Oracle® Hyperion Disclosure Management
Oracle® Hyperion Disclosure Management for Oracle Hyperion Financial Close Suite

Administrator's Guide
Release 11.1.2.3.00
# Contents

**Documentation Accessibility** ........................................................... 5

**Chapter 1. Deploying and Configuring Disclosure Management** ................................. 7
  Prerequisites .......................................................................................... 7
  Disclosure Management Components ....................................................... 7
    Client Components ............................................................................... 8
    Disclosure Management Integration With Financial Reporting, Smart View (APS) and Data Source Access ........................................................................ 9
  Middle-Tier Services ............................................................................. 10
  Databases .......................................................................................... 10
  Disclosure Management XBRL Taxonomy Designer ................................. 11
  Server Configuration Options ................................................................ 11
    Registering XBRL Taxonomies .............................................................. 11
    Taxonomy Caching ............................................................................. 21
    iXBRL Instance Generation for Large Number of Mappings ..................... 29
  Client Configuration Options .................................................................. 30
    Setting up Server Information .............................................................. 30
    Preview Options ............................................................................... 31
    Default Formatting ............................................................................. 32
    Mappings .......................................................................................... 33
   Launching the Viewer ........................................................................... 33

**Appendix A. Disclosure Management Server Logs** ................................................. 35
Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Deploying and Configuring Disclosure Management

In This Chapter

- Prerequisites ................................................................. 7
- Disclosure Management Components .................................. 7
- Server Configuration Options ............................................ 11
- Client Configuration Options .............................................. 30
- Mappings ....................................................................... 33
- Launching the Viewer ...................................................... 33

Prerequisites

The following components must be installed to use Oracle Hyperion Disclosure Management:

- Installed and configured Disclosure Management Release 11.1.2 or later
- Installed and configured Oracle Hyperion Smart View for Office and Disclosure Management Microsoft Office client components (Office 2007, or Office 2010)
- Microsoft Internet Explorer 7 or later

Note: To use the Disclosure Management client, when you install Microsoft Office, select .NET programmability support for Microsoft Word and Excel.

Note: When an upgrade to Disclosure Management is made, the xbrlData.properties file is overridden with a newer version. Additionally, a backup copy of the previous version is created automatically with a .bak extension in the same directory as the new version. However, any custom settings from the previous version file are not merged or retained by the new version. If you have customized the previous version (such as the changing the proxy settings), reapply changes to the new version manually.

Disclosure Management Components

This section includes information on the following Disclosure Management client and server components:
Client Components

This section includes information on the following client components of Disclosure Management:

- “Disclosure Management XBRL Add-in for Microsoft Office” on page 8
- “Disclosure Management Mapping Tool ” on page 8
- “Financial Reporting HTML Preview” on page 9
- “Smart View (APS) and Data Source Access” on page 10

Disclosure Management XBRL Add-in for Microsoft Office

The Disclosure Management add-in for Microsoft Office provides an interface to all of its features within the Microsoft Office framework (specifically, Microsoft Excel and Word). The Disclosure Management XBRL add-in uses the Smart View extensions API for integrating with Microsoft Office and accessing metadata from EPM data sources.

Disclosure Management Mapping Tool

The Disclosure Management Mapping Tool maps information between the items of a taxonomy and to the selected data value or values in an Office document or Oracle Hyperion Financial Reporting. The Disclosure Management Mapping Tool is the central piece for most client-side user interactions. The Disclosure Management Mapping Tool is an add-in component for Office applications (Microsoft Word or Excel), and is bundled with Financial Reporting. Users can easily select taxonomies, manage mappings, add variables, and validate XBRL instance documents using the Disclosure Management Mapping Tool.

The Disclosure Management Mapping Tool:

- Renders XBRL taxonomies and provides mapping functionality
- Provides the ability to create, edit, and delete variables in doclets and Master Documents
- Exposes most client-side user interactions
- Is used by the Office Add-in and Financial Reporting
- Provides a uniform user interface across all client applications
- Includes taxonomy search capability
- Enables reviewing and validation of all mappings

The add-in for Microsoft Word and Excel provides the following mapping features:

- XBRL Taxonomy Concepts
Additionally, you can generate a document in iXBRL format, which enables you to view submissions in a human-readable format while retaining the machine-readable formats—within the same document. An iXBRL document is an HTML document embedded with special XBRL tags, allowing the human-readable part in HTML and the machine-readable bits in XBRL.

**Report Manager**

Disclosure Management Report Manager report writers can use the Report Manager interface to manage and produce reports using Master Documents and doclets.

A Master Document acts as a container file for subdocuments called “doclets.” Using Master Documents Disclosure Management enables you to leverage your last report as the starting point for your next report by using the Master Document feature. With a Master Document in Microsoft Word, an administrator can easily copy the last report, embed a Microsoft Word and Excel sections into the document, rename it, and update the report view to the current Period or Year. Any registered Microsoft Word document may be used as a Master Document. A Disclosure Management document becomes a Master Document when at least one doclet is inserted into it. A doclet is a separate Microsoft Word or Excel file that includes selected data that you want to include in the Master Document.

**Disclosure Management Integration With Financial Reporting, Smart View (APS) and Data Source Access**

This section includes information on:

- “Financial Reporting HTML Preview” on page 9
- “Financial Reporting Web Application Service” on page 10
- “Smart View (APS) and Data Source Access” on page 10

**Financial Reporting HTML Preview**

The Disclosure Management Mapping Tool is integrated in the Financial Reporting HTML Preview. You can map XBRL concepts to report data in a grid, including data from data sources (such as Oracle Hyperion Financial Management, Oracle Hyperion Planning, and Oracle Essbase) as well as formula and text cells. Data with XBRL maps from a Financial Reporting grid can be reused and imported into Microsoft Word or Excel documents through Smart View. After the data is imported into an Office document, the Disclosure Management add-in determines and consumes all relevant XBRL maps from the Financial Reporting function grid.
Financial Reporting Web Application Service

The Financial Reporting Web application runs, stores, and schedules reports and batches. For more information, see the Oracle Hyperion Financial Reporting, Administrator's Guide.

Smart View (APS) and Data Source Access

Data from Oracle Hyperion data sources such as Financial Management, Planning, and Essbase can be imported into a Microsoft Word or Excel document through the Smart View Analytic Provider Services (APS). After the data is in the Office document, the data source members can be associated with XBRL concepts through the Disclosure Management Mapping Tool. When this association occurs, the XBRL concepts are recognized from the data source member when it is part of a Financial Reporting grid or another Office document. Therefore a “data source XBRL map” can be associated once and reused in Office documents and Financial Reporting grids.

Middle-Tier Services

The Disclosure Management Web application interacts with several middle-tier components which can exist on a distributed environment. This section highlights the most important middle-tier components:

- “Disclosure Management Web Application Service” on page 10
- “Financial Reporting Web Application Service” on page 10

Disclosure Management Web Application Service

A J2EE-based Web application provides services to most of the components in Disclosure Management. This service interacts with client-layer components, other middle-tier services, and data storage components.

Databases

Disclosure Management database sources includes the Mapping Reporting and Oracle Hyperion data sources:

- “Mapping Repository” on page 10
- “Oracle Hyperion Data Sources” on page 11

Mapping Repository

The Mapping Repository is a server-side application responsible for storing and retrieving the XBRL taxonomy mappings created by the Mapping Tool. When a user creates, modifies, or deletes a mapping, the mappings are centrally stored in the Mapping Repository. Users cannot load and view mappings directly from the Mapping Repository.
Oracle Hyperion Data Sources

Disclosure Management supports Enterprise Performance Management (EPM) data sources such as Oracle Hyperion Planning, Oracle Hyperion Financial Management, and Oracle Essbase. Data sources can be reused and imported into Microsoft Word or Excel documents through Oracle Hyperion Smart View for Office.

Data from non-EPM sources, such as Enterprise Resource Planning (ERP) systems, can also be used when they are imported into Microsoft Word or Excel.

Disclosure Management XBRL Taxonomy Designer

Disclosure Management XBRL Taxonomy Designer is a desktop application designed for building, extending, and maintaining XBRL taxonomies. The Disclosure Management XBRL Taxonomy Designer user interface offers multiple views, including concept relationships, calculations, languages, and properties that stay synchronized as the taxonomy is browsed. Disclosure Management XBRL Taxonomy Designer offers support for the most current XBRL 2.1 specification, including dimensions and tuples.

The Disclosure Management XBRL Taxonomy Designer is installed with the Disclosure Management application. For more information, see the Disclosure Management XBRL Taxonomy Designer online help.

Server Configuration Options

This section includes information on the Disclosure Management server configuration options:

- “Registering XBRL Taxonomies” on page 11
- “Downloading the Taxonomies” on page 12

Registering XBRL Taxonomies

XBRL Taxonomies must be registered in the Disclosure Management Web application server. Once registered, the taxonomy can be accessed in the Disclosure Management Mapping Tool for mapping and generating XBRL instance documents. Registered taxonomies must be valid according to the XBRL specifications and include or correctly reference any taxonomy dependencies. The taxonomies that are registered are available to all Disclosure Management users in the client components. After Disclosure Management is installed, administrators must download the XBRL taxonomies manually and configure the mappingtool.properties file located in the DISCMAN_INSTANCE/config folder.

Note: The administrator is responsible for installing and registering the taxonomies that the Disclosure Management Mapping Tool uses.
**Downloading the Taxonomies**

Official XBRL taxonomies are usually downloaded from official taxonomy sites, such as: [www.xbrl.org](http://www.xbrl.org)

Required taxonomies are generally available from your regulator. You should always refer to the regulator mandates and websites for instructions on which taxonomies to download:

- [http://www.xbrl.us](http://www.xbrl.us)—US taxonomies
- [http://www.iasb.org/XBRL/IFRS+Taxonomy/IFRS+Taxonomy.htm](http://www.iasb.org/XBRL/IFRS+Taxonomy/IFRS+Taxonomy.htm)—Current IFRS taxonomy, such as IFRS

**Extracting the Taxonomies**

Taxonomies must be stored and registered at the computer hosting the web application. Typically, taxonomies are downloaded in a compressed file format. When extracting a taxonomy, maintain the folder structure of the taxonomy files.

- To extract a taxonomy:
  1. **If the taxonomy does not already exist in the Disclosure Management folder, locate the DISCMAN_INSTANCE/XbrlFiles folder.**
  2. **Uncompress the taxonomy files to the DISCMAN_INSTANCE/XbrlFiles folder.**
     
     Ensure that the folder structure is maintained.
  3. **Ensure that the Disclosure Management Web application has read access to the XbrlFiles folder and its files.**

**Extending Taxonomies**

To extend a taxonomy, you can create or edit the taxonomy in the Disclosure Management XBRL Taxonomy Designer, which is a standalone client application. Disclosure Management XBRL Taxonomy Designer is a full-featured taxonomy and instance creator. Disclosure Management XBRL Taxonomy Designer includes a suite of robust creation, editing, and validation for managing complex taxonomies, both with single and in-reference taxonomy usage patterns and validation capabilities. With Taxonomy Designer, you can:

- Create or rename concepts to closely match the nomenclature in your financial states
- Change the data type, balance, and period type of concepts
- Change the relationship of concepts
- Change the file path where taxonomies are saved

After a taxonomy is modified, it can be registered in Disclosure Management as described in "Registering XBRL Taxonomies" on page 11.
Registering and Viewing the XBRL Taxonomy Structure

When the Disclosure Management Web application is installed, a properties file named “mappingtool.properties” is placed in the DISCMAN_INSTANCE/config folder. The DISCMAN_INSTANCE pertains to the computer where the Disclosure Management Web application is installed.

After the taxonomies files are unzipped on the Disclosure Management server in the XbrlFile folder, they are registered and recognized by Disclosure Management and listed in the Disclosure Management Mapping Tool. (There is no DISCMAN_INSTANCE folder in a client, for example.)

**Note:** You can view and edit the mappingtool.properties file using any text editor.

**Note:** Non-ASCII characters are not supported by mappingtool.properties. To use non-ASCII characters, use a unicode format (for example, \u00D2).

The mappingtool.properties file contains the following properties:

- taxonomy_.#.prefix
- taxonomy_.#.entryPoint_
- taxonomy_.#.label_
- # taxonomy_.#.addlinkbases
- taxonomy_.#.extLinkLabel_
- taxonomy_.#.formatted_
- # taxonomy_.#.unformatted
- taxonomy_.#.overridable_
- taxonomy_.#.unoverridable_
- taxonomy_.#.package_
- taxonomy_.#.schemaRef_

**Note:** # represents a placeholder for a numeric value. The above properties are case-sensitive.

**Taxonomy Properties Example**

This example shows how a mappingtool.properties file might be specified. Four taxonomies are registered:

- US GAAP 2009
- US GAAP 2008
- Oracle Extension (that is, a custom taxonomy) to the US GAAP 2008 taxonomy
- IFRS 2009
The US GAAP 2009 taxonomy defines five entry points, but based on the schema below, the Disclosure Management Mapping Tool shows only two (“Banking and Saving” and “Commercial and Industrial”). The US GAAP 2008 taxonomy shows three entry points (“Banking and Saving”, “Commercial and Industrial”, and “Real Estate”):

taxonomy_1.prefix=us-gaap-2009
taxonomy_1.entryPoint_1=XBRLUSGAAP/2009-01-31/ind/basi/us-gaap-basi-stm-dis-all-2009-01-31.xsd
taxonomy_1.label_1=Banking and Savings
taxonomy_1.label_2=Commercial and Industrial

taxonomy_2.prefix=us-gaap-2008
taxonomy_2.entryPoint_1=XBRLUSGAAP/2008-03-31/ind/ci/us-gaap-ci-stm-dis-all-2008-03-31.xsd
taxonomy_2.label_1=Commercial and Industrial
taxonomy_2.entryPoint_2=XBRLUSGAAP/2008-03-31/ind/basi/us-gaap-basi-stm-dis-all-2008-03-31.xsd
taxonomy_2.label_2=Banking and Savings
taxonomy_2.entryPoint_3=XBRLUSGAAP/2008-03-31/ind/re/us-gaap-re-stm-dis-all-2008-03-31.xsd
taxonomy_2.label_3=Real Estate

taxonomy_3.prefix=orcl-2008
taxonomy_3.entryPoint_1=oracle/2008-04-01/oracle-ext_2008-04-01.xsd
taxonomy_3.label_1=Oracle GAAP Extension

taxonomy_4.prefix=ifrs-2009
taxonomy_4.entryPoint_1=ifrs-2009-04-01/ifrs-cor_2009-04-01.xsd
taxonomy_4.label_1=IFRS Core

Viewing the mappingtool.properties file

![Mapping Tool Properties File](image)

The mappingtool.properties file includes these properties:
The prefix or “short name” is used in the Disclosure Management Mapping Tool user interface and instance documents. The prefix value must have these characteristics:

- Has a unique value—two or more taxonomies should not use the same prefix
- Starts with a letter or underscore character
- Contains no spaces
- Is short because it is used repeatedly within instance documents

### entryPoint_

The taxonomy entry point is the path to a taxonomy’s .xsd file, relative to the DISCMAN_INSTANCE/XbrlFiles folder.

Taxonomies can have multiple entry points. Administrators control which entry points are registered and shown by the Disclosure Management Mapping Tool. For example, the US GAAP taxonomy has five entry points, but an administrator can choose to register only three. The website from which the taxonomy is downloaded usually contains details about its entry points.

The path to the .xsd file should use the “/” character as a path separator. Alternately a double “\” can be used, but not a single “\”, for example:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-gaap/ci/us-gaap-ci-all.xsd</td>
<td>Valid</td>
</tr>
<tr>
<td>us-gaap\ci\us-gaap-ci-all.xsd</td>
<td>Valid</td>
</tr>
<tr>
<td>us-gaap\ci\us-gaap-ci-all.xsd</td>
<td>Invalid</td>
</tr>
</tbody>
</table>

### label_

`label_#` is the user-readable label associated with the entry point.

The label is shown in the Disclosure Management Mapping Tool user interface.
Each entry point value should have a corresponding label entry.

**extLinkLabel_#**

Administrators can indicate the extended link label value to be shown for extended links in the taxonomy. Two values are available: “title” or “definition”. The extended link value is defined in the `extLinkLabel_1=[definition][title]` property of `mappingtool.properties`. When one value is not available, the other is used. For example, when the value is set to “definition” and the taxonomy has only title labels, titles are used. This property is optional. If the property is not provided, the default value is “title”.

**schemaRef_#**

The schema name (schemaRef property) in the instance document is determined by the `schemaRef_#` value specified in `mappingtool.properties`. Because this information is not supplied by the taxonomy itself, the administrator must provide the schemaRef property. The pattern for this property is: `schemaRef_#=[SomeTaxonomyURI]`

Note the following when specifying the schemaRef property:

- The schemaRef property is normally a URI to the entry point of the taxonomy referenced by an instance document. The SEC requires that the schemaRef property point only to the taxonomy file name (see “orcl-20100831.xsd”). However, the UK--IFRS requires that a full URI, (for example, `http://www.xbrl.org/uk/ifrs/core/2009-09-01/uk-ifrs-full-2009-09-01.xsd`) be used.

- This property is optional. If it is not provided, the schema value from the `correspondingEntryPoint_#` property is used.

Disclosure Management supports multiple schema reference (SchemaRef) declarations in an instance document. For example, the following schema reference declarations can be specified in the `mappingtool.properties` file using the `schemaRef_#` parameter and spaces as separators: Note that the three `schemaRef_#` values are separated by spaces.

```
```

**package_#**

The `package_#` property determines whether the taxonomy files are included when users selects the “Generate XBRL” option from Microsoft Excel or Word. When this property is enabled, Disclosure Management produces the XBRL instance document on the Disclosure Management server and includes the additional documents within the compressed file (with a “dmr” extension). The dmr file is then serialized to the client machine and saved to the file system (as indicated by the user). When the package property is “false” Disclosure Management does...
not include the dependent taxonomy files within the dmr file. Disclosure Management includes only the XBRL instance document and a few other proprietary files.

The `package_#` property accepts a Boolean flag value:

- A “true” Boolean value indicates that the taxonomy files is packaged.
- A “false” Boolean value indicates that the taxonomy file is not packaged.

This property is optional; if it is not provided, “true” is the default.

**formatted**

The `formatted` property is used to automatically apply a Rich Text Format to specified data types.

Each data type must be space-separated and represented the following way: "xsd_target_namespace#dataType". The pattern for the properties is:

```
taxonomy_#.formatted=[Space separated data types]
```

For example, to indicate that concepts which are of the `textBlockItemType` data type, always use Rich Text Formatting for "taxonomy_1", add the following entry:

```
taxonomy_1.formatted=http://xbrl.us/us-types/2009-01-31#textBlockItemType
```

This property is optional. If it is not provided, plain text formatting is always used.

**unformatted**

The `unformatted` property is useful to automatically apply plain text format to specified data types.

Each data type must be space separated, and represented in the following way: "xsd_target_namespace#dataType". The pattern for the properties is:

```
taxonomy_#.unformatted=[Space separated data types]
```

For example, to indicate that concepts which are of the `textBlockItemType` data type always use plain text formatting for "taxonomy_1", add the following entry:

```
taxonomy_1.unformatted=http://xbrl.us/us-types/2009-01-31#textBlockItemType
```

This property is optional. If it is not provided, plain text formatting is used.

**overridable/unoverridable**

You can enable or disable `override` functionality for a particular concept type in the `mappingtool.properties` file. The “overridable” and “unoverridable” properties govern whether it is possible to override all facts based on concepts of a specified type and its derived types on the Review tab.

Each item of the list in the `mappingtool.properties` file must be in the form of: `<target-name-space>#$<dataTypeName>`
You need not enumerate all data types for which the override setting is enabled. Because data types are usually organized hierarchically, specify the override setting for the common parent type. For example, you could enable the override setting for the decimalItem Type and its children by entering: `taxonomy_1.overridable_1=http://www.xbrl.org/2003/instance#decimalItemType`.

In this case, all facts based on concepts of all types inherited from `decimalItemType` (for example numeric, monetary, or `volumeItem Type`) are overridable.

You can also set global override settings in addition to taxonomy specific settings, for example: `global.overridable=http://www.xbrl.org/2003/instance#decimalItemType http://www.xbrl.org/2003/instance#booleanItemType http://www.xbrl.org/2003/instance#dateItemType`.

The unoverridable setting enables you to disable the ability to override types in the hierarchy. For example if you want to disable the ability to override formatted items in the US GAAP extension, you would specify: `taxonomy_1.unoverridable_1= http://xbrl.us/us-types/2009-01-31#textBlockItemType`.

**addlinkbases**

Use the `addlinkbases` property to add documentation for concepts in extension taxonomies. The documentation refers to the actual meaning of the concept being created. The `addlinkbases` property is set by specifying a space-delimited list of one or more linkbases, which you attach to a registered taxonomy. While the linkbases listed do not have to be for documentation only, it is the only resource supported at this time.

The most common documentation that SEC filers might attach to their extension taxonomies:

<table>
<thead>
<tr>
<th>Documentation Linkbase</th>
<th>File Location</th>
</tr>
</thead>
</table>
The linkbases in this are not exclusive. Several additional documentation linkbases are available for the US GAAP taxonomy. The `addlinkbases` property is case-sensitive (the file name should only be in lowercase characters). Additionally, the Disclosure Management service is normally restarted when the `mappingtool.properties` file is modified.

In the following example, the taxonomy "orcl-20101130" is a 2009 US GAAP extension taxonomy. Two documentation linkbases are attached, including one for the US GAAP concepts and one for the DEI (Document & Entity Information) concepts. Adding the two linkbases, shows the documentation (where available) when a US GAAP or a DEI concept is selected in the mapping tool. Note that the two linkbases are space separated:

```
taxonomy_1.prefix=Oracle
taxonomy_1.label_1=Oracle 10-Q 20101130
taxonomy_1.entryPoint_1=orcl-20101130/abc-20101130.xsd
```

### Configuring the Unit Type List

The units or currency list that is displayed when creating a unit in the Disclosure Management Mapping Tool is derived and configured in the `mappingtool.properties` file. Units types are available in the Measure field. The Unit type code corresponds to the ISO (International Organization for Standardization) 4217 standard. In the `mappingtool.properties` file, the current unit values:

- `unit_type1=share`
- `unit_type2=pure`
- `unit_type3=iso4217:AED`
- `unit_type4=iso4217:AUD`
- `unit_type5=iso4217:CAD`
- `unit_type6=iso4217:CAF`
- `unit_type7=iso4217:SGD`
- `unit_type8=iso4217:USD`
- `unit_type9=iso4217:DEM`
- `unit_type11=iso4217:NZD`
- `unit_type12=iso4217:PLN`
- `unit_type13=iso4217:PLN`
When you create a unit type, the default unit type code is: `unit_type8=iso4217:USD`.

To add or change a unit type:

1. Navigate to `mappingtool.properties` file in the `DISCMAN_INSTANCE/config` folder.
2. Using any text editor, open the `mappingtool.properties` properties file.
3. Scroll down to `# the unit type sections`.
4. Add the new unit using the format: `unit_type[number]=iso4217:[currency code]`.
   
The currency code consists of the two-character country code and a character that represents the currency unit.
5. Save the `mappingtool.properties` file.
   
Unit types are validated in Review mode.

**Updates to the XBRL Taxonomy**

When the administrator shuts down and restarts services for Disclosure Management, the web application examines the `mappingtool.properties` file and detects the following changes:

- A new taxonomy was added (that is, registered).
- The taxonomy label or prefix is modified.
- The content of an existing taxonomy is modified.
- A previously registered taxonomy is removed.
Viewing Taxonomy Structure

You can view the structure of registered taxonomies in the Disclosure Management Mapping Tool in the Select Taxonomy pane. The taxonomies are shown in alphabetical order (case sensitive) in the Select Taxonomy pane.

Taxonomy Caching

Disclosure Management provides a taxonomy caching system that manages the lifecycle of a taxonomy that is loaded into memory. The taxonomy caching system can be tuned using various properties.

Overview

The Disclosure Management Web application manages the loading and unloading of the XBRL taxonomies that are registered in the Disclosure Management system. Because XBRL taxonomies can be large, they tend to take up a lot of memory resources available to the Java process. Additionally, every time a taxonomy is loaded (into memory), performance is affected. Disclosure Management has a taxonomy caching system that keeps loaded taxonomies in memory so subsequent requests for taxonomy resources can be derived from the cache rather than reloading the taxonomy; the taxonomy system works as follows:

- At startup, the taxonomy broker reads the list of registered taxonomies from the mappingtool.properties file.
- A taxonomy cache object is created for each registered taxonomy. This does not mean that the taxonomy is loaded at this time - taxonomy loading is done on demand.
- When a user requests a particular taxonomy, the taxonomy broker checks the corresponding taxonomy cache object:
  - If the taxonomy is already loaded, the request is fulfilled by providing the cached taxonomy.
  - If the taxonomy is not already loaded, the taxonomy is loaded into memory. (Note that this requires the additional overhead of loading the taxonomy before the user request is fulfilled.)
- After the user request is fulfilled, the loaded taxonomy remains in memory. Any subsequent requests on the loaded taxonomy are fulfilled from the cache.
- When a request is made on a cache taxonomy, a timestamp is registered in order to determine the “last accessed time” of the taxonomy.
- The time stamp of the taxonomy subsequently helps to determine when it is safe to unload the taxonomy.
- When certain criteria is met, a taxonomy is unloaded from memory. This action releases the associated resources from the web application.

The criteria used to determine if a given taxonomy should be unloaded:
1. Available Memory—When the memory available to the Java Virtual Machine (JVM) reaches a certain threshold, the least used taxonomies are unloaded until a certain amount of memory is recovered.

2. Unused Taxonomy—When a certain time has elapsed since a loaded taxonomy was last used or accessed, the taxonomy is unloaded.

3. Maximum Taxonomies Loaded—When the number of taxonomies that have been loaded meets or exceeds a specified threshold, the least used taxonomies are unloaded automatically.

**Taxonomy Cache Polling Feature**

After a taxonomy is loaded into memory, a polling feature is provided to determine when a taxonomy can be unloaded. The polling system works in this way:

- Every time a request is made on a taxonomy cache object, a time stamp is registered to determine the “last accessed time” of the taxonomy.
- The time stamp subsequently helps to determine when a taxonomy cache object is a candidate for unloading; that is, Disclosure Management applies the “least recently used” or the LRU cache algorithm.
- Disclosure Management spins two threads that are responsible for polling the taxonomy cache objects which have loaded taxonomies (in memory).
  - The first thread automatically runs every 60 seconds. It tests the amount of free memory that is available to the JVM (using the Runtime.freeMemory() Java API). If the amount of free memory is less than 1 MB, Disclosure Management automatically unloads the least recently used taxonomy cache objects until Disclosure Management has freed more than 1 MB of memory.
  - The second thread runs at a user-defined interval (using the `taxonomy_cache_poll` property). When this thread is enabled, the thread polls the taxonomy cache objects (with loaded taxonomies) and performs three tests to determine whether a taxonomy should be unloaded:
    - **Available Memory**—When the memory available to the JVM reaches a certain threshold, the least recently used taxonomies are unloaded until a certain amount of memory is recovered. This is the same test as the one performed by the first thread as discussed above. This test is covered in detail in “JVM Memory Threshold” on page 23.
    - **Unused Taxonomy**—When a certain amount of time has elapsed since a loaded taxonomy was last used or accessed, the taxonomy is unloaded.
    - **Maximum Taxonomies Loaded**—When the number of taxonomies loaded meets or exceeds a user specified threshold, the least used taxonomies are unloaded automatically.

This process is described at: “Maximum Taxonomies Loaded Threshold” on page 24.
Cache Poll Interval

The cache poll interval property indicates the frequency or interval in which the system inspects the cached taxonomies to determine whether a taxonomy is unloaded. In the file, this property is named: `taxonomy_cache_poll`.

Settings for this property include:

- **Value**—The value for this property is specified as an integer representing minutes.
- **Default**—The default value is 5 minutes. For example, setting the property to `taxonomy_cache_poll=5` means that all taxonomies loaded in memory are polled every 5 minutes. The thread runs every 5 minutes, after which the threshold tests (described below) are performed. If the interval is longer than the Maximum value (10 hours), Disclosure Management starts the thread every 10 hours instead of what is specified by this property.
- **Maximum**: The system maximum value is 10 hours.
- **Disable**—Setting the value to zero disables the polling feature. Oracle does not recommend that this feature be disabled. Other caching properties depend on the polling feature to be enabled. If this property is disabled, the only way a taxonomy is unloaded is when the JVM Memory Threshold is exceeded—or if the Disclosure Management web application is shut down or restarted.

JVM Memory Threshold

The JVM (Java Virtual Machine) memory threshold is not user configurable. When either the cache poll routines run, the first test checks how much free memory that is available of the JVM of the Disclosure Management web application. If the free memory is less than 1 MB, the least used taxonomies are automatically unloaded until the amount of available memory exceeds the threshold (1 MB). The least recently used taxonomies are determined by examining the time stamp of when a taxonomy was last used or accessed. The more time that has elapsed since a taxonomy was last used, the greater the chance that it is unloaded. The most recently used taxonomies have the best chance to remain in memory.

Least Recently Used Taxonomy Threshold

The least recently used taxonomy threshold property indicates the maximum time that can elapse since a taxonomy was last accessed before it is unloaded. In the properties file, this property is named `taxonomy_cache_threshold`.

Settings for this property include:

- **Value**—In minutes.
- **Default**—The default value is 30 minutes. For example, setting the value to 30 means that a loaded taxonomy remains in the cache (memory) for up to 30 minutes of inactivity before it is unloaded. When a new user request, which accesses a taxonomy occurs, its time stamp is reset. In this example, 30 minutes of no user requests must occur before the taxonomy is unloaded.
- **Disable**—Setting the value to zero disables this feature.
Maximum Taxonomies Loaded Threshold

The maximum taxonomies loaded threshold property indicates the maximum number of taxonomies that can be loaded in the cache (memory) before the least recently used taxonomies are unloaded. In the properties file, this property is named: max_taxonomy_cached.

Settings for this property:

- **Value**—Specified as a positive integer.
- **Default**—The default value is 10 taxonomies. For example, setting the value to 10 means that the number of loaded taxonomies that can remain in the cache (memory) cannot exceed 10. If 10 taxonomies are currently loaded in the cache, and a request is made to load an 11th taxonomy, the least used taxonomy is unloaded.
- **Disable**—Setting the value to zero disables this feature.

The “least recently used taxonomy” is determined by examining the time stamp of when a taxonomy was last used or accessed. The more time that has elapsed since a taxonomy was last used, the greater the chance that it is unloaded. The most recently used taxonomies have the best chance to remain in memory.

UBmatrix XBRL Processing Engine Settings

Disclosure Management uses the UBmatrix XBRL Processing Engine© (XPE) as the back-end engine for the majority of the XBRL processing. XPE provides a rich set of APIs that enable Disclosure Management to process and create XBRL documents. Disclosure Management uses XPE within the web application. The following section describes the settings exposed by XPE for performance and caching of XBRL documents.

The majority of the performance and caching settings for XPE can be found at: http://docs.ubmatrix.com/webhelp/XPE/3_5/.

Note that the this site should be viewed with Microsoft Internet Explorer. There are some known issues when viewing the documentation with Mozilla FireFox.

While the XPE online documentation provides details for XPE performance tuning, note the following settings:

- **Configuring the Web Cache**—http://docs.ubmatrix.com/webhelp/XPE/3_5/Configuration/configuring_the_web_cache.htm.
- **Configuring the JVM**—http://docs.ubmatrix.com/webhelp/XPE/3_5/.

XPE Taxonomy Caching Options

XPE provides three types of caching options:
- Preload—A commonly used taxonomy can be preloaded every time XPE is initiated, which is useful with frequently used taxonomies. After XPE is initiated, the preloaded taxonomies are already loaded in memory and available for processing. See: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/Preload.htm.

- Web Caching—Some taxonomies have external references to other taxonomies or XBRL documents that must be fetched though the Internet when they are not locally available. After XPE retrieves these external resources, they are saved locally the next time they are required. The web cache feature in Disclosure Management is described below. See: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/web_caching.htm.

- Redirection—XPE provides a mechanism to redirect external taxonomy references to local resources. This feature prevents XPE from fetching the external taxonomy resources from the Internet; instead, local resources are used. See: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/Redirection.htm.

The following are the usage points with Disclosure Management:
- Preload—While preloading taxonomies might be useful for some users, the Disclosure Management caching system can better manage loading and unloading taxonomies. A preloaded taxonomy can eventually be unloaded by Disclosure Management (per the caching feature described above). The use of this feature is not recommended.
- Web caching—Web caching is the recommended caching mechanism. See “XPE Taxonomy Caching Overview” on page 25.
- Redirection—Disclosure Management does not encourage the user of redirection, which is unreliable and difficult to configure. UBmatrix recommends web caching instead of redirection.

**XPE Taxonomy Caching Overview**

This section provides a brief overview of the taxonomy caching framework. It is important to understand the process that XPE employs when attempting to load a taxonomy:

- When initialized, XPE loads preloads into the document cache.
- When a request is made to load a new taxonomy (which is not already in the document cache), XPE takes the following actions:
  - Checks the web cache first.
  - If the requested documents are not found in the web cache, XPE uses the following built-in resolver settings:
    - The documents are searched in the file system (that is, File Resolver).
    - The documents are searched in the web (that is, HTTP resolver).
    - The documents are searched using other resolvers (none of which applies to Disclosure Management).
- If the documents are not found in the built-in resolver locations (that is, Preload and Redirection), then the document fails to load, and XPE generates an error.
Additional details about the XPE caching framework are available at: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/How_does_document_caching_work.htm.

**XPE Web Caching**

Web caching is the only XPE taxonomy caching mechanisms recommended for use with Disclosure Management.

Under the Disclosure Management system, XPE typically loads a registered taxonomy from the file system. Disclosure Management registered taxonomies are installed by the Administrator under the XbrlFiles folder. Most taxonomies are self-contained when downloaded from an official taxonomy repository (such as xbrl.org.) Some have external references to other taxonomies or XBRL documents. When this condition exists, XPE must resolve the external references to obtain the external documents. The first place XPE searches for these external documents is within its local “web cache”. If the documents are not found in the web cache, it searches the file system and ultimately the Internet—if access is provided to XPE.

**Web Cache Folder**

The web cache is a folder on the machine hosting the XPE process. In the case of Disclosure Management, it is on the server hosting the Disclosure Management Web application. Particularly, the web cache in: %DISCMAN_HOME%\resources\System\cache.

When XPE needs to fetch any XBRL resources (external documents and/or taxonomies) from the Internet, the download files are automatically stored in the Web Cache folder. In this manner, the next time these documents are required, XPE looks for them in the Web Cache folder before attempting to obtain them from another location. Additional details about the XPE cache folder are available at: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/How_does_web_caching_work.htm.

**Configuring the Web Cache**

If you need to enable XPE to fetch requested XBRL documents from the Internet, modify the xbrlData.properties file in the following folder: %DISCMAN_HOME%\lib\xbrlData.properties.

In the xbrlData.properties file, the following properties control the Web Cache feature:

- **WorkOffline**—Controls whether XPE has access to the Internet. This property is set to true by default. While it is recommended that this property is set to true, some clients do not enable Internet access to processes, especially in a server environment. When this property is set to false, administrators must ensure that the Web Cache folder contains all XBRL documents used by their registered taxonomies (for details see below).

**Note:** If a requested document is not available to XPE, and this property is set to false, loading the requesting taxonomy may fail. See: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Work_Offline.htm.
- **useCache**—Enable or disable the use of the web cache folder. When this property is set to False, the web cache folder is completely disabled. Oracle highly recommends setting this property default to True. See: http://docs.ubmatrix.com/webhelp/XPE/3_5/Configuration/configuring_the_web_cache.htm.

- **proxyHost**—Specify the proxy for XPE to use if Internet access is provided to it. Setting this property is important if a proxy must be used for XPE to get Internet access. By default, this property does not exist. For example, on the Oracle network, the following proxy setting can be specified: `proxyHost=www-proxy.us.oracle.com:80`.

  See: http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Configuration/Configuring_a_proxy_server.htm.

**Note:** Changing any of these properties requires that XPE is reinitialized, which requires a restart of the Disclosure Management Web application.

### Recommended Usage in Disclosure Management

This section provides the several usage scenarios for using the XPE web cache feature.

XPE copies the external Web resources into the web cache folder only if the `useCache` property in the `xbrlData.properties` file (on the Disclosure Management web server) is set to “true”. This setting enables XPE to copy any external taxonomy files that it retrieved from the web into the local web cache folder on the Disclosure Management web server. This setting also forces XPE to look for the externally referenced taxonomy resources in the web cache folder before attempting to fetch them from the Internet. In this case, you must have successfully rendered the taxonomy in question at least once so that any external files were copied to the web cache folder.

Subsequent requests to render the taxonomy results in XPE looking for the external resources in the web cache folder; thus, no Internet connection should be required. Alternately, you can manually copy the externally referenced taxonomy files to the web cache folder. However, this is tricky because the folder structure for those files must follow the resources' namespace sequence. For example, if the namespace of the external file is `http://external.com/2010/04/30/ExternalTaxonomy.xsd`, copy the `ExternalTaxonomy.xsd` under the following folder structure: `web cache folder\http\external.com\2010\04\30` (the `http` folder must be included). Note that all of these scenarios assume that the `useCache` property (in `xbrlData.properties`) is set to true. Turning off the web cache feature is not recommended.

### Allow Internet Access

The easiest solution is to allow XPE Internet access so externally referenced XBRL documents are automatically downloaded and available in the web cache folder:

- The `workOffline` property (in the `xbrlData.properties` file) is set to False.
- The `useCache` property is set to True.
- Ensure that `proxyHost` property contains a proxy server if necessary.
**WorkOffline**

You can block XPE from access to the Internet, which is preferable on a secured server environment.

- Set `workOffline` property (in the `xbrlData.properties` file) to true. When working offline, the administrator must ensure that all externally referenced XBRL documents within the registered taxonomies are available in the web cache folder.
- Set `useCache` property to true to ensure that the necessary files in the web cache folder are used.

When using this solution, administrators can manually fill the web cache folder using any file transfer technique preferred (for example, FTP, copy and paste). The folder structure must represent the XBRL URI of the document. Sometimes the URI is not apparent. Administrators may need to open the XBRL document in a text editor to determine the folder structure.

For additional details on the Web Cache folder structure, see: [http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/How_does_web_caching_work.htm](http://docs.ubmatrix.com/webhelp/XPE/3_5/default.htm#Caching/How_does_web_caching_work.htm).

**Provide Temporary Internet Access**

Another solution is to enable XPE temporary access to the Internet so that the required external XBRL files are automatically downloaded to the web cache folder (with the appropriate folder structure). To implement this solution:

1. Give XPE Internet access as described in “Allow Internet Access” on page 27.
2. Restart the Disclosure Management Web application if necessary.
3. In the Disclosure Management add-in or in Oracle Hyperion Financial Reporting, load the taxonomy containing links to external XBRL resources. When the taxonomy is fully rendered, externally referenced XBRL documents are downloaded to the web cache folder.
4. Disable Internet access for XPE by setting the `workOffline` property (in the `xbrlData.properties`) to true.
5. Restart the Disclosure Management Web application.

**Note:** The steps may be required when a new taxonomy is registered in the Disclosure Management system.

**Copy Folder Structure**

Another solution is to enable Internet access (as described previously) on a development environment. In this environment, the administrator can access and use the taxonomies that they want to cache in the web cache folder. The XBRL files are downloaded and installed on the development environment, the administrator can copy the entire web cache folder from the development and put it into the web cache folder of the production server. The production server can have the `workOffline` property permanently turned off. Table 3 describes the `workOffline` property:
Table 3  workOffline property

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>XPE cannot fetch from the Internet external XBRL resources referenced within taxonomies. The danger in setting this option is that XPE does not properly process the taxonomy in question if those resources are not cached in the web cache folder.</td>
</tr>
<tr>
<td>False</td>
<td>XPE is allowed Internet access to fetch external XBRL resources referenced within taxonomies. When external XBRL resources are required, XPE first checks the web cache folder for the resources. If they are not there, it attempts to fetch them from the Internet. This setting assumes that XPE has an Internet connection. In many environments (such as Oracle) this requires that the HTTP proxy setting must be indicated (with the proxyHost property in the xbrlData.properties). Note that this is the default setting (when you install Disclosure Management, this property is set to False). However, some companies do not allow services (such as Disclosure Management) Internet access. For these clients, Copy Folder Structure is a viable solution.</td>
</tr>
</tbody>
</table>

Using Registered Taxonomies in the Web Cache

You can employ the XPE web cache feature and register the taxonomies that exist in the web cache, which might be useful when a base taxonomy is commonly used. For example, suppose you work with taxonomy extensions that are based on the US GAAP taxonomy. While you usually work with the US GAAP taxonomy extensions, you occasionally work with the base US GAAP taxonomy.

In this scenario, having US GAAP base files in the web cache folder makes sense. But rather than having two copies of the US GAAP taxonomy (one in the web cache folder and the other in the XbrlFiles folder), you can keep the US GAAP taxonomy in the web cache folder and put a reference to the entry point in mappingtool.properties (for details on registering a taxonomy, see “Registering XBRL Taxonomies” on page 11):

- Download or copy the base taxonomy files to the web cache folder. The folder structure must be maintained. For example, if the 2009 US GAAP taxonomy is installed in the web cache, it might exist in the following folder: %DISCMAN_HOME%\resources\System\cache\http\taxonomies.xbrl.us\us-gaap\2009.

- Modify mappingtool.properties so a relative path is used to the new entry point of the base taxonomy. For example, to register the 2009 US GAAP Commercial & Industrial taxonomy after completing the previous step, indicate the following:

    taxonomy_X.label_Y=Commercial and Industrial 2009

    **Note:** Note the use of the relative path (’../’) at the beginning of the taxonomy_X.entryPoint_Y property.

iXBRL Instance Generation for Large Number of Mappings

If performance issues occur when generating an iXBRL instance document with a large number of mappings, increase the timeout period between the Oracle HTTP Server (OHS) and Oracle WebLogic app server (WL).
To increase the timeout period for iXBRL instance documents:

1. With any text editor, open the `EPM_INSTANCE\httpConfig\ohs\config\OHS\ohs_component\mod_wl_ohs.conf` file.
2. Set the `WLIOTimeoutSecs` parameter to a relatively large number of seconds for the `/discmanwebservices` context.

For example, you could change `WLIOTimeoutSecs;` to `60000` (seconds) as shown below:

```
/discmanwebservices context
<LocationMatch ^/discmanwebservices/>
SetHandler weblogic-handler
WeblogicCluster
  epbminw0076.epminsk.hyperion.com:8600, epbminw0076.epminsk.hyperion.com:8601
DynamicServerList OFF
WLIOTimeoutSecs 60000
</LocationMatch>
```

### Client Configuration Options

This section includes information on setting Disclosure Management options:

- “Setting up Server Information” on page 30
- “Preview Options” on page 31

### Setting up Server Information

Disclosure Management server information is stored in the Oracle's Hyperion Shared Services Registry. Initially, the server (host) name and server (host) port fields are empty. To set up the server information, users must specify the server name and port in the Disclosure Management Options dialog box. The server name and port should be the same as those used to download the extension. If you need to set up or point to another Disclosure Management server, use the Services Option to specify the server name and port and server access URLs.

2. From the navigation pane, select Services.
3. In Server Name, enter the server name.
4. In Port, enter the port number associated with server.
5. To enter the server access URLs manually for the following service access URL, select the URL and enter the address.
The Disclosure Management Service access URLs:

- XBRL Map Tool URL
- Session Service URL
- Report Service URL
- Mapping Service URL

6 Select OK.

Preview Options

For instance documents derived from the US GAAP taxonomy (or a related extension) Disclosure Management uses the SEC Interactive Financial Report Viewer (also known as the SEC viewer). The dependent files required to render the US GAAP-based instances are not shipped by Oracle. Disclosure Management users must download the source code for the SEC viewer. The source code can be downloaded from: http://www.sec.gov/spotlight/xbrl/xbrlvviewerlicense.htm.

Note: The SEC Viewer can be used only to display instances derived from a US GAAP taxonomy. It should not be used to view an instance that directly references the US GAAP taxonomy for SEC submission. When viewing instances that reference taxonomies not located in the same directory, the SEC viewer may not contain the full set of taxonomies. For example, if taxonomy “A” is in the ADir directory and it references taxonomy “B” in the ADir\BDir directory, the instance package may not include all the referenced taxonomies. The limitation exists because the taxonomy references may become too large, and including them all in the instance package is impractical. To resolve this issue, copy the taxonomies from the file system of the server to the file system of the client and maintain the original directory structure.
To download the latest version of the SEC viewer:


   Three downloads are available. Although Disclosure Management supports all three downloads, Oracle recommends that users download the personal renderer, which is the smallest file (12MB).

2. Select **Download the Rendering Engine configurable binary distribution**.


4. Unzip the source code files to your installation root source folder.

   **Note:** It is recommended that you unzip the SEC source code files to a local drive instead of a shared network drive.

5. From the Disclosure Management ribbon, select **Options**, and then **XBRL**.

6. Select the following:

   - **Auto Preview Published Documents**. Select one:
     - Select this check box to launch the Instance Viewer immediately after the report is generated.
     - Clear this check box to inhibit the Instance Viewer from launching immediately after the report is generated. A dialog confirms that the Disclosure Management Report is generated successfully.

   - **Preview with SEC Instance Viewer**. Select one:
     - Select this check box to use the SEC Interactive Financial Report Viewer to render the US GAAP-based instances (recommended). Then enter the root location of the source folder in the **SEC Instance Viewer Path** field.
     - Clear this check box to view the XBRL report in the generic Instance Viewer.

7. Click **OK**.

### Default Formatting

You can preserve the formatting for detailed tagging in the instance document. Formatting options are associated with nonnumeric XBRL mappings in which the string is based on concepts of the “xbrl.us:TextBlockItemType” type and its derivatives. The available formatting options include:

- **Rich text**—Disclosure Management extracts the HTML formatted content from Office document and applies this formatting to the nonnumeric item.

- **Plain text**—Disclosure Management uses the formatting value of the corresponding fact and applies the formatting to the nonnumeric item.
• Default—A plain text format is applied to nonnumeric items by default. However the default configuration can be set to use the rich text format for certain string types and their derivatives.

To include formatting in detailed tagging:
1. Select **Options**, and then select **XBRL**.
2. Select **Include formatting in detailed tagging**.
3. Click **OK**.

**Mappings**

You can instruct Disclosure Management to preserve data source selections as a default for future data source mappings.

To preserve data source selections as a default for future data source mappings:
1. Select **Options**, and then select **XBRL**.
2. Select **Remember choice for Select data source dialog**.
3. Select **Show Virtual Contexts**. When selected, show the context that Disclosure Management automatically generates. When not selected, show only the context that you created.
4. Click **OK**.

**Launching the Viewer**

When selecting **Generate XBRL** or **Generate iXBRL** to generate an instance report, you can choose whether to launch the Instance Viewer immediately after the instance report is generated.

To set the instance viewer option.
1. On the Disclosure Management menu bar, select **Options**, and then select **XBRL**.
2. On the **Select Preview Instance Document After Generation** check box, do one of the following:
   - Click the check box, to launch the Instance Viewer after the instance report is generated.
   - Clear the check box to ignore launching the Instance Viewer after the instance report is generated.
Disclosure Management Server Logs

The following server logs are located in the MIDDLEWARE_HOME/user_projects/domains/EPMSYSTEM/servers/DisclosureManagement0/logs directory:

- DisclosureManagement0.log—Disclosure Management Web tier activity
- DiscMan.log—Oracle Hyperion Disclosure Management activity
- DiscManAuditService.log—Audit service activity
- DiscManMappingTool.log—Mapping tool activity
- DiscManReportService.log—Report service activity
- DiscManRepository.log—Repository activity
- DiscManRepositoryService.log—Repository services activity
- DiscManSessionService.log—Session service activity