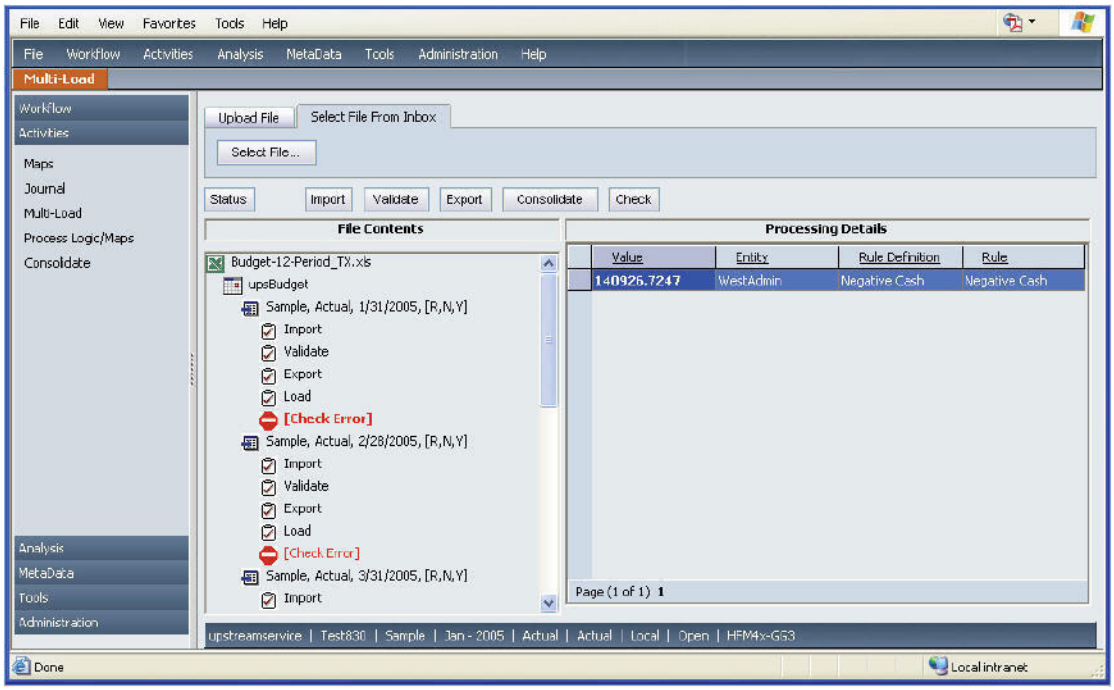
Consolidating Multiload Files

After a multiload template is exported and loaded to the target system, you run a consolidation for each period by clicking the Consolidate button. If errors occur, the  icon is displayed in the tree on the left.

Running Validation Reports for Multiload Files

After a multiload template is consolidated, you process the validation reports for each period by clicking the Check button. The reports are processed in the background.

You can view validation report accounts that do not pass validation by clicking the  icon for each period. Failed account rules are displayed in the Processing Details grid (on the right).



Multiload Security

FDM end-users can load multiload files to any FDM period and category and to any location to which they have access. FDM administrators can load multiload files to any location, period, or category.

Multiload Data Fields

Each amount data field must contain a value. An amount field that does not contain a loaded value must contain a zero (0). Multiload templates that contain a blank amount field do not process successfully.

16 Running and Maintaining Reports

In This Chapter

Running Reports

➤ To run reports:

1 From the FDM Web client, select **Analysis > Reports**.

The Reports screen is displayed.

2 From **Groups**, select a group of reports.

The Reports area lists all reports associated with the selected group.

3 From **Reports**, select a report.

4 From the output format list, (upper-right of the Reports screen), select an output format in which the report will be written.

5 Click **Publish**.

The report parameters screen may be displayed (depending on the report selected).

6 If prompted, enter parameter values and click **OK**.

Note: By default, base trial balance reports do not import or display the target account descriptions.

Running the Process Monitor Report

A process monitor report displays, for the current report, a list of locations and their positions within the data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

➤ To display process monitor reports.

1 From the **Reports** tab, expand the **Process Monitor** report group.

2 Select **Process Monitor Report**.

Maintaining Reports

From Workbench, you select the Reports tab to modify and create reports. FDM provides a built-in report creation tool that enables creation of Active reports. Creating and integrating reports requires advanced knowledge of RDBMS syntax and the FDM Software Data Window Active-X component.

Modifying Report Group Descriptions

The following procedure enables you to edit the names of report groups.

➤ To modify report group parameter descriptions:

1 From FDM Workbench, select the **Reports** tab.

2 Right-click a report group and select **Properties**.

3 The Report Group Properties dialog box is displayed.

4 In the **Description** field, edit the title of the report group.

5 Click **OK**.

Modifying Report Descriptions

The following procedure enables you to edit the names of reports.

➤ To modify report parameter descriptions:

1 From FDM Workbench, select the **Reports** tab.

2 Right-click a report and select **Report Definition Properties**.

The Report Properties dialog box is displayed

3 Select the **General** tab.

4 In the Description field, edit the title of the report.

5 Click **OK**.

Modifying Report Component Descriptions

The following procedure enables you to edit the names of report components.

➤ To modify report component parameter descriptions:

1 From FDM Workbench, select the **Reports** tab.

2 Right-click a report component and select **Properties**

The Report Parameter Properties dialog box is displayed.

3 Select the **General** tab.

4 In the **Title** field, edit the title of the report component.

5 Click **OK**.

Modifying and Creating Reports

➤ To modify Active reports:

1 From the Reports tab, right-click a report.

2 Select **Design Report**.

3 Modify the report.

You can also use Design Report to create reports. For modifying or creating reports, advanced knowledge of FDM reporting components is required.

Setting Report Group Security

Report groups can be assigned security levels. A report-group security assignment restricts access to users who are assigned an equivalent application security level.

➤ To assign security to report groups:

1 From Workbench, select the **Reports** tab.

Reports are listed in the left pane of the Workbench desktop.

2 Right-click a report group, and select **Properties**.

3 From **Security Level**, select the minimum level for the report group.

4 Click **OK**.

The security level that is assigned to a report group is evaluated against the application security level that is assigned to an end user. Access to reports within the report group is granted to users with an application security level that is equal to or less than the level assigned to the report.

Security for Process Explorer reports is set on individual reports. For Process Explorer reports, to modify user access, you must right-click a report, rather than a report group.

Running the Timeline Viewer

The Timeline Viewer enables you to look at FDM events in a linear graphical format. Timeline Viewer can be filtered to view events that have taken place for specific dates, periods, and categories.

➤ To run Timeline Viewer

1 **From within the FDM Web client, select Analysis > Timeline.**

The Timeline Viewer screen is displayed .

2 **At the top of the left pane of the Timeline Viewer screen, select a tab:**

- To filter the timeline by a specific date range, choose the **Dates** tab.
- To filter the timeline by periods, select the **Period** tab.

3 **Select a start date and end date (when using Dates tab), or select a period (when using Period tab).**

4 **Select the events you want to appear on the timeline:**

- From Category, select a single category, or select **All Categories**.
- From Workflow Events, select from the available events.
- From Mapping Events, select **Import**, **Changes**, both, or none.
- Select Multiload to include Multiload events.

5 **From Locations, select the locations to include from the Locations directory structure.**

6 **Click Get Timeline.**

The timeline is displayed in the right pane of the Timeline Viewer screen.

- The events on the timeline are color-coded to the selectable events in the left pane.
- Roll the cursor over each of the event in the timeline to reveal the tool tip which contains the details of each event.
- Black markers on the timeline designate multiple, overlapping events. Left-click on the marker to display all events that correspond with that marker.
- By default, the timeline only shows the last instance of an event. To view all instances of an event, left click the event, then select **All <name of event> Events** (for example, All Import Events), or **All <name of event> Events with Sub-events** (example, All Export Events with Sub-events). This displays one timeline with all instances of the selected event, with or without sub-events depending on the option selected. Sub-events include processing tasks such as processing maps, writing to and deleting temporary tables, exporting data, and loading files.
- When you view an expanded event (All Events or All Events with Sub-events), a Back button is displayed. Click **Back** to return to the previous view.

Setting Number of Locations for Timeline Viewer

You can change the number of locations that are displayed on each page of the Timeline Viewer. The default is 3.

➤ To change the number of locations displayed on each page of the Timeline Viewer:

1 Select **Tools > Configuration Settings**.

The Configuration Settings dialog box is displayed.

2 In the Configuration Settings dialog box, from **Options** select **Number of Locations per Page**.

3 From **Value**, select the number of locations to be displayed (2–10)

4 Click **Save and Close**.

17 Processing Batch Files

In This Chapter

Overview of Batch Loader

To process batch files, you use Batch Loader, which you initiate from FDM Workbench or FDM Task Manager. The Batch Processing screen is only available in the Workbench and Task Manager only runs the batch script.

Note: Batch Loader is only available with the purchase the FDM Adapter Suite. Contact your Oracle sales representative for information regarding Batch Loader licensing options.

Overview of Batch Loader Files

The batch files processed by Batch Loader are of two types:

- Standard—Text files that can contain only one month and that are usually loaded through the Import screen
- Multiload—Text or Excel files that can contain multiple periods and locations

Name Format for Batch Files

The names of standard batch files consist of the following segments in the following order:

1. **File ID**—A free-form field that can be used to control load order (Batch files load in alphabetic order by file name. Thus, for example, a file named `a_Texas_Actual04_Jan-2004_RR.txt` loads before a file named `b_Texas_Actual04_Jan-2004_RR.txt`.)
2. **Location**
3. **Category**

4. **Period**
5. **Load Method**—A two-character item (Character 1 = FDM append or replace, and character 2 = target append or replace. Valid values are `A` and `R`.

   Because Multiload files contain metadata, they do not require a specific file name format.

Directories for Batch Files

You place files that are to be loaded using Batch Loader in the `OpenBatch` directory of the application `Inbox\Batches` subdirectory.

After a batch is processed, a directory is created, all files within the `OpenBatch` directory are moved into the new directory, and the new directory is assigned a unique batch ID

The names of directories that contain multiload batch files are prefixed with `ML`. Batch IDs are referenced in Import logs.

Running Batch Loader

➤ To initiate the batch load process: :

1 **Create the batch files, and place them in the appropriate FDM directory.**

See "Name Format for Batch Files" on page 173.

2 **Perform an action:**

- From Workbench, select **Tools** > **Batch Processing**.
- From Task Manager, select **Start** > **Programs** > **Oracle EPM System**> **Financial Data Quality Management** > **Task Manager** > **Task Manager**.

  The Batch Processing screen is displayed.

3 **Set the batch controls.**

4 **Perform an action:**

- To run the batch with the selected controls, click **Execute**.

  The Batch Monitor section lists the files that Batch Loader is processing and updates with the results for each file.

- To create a script with the selected controls, click **Create Script**.

  You are prompted for a script name. You can add the script name to a scheduled task and, thus, set the script to run automatically.

  Note: Custom scripts do not support multibyte or extended ASCII characters in the script name or in the script parameters.

See "Batch Controls" on page 175 and "Batch Processing Screen Results" on page 175for information related to steps 3 and 4.

Batch Controls

- **Batch Type**—Determines whether the current batch is a standard or multiload batch
- **Process Level**—Determines the level of processing to be performed (for example, import, process, or run the check report step)
- **Process Method**—Determines whether the batch is loaded in serial or parallel mode (Serial mode processes files sequentially, requiring that one file complete its process before the next file starts its process . Parallel mode processes files simultaneously.)
- **No. of Parallel Processes**—If Process Method is set to parallel, determines the number of processes that the batch uses concurrently (For example, if a batch contains 20 files and No. of Parallel Processes is set to 5, five sequential processes, each with four files, run concurrently.

Note: When processing batch files in parallel, FDM groups files that are loading to one FDM location into one process, thus ensuring that multiple processes do not attempt to load data to one location simultaneously and thus avoiding invalid results.

Load Balance Server—Enables the administrator to select, as the load balance server that is used for processing the batch, any load balance server of the Workbench Load Balances Server group

AutoMap Correct—Enables or disables automapping for all files of the batch being processed. See for additional information about Automap correction."AutoMap Correct" on page 175

File Name Delimiter—Defines what character is used to separate the five segments of a standard batch file name.

AutoMap Correct

FDM can automap all file items that are not included in dimension maps. For unmapped members, map entries (based on predetermined suspense accounts) are created in the location maps associated with the unmapped members. Before you can set up suspense accounts, you must add a location named *AutoMapCorrect* to the FDM application. Then, in the Explicit map, you must create a map for each dimension. The source value for each map must be named *AutoCorrect*, but the target value can be any member of the relevant dimension.

Batch Processing Screen Results

As a batch is processed, the Batch Monitor field of the Batch Processing screen provides a graphical display of the batch results. Each completed load displays a symbol that indicates which processes failed and which processes passed:

 **Auto Corrected**—Failed steps are automatically corrected.

 **Successful**—All steps completed successfully.

 **Failure**—One of more steps failed.

 **Invalid POV**—Process is being loaded to an invalid period, category, or location.

Batch Processing for Source Adapters

The source adapters support batch processing using the Source Controls tables. When launching a batch process on a location that is using a source adapter, place an empty file in the `Open Batch` directory. Doing so launches batch processing. This file must follow the file-naming conventions outlined in "Name Format for Batch Files" on page 173.

Note: Batch multiload processing is not supported for source adapters.

Scripting for Batch Loader

You can run Batch Loader from within Workbench or from within a script environment, manually or through Task Manager. You can create scripts manually, by copying the following scripts, or by using the Batch Processing screen.

Script for Standard Batch Loads

```
Sub BatchLoad()
'--------------------------------------------------------------------------
' FDM Custom Script:
'
'Created By:      ADMIN
'Date Created:     3/24/2006 09:00
'
'Purpose:        Execute a STANDARD Serial Processing FDM Batch
'--------------------------------------------------------------------------
'Declare Local Variables
Dim lngProcessLevel
Dim strDelimiter
Dim blnAutoMapCorrect
'Initialize Variables
lngProcessLevel = 12  'Up-To-Check
strDelimiter = "_"
blnAutoMapCorrect = 0
'Create the file collection
Set BATCHENG.PcolFiles = BATCHENG.fFileCollectionCreate(CStr(strDelimiter))
'Execute a Standard Serial batch
BATCHENG.mFileCollectionProcess BATCHENG.PcolFiles, CLng(lngProcessLevel), ,
CBool(blnAutoMapCorrect)
End Sub
```

After a script is created, you can create a FDM Task Manager event to run Batch Loader at scheduled intervals.

Script for Parallel Batch Loads

Because FDM supports parallel processing of batch files, (up to 50 files) can run simultaneously. FDM groups files into processes. The files within each process run sequentially (requiring that one file completes its run before the next file starts its run), and the processes run simultaneously.

Example: For example, if parallel processing is enabled for a batch that contains 20 files and the number of processes is set to five, five processes (each including 4 files that run sequentially) run simultaneously.

Note: When processing batch files in parallel, FDM groups files that are loading to one FDM location into one process, ensuring that multiple processes do not attempt to load data simultaneously to avoid inaccurate results.

```
Sub ParallelBatch()
'-----------------------------------------------------------------
' FDM Custom Script:
'
'Created By:        Admin
'Date Created:      3/23/2006 13:50
'
'Purpose:           Execute a FDM Batch Loader with parallel processing
'-----------------------------------------------------------------
'Declare Local Variables
Dim lngProcessLevel
Dim strDelimiter
Dim blnAutoMapCorrect
Dim lngParallelProcessCount
Dim strLoadBalanceServerName
'Initialize Variables
lngProcessLevel = 12   'Up-To-Check
strDelimiter = "_"
blnAutoMapCorrect = 0
lngParallelProcessCount = 5
strLoadBalanceServerName = "LocalHost"
'Create the file collection
Set BATCHENG.PcolFiles = BATCHENG.fFileCollectionCreate(CStr(strDelimiter))
'Execute a Standard Parallel batch
BATCHENG.mFileCollectionProcessParallel BATCHENG.PcolFiles, CLng(lngProcessLevel),
CLng(lngParallelProcessCount), CStr(strLoadBalanceServerName), ,
CBool(blnAutoMapCorrect)
End Sub
```

18 Performing Miscellaneous FDM Tasks

In This Chapter

Exporting Applications and Application Components to XML

FDM enables you to export applications or selected application components to XML files.

➤ To export applications or selected application components:

1 **From Workbench, log on an application.**

2 **Select File > Export.**

The Save Metadata Export File dialog $_{\text{dialog box}}$box is displayed.

3 **Enter a name for the exported file.**

4 **Click Save.**

The Export dialog box is displayed.

5 **Select the components to export.**

6 **Click Save.**

Note: If you export locations, you can export the maps associated with the locations by selecting, on the Options tab, Export Maps with Locations.

Viewing Transaction Information in System Logs

A system log lists all system transactions, showing the activities that occurred , which user performed each activity, and at what time each activity was performed.

Three fields present detailed information about each transaction:

- **Event Info**—Statistics or information
- **Error Info**—A Visual Basic or Microsoft® DAO error code (may be required by Customer Support to determine cause and resolution of errors)
- **IO Source (Input/Output)**—The input and output RDBMS statements that were run and the external files that were created or imported

➤ To view the system log

1 From the Web client, select **Analysis > Log**.

2 From **Transaction Keys**, select a transaction type.

Importing Metadata from Excel

You can use the Import XLS tool to import metadata directly from an Excel spreadsheet into any table of the current FDM database. Therefore, you can import large amounts of application information without entering each line manually in the Web client setup grids.

➤ To import metadata from Excel spreadsheets:

1 From Workbench or the Web client, select **Tools > Import XLS**.

The Import XLS dialog box is displayed.

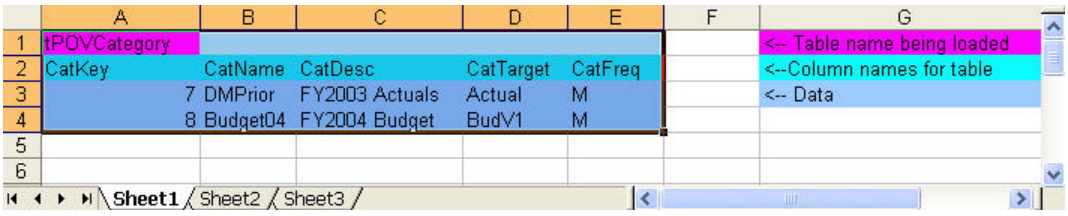
2 Browse to locate and select an Excel spreadsheet.

3 Click **Open**.

Note: Uploaded Excel files should be MS Office 2002-2003 (.xls) format compliant.

Ranges Within Excel Spreadsheets

An Excel spreadsheet that is used as an import source represents one or more ranges. The first cell of a range contains the name of the FDM table to be imported. The second row of a range contains the column names of the table identified in the first row. Rows 3 and higher contain the data to be loaded to FDM.



Range names begin with the FDM import identifier `ups`, for example, `upsCategory`. If imported tables are related, the parent table must be imported prior to the child table. Range names process in alphabetical order. Therefore, to ensure correct parent-child order, assign range names such as `upsAParent`and `upsBChild`.

Using `UpReplace` Within Excel Spreadsheets

If you create a Visual Basic (VB) function named `UpReplace` within an Excel spreadsheet and set its return value to True, you instruct FDM to delete all data from the `TDim` table before data is loaded.

Example:

```
Public Function UpReplace() as Boolean
'Set value to true to force delete prior to load
UpReplace = True
End Function
```

Note: The UpReplace function can be used only for the `tDim` table.

Using Map Converter

When you replace the current target application with another target application, you can use Map Converter to triangulate dimension maps. For example, if the Texas location contains a map between the general ledger and the current target application and the Michigan location contains a map between the current target application and the new target application, Map Converter creates a map from the general ledger to the new target application.

➤ To use Map Converter:

1 From Workbench or the Web client, select **Tools > Map Converter**

The Map Converter screen is displayed.

2 Select the **Dimension** tab.

3 Select a dimension.

4 Select the **Source Location** tab.

5 Select the FDM location that contains the map between the current target application and the new target application.

6 Select the **Target Location** tab.

7 Select the FDM location that contains the map between the general ledger and the current target application.

8 Select the **Preview and Convert** tab.

9 Click **Preview the Conversion**.

Preview the Conversion enables you to view the new map without performing the change.

10 Click **Perform the Conversion** to convert the map.

The location that contained the general ledger is updated to store the converted map.

Using Text Editor to Open and Edit Text Files

You can use Text Editor to open and edit any text file.

➤ To open Text Editor, from Workbench, select **Tools** > **Text Editor**.

Creating, Deleting, Recreating, and Reassigning Data Segments

When an FDM application is created, fifty data segments are automatically created for the application. Data segments are used to store data for FDM locations. When a location is created, FDM assigns it to the next available data segment. Locations will share data segments when the number of locations exceeds fifty. It is recommended that if you will have more than one hundred locations, you create additional data segments before creating locations.

From Workbench, you can create, delete, recreate, and reassign data segments

Creating Segments

You can create data segment tables within a database without recreating the application's database.

➤ To create data segments:

1 From Workbench, select **Tools > Manage Data Segments > Create New Segments.**

The Create New Segments dialog box is displayed.

2 From **Segments**, select the number of segments to create.

3 Click **Save and Close**.

Note: This procedure does not assign existing locations to the new segments. All locations remain in their original data segment.

Deleting, Recreating, and Reassigning Data Segments

When you create segments in an application where locations already exist, only newly created locations will be assigned to the new data segments. Existing locations are not reassigned. Reassigning data segments prompts FDM to reallocate (recreate) all locations.

Reports are run from a view that combines all data segments in an application. An excessive number of data segments can slow down performance when running reports. Thus, you may wish to delete unnecessary data segments to improve performance.

Note: When you delete, recreate, or reassign data segments, all data associated with existing locations (including maps) is deleted

➤ To delete, re-create, and reassign data segments:

1 From Workbench, select **Tools > Manage Data Segments > Delete, Recreate, or Reassign All Segments**.

The Recreate Segments dialog box is displayed.

2 From **Segments**, select the total number of segments you want in the application.

3 Click **Save and Close**.

Backing Up Application Files

The Backup Application Files function compresses all files of the FDM `Application` directory into one `.zip` file that includes the following types of files:

- Scripts
- Reports
- Excel templates
- Logs
- Archives
- All other files created or saved in the application directory

This functionality is useful for archiving applications.

➤ To back up application files:

1 From Workbench, select **File > Backup Application Files**.

2 Enter a name for the archive file.

3 Click **OK**.

19 Creating and Using Scripts

In This Chapter

Overview of Scripts

FDM uses the Visual Basic scripting engine to enhance the flexibility and power of FDM. Using standard VBScript syntax, you can create any of three types of scripts—import (including integration import), event, or custom. The internal object model (functions specific to FDM) can also be referenced within FDM scripts.

Using Script Editor to Create and Edit Scripts

Script Editor is used to define Visual Basic scripts that run in response to FDM events or custom-menu selections or during file import processes. Scripts are saved in the `Data\Scripts` directory of the FDM application (with a `.uss` extension). Scripts can be copied to other FDM applications and edited using a text or XML editor.

Launching Script Editor

During certain script-related procedures, Script Editor is launched automatically. However, it can be accessed manually.

➤ To access Script Editor from the Web client, select **Tools** > **Script Editor.**

➤ To access Script Editor from Workbench, select the **Scripts** tab.

Overview of Script Editor

The upper-left tree of Script Editor (in Workbench) lists the directories that contain the three types of FDM scripts. The bottom-left tree displays a hierarchy of the accessible FDM objects.

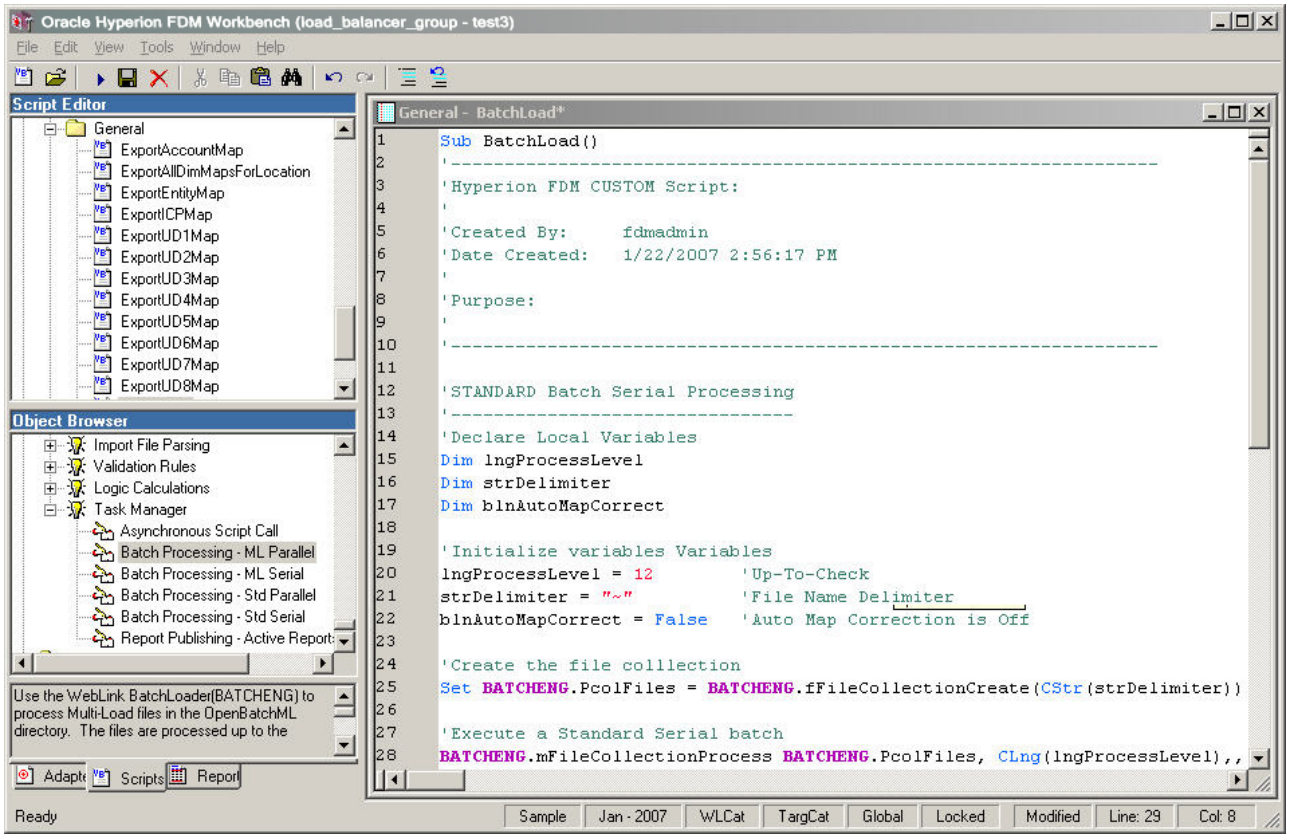
Each type of script references all FDM objects (except the API object, which cannot be referenced in import scripts). Each object corresponds to a public class module within a FDM dll file.

Methods, Functions, and Properties

FDM objects contain subordinate objects and methods, functions, and properties. When, within Script Editor, you double-click a method, function, or property, code is inserted directly into scripts.

Accelerators

The Object Browser pane of Script Editor lists accelerators (code modules that simplify the process of creating scripts). When you double-click an accelerator name, the accelerator code is placed at the cursor position in the open script.



Using Import Scripts to Import Data

Import scripts are executed as source files are imported. All FDM objects, except the FDM API object, are supported in import scripts.

Creating Import Scripts

➤ To create import scripts:

1 From the FDM Web client, select **Metadata > Import Formats**.

2 In the **Import Formats** screen, in the **Expression** field of a row, double-click, and select **Build Script**.

Script Editor is displayed.

3 Click **New.**

4 From **Script Type**, select **Import (DataPump)**.

5 In **File Name**, enter a name.

Note: Custom scripts do not support multibyte or extended ASCII characters in the script name or in the script parameters.

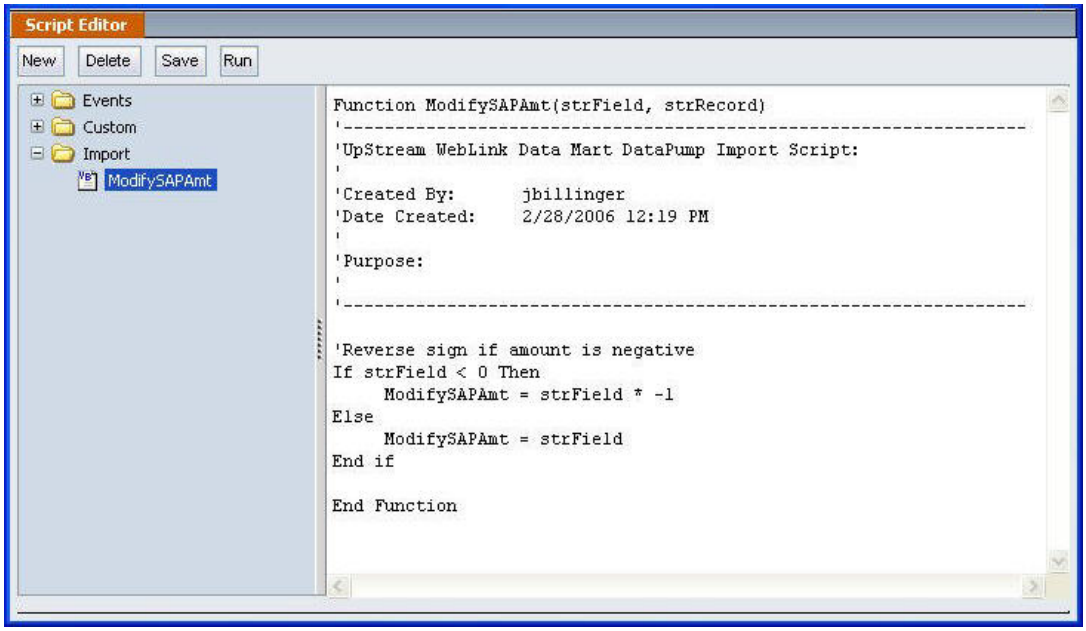
6 Click **OK**.

7 Write the script in the Script Editor.

8 Click **Save**.

Import Script Parameters

- **strField**—The values of the source-file column that is defined in the import format (For example, for import scripts assigned to the Amount column, `strField` contains amount values, and, for import scripts assigned to the Account column, `strField` contains account values.)
- **strRecord**—The source-file line (record) that is being scanned



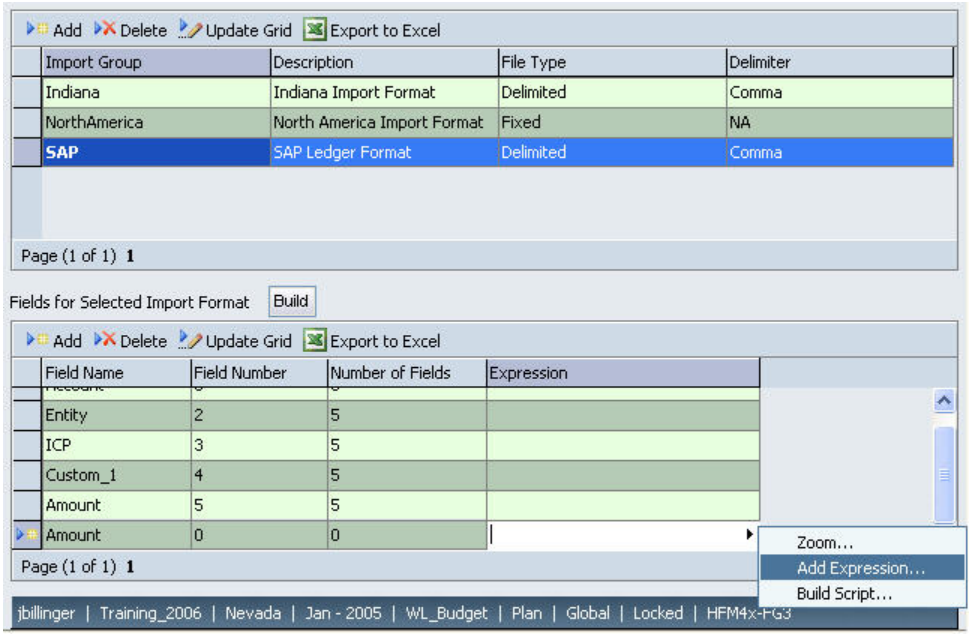
Assigning Import Scripts to Import Formats

After creating and saving a script in Script Editor, you assign the script to an import field—any import field within any import format.

➤ To assign import scripts to import fields:

1 **Within the Web client, select Metadata > Import Formats.**

The Import Formats screen is displayed.



2 **Select an import format group (top grid).**

3 Right-click the **Expression** field of a row (bottom grid), and select **Add Expression**.

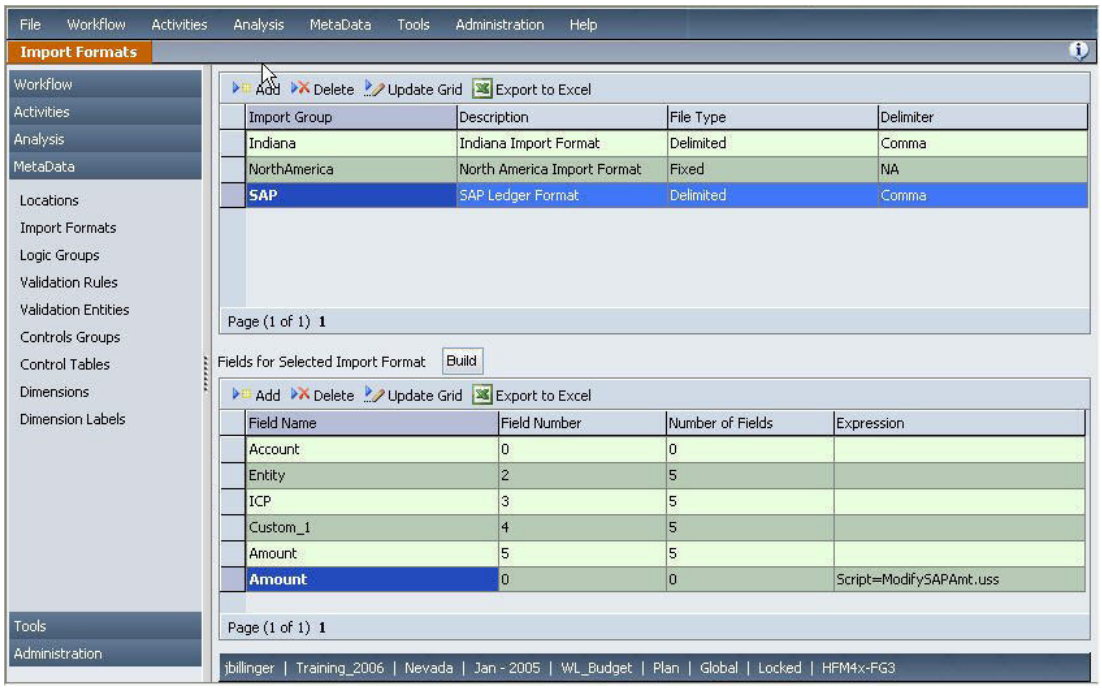
The Script Editor dialog box is displayed.

4 From **Expression Type**, select **Script**.

5 Browse for and select an import script.

6 Click **OK**.

The script is assigned to the import field. The name of the import script is displayed in the Expression column.



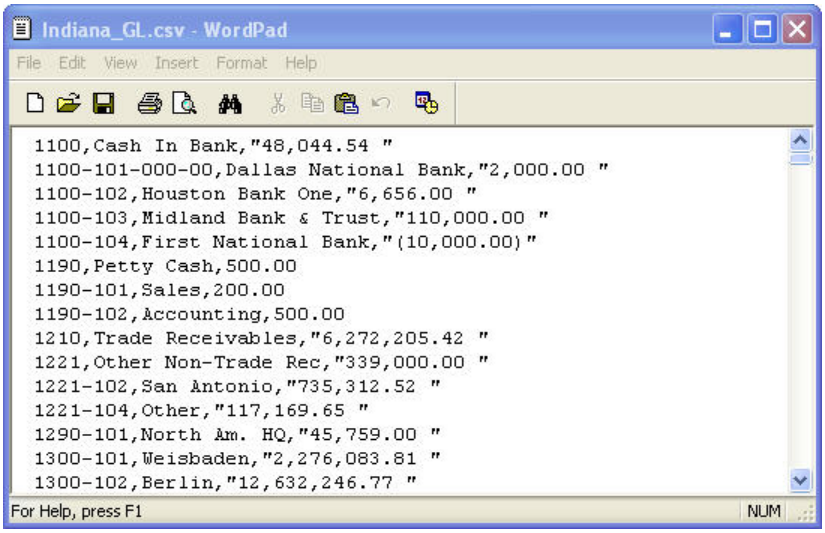
Using Import Script Functions

Within import scripts, you use functions to manipulate the data that FDM imports.

Left Function

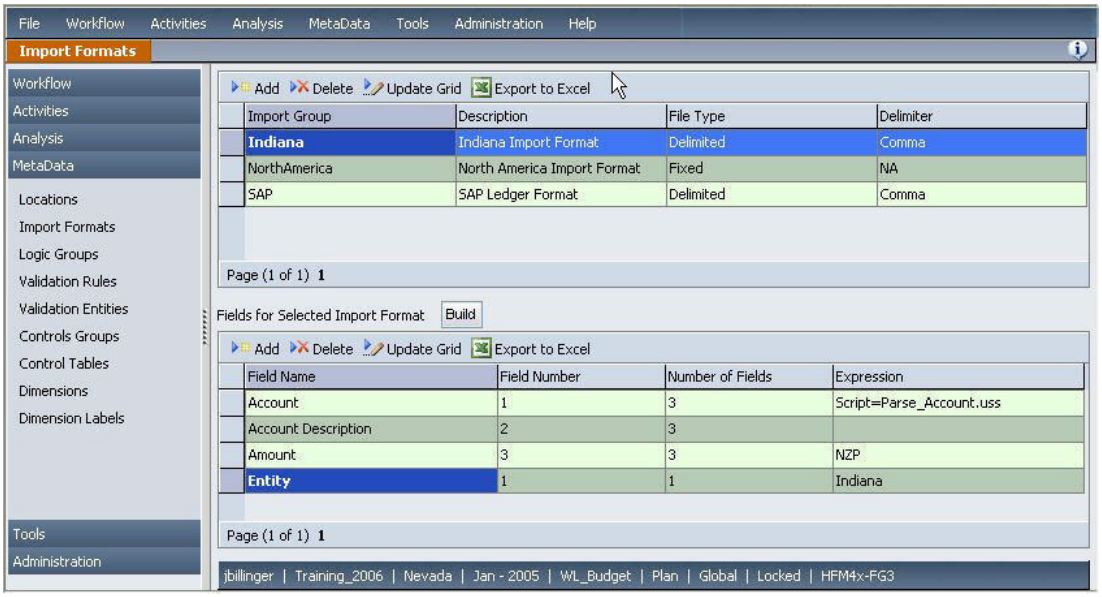
```
Left (string, # of characters)
```

Problem: The account numbers of the `Indiana_GL` file, which are in the first field of each line, include varying numbers of digits. However, the mapping associated with the file uses only four digits.

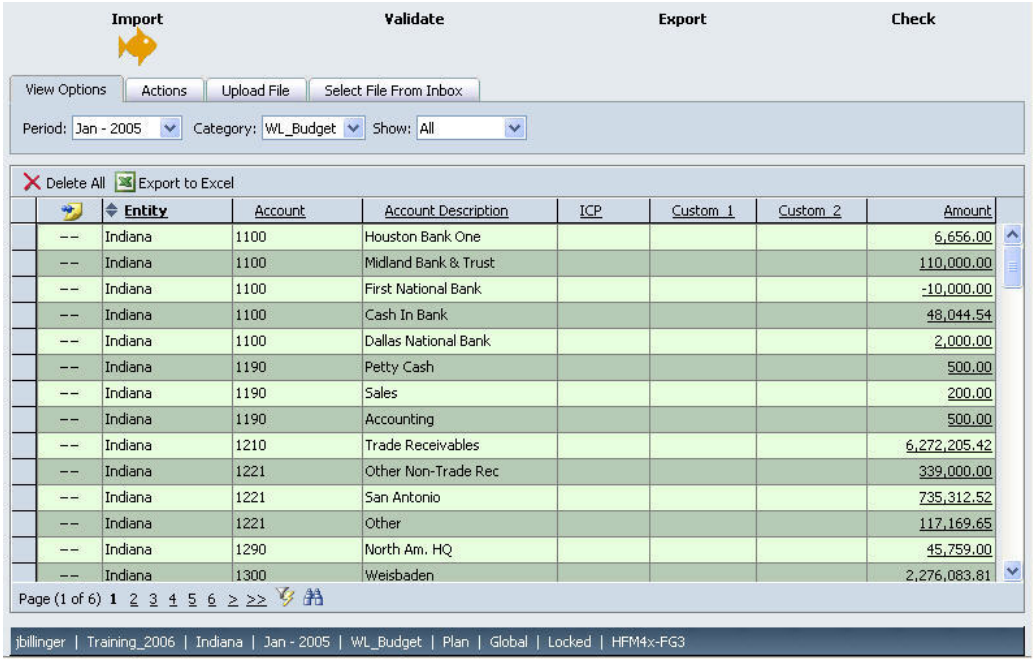


Solution: In the Import Scripts screen, assign the following script, which uses the `Left` function, to the Expression field of the Account row. The script assigns the first four digits of each account number (the first four digits on the left) to `Parse_Account`.

```
------------------------------------------------------------------
Function Parse_Account [strField, strRecord]
'------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------
Parse_Account = Left (strField, 4)
End Function
------------------------------------------------------------------
```



Result: The import file displays only the first four digits of each account number.

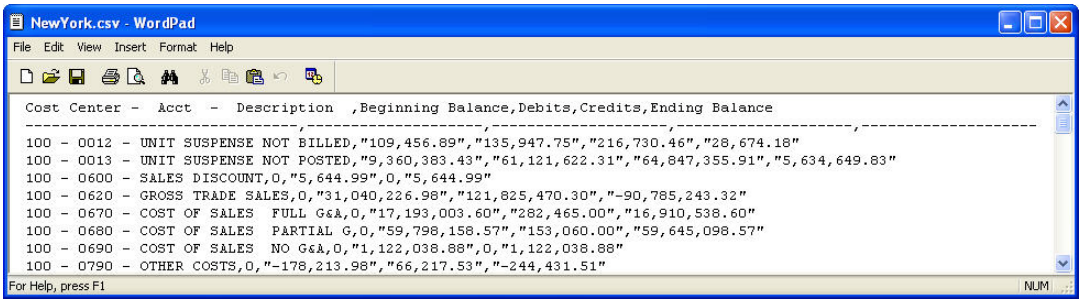


	Entity	Account	Account Description	ICP	Custom 1	Custom 2	Amount
--	Indiana	1100	Houston Bank One				6,656.00
--	Indiana	1100	Midland Bank & Trust				110,000.00
--	Indiana	1100	First National Bank				-10,000.00
--	Indiana	1100	Cash In Bank				48,044.54
--	Indiana	1100	Dallas National Bank				2,000.00
--	Indiana	1190	Petty Cash				500.00
--	Indiana	1190	Sales				200.00
--	Indiana	1190	Accounting				500.00
--	Indiana	1210	Trade Receivables				6,272,205.42
--	Indiana	1221	Other Non-Trade Rec				339,000.00
--	Indiana	1221	San Antonio				735,312.52
--	Indiana	1221	Other				117,169.65
--	Indiana	1290	North Am. HQ				45,759.00
--	Indiana	1300	Weisbaden				2,276,083.81

Mid Function

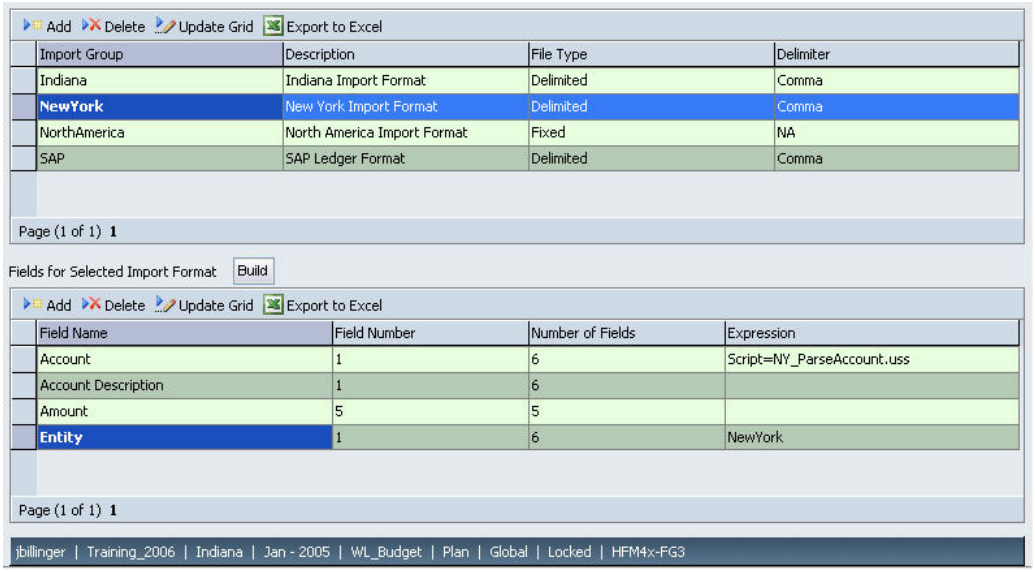
`Mid (string, start character, # of characters)`

Problem: The `NewYork` file presents cost center, account, and account description as a continuous string, rather than as three strings. You want to separate the account strings from the cost center and account description strings.



Solution: In the Import Formats screen, assign the following script, which uses the `Mid` function, to the Expression field of the Account row. The script assigns the characters of the account strings (the characters from positions 7 to 10 of the string) to `NY_ParseAccount`.

```
------------------------------------------------------------------
Function NY_ParseAccount [strField, strRecord]
'------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------
NY_ParseAccount = Mid (strField, 7,4)
End Function
------------------------------------------------------------------
```

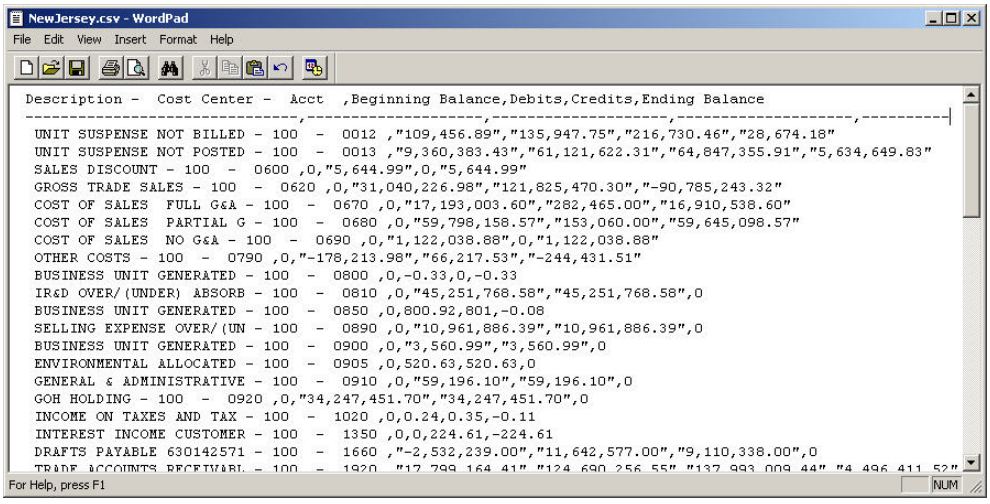


Result: Account strings are separated from cost center and account description strings.

Right Function

`Right (string, # of characters)`

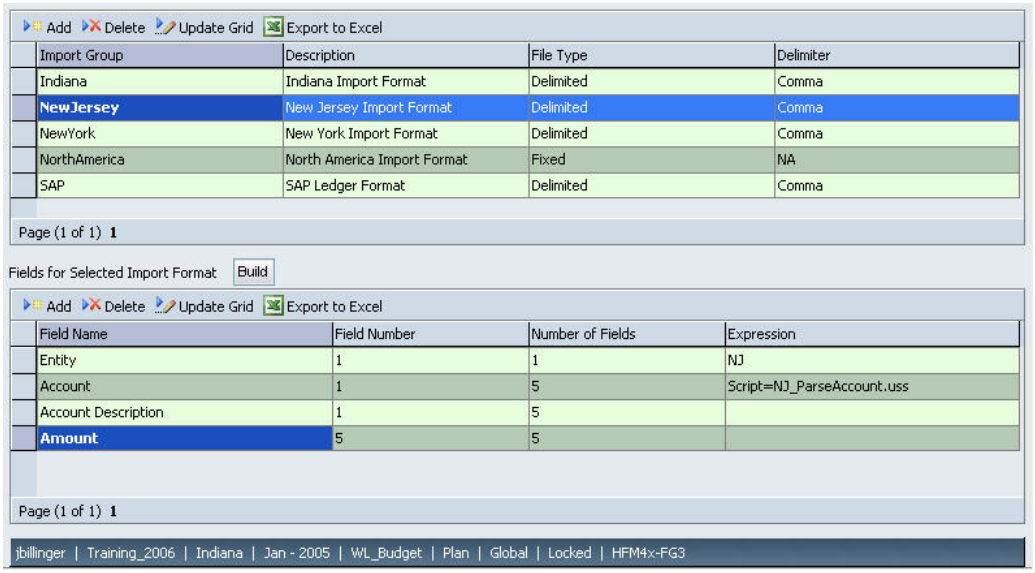
Problem: The `NewJersey` file presents account numbers as the last four characters of account fields. You want to extract only account numbers.



Solution: In the Import Formats screen, assign the following script, which uses the `Right` function, to the Expression field of the Account row. The script assigns the account numbers (the last four characters of the account fields) to `NJ_ParseAccount`.

```
'-----------------------------------------------------------------------
Function NJ_ParseAccount [strField, strRecord]
'-----------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'-----------------------------------------------------------------------
```

```
NJ_ParseAccount = Right (strField,4)
End Function
-------------------------------------------------------------------
```



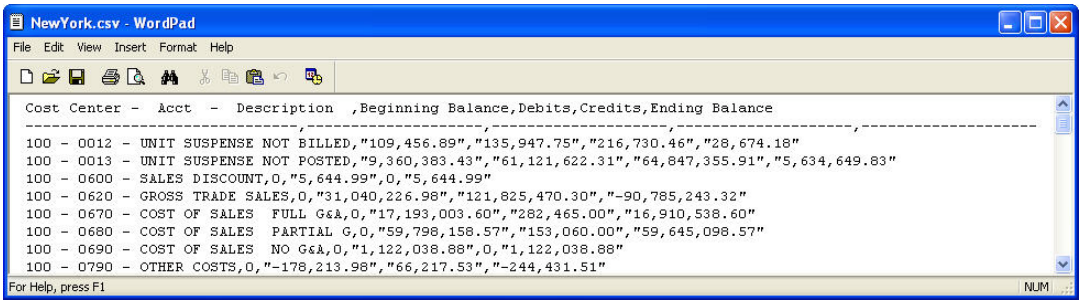
Result: In the Account column of the import file, only account numbers are displayed.

Parsing Function

`DW.Utilities.fParseString (`*`string, total field count, field # to return, delimiter`*`)`

Note: The `fParseString` function, which is used to parse strings that contain delimiters, is a FDM function, not a Visual Basic function.

Problem: The `NewYork` file presents cost center, account, and account description in one field, rather than in three fields. The three strings are separated by dashes (-). You want to present the three strings as three fields.

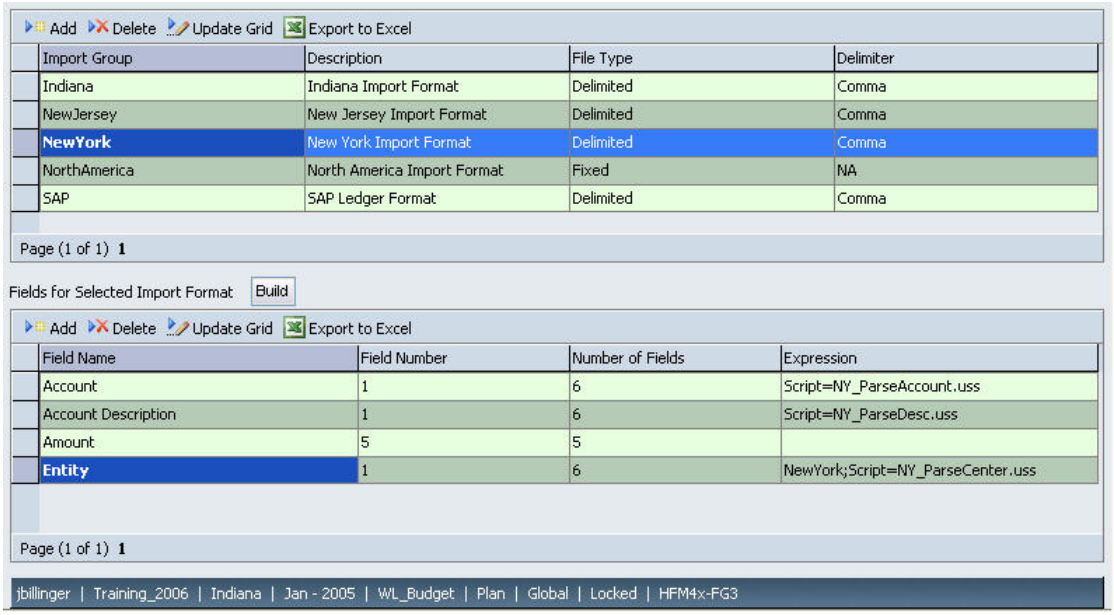


Solution: In the Import Formats screen, assign the following scripts, each of which uses the `fParseString` function, to the Expression fields of the Entity, Account, and Account Description rows (first script to Entity, second script to Account, and third script to Account Description). The first script returns the set of characters before the first hyphen (a cost center value), the second script returns the set of characters after the first hyphen (an account value),

and the third script returns the set of characters after the second hyphen (an account description value).

```
------------------------------------------------------------------
Function NY_ParseCenter [strField, strRecord]
'------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------
NY_ParseCenter = DW.Utilities.fParseString (strField, 3, 1, "-")
End Function
------------------------------------------------------------------


Function NY_ParseDesc [strField, strRecord]
'------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------
NY_ParseDesc = DW.Utilities.fParseString (strField, 3, 3, "-")
End Function
```

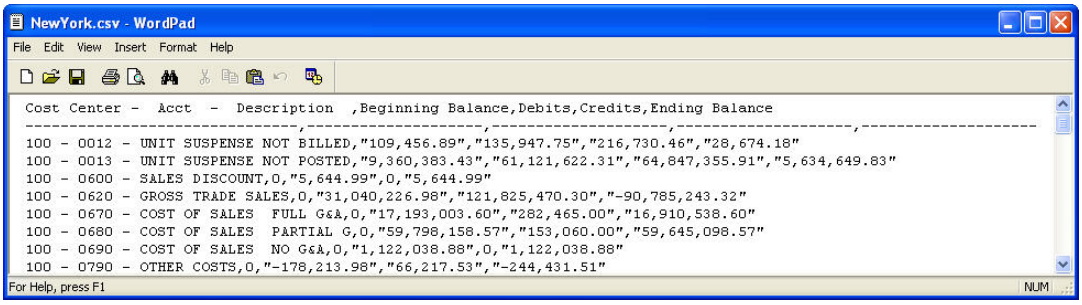


Result: In the import file, cost center, account, and account description strings are presented in three separate fields.

Skip Function (Conditional Skip)

`RES.PblnSkip`

Problem: You want FDM to skip all lines of the `NewYork` file that contain an entity value that begins with *06*.



Solution: In the Import Scripts screen, you assign the following script, which uses the `fParseString` function, to the Expression field of the Entity row. The script parses the entity column and uses a local variable that is discarded after the script executes.

```
------------------------------------------------------------------------
Function NY_Skip06Center [strField, strRecord]
'------------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------------
Dim strEntity
'Store first entity in a variable
strEntity = DW.Utilities.fParseString (strField, 3, 1, "-")
'Check first two characters of entity
If Left(strEntity, 2) = "06" then
    'Skip line
    Res.PblnSKip = True
Else
    NY_Skip06Center = strEntity
End if
End Function
------------------------------------------------------------------------
```

Result: No line that contains entity values that begin with *06* are imported.

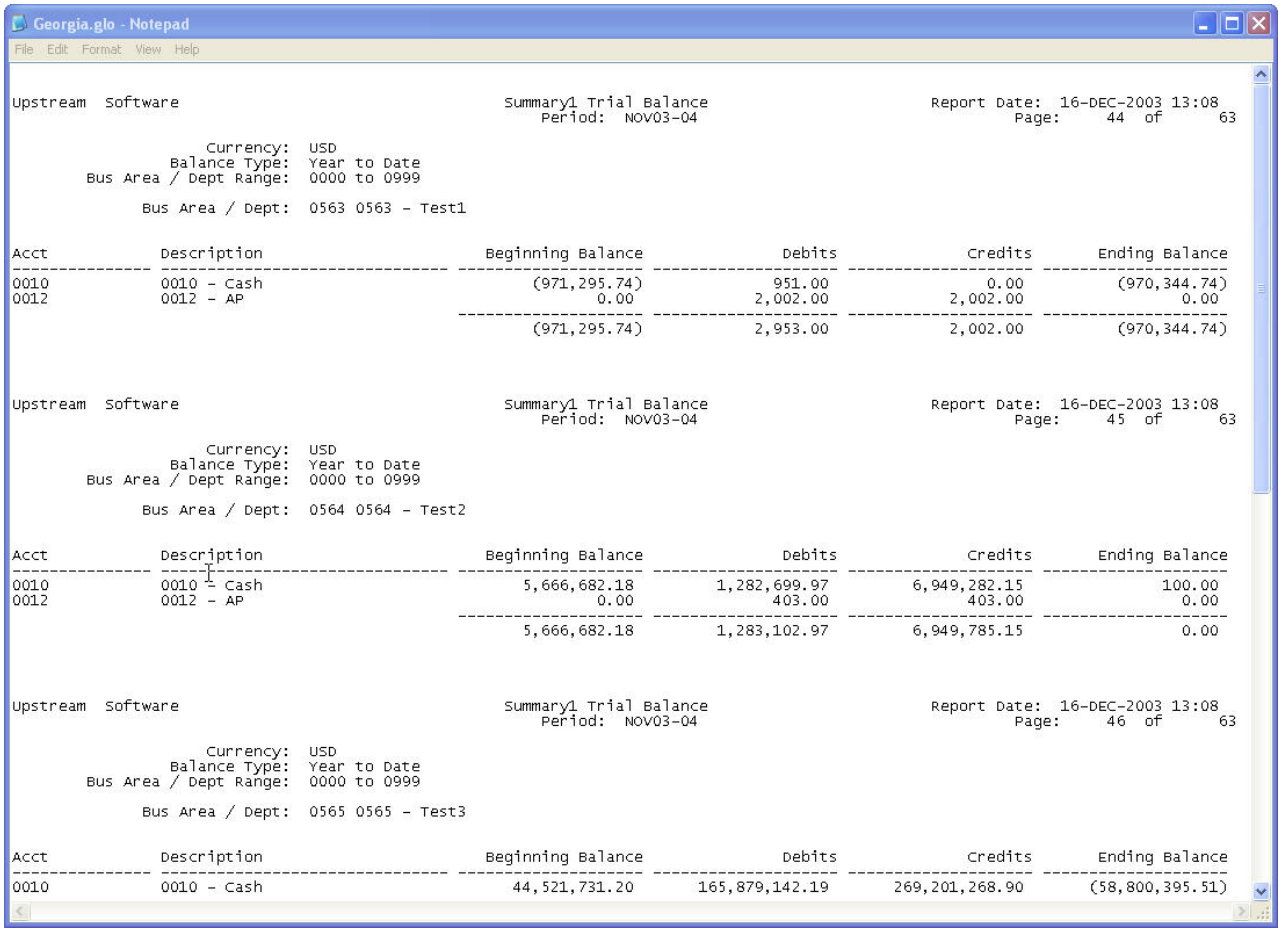
Storing and Retrieving Temporary Variables

`RES.PvarTemp1.....RES.PvarTemp5`

When FDM imports a source file, it skips lines that do not contain valid amounts but executes all scripts assigned to the Amount column, regardless of whether amounts are valid. Therefore, you can use scripts that run for lines that FDM would otherwise skip to store temporary variables that can be retrieved by other scripts. FDM can store up to five temporary variables simultaneously.

Storing Temporary Variables

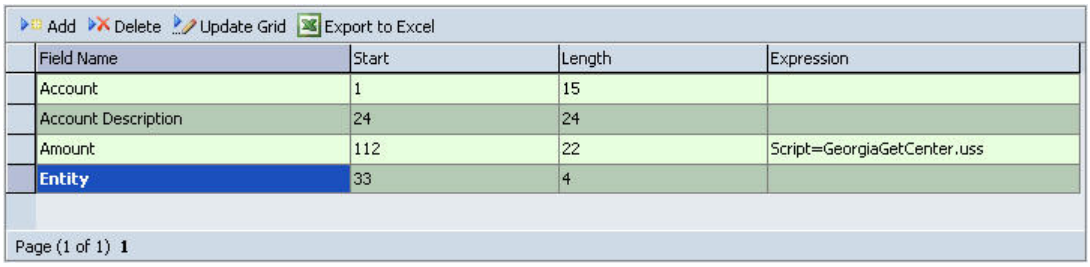
Within source files, not all lines contain all fields. For example, in the `Georgia` file, entity values, such as `0565 0565 Test 3`, are presented in the header of each report section after the `Bus Area / Dept` label but are not presented in every line. Therefore, entity values must be stored in temporary variables and assigned to the lines of the source file.



For the `Georgia` file, to store entity values in temporary variables, in the Import Formats screen, you assign the following script to the Expression field of the Amount row. The script uses an `If...Then` statement and the `Mid` function to determine whether lines contain the `Bus Area / Dept` label. If a line contains the label, the script stores the entity value (located in position 33 and including 4 characters) in a temporary variable. If a line does not include the label, `strField` is assigned to the `GetCenter` function (`GeorgiaGetCenter=strField`).

The `RES.PvarTemp1` through `RES.PvarTemp5` variables are global, and, therefore, remain available throughout the current FDM session (unlike local variables, which lose their values when the current script is closed). Therefore, values stored from within one script can be retrieved by other scripts.

```
------------------------------------------------------------------
Function GeorgiaGetCenter [strField, strRecord]
'-----------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'-----------------------------------------------------------------
If Mid(strRecord, 15,16) = "Bus Area / Dept:" then
    RES.PvarTemp1 = Mid(strRecord,33,4)
End if
GeorgiaGetCenter = strField
End Function
------------------------------------------------------------------
```



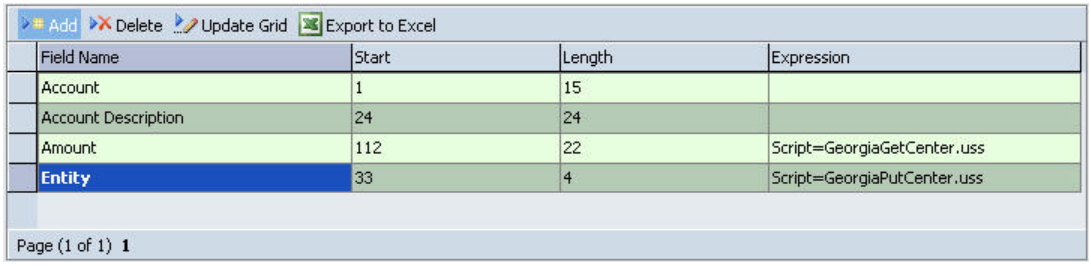
Retrieving Temporary Variables

You use scripts to assign temporary, stored variables to the fields of a column.

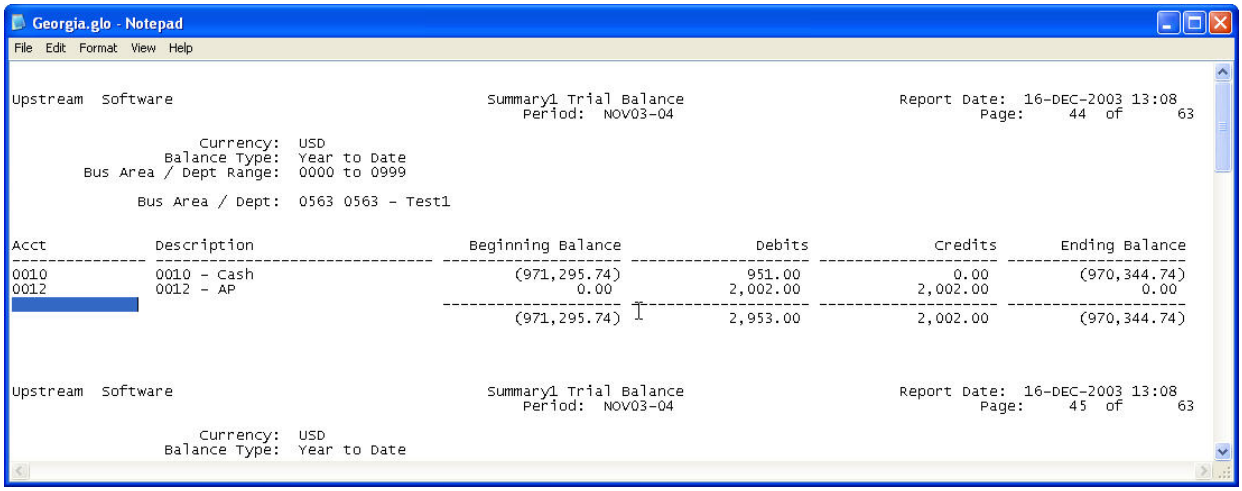
For example, if you are working with the `Georgia` file, you begin by using the following script to assign the temporary variable `PvarTempt1` to the `GeorgiaPutCenter` function.

```
------------------------------------------------------------------
Function GeorgiaPutCenter [strField, strRecord]
'------------------------------------------------------------------
' FDM DataPump Import Script:
'Created by:    FDM_Admin
'Date created: 2/28/2006
'------------------------------------------------------------------
GeorgiaPutCenter = RES.PvarTemp1
End Function
------------------------------------------------------------------
```

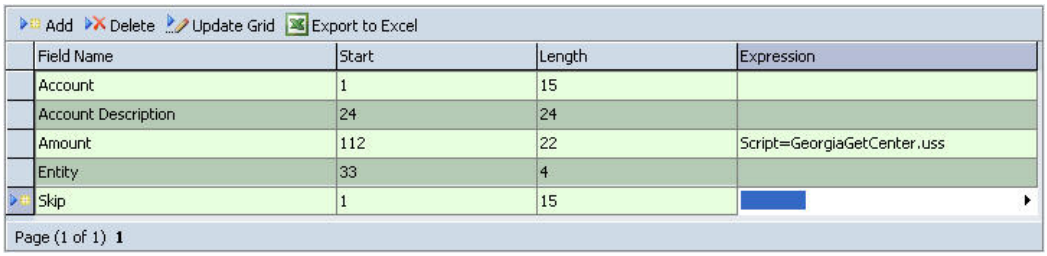
Then, in the Import Formats screen, you assign the script to the Expression field of the Entity row and, thus, assign the values of the temporary variables to the Entity fields. In this case, entity values are not read from the source file.



Because the `Georgia` file includes subtotal lines that must not be imported, the `Skip` function is required.



To direct FDM to skip lines without account numbers, you configure the `Skip` function to recognize blank Account fields (15 blank spaces), and you drag and drop to define the start and length fields for the expression.



Using Integration Import Scripts to Import Data from Specific Types of Data Sources

When data is imported directly from ODBC- or OLEDB-compliant data sources, integration import scripts (rather than ASCII files) must be used. An integration import script is responsible for connecting to source data and importing it into the FDM database.

➤ To run integration import scripts, in the **Import** screen, select a script, and click **Import**.

All FDM objects, except API objects, are supported in integration import scripts.

Creating Integration Import Scripts

When defining import groups that may be used to create integration import scripts, select Script as the file type.

When creating integration import scripts, you do not specify multiple fields in the bottom grid, as you do with other import formats. Instead, you specify only a script name.

➤ To create integration import scripts:

1 From the Web client, select **Metadata > Import Formats**.

2 In the **Import Formats** screen, from the top grid, select an import format group that is configured with Script as the file type.

3 In the bottom grid, click the arrow to the right of a row, and select **Build Script**.

Script Editor is displayed.

4 Click **New**.

5 From **Script Type**, select **Import (Integration)** script.

6 Enter a name for the script.

Note: Custom scripts do not support multibyte or extended ASCII characters in the script name or in the script parameters.

7 Click **OK**.

Integration Import Script Parameters

Four parameters are passed into an integration import script:

- **strLoc**—Active FDM location key; referenced when the FDM worktable is updated with source data
- **lngCatKey**—Active FDM category key; referenced when the FDM worktable is updated with source data
- **dblPerKey**—Date serial key of the active FDM period; referenced when the FDM worktable is updated with source data
- **strWorkTableName**—Name of the worktable into which the source data is imported for the current FDM location

FDM Worktable Fields

When creating an integration import script, you must populate the FDM worktable with the values retrieved from the data source.

Worktable Field Name	Worktable Field Information	Note
PartitionKey	[Type=Long, Size=0, Required=Yes]	WL location name (Use the `strLoc` parameter.)
CatKey	[Type=Long, Size=0, Required=Yes]	WL category key (Use the `lngCatKey` parameter.)
PeriodKey	[Type=TimeStamp, Size=0, Required=Yes]	WL period key (Use the `dblPerKey` parameter.)
DataView	[Type=VarWChar, Size=5, Required=Yes]	Load frequency; defaults to YTD
Amount	[Type=Double, Size=0, Required=Yes]	Source amount

Worktable Field Name	Worktable Field Information	Note
Desc1	[Type=VarWChar, Size=75, Required=No]	Source account description
Account	[Type=VarWChar, Size=75, Required=Yes]	Source account
Entity	[Type=VarWChar, Size=75, Required=Yes]	Source entity
ICP	[Type=VarWChar, Size=75, Required= No]	Source ICP
UD1 through UD20	[Type=VarWChar, Size=75, Required= No]	Source dimension—UserDefined1 through UserDefined20
Attr1 through Attr14	Type=VarWChar, Size=20, Required=No]	Source attribute dimension—1 through 14
ArchiveID	[Type=Bigint, Size=8, Required=No]	Archive ID number from the data directory, if an archive is to be associated

Assigning Integration import Scripts to Import Formats

After you create and save an integration import script in Script Editor, you must add the script to an import format. For instructions, see "Adding Import Expressions" on page 84.

Integration Import Script Example

The following integration import script opens a SQL server source database (Northwind) and writes data directly into FDM.

If the import is successful, True is returned. If an error occurs, False is returned.

```
'------------------------------------------------------------------
Function SQLIntegration(strLoc, lngCatKey, dblPerKey, strWorkTableName)
'------------------------------------------------------------------
' FDM Integration Import Script:
'Created By:      Admin
'Date Created: 04/19/2004 2:18:39 PM
'Purpose: This import integration script connects to the sample
'        Northwind SQL Server database and imports sample
'        source data from the Orders table into FDM.
'------------------------------------------------------------------
Dim cnSS        'ADO connection object
Dim strSQL     'SQL string
Dim rs       'Source system recordset
Dim rsAppend     ' FDM recordset
'Initialize ADO objects
Set cnSS = CreateObject("ADODB.Connection")
Set rs = CreateObject("ADODB.Recordset")
'Open  FDM work table recordset for appending
Set rsAppend = DW.DataAccess.farsTableAppend(strWorkTableName)
'Connect to Northwind SQL Server database (our data source)
Dim strConn
strConn="Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;"
strConn=strConn & "Initial Catalog=Northwind;Data Source=LocalHost;"
cnss.open strConn
'Create source query string
```

```
strSQL = "Select * "
strSQL = strSQL & "FROM Orders "
'Open source recordset
rs.Open strSQL, cnSS
'Check for data in source system
If rs.bof AND rs.eof Then
'Give error message
    RES.PlngActionType = 2
RES.PstrActionValue = "No records to load!"
    'Assign Return value of function
    SQLIntegration = False
    Exit Function
End If
'Loop through source records in Northwind database (Orders table) and append to FDM work
table
If Not rs.bof And Not rs.eof Then
    Do While Not rs.eof
        rsAppend.AddNew
        rsAppend.Fields("PartitionKey") = RES.PlngLocKey
        rsAppend.Fields("CatKey") = lngCatKey
        rsAppend.Fields("PeriodKey") = dblPerKey
        rsAppend.Fields("DataView") = "YTD"
        rsAppend.Fields("Amount") = rs.fields("Freight").Value
        rsAppend.Fields("Account") = rs.fields("CustomerID").Value
        rsAppend.Fields("Entity") = rs.fields("ShipCountry").Value
        rsAppend.Fields("Desc1") = rs.fields("ShipName").Value
        rsAppend.Update
        rs.movenext
    Loop
End If
'Give success message
RES.PlngActionType = 2
RES.PstrActionValue = "SQL Import successful!"
'Assign Return value
SQLIntegration = True
End Function
---------------------------------------------------------------------
```

Using Custom Scripts to Execute Scripts Manually

Custom scripts, which enable manual execution of scripts, are executed through a custom link on the Task Flow menu.

All FDM objects are supported in custom scripts.

Custom Script Types

- **General**—Used within Workbench
- **Web**—Used within FDM Web client
- **Plug-in**—Custom integration solutions provided by Oracle Hyperion

Creating Custom Scripts

➤ To create custom scripts:

1 **From Workbench, select File > New Script.**

The New Script dialog box is displayed.

2 **From Script Type, select Custom (Web).**

3 **In Script Name, enter a name.**

Note: Custom scripts do not support multibyte or extended ASCII characters in the script name or in the script parameters.

4 **Click OK.**

FDM prefixes *web* to the name of the script.

Using Event Scripts to Respond to FDM Events

Event scripts are executed in response to FDM events. The `Event` directory of Script Editor provides a list of trigger events. All FDM objects are supported in event scripts.

Creating Event Scripts

➤ To create event scripts:

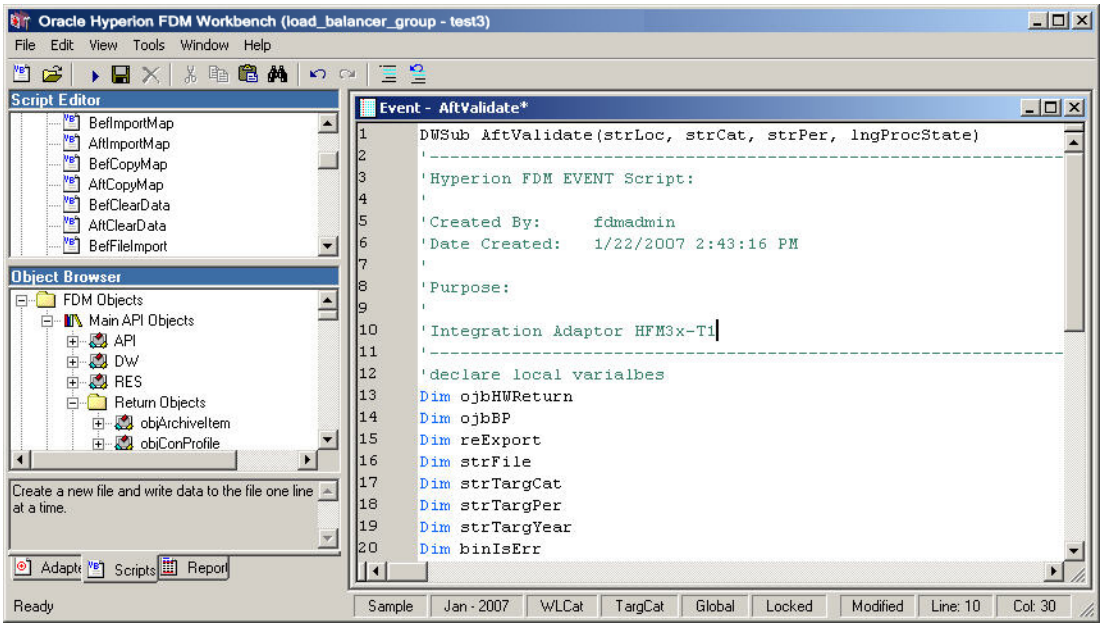
1 **From FDM Workbench, open Script Editor, and double-click an event.**

A blank event script is displayed in Script Editor.

2 **Write the script in Script Editor.**

Event Script Parameters

Each FDM event script includes a set of parameters. Scripts use their parameters—rather than FDM functions, methods, or properties—to return values. For example, using the parameter `strLoc` to return the name of the active FDM location is more efficient than using the `API.POVMgr.PPOVLocation` property.



Dynamically Changing Import Formats

You can use the `BefFileImport` event to modify the import format for a location dynamically. The following script changes the import group; the change is based on the file name.

To implement the script, you create two import formats and assign one of them to the FDM location. The script evaluates the file name and, if necessary, changes the import format.

```
---------------------------------------------------------------------
Sub BefFileImport(strLoc, strCat, strPer, strFile)
'---------------------------------------------------------------------
'Hyperion EVENT Script:
'
'Created By:      Admin
'Date Created:     10/28/2004 4:29:54 PM
'
'Purpose: Change the import group if importing the B/S
'
'---------------------------------------------------------------------
'Check if the file name contains "BS"
If InStr(strFile,"BS")>0 Then
    'Override default import group with B/S import group
    RES.PblnImportGroupOverride=True
    RES.PstrImportGroup="BSImportFormat"
End If
End Sub
---------------------------------------------------------------------
```

Dynamically Replacing Location Attributes

You can use even scripts to replace validation groups, validation rules, and log groups dynamically before they are processed.

Event Name	FDM	Location Attribute that is Replaced
BefFileImport	RES.PbInImportGroupOverride=True RES.PstrImportGroup=”MyNewImportFormat”	Import Format, before data is imported
BefProcLogicGrp	RES.PbInLogicGroupOverride=True RES.PstrLogicGroup=”MyNewLogicGroup”	Logic Group, before logic is processed
BefConsolidate	RES.PbInValEntGroupOverride=True RES.PstrValEntGroup=”MyNewEntityGroup”	Validation Entity Group, before data is consolidated
BefCheck	RES.PbInValRuleGroupOverride=True RES.PstrValRuleGroup=”MyNewValRuleGroup”	Validation Rule Group, before the validation report is run

Using the File System Object in Event Scripts

You can use the Visual Basic file system object to process files and folders. The following example uses the file system object to create a file and to copies the contents of an existing file to the new file.

```
'------------------------------------------------------------------
Sub AftExportToDat(strLoc, strCat, strPer, strTCat, strTPer, strFile)
'------------------------------------------------------------------
' FDM EVENT Script:
'
'Created By:      Admin
'Date Created:     3/18/2004 4:17:58 PM
'
'Purpose: This script loops through the newly created Hyperion
'     Enterprise .dat file And creates a new .dat file with
'     a different category and multiplies the amount by .75
'------------------------------------------------------------------
'Declare local variables
Dim strline
Dim fso
Dim f1
Dim f2
Dim strNewFileName
Dim strEntity
Dim strAcct
Dim dblAmt
'Declare file system object
Set fso = CreateObject("Scripting.FileSystemObject")
'Open existing dat file for reading
Set f1 = fso.OpenTextFile(strFile, 1)
'Create new .dat file with a "-FAS" suffix
strNewFileName=Left(strFile,Len(strFile)-4) & "-FAS" & Right(strFile,4)
Set f2=fso.CreateTextFile(strNewFileName,True)
'Write category and beginning and ending periods to new file
f2.writeLine "FAS"
f2.writeLine strTPer
f2.writeLine strTPer
'Skip first 3 header lines of existing dat file
```

```
f1.SkipLine
f1.SkipLine
f1.SkipLine
'Loop through existing .dat file
Do While f1.AtEndOfStream <> True
'Store line in a variable
     strline = f1.ReadLine
     'Parse the entity from the line
     strEntity = DW.Utilities.fParseString(strline, 3, 1, ",")

     'Parse the account from the file
     strAcct = DW.Utilities.fParseString(strline, 3, 2, ",")
     'Parse the amount from the file
     dblAmt = DW.Utilities.fParseString(strline, 3, 3, ",")

     'Write out line to new .dat file but multiply amt by .75
     f2.writeline strEntity & "," & strAcct & "," & dblAmt * .75
Loop
'Close the files
f1.Close
f2.Close

'Destroy file system object
Set fso=Nothing
End Sub
-----------------------------------------------------------------------
```

See Appendix A for advanced information about scripting in FDM.

Using Menu Maker to Create Menu Options That Invoke Scripts

You can use Menu Maker to assign custom Web scripts to menu commands. Then, the menu commands can be used to execute the scripts.

➤ To create menu commands that can be used to execute scripts:

1 **Within the Web client, select Tools > Menu Maker.**

The Menu Maker screen is displayed.

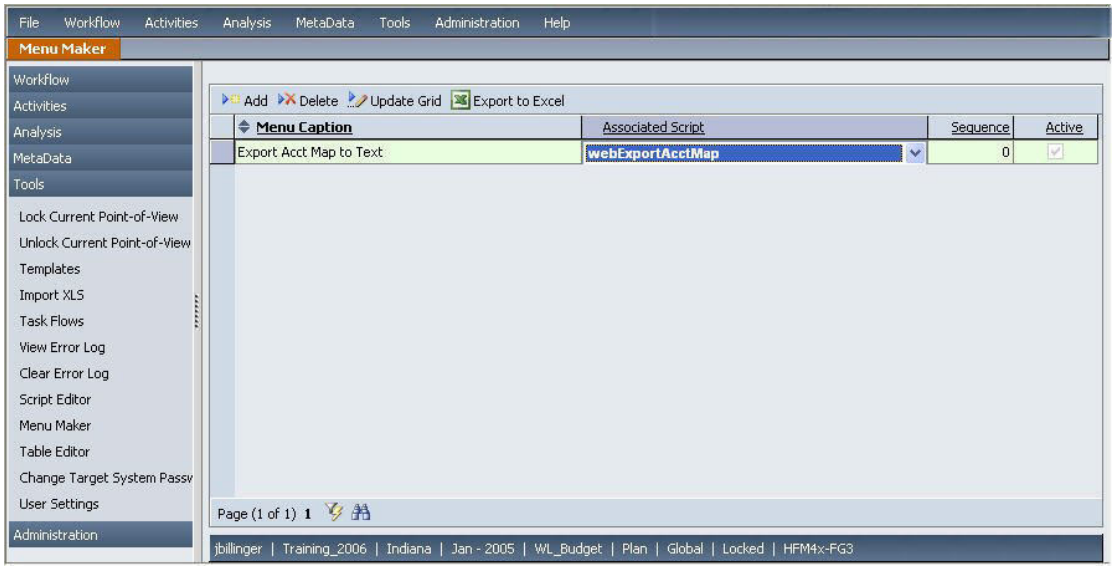
2 **Click Add.**

A row is added to the grid.

3 **Under Menu Caption, enter the text of a menu command.**

4 **In the Associated Script column, double-click, and select a Web script.**

5 **Select Active to list the script on the Task Flows screen of the Web client.**



➤ To execute scripts from the Task Flows screen:

1 **From the Web client, select Tools > Task Flows.**

2 **Click a menu option that identifies a script.**

The script runs immediately.

Executing Scripts in Parallel Mode

FDM can execute scripts in parallel by using a command line interface object. Scripts that are run from within asynchronous script shells execute in parallel mode.

Execution in parallel mode is also supported in Task Manager. Task Manager enables scripts to execute simultaneously or asynchronously.

```
Sub AsynchScript()
'--------------------------------------------------------------------
'FDM Custom Script:
'
'Created By:      Admin
'Date Created:     3/23/2006 13:57
'
'Purpose:        Asynchronous Script Processing
'
'--------------------------------------------------------------------
'Declare Local Variables
Dim strScriptToRun
Dim strLoadBalanceServerName
'Initialize variables Variables
strScriptToRun = "YourScriptName"  'Name of the script to execute (Type Custom General
Only)
strLoadBalanceServerName = "LocalHost"     'Load balance server for Asynchronous process
to use
'Execute the script
```

```
If API.DataWindow.Utilities.fExecuteCustomScriptAsync(CStr(strScriptToRun),
CStr(strLoadBalanceServerName)) = True Then
    'Async Script Started, you can make another call while it is executing
Else
    'Async process failed to start
End If
End Sub
-----------------------------------------------------------------
```

Using UPSShell.exe

UPSShell.exe is a utility contained within the FDM installation. UPSShell.exe assists you with writing a command line that can be used to launch an FDM script, much in the same way that the FDM Task Manager would be used.

➤ Using UPSShell.exe to Write a Command Line:

1 **Locate UPSShell.exe**

UPSShell is located in `\\<fdm home>\shared components\`.

2 **Double-click UPSShell.exe**

This action copies a list of parameters to the clipboard.

3 **Open a new document in a text editor (WordPad, for example) and paste the contents of the clipboard**

The pasted text will look something similar to the following: `<fdm home>\SHARED~1\upsShell.exe CustomScriptEx=AppName~UserID~PW~DOMAIN~LoadBalancerName~LogDirectory~ScriptName~LanguageCode(Default=1033)~EncodeUnicode(1).`

4 **Replace the following fields with the corresponding information from you application:**

- AppName – Name of the FDM application.
- UserID – User ID to use to gain access to FDM.
- PW – Password of the User ID used to gain access to FDM.
- Domain – Domain entered by the user in the UserID parameter.
- Load Balancer Name – Name of the load balance server used with the FDM application.
- Log Directory – File path where you want the UPSShell.exe log file written.
- Script Name – Name of the FDM script to execute. The new text (with fields replaced) can now be used from a command line or executed from within another program or script.
- Language Code – English = 1033, French = 1036, German = 1031, Danish = 1030.
- Encode Unicode – 1 = Unicode. This must always be set to one (1).

Frequently Used Visual Basic Scripting Functions

In This Appendix

About This Appendix

This appendix describes some of the more frequently used Visual Basic scripting functions.

For a thorough discussion of Visual Basic scripting, see

`http:\\www.msdn.microsoft.com\\vbscript.`

InStr and InStrRev

Returns the position of the first occurrence of one string within another string (searching from the beginning of the string (InStr), the end of a string (InStrRev) or from a specified character within the string)

```
InStr([start, ]string1, string2[, compare])InStrRev(string1, string2[, start[,
compare]])
```

Arguments

- `start`—Numeric expression that sets the starting position for each search (If the `start` expression is omitted, the search begins with the first character of a string (InStr), or –1 (InStrRev). If `start` contains `Null`, an error occurs. The `start` argument is required if `compare` is specified.)
- `string1`—String expression being searched
- `string2`—String expression being searched for
- `compare`—Numeric expression that specifies the kind of comparison to use when substrings are evaluated (If `compare` is omitted, a binary comparison is performed.)

Settings

`Compare` Argument Constant	**`Compare` Argument Value**	**Action Performed**
`vbBinaryCompare`	0	Binary comparison
`vbTextCompare`	1	Textual comparison

Return Values

Value of `InStr` Argument	**Value Returned by `InStr`**
`string1` is zero-length	0
`string1` is `Null`	`Null`
`string2` is zero-length	`start`
`string2` is `Null`	`Null`
`string2` is not found	0
`string2` is found within string1	Position at which a match is found
`start > Len (string2)`	0

Examples

```
Dim SearchString, SearchChar, MyPos
SearchString ="XXpXXpXXPXXP"   ' String to search in.
SearchChar = "P"   ' Search for "P".
MyPos = InStr(4, SearchString, SearchChar, 1)   ' A textual comparison starting at
position 4. Returns 6.
```

```
MyPos = InStr(1, SearchString, SearchChar, 0)  ' A binary comparison starting at
position 1. Returns 9.
MyPos = InStr(SearchString, SearchCharb  ' Comparison is binary by default (last
argument is omitted). Returns 9.
MyPos = InStr(1, SearchString, "W")  ' A binary comparison starting at position 1.
Returns 0 ("W" is not found).
MyPos = InStrRev(SearchString, SearchChar, 10, 0)  ' A binary comparison starting at
position 10. Returns 9.
MyPos = InStrRev(SearchString, SearchChar, -1, 1)  ' A textual comparison starting at
the last position. Returns 12.
MyPos = InStrRev(SearchString, SearchChar, 8)  ' Comparison is binary by default (last
argument is omitted). Returns 0
```

Note: The `InStrB` function, used with byte data that is contained in strings, returns byte position, rather than character position.

Note: The syntax for the `InStrRev` function differs from the syntax for the `InStr` function.

IsNumeric

Returns a Boolean value that indicates whether an expression can be evaluated as a number

`IsNumeric(`*`expression`*`)`

Argument

`expression`—any expression

Remarks

For expressions recognized as numbers, `IsNumeric` returns True. For expressions not recognized as numbers, including dates, `IsNumeric` returns False.

Examples

```
Dim MyVar, MyCheck
MyVar = 53  ' Assign a value.
MyCheck = IsNumeric(MyVar)  ' Returns True.
MyVar = "459.95"  ' Assign a value.
MyCheck = IsNumeric(MyVar)  ' Returns True.
MyVar = "45 Help"  ' Assign a value.
MyCheck = IsNumeric(MyVar)  ' Returns False.
```

Len

Returns the number of characters in a string

`Len(string)`

Returns the number of bytes required to store a variable

`Len (varname)`

Arguments

- `string`—Any valid string expression (For strings that contain `Null`, `Null` is returned.)
- `varname`—Any valid variable name (For variable names that contain `Null`, `Null` is returned.)

Example

```
Dim MyString
MyString = Len("VBSCRIPT") ' MyString contains 8.
```

Note: The `LenB` function, used with byte data that is contained in strings, returns the number of bytes represented by a string, rather than the number of characters in a string.

Left

Returns a specified number of characters from the left side of a string

`Left(string, length)`

Arguments

- `string`—String expression from which the left-most characters are returned (For strings that contain `Null`, `Null` is returned.)
- `length`—Numeric expression that specifies how many characters to return (If length is 0, a zero-length string ("") is returned. If length is greater than or equal to the number of characters in a string, all characters within the string are returned.)

Example

```
Dim MyString, LeftString
MyString = "VBScript"
LeftString = Left(MyString, 3) ' LeftString contains "VBS".
```

Note: In the `LeftB` function, used with byte data that is contained in strings, `length` specifies number of bytes, rather than number of characters.

LCase

Returns a string that was converted to lowercase

```
LCase(string)
```

The string argument is any valid string expression. For strings that contain `Null`, `Null` is returned.

Argument

`string`—Any valid string expression (For strings that contain `Null`, `Null` is returned.)

Remarks

Only uppercase letters are converted to lowercase. Lowercase letters and non-letter characters are unchanged.

Example

```
Dim MyString
Dim LCaseString
MyString = "VBScript"
LCaseString = LCase(MyString) ' LCaseString contains "vbscript".
```

LTrim, RTrim, and Trim

Returns a copy of a string without leading spaces (`LTrim`) or without trailing spaces (`RTrim`) or with neither leading nor trailing spaces (`Trim`)

```
LTrim(string)
RTrim(string)
Trim(string)
```

Argument

`string`—Any valid string expression (For strings that contain `Null`, `Null` is returned.)

Examples

```
Dim MyVar
MyVar = LTrim("  vbscript ")  ' MyVar contains "vbscript "
MyVar = RTrim("  vbscript ")  ' MyVar contains "  vbscript"
MyVar = Trim("  vbscript ")  ' MyVar contains "vbscript"
```

Mid

Returns a specified number of characters from a string

`Mid(string, start[, length])`

Arguments

- `string`—String expression from which characters are returned (For strings that contain `Null`, `Null` is returned.)
- `start`—Position of the first character to be returned (If the `start` value is greater than the number of characters in the string, `Mid` returns a zero-length string.)
- `length`—Number of characters to be returned (If `length` is omitted or if the value of `length` exceeds the number of characters between the first character and the end of the string (including the first character), all characters from the `start` position to the end of the string are returned.)

Example

```
Dim MyVar
MyVar = Mid("VBScript is fun!", 4, 6) ' MyVar contains "Script".
```

Note: In the `MidB` function, used with byte data that is contained in strings, arguments specify bytes and number of bytes, rather than characters and number of characters.

MsgBox

Displays a message in a dialog box, waits for the user to click a button, and returns a value that indicates which button the user clicked

`MsgBox(prompt[, buttons][, title][, helpfile, context])`

Note: Messages that use the Web client must be displayed in a new window. Therefore, in the Web client, use the Show Message accelerator provided in Workbench Object Browser, not the `MsgBox` function.

Arguments

- `prompt`—String expression that is displayed as the message in the dialog box (maximum length determined by character width; approximate maximum of 1024 characters; lines of messages with multiple lines separated by carriage returns (`Chr(13)`), linefeeds (`Chr(10)`), or combinations of carriage returns and linefeeds)
- `buttons`—Numeric expression that specifies the number and type of buttons to be displayed, the icon style to be used, the identity of the default button, and the modality of

the message box (If `buttons` is omitted, the `buttons` value is 0). Refer to Microsoft MSDN information for more details about this feature.

- `title`—String expression that is displayed in the title bar of the dialog box (If `title` is omitted, the application name is placed in the title bar.)
- `helpfile` (not available on 16-bit platforms)—String expression that identifies the help file that provides context-sensitive help for the dialog box (If `helpfile` is provided, `context` must be provided.)
- `context` (not available on 16-bit platforms)—Numeric expression that identifies the help context number assigned by the help author to the relevant help topic (If `context` is provided, `helpfile` must be provided.)

Remarks

When `helpfile` and `context` are provided, users can press F1 to view the help topic identified by the `context` value.

Clicking a Cancel button and pressing Esc produce the same result. For dialog boxes that contain Help buttons, context-sensitive help is available

When the `MsgBox` function is used with Internet Explorer, the titles of all dialog boxes contain *VBScript*—to differentiate the boxes from standard system dialog boxes.

Example

```
Dim MyVar

MyVar = MsgBox ("Hello World!", 65, "MsgBox Example")

   ' MyVar contains either 1 or 2, depending on which button is clicked.
```

Now

Returns the current date and time, as reported by your computer

```
Now
```

Argument

The `Now` function does not use arguments.

Example

```
Dim MyVar
MyVar = Now ' MyVar contains the current date and time
```

Replace

Returns a string in which a specified substring was replaced with another substring a specified number of times

```
Replace(expression, find, replacewith[, start[, count[, compare]]])
```

Arguments

- `expression`—String expression that contains the substring to replace
- `find`—Substring being searched for
- `replacewith`—Replacement substring
- `start` (used with `count`)—Position within `expression` where the substring search begins (If `start` is omitted, 1 is assumed.)
- `count` (used with `start`)—Number of substring substitutions to be performed (If `count` is omitted, the default value is -1; thus, all possible substitutions occur.)
- `compare`—Numeric value that specifies the kind of comparison to use when substrings are evaluated (If `compare` is omitted, the default value is 0; thus, a binary comparison is performed.)

Settings

`Compare` Argument Constant	**`Compare` Argument Value**	**Action Performed**
`vbBinaryCompare`	0	Binary comparison
`vbTextCompare`	1	Textual comparison

Return Values

Value of `Replace` Argument	**Value Returned by `Replace`**
`expression` is zero-length	Zero-length string (“”)
`expression` is `Null`	An error
`find` is zero-length	Copy of expression
`replacewith` is zero-length	Copy of expression with all occurrences of `find` removed
`start > Len(expression)`	Zero-length string
`count` is 0	Copy of expression

Remarks

The return value of the `Replace` function is a string, with substitutions made, that begins at the position specified by `start` and concludes at the end of the expression string. The return value is not a copy of the original string.

Example

```
Dim MyString
' A binary comparison starting at the beginning of the string.
MyString = Replace("XXpXXPXXp", "p", "Y")
' Returns "XXYXXPXXY".
' A textual comparison starting at position 3.
MyString = Replace("XXpXXPXXp", "p", "Y", 3, -1, 1)
' Returns "YXXYXXY".
```

Right

Returns a specified number of characters from the right side of a string.

`Right(`*`string, length`*`)`

Arguments

- `string`—String expression from which the right-most characters are returned (If `string` contains `Null`, `Null` is returned.)
- `length`—Numeric expression that indicates how many characters to return (If `length` is 0, a zero-length string is returned. If `length` is greater than or equal to the number of characters in `string`, all characters of the string are returned.)

Example

```
Dim AnyString, MyStr
AnyString = "Hello World"    ' Define string.
MyStr = Right(AnyString, 1)  ' Returns "d".
MyStr = Right(AnyString, 6)  ' Returns " World".
MyStr = Right(AnyString, 20)  ' Returns "Hello World".
```

Note: In the `RightB` function, used with byte data that is contained in strings, `length` specifies number of bytes, rather than number of characters.

UCase

Returns a string that was converted to uppercase

`UCase(`*`string`*`)`

Argument

`string`—Numeric expression that specifies how many characters to return (If string is 0, a zero-length string is returned. If `string` is greater than or equal to the number of characters in the current string, all characters of the string are returned.)

Remarks

Only lowercase letters are converted to uppercase. Uppercase letters and non-letter characters remain unchanged.

Example

```
Dim MyWord
MyWord = UCase("Hello World")   ' Returns "HELLO WORLD".
```

Do...Loop

Repeats a block of statements while a condition is True

```
Do [While condition]
  [statements]
  [Exit Do]
  [statements]
Loop
```

OR

```
Do
  [statements]
  [Exit Do]
  [statements]
Loop [While condition]
```

Repeats a block of statements until a condition becomes True:

```
Do [Until condition]
  [statements]
  [Exit Do]
  [statements]
Loop
```

OR

```
Do
    [statements]
    [Exot Do]
    [statements]
Loop [Until condition]
```

Arguments

- `condition`—Numeric or string expression that is True or False (If `condition` is `Null`, its expression is treated as False.)
- `statements`—One or more statements that are repeated while or until `condition` is True

Remarks

`Exit Do`, an alternate way to exit a `Do...Loop` structure, can be used only within `Do...Loop` structures. Any number of `Exit Do` statements can be placed anywhere within a `Do...Loop` structure.

Often used with the evaluation of a condition (for example, `If...Then`), `Exit Do` transfers control to the statement that immediately follows `Loop`.

When used within nested `Do...Loop` statements, `Exit Do` transfers control to the loop that is nested one level above the loop in which it occurs.

Example

```
Do Until DefResp = vbNo
  MyNum = Int(6 * Rnd + 1)  ' Generate a random integer between 1 and 6.
  DefResp = MsgBox (MyNum & " Do you want another number?", vbYesNo)
Loop
Dim Check, Counter
Check = True: Counter = 0  ' Initialize variables.
Do              ' Outer loop.
  Do While Counter < 20   ' Inner loop.
   Counter = Counter + 1  ' Increment Counter.
   If Counter = 10 Then  ' If condition is True...
     Check = False    ' set value of flag to False.
     Exit Do       ' Exit inner loop.
   End If
  Loop
Loop Until Check = False   ' Exit outer loop immediately.
```

If...Then...Else

Conditionally executes a group of statements, depending on the value of an expression.

```
If condition Then statements [Else elsestatements]
```

OR

```
If condition Then
  [statements]
[ElseIf condition-n Then
  [elseifstatements]] . . .
[Else
  [elsestatements]]
End If
```

Arguments

- `condition`—One or both of two types of expressions
  - Numeric or string expression that evaluates to True or False (If `condition` is `Null`, the expression is treated as False.)
  - Expression of the form `TypeOf` *`objectname`* `Is` *`objecttype`*. (`objectname` is any object reference, and `objecttype` is any valid object type. The expression is True if `objectname` is of the object type specified by `objecttype`; otherwise, the expression is False.)
- `statements`—One or more statements separated by colons; executed if `condition` is True
- `condition-n`—Same as `condition`
- `elseifstatements`—One or more statements executed if the associated `condition-n` is True
- `elsestatements`—One or more statements executed if no previous `condition` or `condition-n` expression is True

Remarks

You can use the single-line form (first syntax) for short, simple tests. However, the block form (second syntax) provides more structure and flexibility and is usually easier to read, maintain, and debug.

Note: With the single-line form, you can execute multiple statements as the result of an `If...Then` decision, provided that you place the statements on one line and separate the statements by colons, as in the following statement:

```
If A > 10 Then A = A + 1 : B = B + A : C = C + B
```

When a block `If` (second syntax) is executed, `condition` is tested. If `condition` is True, the statements following `Then` are executed. If `condition` is False, each `ElseIf` (if any) is evaluated in turn. When a True condition is found, the statements following the associated `Then` are executed. If no `ElseIf` is True (or if no `elseifstatements` exist), the statements following `Else` are executed. After the statements following `Then` or `Else` are executed, execution continues with the statement that follows `End If`.

`elsestatements` and `elseifstatements` are optional. An `If` block can contain multiple `elseifstatements`, but no `elseifstatement` can be displayed after an `elsestatement`. Block `If` statements can be nested.

What follows the `Then` keyword is examined to determine whether a statement is a block `If`. Any statement, other than a comment, that is displayed after and on the same line as `Then` is treated as a single-line `If` statement.

A block `If` statement must be the first statement on a line. A block `If` statement must end with an `End If` statement.

Processing Files

To process files, you can use the `fso` (filesystemobject) object model, which combines object-method syntax with a rich set of properties, methods, and events.

Creating Files

Three ways to create empty text files (sometimes referred to as `TextStream`):

- `CreateTextFile` method

```
[VBScript]
Dim fso, f1
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
```

- `OpenTextFile` method of the FileSystemObject object with the `ForWriting` flag set

```
[VBScript]
Dim fso, ts
Const ForWriting = 2
Set fso = CreateObject("Scripting.FileSystemObject")
Set ts = fso.OpenTextFile("c:\test.txt", ForWriting, True)
```

- `OpenAsTextStream` method with the `ForWriting` flag set

```
[VBScript]
Dim fso, f1, ts
Const ForWriting = 2
Set fso = CreateObject("Scripting.FileSystemObject")
fso.CreateTextFile ("c:\test1.txt")
Set f1 = fso.GetFile("c:\test1.txt")
Set ts = f1.OpenAsTextStream(ForWriting, True)
```

Adding Data to Files

➤ To add data to text files:

1 **Using the `OpenTextFile` method of the `fso` object or the `OpenAsTextStream` method of the file object, open a file.**

2 **Using the `Write`, `WriteLine`, or `WriteBlankLines` methods of the `TextStream` object, write data to the file:**

- `Write`—Without a trailing newline character
- `WriteLine`—With a trailing newline character
- `WriteBlankLines`—With one or more blank lines

3 **Using the `Close` method of the `TextStream` object, close the file.**

Note: The newline character contains a character or characters (depending on the operating system) that advances the cursor to the beginning of the next line. The ends of some lines

may include non-printing characters; in such cases, the newline character may duplicate an existing character or characters.

Example of Opening, Adding Data to, and Closing Files

```
[VBScript]
Sub CreateFile()
  Dim fso, tf
  Set fso = CreateObject("Scripting.FileSystemObject")
  Set tf = fso.CreateTextFile("c:\testfile.txt", True)
  ' Write a line with a newline character.
  tf.WriteLine("Testing 1, 2, 3.")
  ' Write three newline characters to the file.
  tf.WriteBlankLines(3)
  ' Write a line.
  tf.Write ("This is a test.")
  tf.Close
End Sub
```

Reading Files

To read data from text files, use one of the following `TextStream` object methods:

- `Read`—Read a specified number of characters
- `ReadLine`—Read one line (up to, but not including, the newline character)
- `ReadAll`—Read all contents

If using the `Read` or `ReadLine` method, to skip to a particular portion of data, use the `Skip` or `SkipLine` method. The text produced by the read method is stored in a string. The string can be displayed in a control, parsed by string functions (such as `Left`, `Right`, and `Mid`), concatenated, and so forth.

The following example opens, writes data to, and reads from a file:

```
[VBScript]
Sub ReadFiles
  Dim fso, f1, ts, s
  Const ForReading = 1
  Set fso = CreateObject("Scripting.FileSystemObject")
  Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
  ' Write a line.
  Response.Write "Writing file <br>"
  f1.WriteLine "Hello World"
  f1.WriteBlankLines(1)
  f1.Close
  ' Read the contents of the file.
  Response.Write "Reading file <br>"
  Set ts = fso.OpenTextFile("c:\testfile.txt", ForReading)
  s = ts.ReadLine
  Response.Write "File contents = '" & s & "'"
  ts.Close
```

```
End Sub
```

Moving, Copying, and Deleting Files

The `fso` object model provides two methods each for moving, copying, and deleting files:

- `File.Move or FileSystemObject.MoveFile`
- `File.Copy or FileSystemObject.CopyFile`
- `File.Delete or FileSystemObject.DeleteFile`

To run the following example, you must create directories named `\tmp` and `\temp` in the root directory of drive C. The example performs the following actions:

1. Creates a text file in the root directory of drive C.
2. Writes information to the file.
3. Moves the file to a directory named `\tmp`.
4. Copies the file to a directory named `\temp`.
5. Deletes the copies from both directories.

```
[VBScript]
Sub ManipFiles
  Dim fso, f1, f2, s
  Set fso = CreateObject("Scripting.FileSystemObject")
  Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
  Response.Write "Writing file <br>"
  ' Write a line.
  f1.Write ("This is a test.")
  ' Close the file to writing.
  f1.Close
  Response.Write "Moving file to c:\tmp <br>"
  ' Get a handle to the file in root of C:\.
  Set f2 = fso.GetFile("c:\testfile.txt")
  ' Move the file to \tmp directory.
  f2.Move ("c:\tmp\testfile.txt")
  Response.Write "Copying file to c:\temp <br>"
  ' Copy the file to \temp.
  f2.Copy ("c:\temp\testfile.txt")
  Response.Write "Deleting files <br>"
  ' Get handles to files' current location.
  Set f2 = fso.GetFile("c:\tmp\testfile.txt")
  Set f3 = fso.GetFile("c:\temp\testfile.txt")
  ' Delete the files.
  f2.Delete
  f3.Delete
  Response.Write "All done!"
End Sub
```

B

In This Appendix

Tips and Troubleshooting

This section addresses common FDM issues.

Mapping Table Export

Problem: When attempting to export a mapping table to Excel using the "Export to Excel" button, the export fails. The process is suspended on the "please wait" screen.

Cause: The Essbase add-in for Excel is registered within Excel on the FDM application server and causes the export process to be suspended.

Solution: Unregister the add-in.

➤ To unregister the add-in:

1. Log on to the FDM application server as the service account set in all the FDM configuration items.
2. Select **Start**, then **Programs**, then **Oracle EPM System**, then **Essbase**, and then **Essbase Add-in**.

   A command window is displayed.
3. Right-click in the Windows taskbar and select **Task Manager**.
4. Select the Processes tab and, to sort by image, click on the **Image Name** column header.
5. Right-click on all EXCEL.EXE processes and Select **End Process**.
6. Log in to the FDM Web Client and rerun the map export function.

Data Export to Essbase

Problem: When attempting load data to Essbase or browsing for target values in the mapping table, the process returns the error message: `Error: Export failed. Detail: File not found: esbapin.DLL`

Cause: FDM Essbase adapter `.exe` cannot locate `esbapin.dll`, which is part of the Essbase API.

Detail: esbapin.dll is a file of the Essbase client API that FDM uses to connect to Essbase. It is located in the *Essbase home*\Bin directory in the Essbase client installation directory. The Essbase adapter .exe file is registered within the FDM adapters folder and cannot locate esbapin.dll.

Solution: Register the Essbase adapter .exe file in the Essbase\Essbase Client\Bin directory.

➤ To register the adapter:

1. **Copy ES11xG4G.exe (or appropriate FDM Essbase adapter version .exe) to *Oracle home*\products\Essbase\Essbase Client\Bin.**
2. **Launch FDM Workbench.**
3. **Log in to the FDM application using an administrator account.**
4. **Select File, then Register Adapter, and then browse to the *Oracle home*\products\essbase\essbase client\bin directory and select the Oracle Essbase adapter .exe file.**
5. **Click Open.**

Delete and Trim tLogActivity Table Entries

➤ To deleting or trim entries from the tLogActivity table:

1. **Using the FDM Web client, log in to an FDM application.**
2. **Select Analysis, and then Log Option.**
3. **Press the "Delete" key on the keyboard.**
4. **Enter a threshold date in the format MM/DD/YYYY.**
5. **Click OK.**
6. **Confirm delete when "Delete all log entries, for all locations, prior to and including threshold date: MM/DD/YYYY?" is displayed.**

Application Error on Login Page

Problem: The FDM Web logon page does not display. The browser shows "Application Error."

Cause: When a browser pulls information from the Web, it sends an HTTP request to the server, which contains information about its language preference settings [Accept-Language request-header]. An error occurs if this language setting is not correct.

Solution: Open Internet Explorer and Select Tools, then Internet Options, and then Languages, and verify that "English (United States) [un-es]" is selected.

Automation Error During Validation

Problem: The Validation step in the FDM workflow, the user experiences "Automation Error" even though the Validate step passes successfully.

Note: This applies only when integrating with a Financial Management application.

Cause: The FDM user cannot be found in the Financial Management application. The user logging in to FDM enters the domain name into the domain field. This login is passed to Financial Management as *domain entered**username* causing Oracle Hyperion Financial Management, Fusion Edition to find the user invalid for the application.

Solution: Log in to FDM with the domain field empty and rerun the Validation step.

View File Error on Export Screen

Problem: Clicking View File on the Export screen results in a "Page Cannot Be Displayed" error.

Solution: Add the `.dat` file to the MIME types within IIS on the FDM Web Server.

➤ To add the `.dat` file:

1. From Windows **Select Start**, then **Settings**, then **Control Panel**, and then **Administrative Tools**.

   The Administrative Tools screen is displayed.

2. Double-click Internet Information Services.

   The Internet Information Services screen is displayed.

3. Expand Web Sites.
4. Expand Default Web Site.
5. Expand the HyperionFDM virtual directory.
6. Right-click HyperionFDM virtual directory and select **Properties**.
7. Select the **HTTP Headers** tab.
8. In the MIME MAP section, click **File Types**.

   The File Types dialog box is displayed.

9. Click New Type.

   The New Type dialog box is displayed.

10. Enter `.dat` in the Associated Extension field and enter `.txt` in the Content Type (MIME) field.

Drill Through Dimension Names Incorrect when Using an Alternate Adapter

Problem: When drilling through to a target system assigned to an alternate (nondefault) adapter, the Drill Through page displays dimension names for the default adapter.

Cause: This behavior occurs when both the default adapter and alternate adapter have the same value in the AppName option.

Solution: Ensure that all adapters have unique AppName values.

Password-Protected Excel Files

Problem: If an Excel file is password protected, it could cause FDM to stop responding when the file is accessed.

Solution: Set the Excel sheet that contains the UPS range as the current sheet.

Unpopulated tPOVCategoryAdapter Tables

Problem: When using user-defined Excel templates to populate the Period and Category tables, the tPOVCategoryAdapter and tPOVPeriodAdapter tables are not populated automatically.

Solution: Use your Excel template to populate the tPOVCategoryAdapter and tPOVPeriodAdapter tables.

One-Hour Web Page Timeouts

Problem: Internet Explorer times out after an hour.

Solution: See the Microsoft Knowledge Base article: http://support.microsoft.com/kb/181050 to change Internet Explorer timeout settings.

File Import Error

Problem: When attempting to perform an import of a GL file into FDM, the following error message (or similar) is returned: `Error: An error occurred importing the file! Detail: File [C:\DOCUME~1\! <serviceaccount>\LOCALS~1\Temp \tWhypadm10936117046556-SL.log] does not exist!`

Cause: When you use the Oracle SQL Loader to import data into the FDM schema within the Oracle Instance, a temp file is generated within the user profiles (`C:\documents and settings\\applicationdata`). When the SQL Loader is attempting to write the log to this directory it encounters an issue and the import fails.

➤ To verify the cause, perform the following tests:

1 **Open the FDM application.**

2 Select **Metadata**, and then **Locations**.

3 Find the location set in the POV.

4 Change **Insert Type** from **Bulk Insert** to **SQL Insert**.

5 Click **Save**.

6 Rerun the import process and verify that the import completes successfully.

When using the SQL Insert option, the SQL Loader is not used during the import operation.

Solution: Reinstall the SQL Loader Database Utility on the FDM server by performing the following procedure:

1. Open the FDM application.
2. Working from the Oracle Hyperion Financial Data Quality Management, Fusion Edition server, launch the Oracle Client Installer.
3. Choose Custom Install and click Next.
4. Expand Database Utilities, and select the SQL Loader option or choose to install all database utilities and click Next.
5. Choose the current Oracle home directory to install the components and click Next.
6. When the installation is complete, click Finish.
7. Reset the insert type that you changed in the test procedure back to Bulk Insert.

Glossary

! See bang character.

#MISSING See missing data.

access permissions A set of operations that a user can perform on a resource.

accessor Input and output data specifications for data-mining algorithms.

account blocking The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations Accounts which have their values set to zero in the consolidated file during consolidation.

account type A property that determines how an account's value flows over time and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

active service A service whose Run Type is set to Start rather than to Hold.

active-active high availability system A system in which all the available members can service requests, and no member is idle. An active-active system generally provides more scalability options than an active-passive system. Contrast with active-passive high availability system.

active-passive high availability system A system with active members, which are always servicing requests, and passive members that are activated only when an active member fails. Contrast with active-active high availability system.

activity-level authorization Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report An online analytical query that an end user creates dynamically.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

adaptive states Interactive Reporting Web Client level of permission.

adjustment See journal entry.

Advanced Relational Access The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data in the Essbase database.

agent An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

aggregate cell A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

aggregate function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit A limit placed on an aggregated request line item or aggregated metatopic item.

aggregate storage database The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

aggregate view A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.

aggregation The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

aggregation script In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

alias table A table that contains alternate names for members.

alternate hierarchy A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

ancestor A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

appender A Log4j term for destination.

application 1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. 2) A related set of dimensions and dimension members that are used to meet a specific set of analytical requirements, reporting requirements, or both.

application administrator A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

application currency The default reporting currency for the application.

Application Migration Utility A command-line utility for migrating applications and artifacts.

application server cluster A loosely joined group of application servers running simultaneously, working together for reliability and scalability, and appearing to users as one application server instance. See also vertical application cluster and horizontal application cluster.

area A predefined set of members and values that makes up a partition.

arithmetic data load A data load that performs operations on values in the database, such as adding 10 to each value.

artifact An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

assemblies Installation files for EPM System products or components.

asset account An account type that stores values that represent a company's assets.

assignment The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

attribute A characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

attribute association A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

Attribute Calculations dimension A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

attribute dimension A type of dimension that enables analysis based on the attributes or qualities of dimension members.

attribute reporting A reporting process based on the attributes of the base dimension members. See also base dimension.

attribute type A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.

authentication Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

authentication service A core service that manages one authentication system.

auto-reversing journal A journal for entering adjustments that you want to reverse in the next period.

automated stage A stage that does not require human intervention; for example, a data load.

axis (1) A straight line that passes through a graphic used for measurement and categorization. (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

backup A duplicate copy of an application instance.

balance account An account type that stores unsigned values that relate to a particular time.

balanced journal A journal in which the total debits equal the total credits.

bang character (!) A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

base currency The currency in which daily business transactions are performed.

base dimension A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

base entity An entity at the bottom of the organization structure that does not own other entities.

batch calculation Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

batch file An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

Batch Loader An FDM component that enables the processing of multiple files.

batch POV A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

batch processing mode A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

block The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.

block storage database The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

Blocked Account An account that you do not want calculated in the consolidated file because you want to enter it manually.

book 1) In Financial Reporting, a container that holds a group of similar documents. Books may specify dimension sections or dimension changes. 2) In Data Relationship Management, a collection of exports that can be run together as a group. Export results can be combined together or output separately.

book POV The dimension members for which a book is run.

bookmark A link to a reporting document or a Web site, displayed on a personal page of a user. The types of bookmarks are My Bookmarks and image bookmarks.

bounding rectangle The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

broadcast message A simple text message sent by an administrator to a user who is logged on to a Planning application. The message details information such as system availability, notification of application refresh, or application backups.

build method A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

business process A set of activities that collectively accomplish a business objective.

business rules Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

cache A buffer in memory that holds data temporarily.

calc script A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

Calculated Accounts Accounts with formulas that you cannot alter. These formulas are fixed to maintain the accounting integrity of the model that you are building. For example, the formula for Net Income, a Calculated Account, is modeled into Strategic Finance and cannot be changed in historical or forecast periods.

calculated member in MaxL DML A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

Calculation Manager A module of Enterprise Performance Management Architecture (EPMA) that Planning and Financial Management users can use to design, validate, and administrate business rules in a graphical environment. c

calculation status A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

calendar User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

cascade The process of creating multiple reports for a subset of member values.

Catalog pane An area that displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

categories Groupings by which data is organized. For example, Month.

cause and effect map A map that depicts how the elements that form your corporate strategy relate and how they work together to meet your organization's strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

CDF See custom-defined function.

CDM See custom-defined macro.

cell (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

cell note A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

CHANGED status Consolidation status that indicates data for an entity has changed.

chart template A template that defines the metrics to display in Workspace charts.

child A member with a parent above it in the database outline.

choice list A list of members that a report designer can specify for each dimension when defining the report's point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

clean block A data block in which the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

cluster An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

cluster interconnect A private link used by a hardware cluster for heartbeat information, to detect node failure.

cluster services Software that manages cluster member operations as a system. With cluster services, you can define a set of resources and services to monitor through a heartbeat mechanism between cluster members and to move these resources and services to a different cluster member as efficiently and transparently as possible.

clustered bar charts Charts in which categories are viewed side-by-side; used only with vertical bar charts.

code page A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. See also encoding.

column In Data Relationship Management, a field of data associated with an import source or the results of a query, compare, validation, or export.

committed access An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

computed item A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

connection file See Interactive Reporting connection file (.oce)

consolidated file (Parent) A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

consolidation The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

consolidation file (*.cns) A graphical interface that enables you to add, delete, or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

consolidation rule The rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer-specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

content Information stored in the repository for any type of file.

content browser A component that enables users to browse and select content to be placed on a Workspace Page.

context variable A variable that is defined for a particular task flow to identify the context of the taskflow instance.

contribution The value added to a parent from a child entity. Each child has a contribution to its parent.

controls groups Groupings used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

conversion rate See exchange rate.

cookie A segment of data placed on your computer by a Web site.

correlated subqueries Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

critical business area (CBA) An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

critical success factor (CSF) A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

crosstab reporting Reporting that categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

cube A block of data that contains three or more dimensions. An Essbase database is a cube.

cube deployment In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

cube schema In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

currency conversion A process that converts currency values in a database from one currency into another. For example, to convert one U. S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied by the dollar (1* 0.923702). After conversion, the European euro amount is .92.

Currency Overrides A feature allowing the selected input method for any input period to be overridden to enable input of that period's value as Default Currency/Items. To override the input method, enter a pound sign (#) before or after the number.

currency partition A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

custom calendar Any calendar created by an administrator.

custom dimension A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

custom property A property of a dimension or dimension member that is created by a user.

custom report A complex report from the Design Report module, composed of any combination of components.

custom-defined function (CDF) Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. See also custom-defined macro.

custom-defined macro (CDM) Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. See also custom-defined function.

cycle through Perform multiple passes through a database while calculating it.

dashboard A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

data cache A buffer in memory that holds uncompressed data blocks.

data cell See cell.

data file cache A buffer in memory that holds compressed data (PAG) files.

data form A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

data function Function that computes aggregate values, including averages, maximums, counts, and other statistics that summarize groupings of data.

data load location In FDM, a reporting unit responsible for submitting source data into the target system. Typically, one FDM data load location exists for each source file loaded to the target system.

data load rules A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

data lock A feature that prevents changes to data according to specified criteria, such as a period or scenario.

data mining The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

data model A representation of a subset of database tables.

data value See cell.

database connection A file that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

date measure In Essbase, a member tagged as Date in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for analysis types that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. See also typed measure.

Default Currency Units The unit scale of data. For example, If you select to define your analysis in thousands and enter 10, this unit is interpreted as 10,000.

dense dimension In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. Contrast with sparse dimension.

dependent entity An entity that is owned by another entity in the organization.

derived text measure In Essbase Studio, a text measure whose values are governed by a predefined rule expressed as a range. For example, a derived text measure, called "Sales Performance Index," based on a measure Sales, could consist of the values "High," "Medium," and "Low." This derived text measure is defined to display "High," "Medium," and "Low" depending on the range in which the corresponding sales values fall. See also text measure.

descendant Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

Design Report An interface in Web Analysis Studio for designing custom reports, from a library of components.

destination 1) In Business Rules, a block of the database where calculated values are stored; 2) In Profitability and Cost Management, the association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

destination currency The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

detail chart A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

dimension A data category used to organize business data for the retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

dimension build The process of adding dimensions and members to an Essbase outline.

dimension build rules Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

dimension tab In the Pivot section, the tab that enables you to pivot data between rows and columns.

dimension table (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

dimension type A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

dimensionality In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality, because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

direct rate A currency rate that you enter in the exchange-rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, in the exchange-rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

dirty block A data block containing cells that have been changed since the last calculation. Upper-level blocks are marked as dirty if their child blocks are dirty (that is, if they have been updated).

display type One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

dog-ear The flipped page corner in the upper-right corner of the chart header area.

domain In data mining, a variable representing a range of navigation within data.

drill-down Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

drill-through The navigation from a value in one data source to corresponding data in another source.

driver In Profitability and Cost Management, an allocation method that describes the mathematical relationship between the sources that use the driver and the destinations to which those sources allocate cost or revenue. For Business Modeling, see also cost driver and activity driver.

duplicate alias name A name that occurs more than once in an alias table and can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

duplicate member name Multiple occurrences of a member name in a database, with each occurrence representing a different member. For example, a database has two members named New York. One member represents New York state and the other member represents New York city.

duplicate member outline A database outline containing duplicate member names.

Dynamic Calc and Store members Members in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

Dynamic Calc members Members in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

dynamic calculation In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member's values are calculated at retrieval time instead of being precalculated during batch calculation.

dynamic hierarchy In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

dynamic member list A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

dynamic reference A pointer in the rules file to header records in a data source.

dynamic report A report containing data that is updated when you run the report.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

dynamic view account An account type indicating that account values are calculated dynamically from the data that is displayed.

Eliminated Account An account that does not appear in the consolidated file.

elimination The process of zeroing out (eliminating) transactions between entities within an organization.

employee A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts, for authorization purposes.

encoding A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. See also code page, locale.

ending period A period enabling you to adjust the date range in a chart. For example, an ending period of "month" produces a chart showing information through the end of the current month.

Enterprise View An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

entity A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

EPM Oracle home A subdirectory of Middleware home containing the files required by EPM System products. The EPM Oracle home location is specified during installation with EPM System Installer.

Equity Beta The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called "beta." For example, if a stock's return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

essbase.cfg An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

EssCell A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

ESSCMD A command-line interface for performing Essbase operations interactively or through batch script files.

ESSLANG The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

ESSMSH See MaxL Shell.

exceptions Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

exchange rate type An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to a specific time.

expense account An account that stores periodic and year-to-date values that decrease net worth if they are positive.

Extensible Markup Language (XML) A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

external authentication Logging on to Oracle EPM System products with user information stored outside the application. The user account is maintained by the EPM System, but password administration and user authentication are performed by an external service, using a corporate directory such as Oracle Internet Directory (OID) or Microsoft Active Directory (MSAD).

externally triggered events Non-time-based events for scheduling job runs.

Extract, Transform, and Load (ETL) Data-source-specific programs for extracting data and migrating it to applications.

extraction command An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less-than (<) character.

fact table The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

failover The ability to switch automatically to a redundant standby database, server, or network if the primary database, server, or network fails or is shut down. A system that is clustered for failover provides high availability and fault tolerance through server redundancy and fault-tolerant hardware, such as shared disks.

Favorites gadget A gadget that contains links to Reporting and Analysis documents and URLs. See also gadget.

file delimiter A character, such as a comma or tab, that separates fields in a data source.

filter A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

flow account An unsigned account that stores periodic and year-to-date values.

footer Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

format string 1) In Essbase, a method for transforming the way cell values are displayed. 2) In Data Relationship Management, a parameter of a Format or Formatted Date derived property that indicates the format in which a property value should be returned.

formula In Data Relationship Management, business logic used by a derived property to dynamically calculate a property value.

frame An area on the desktop. Two main areas: the navigation and workspace frames.

free-form grid An object for presenting, entering, and integrating data from different sources for dynamic calculations.

free-form reporting Creating reports by entering dimension members or report script commands in worksheets.

function In Data Relationship Management, a syntactic element of a derived property formula that accepts parameters and returns dynamic values.

gadget A simple, specialized, lightweight application that provides easy viewing of EPM content and enables access to core Reporting and Analysis functionality.

geneology data Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

generation A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

generic jobs Non-SQR Production Reporting or non-Interactive Reporting jobs.

global report command A command in a running report script that is effective until it is replaced by another global command or the file ends.

grid POV A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

group A container for assigning similar access permissions to multiple users.

GUI Graphical user interface

hardware cluster a collection of computers that provides a single view of network services (for example, an IP address) or application services (such as databases and Web servers) to clients of these services. Each node in a hardware cluster is a standalone server that runs its own processes. These processes can communicate with one another to form what looks like a single system that cooperatively provides applications, system resources, and data to users.

high availability A system attribute that enables an application to continue to provide services in the presence of failures. This is achieved through removal of single points of failure, with fault-tolerant hardware, as well as server clusters; if one server fails, processing requests are routed to another server.

Historical Average An average for an account over a number of historical periods.

holding company An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

horizontal application server cluster A cluster with application server instances on different machines.

host A server on which applications and services are installed.

host properties Properties pertaining to a host, or if the host has multiple Oracle EPM homes, to an Oracle EPM home.

Hybrid Analysis An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

hyperlink A link to a file, a Web page, or an intranet HTML page.

Hypertext Markup Language (HTML) A programming language specifying how Web browsers display data.

identity A unique identification for a user or group in external authentication.

image bookmarks Graphic links to Web pages or repository items.

IMPACTED status A status that indicates changes in child entities consolidating into parent entities.

implied share A member with one or more children but only one that is consolidated, so the parent and child share a value.

import format In FDM, the definition of the structure of the source file that enables the loading of a source data file to an FDM data-load location.

inactive group A group for which an administrator has deactivated system access.

INACTIVE status A status that indicates entities deactivated from consolidation for the current period.

inactive user A user whose account was deactivated by an administrator.

income account An account storing periodic and year-to-date values that, if positive, increase net worth.

index (1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. (2) The index file.

index cache A buffer containing index pages.

index entry A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

index file An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

index page A subdivision in an index file. An index page contains pointers to data blocks.

input data Data loaded from a source rather than calculated.

installation assemblies Product installation files that plug in to EPM System Installer.

integration A process that is run to move data between Oracle's Hyperion applications using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and they enable the data movements to be grouped, ordered, and scheduled.

intelligent calculation A calculation method tracking updated data blocks since the last calculation.

Interactive Reporting connection file (.oce) Files encapsulating database connection information, including the database API (ODBC, SQL*Net, and so on), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection (.oce) files.

intercompany elimination See elimination.

intercompany matching The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization's consolidated totals.

intercompany matching report A report that compares intercompany account balances and indicates whether the accounts are in balance.

interdimensional irrelevance A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the nonintersecting dimensions, the nonintersecting dimensions are not relevant to that dimension.

intersection A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

intrastage assignment An assignment in the financial flow to an object within the same stage.

introspection A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. Contrast with scraping.

Investigation See drill-through.

isolation level An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration A pass of the budget or planning cycle in which the same version of data is revised and promoted.

Java application server cluster An active-active application server cluster of Java Virtual Machines (JVMs).

Java Database Connectivity (JDBC) A client-server communication protocol used by Java-based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output Files or reports produced from running a job.

jobs Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

join A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

journal entry (JE) A set of debit-credit adjustments to account balances for a scenario and period.

JSP Java Server Page.

KeyContacts gadget A gadget that contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team. See also gadget.

latest A spreadsheet keyword used to extract data values from the member defined as the latest time period.

layer (1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in same layer, though they are in the same generation.

layout area An area on a Workspace Page where content can be placed.

legend box A box containing labels that identify the data categories of a dimension.

level A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

level 0 block A data block for combinations of sparse, level 0 members.

level 0 member A member that has no children.

liability account An account type that stores "point in time" balances of a company's liabilities. Examples: accrued expenses, accounts payable, and long-term debt.

lifecycle management The process of migrating an application, a repository, or individual artifacts across product environments.

line item detail The lowest level of detail in an account.

lineage The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. See also traceability.

link (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a taskflow, the point where the activity in one stage ends and another begins.

link condition A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

linked data model Documents that are linked to a master copy in a repository

linked partition A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

linked reporting object (LRO) A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) Contrast with local report object.

load balancer Hardware or software that directs the requests to individual application servers in a cluster and is the only point of entry into the system.

load balancing Distribution of requests across a group of servers, which helps to ensure optimal end user performance.

local currency An input currency type. When an input currency type is not specified, the local currency matches the entity's base currency.

local report object A report object that is not linked to a Financial Reporting report object in Explorer. Contrast with linked reporting object.

local results A data model's query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

locale A computer setting that specifies a location's language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. See also encoding, ESSLANG.

locale header record A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

location alias A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

locked A user-invoked process that prevents users and processes from modifying data.

locked data model A data model that cannot be modified by a user.

LOCKED status A consolidation status indicating that an entity contains data that cannot be modified.

Log Analyzer An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

logic group In FDM, one or more logic accounts generated after a source file is loaded into FDM. Logic accounts are calculated accounts derived from the source data.

logical Web application An aliased reference used to identify the internal host name, port, and context of a Web application. In a clustered or high-availability environment, this is the alias name that establishes a single internal reference for the distributed components. In EPM System, a nonclustered logical Web application defaults to the physical host running the Web application.

LRO See linked reporting object.

managed server An application server process running in its own Java Virtual Machine (JVM).

manual stage A stage that requires human intervention.

Map File A file that stores the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

Map Navigator A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

Marginal Tax Rate The rate used to calculate the after-tax cost of debt; represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state, and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

Market Risk Premium The additional rate of return paid over the risk-free rate to persuade investors to hold "riskier" investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

master data model An independent data model that is referenced as a source by multiple queries. When used, "Locked Data Model" is displayed in the Query section's Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

mathematical operator A symbol that defines how data is calculated in formulas and outlines. Can be any of the standard mathematical or Boolean operators; for example, +, -, *, /, and %.

MaxL The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). See also MaxL DDL, MaxL DML, and MaxL Shell

MaxL DDL The data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML The data manipulation language used in Essbase for data query and extraction.

MaxL Perl Module A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

MaxL Script Editor A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

MaxL Shell An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh; Windows: essmsh.exe).

MDX (multidimensional expression) A language used for querying and calculation in multidimensional-compliant databases.

measures Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. See also fact table.

member A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include members Jan, Feb, and Qtr1.

member list A named system- or user-defined group that references members, functions, or member lists within a dimension.

member load In Essbase Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

member selection report command A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

member-specific report command A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.

merge A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

metadata A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

metadata elements Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

metadata sampling The process of retrieving a sample of members in a dimension in a drill-down operation.

metadata security Security set at the member level to restrict users from accessing certain outline members.

metaoutline In Essbase Integration Services, a template containing the structure and rules for creating an Essbase outline from an OLAP model.

Middleware home A directory that includes the Oracle WebLogic Server home and can also include the EPM Oracle home and other Oracle homes. A Middleware home can reside on a local file system or on a remote shared disk that is accessible through NFS.

migration audit report A report generated from the migration log that provides tracking information for an application migration.

migration definition file (.mdf) A file that contains migration parameters for an application migration, enabling batch script processing.

migration log A log file that captures all application migration actions and messages.

migration snapshot A snapshot of an application migration that is captured in the migration log.

MIME Type An attribute that describes the data format of an item, so that the system knows which application should open the object. A file's MIME (Multipurpose Internet Mail Extension) type is determined by the file extension or HTTP header. Plug-ins tell browsers which MIME types they support and which file extensions correspond to each MIME type.

mining attribute In data mining, a class of values used as a factor in analysis of a set of data.

minireport A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

minischema A graphical representation of a subset of tables from a data source that represents a data modeling context.

missing data (#MISSING) A marker indicating that data in the labeled location does not exist, contains no value, or was never entered or loaded. For example, missing data exists when an account contains data for a previous or future period but not for the current period.

model 1) In data mining, a collection of an algorithm's findings about examined data. A model can be applied against a wider data set to generate useful information about that data. 2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and nondimensional application objects. 3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

multidimensional database A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions. Contrast with relational database.

Multiload An FDM feature that allows the simultaneous loading of multiple periods, categories, and locations.

My Workspace Page Customizable Workspace Pages created by users. They are marked specially so that they can be easily accessed from one single place without having to navigate the repository.

named set In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

native authentication The process of authenticating a user name and password from within the server or application.

nested column headings A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

NO DATA status A consolidation status indicating that this entity contains no data for the specified period and account.

non-dimensional model A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

non-unique member name See duplicate member name.

null value A value that is absent of data. Null values are not equal to zero.

numeric attribute range A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

OLAP model In Essbase Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database. See also online analytical processing (OLAP).

online analytical processing (OLAP) A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

Open Database Connectivity (ODBC) Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

Oracle home A directory containing the installed files required by a specific product, and residing within the directory structure of Middleware home. See also Middleware home.

organization An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin The intersection of two axes.

outline The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

outline synchronization For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L) Profit and loss accounts. P&L refers to a typical grouping of expense and income accounts that comprise a company's income statement.

page A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file An Essbase data file.

page heading A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member A member that determines the page axis.

palette A JASC-compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments The journal entries that are posted to a child in relation to its parent.

parents The entities that contain one or more dependent entities that report directly to them. Because parents are entities associated with at least one node, they have entity, node, and parent information associated with them.

partition area A subcube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

partitioning The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild-card values such as a question mark (?) or an asterisk (*). For example, "Find all instances of apple" returns apple, but "Find all instances of apple*" returns apple, applesauce, applecranberry, and so on.

percent consolidation The portion of a child's values that is consolidated to its parent.

percent control The extent to which an entity is controlled within the context of its group.

percent ownership The extent to which an entity is owned by its parent.

performance indicator An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA) A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission A level of access granted to users and groups for managing data or other users and groups.

persistence The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

personal pages A personal window to repository information. You select what information to display and its layout and colors.

personal recurring time events Reusable time events that are accessible only to the user who created them.

personal variable A named selection statement of complex member selections.

perspective A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pinboard One of the three data object display types. Pinboards are graphics composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

pins Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot Alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner A user who can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View. Planners comprise the majority of users.

planning unit A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account An account in which the system stores any out-of-balance differences between intercompany account pairs during the elimination process.

post stage assignment Assignments in the allocation model that are assigned to locations in a subsequent model stage.

POV (point of view) A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precalculation Calculating the database before user retrieval.

precision Number of decimal places displayed in numbers.

predefined drill paths Paths used to drill to the next level of detail, as defined in the data model.

presentation A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

preserve formulas User-created formulas kept within a worksheet while retrieving data.

primary measure A high-priority measure important to your company and business needs. Displayed in the Contents frame.

Process Monitor Report A list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

product In Shared Services, an application type, such as Planning or Performance Scorecard.

Production Reporting See SQR Production Reporting.

project An instance of Oracle's Hyperion products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting Server instance.

provisioning The process of granting users and groups specific access permissions to resources.

proxy server A server acting as an intermediary between workstation users and the Internet to ensure security.

public job parameters Reusable named job parameters created by administrators and accessible to users with requisite access privileges.

public recurring time events Reusable time events created by administrators and accessible through the access control system.

PVA See periodic value method.

qualified name A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State].[New York] or [Market].[East].[City].[New York].

query governor An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

reciprocal assignment An assignment in the financial flow that also has the source as one of its destinations.

reconfigure URL A URL that is used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

record In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

recurring template A journal template for making identical adjustments in every period.

recurring time event An event specifying a starting point and the frequency for running a job.

redundant data Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

regular journal A feature for entering one-time adjustments for a period. A regular journal can be balanced, balanced by entity, or unbalanced.

Related Accounts Accounts related to the main account and grouped under the same main account number. The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

relational database A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

replace A data load option that clears existing values from all accounts for periods specified in the data load file and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

replicated partition A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

Report Extractor An Essbase component that retrieves report data from the Essbase database when report scripts are run.

report object In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

report script A text file containing Essbase Report Writer commands that generate one or more production reports.

Report Viewer An Essbase component that displays complete reports after report scripts are run.

reporting currency The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

repository Storage location for metadata, formatting, and annotation information for views and queries.

resources Objects or services managed by the system, such as roles, users, groups, files, and jobs.

restore An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

restructure An operation to regenerate or rebuild the database index and, in some cases, data files.

result frequency The algorithm used to create a set of dates to collect and display results.

review level A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

Risk Free Rate The rate of return expected from "safer" investments such as long-term U.S. government securities.

role The means by which access permissions are granted to users and groups for resources.

roll-up See consolidation.

root member The highest member in a dimension branch.

runtime prompt A variable that users enter or select before a business rule is run.

sampling The process of selecting a representative portion of an entity to determine the entity's characteristics. See also metadata sampling.

saved assumptions User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

scaling Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

scenario A dimension for classifying data; for example, Actuals, Budget, Forecast1, or Forecast2.

schema In relational databases, a logical model that represents the data and the relationships between the data.

scope The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. The levels, from highest to lowest: the entire system (Essbase Server), applications on Essbase Server, or databases within Essbase Server applications. See also persistence.

score The level at which targets are achieved, usually expressed as a percentage of the target.

scorecard A business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

scraping An inspection of a data source to derive the most basic metadata elements from it. Contrast with introspection.

secondary measure A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

security agent A Web access management provider (for example, Oracle Access Manager, Oracle Single Sign-On, or CA SiteMinder) that protects corporate Web resources.

security platform A framework enabling Oracle EPM System products to use external authentication and single sign-on.

serial calculation The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

services Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

servlet A piece of compiled code executable by a Web server.

shared disks See shared storage.

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services Registry The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.

shared storage A set of disks containing data that must be available to all nodes of a failover cluster; also called shared disks.

Shared Workspace Pages Workspace Pages shared across an organization that are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.

silent response files Files providing data that an installation administrator would otherwise be required to provide. Response files enable EPM System Installer or EPM System Configurator to run without user intervention or input.

single point of failure Any component in a system that, if it fails, prevents users from accessing the normal functionality.

single sign-on (SSO) The ability to log on once and then access multiple applications without being prompted again for authentication.

smart tags Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In Oracle EPM System products, smart tags can also be used to import Reporting and Analysis content and to access Financial Management and Essbase functions.

SmartCut A link to a repository item, in URL form.

snapshot Read-only data from a specific time.

source currency The currency from which values originate and are converted through exchange rates to the destination currency.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. Contrast with dense dimension. For example, not all customers have data for all products.

SPF files Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

Spotlighter A tool that enables color coding based on selected conditions.

SQL spreadsheet A data object that displays the result set of a SQL query.

SQR Production Reporting A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

stage 1) A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated. 2) For Profitability, logical divisions within the model that represent the steps in the allocation process within your organization.

stage action For automated stages, the invoked action that executes the stage.

staging area A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMS.

staging table A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMSs.

standard dimension A dimension that is not an attribute dimension.

standard journal template A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, and then use the template as the basis for many regular journals.

Status bar The bar at the bottom of the screen that displays helpful information about commands, accounts, and the current status of your data file.

stored hierarchy In aggregate storage databases outlines only, a hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions; for example, they cannot contain formulas.

strategic objective (SO) A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

Strategy map Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

structure view Displays a topic as a simple list of component data items.

Structured Query Language A language used to process instructions to relational databases.

Subaccount Numbering A system for numbering subaccounts using nonsequential whole numbers.

subscribe Flags an item or folder to receive automatic notification whenever the item or folder is updated.

Summary chart In the Investigates Section, a chart that rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

supervisor A user with full access to all applications, databases, related files, and security mechanisms for a server.

supporting detail Calculations and assumptions from which the values of cells are derived.

suppress rows A setting that excludes rows containing missing values and underscores characters from spreadsheet reports.

symmetric multiprocessing (SMP) A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users simultaneously connect to an single instance.

sync Synchronization of Shared Services and application models.

synchronized The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

system extract A feature that transfers data from application metadata into an ASCII file.

tabs Navigable views of accounts and reports in Strategic Finance.

target Expected results of a measure for a specified period of time (day, quarter, and so on).

task list A detailed status list of tasks for a particular user.

taskflow The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

taskflow definition Business processes in the taskflow management system that consist of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

taskflow instance A single instance of a taskflow including its state and associated data.

taskflow management system A system that defines, creates, and manages the execution of a taskflow, including definitions, user or application interactions, and application executables.

taskflow participant The resource that performs the task associated with the taskflow stage instance for both manual and automated stages.

Taxes - Initial Balances Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance, and Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.

TCP/IP See Transmission Control Protocol/Internet Protocol.

text measure In Essbase, a member tagged as Text in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure Satisfaction Index may have the values Low, Medium, and High. See also typed measure, text list, derived text measure.

time dimension The time period that the data represents, such as fiscal or calendar periods.

time events Triggers for job execution.

time scale A scale that displays metrics by a specific time span, such as monthly or quarterly.

time series reporting A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

Timeline Viewer An FDM feature that enables users to view dates and times of completed process flow steps for specific locations.

Title bar A bar that displays the Strategic Finance name, the file name, and the scenario name Version box.

toast message A message that fades in the lower-right corner of the screen.

token An encrypted identification of one valid user or group on an external authentication system.

top and side labels Column and row headings on the top and sides of a Pivot report.

top-level member A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. If a hierarchical relationship exists, the top-level member name is generally the same as the dimension name.

trace allocations A Profitability feature that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.

trace level The level of detail captured in a log file.

traceability The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies to its dimension elements, date/time elements, measures, and, ultimately, to its physical source elements. See also lineage.

traffic lighting Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

transformation 1) A process that transforms artifacts so that they function properly in the destination environment after application migration. 2) In data mining, the modification of data (bidirectionally) flowing between the cells in the cube and the algorithm.

translation See currency conversion.

Transmission Control Protocol/Internet Protocol (TCP/IP) A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

transparent login A process that logs in authenticated users without launching the login screen.

transparent partition A shared partition that enables users to access and change data in a remote database as though it is part of a local database.

triangulation A means of converting balances from one currency to another through a third common currency. In Europe, this currency is the euro for member countries. For example, to convert from the French franc to the Italian lira, the common currency is defined as the European euro. Therefore, to convert balances from the French franc to the Italian lira, balances are converted from the French franc to the European euro and from the European euro to Italian lira.

triggers An Essbase feature whereby data is monitored according to user-specified criteria that, when met, cause Essbase to alert the user or system administrator.

trusted user Authenticated user.

tuple MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual]).

two-pass An Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the outline.

unary operator A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

Unicode-mode application An Essbase application wherein character text is encoded in UTF-8, enabling users with computers set up for different languages to share application data.

unique member name A nonshared member name that exists only once in a database outline.

unique member outline A database outline that is not enabled for duplicate member names.

upgrade The process of replacing a software release with a newer release. The term upgrade does not apply to installing a maintenance release. See also maintenance release, migration.

upper-level block A type of data block wherein at least one of the sparse members is a parent-level member.

user directory A centralized location for user and group information, also known as a repository or provider. Popular user directories include Oracle Internet Directory (OID), Microsoft Active Directory (MSAD), and Sun Java System Directory Server.

user variable A variable that dynamically renders data forms based on a user's member selection, displaying only the specified entity. For example, a user variable named Department displays specific departments and employees.

user-defined attribute (UDA) An attribute, associated with members of an outline to describe a characteristic of the members, that can be used to return lists of members that have the specified associated UDA.

user-defined member list A named, static set of members within a dimension defined by the user.

validation The process of checking a business rule, report script, or partition definition against the outline to ensure that the object being checked is valid.

validation rules Rules used in FDM to enforce data integrity. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

value dimension A dimension that is used to define input value, translated value, and consolidation detail.

variance The difference between two values (for example, between planned and actual values).

version A possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

vertical application server cluster A cluster with multiple application server instances on the same machine.

view A year-to-date or periodic display of data.

visual cue A formatted style, such as a font or a color, that highlights specific data value types. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read-only data cells; read-and-write data cells; or linked objects.

WebLogic Server home A subdirectory of Middleware home containing installed files required by a WebLogic Server instance. WebLogic Server home is a peer of Oracle homes.

weight A value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer's scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

wild card Character that represents any single character (?) or group of characters (*) in a search string.

WITH section In MaxL DML, an optional section of the query used for creating reusable logic to define sets or members. Sets or custom members can be defined once in the WITH section and then referenced multiple times during a query.

workbook An entire spreadsheet file with many worksheets.

workflow The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures that all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

Workspace Page A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

write-back The ability for a retrieval client, such as a spreadsheet, to update a database value.

ws.conf A configuration file for Windows platforms.

wsconf_platform A configuration file for UNIX platforms.

XML See Extensible Markup Language.

XOLAP An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

Y axis scale A range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

Zero Administration A software tool that identifies version number of the most up-to-date plug-in on the server.

ZoomChart A tool for viewing detailed information by enlarging a chart. A ZoomChart enables you to see detailed numeric information on the metric that is displayed in the chart.

Index

Symbols

Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X