

ORACLE® HYPERION FINANCIAL MANAGEMENT,
FUSION EDITION

RELEASE 11.1.1.3

WEB DEVELOPER'S GUIDE

ORACLE®
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Financial Management Web Developer's Guide, 11.1.1.3

Copyright © 2002, 2009, 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

Chapter 1. Overview	19
Features Exposed by the Object Model	19
Overview of the Object Model	20
Sequence Diagrams of Object Model	23
About Subcubes	23
Examples of Subcubes	24
About Process Units	24
About Cell Statuses	25
Activity User IDs	25
Error Handling	25
Constant Files that You Should Include	26
Input/Output Arguments	26
Files Containing Supplemental Methods	26
Chapter 2. Getting Started	27
Logging a User on to an Application	27
Changing User Settings	28
Working with Metadata	28
Inserting, Updating, and Deleting Line Items	29
Process Management	29
Using the Document Repository	29
Working with Journals	30
Getting an Application's Security Classes	30
Defining a Data Grid	30
Error Handling	30
Chapter 3. HFMwApplications Type Library	31
HFMwManageApplications Component: Methods	31
CreateApplication	31
CreateApplicationCAS	31
DeleteApplication	32
DeleteXMLErrorsFromDatabase	33

DoesUserHaveCreateApplicationRights	33
DoesUserHaveSystemAdminRights	33
EnableNewConnections	34
EnumApplicationNames	35
EnumClusterNames	35
EnumProhibitConnections	35
EnumProvisioningProjectNames	37
EnumUserAppPreferences	37
EnumUsersOnSystem	37
EnumUsersOnSystemEx	40
EnumUsersOnSystemEx2	42
GetDisableAspTechnicalErrorMessage	44
GetDMEListenerConnectionInfo	44
GetErrorListCount	44
GetErrorListInRange	44
GetHFMErrLogRecordSet	44
GetSSOTokenUsingWebSecurityAgentCredentials	44
GetXMLErrorFromDatabase	45
GetXMLErrorsListFromDatabase	45
IsValidApplication	47
IsWebSecurityAgentEnabled	47
KillUsers	47
OpenApplication	48
RegisterApplicationCAS	49
SetLogonInfo	49
SetLogonInfoSSO	50
UpdateUserAppPreferences	51
ValidateTokenOnClusterSSO	51
WarnUsersForShutDown	51
HFmwManageApplications Component: Properties	51
domain	52
userName	52
Chapter 4. HFmwSession Type Library	53
HFmwSession Component: Methods	53
ApplyUserSettings	53
GetLicenseExpirationStatus	54
GetPOVByTag	54
GetUserSetting	55

HasSystemChanged	55
IsRunningTasks	56
RemoveFile	56
RemovePOVByTag	56
RemoveUserSetting	56
SetPOVByTag	57
SetUserSetting	57
HF MwSession Component: Properties	58
applicationName	58
clusterName	58
decimalSeparator	58
defaultDataGridDef	59
defaultProcessControlGridDef	59
descriptionLanguage	59
extractEncoding	59
IsBusy	60
IsDebug	60
language	60
metadata	60
resourceManager	60
session	60
showsDescriptions	61
thousandsSeparator	61
userName	61
usesApplets	61
Chapter 5. HF MwMetadata Type Library	63
HF MwMetadata Component	64
HF MwMetadata Methods	64
HF MwMetadata Properties	67
HF MwAccounts Component	72
GetTopMemberOfCustomsForAccount	72
dimension	73
HF MwScenarios Component	73
defaultFrequency	74
dimension	74
HF MwYears Component	74
dimension	74
HF MwPeriods Component	75

HFMwPeriods Component: Methods	75
HFMwPeriods Component: Properties	78
HFMwEntities Component	78
GetDefaultCurrenciesForEntities	78
dimension	79
HFMwValues Component	79
dimension	80
HFMwICPs Component	80
dimension	80
HFMwViews Component	80
dimension	80
HFMwCustom1 Component	81
dimension	81
HFMwCustom2 Component	81
dimension	81
HFMwCustom3 Component	81
dimension	82
HFMwCustom4 Component	82
dimension	82
HFMwDimension Component	82
HFMwDimension Methods	82
HFMwDimension Properties	101
HFMwCurrencies Component	101
EnumCurrencies	102
GetCurrencyID	103
GetCurrencyTranslationOperator	103
Chapter 6. HFMwData Type Library	105
HFMwData Component	105
ClearData	105
ClearDataAuditItems	106
ClearInvalidData	107
CopyData	107
DeleteLineItemDetails	109
EnumDataAuditItems	110
EnumDataAuditItems2	113
GetCellHistory	118
GetCellHistory2	120
GetDoubleFromText	123

GetEntityDetails	124
GetEntityDetailsHeader	126
GetFormattedNumber	127
GetInputPeriods	128
GetRegionsForCell	129
GetTextCellLineItems	130
InsertLineItemDetails	131
SetTextCellLineItems	132
SetTextCellsLineItems	134
SetWebSession	135
HFMwManageProcess Component	135
Approve	136
Approve2	137
ChangeProcessManagementStateForMultipleEntities	138
ChangeProcessManagementStateForMultipleEntities2	140
GetHistory	141
GetHistory2	144
GetHistory2UsingPhaseId	145
GetPhaseSubmissionGroupAndPhaseForCell	147
GetPhaseSubmissionReviewLevel	148
GetProcessManagementRoleAccess	150
GetReviewLevel	150
GetReviewLevelUsingPhaseID	151
MapReviewLevelToString	152
PhasedSubmissionProcessManagementChangeStateForMultipleEntities	153
PhaseSubmissionApprove	155
PhaseSubmissionGetHistory	157
PhaseSubmissionPromote	159
PhaseSubmissionReject	161
PhaseSubmissionSignOff	163
PhaseSubmissionStart	165
PhaseSubmissionSubmit	167
Promote	169
Promote2	171
Publish	172
Publish2	173
Reject	175
Reject2	176
SetWebSession	177

SignOff	177
SignOff2	178
Start	179
Start2	180
Submit	182
Submit2	183
HFMwCalculate Component	184
Allocate	185
ChartLogic	185
Consolidate	186
SetWebSession	187
Translate	187
Chapter 7. HFMwMbrSel Type Library	189
HFMwMbrSel Component	189
HFMwMbrSel Methods	190
HFMwMbrSel Properties	192
HFMwMbrSelDim Component	199
Attr	199
Custom	200
Dump	200
Chapter 8. HFMwDataGrid Type Library	201
HFMwDataGrid Component	201
Pages on Grids	202
Creating a Grid	202
HFMwDataGrid Methods	203
HFMwDataGrid Properties	254
Chapter 9. HFMwDocuments Type Library	265
HFMwManageDocuments Component	265
HFMwManageDocuments Methods	266
CreateFolder	266
CreateFolderEx	267
DeleteDocuments	268
DoesDocumentExist	269
EnumDocuments	269
EnumDocumentsEx	270
GetDocument	273
GetDocumentEx	274

LoadDocuments	275
LoadDocumentsEx	276
SaveDocument	277
SaveDocument2	278
SaveDocumentEx	279
SetWebSession	281
HFMwWorkspace Component	281
HFMwWorkspace Methods	282
HFMwWorkspace Properties	288
Chapter 10. HFMwSecurity Type Library	291
HFMwSecurity Component Methods	291
AddSecurityClasses	291
AllowRulesLoadForEPMAApp	292
DeleteSecurityClasses	292
DoesConnectedUserHaveApplicationAdminRights	292
EnumRoles	292
EnumSecurityClasses	293
EnumSecurityClassesForConnectedUser	294
EnumSecurityClassRights	294
EnumSecurityClassRightsForBiPub	295
EnumSecurityClassRightsAndRoles	296
EnumSecurityClassRightsAndRolesForBiPub	297
EnumUsers	298
EnumUsers2	298
EnumUsersInGroup	298
EnumUsersRoles	299
EnumUsersRolesForBiPub	300
EnumUsersInGroupForBiPub	301
EnumUsersWithFilter	302
GetOwner	303
GetRulesMode	303
GetSecurityClassAccessForAllUsers	303
GetSecurityClassAccessForAllUsers2	303
GetSecurityClassID	304
GetSecurityClassRightsForConnectedUser	304
GetUserAccessForAllSecurityClasses	305
GetUserAccessForAllSecurityClasses2	305
GetUserID	306

GetUserInfoFromUniqueID	306
GetUserInfoFromUniqueID2	306
GetUserSID	307
IsClassicHFMAApplication	307
IsConnectedUserAllowedToPerformTask	307
IsConnectedUserInRole	308
LogInfo	308
SaveSecurityClassRights	308
SetWebSession	309
Chapter 11. HFMwSystemInfo Type Library	311
HFMwSystemInfo Component: Methods	311
ClearAuditTasks	311
ClearAuditTasks2	312
EnableNewConnections	314
EnumActivityServers	314
EnumActivityUsers	315
EnumAuditTasks	315
EnumAuditTasks2	317
EnumProhibitConnections	319
EnumRunningTasks	321
EnumRunningTasksEx	324
GetActivityCodeDesc	327
GetKillUserStatus	327
GetRunningTaskLogFromServerFile	327
GetRunningTaskProgress	328
GetRunningTaskStatus	328
KillUsers	329
SetCurrentModule	330
SetCurrentModuleEx	330
SetWebSession	330
StopRunningTasks	331
WarnUsersForShutDown	331
Chapter 12. HFMwJournals Type Library	333
HFMwJournal Component	333
HFMwJournal Methods	334
HFMwJournal Properties	336
HFMwTemplate Component	341
HFMwTemplate Methods	342

HFMwTemplate Properties	343
HFMwManageJournals Component	348
HFMwManageJournals Methods	349
AddJournalGroup	349
ClosePeriodMultiple	349
CreateJournal	350
CreateJournalFromTemplate	351
CreateTemplate	352
DeleteJournal	352
DeleteJournals	353
DeleteTemplate	353
DeleteTemplates	354
EnumJournalGroups	354
ExecuteQuery	355
ExecuteQueryLabelsAndIDs	356
GenerateJournalFromRecurringTemplate	357
GetJournal	357
GetJournalFromXML	358
GetJournalsDisplayData	358
GetJournalsDisplayDataFromIDs	360
GetJournalUsingIDs	361
GetPeriodStatusMultiple	361
GetTemplate	362
GetTemplateFromXML	363
GetTemplatesDisplayData	363
GetTemplatesDisplayDataFromIDs	365
GetVariance	365
GetVarianceForTemplate	366
OpenPeriodMultiple	366
PerformBatchAction	367
RemoveAllJournalGroups	368
RemoveJournalGroup	368
SaveJournal	368
SaveJournal2	368
SaveTemplate	369
SetWebSession	369
ValidateLineItems	369
HFMwQueryDef Component	370
HFMwQueryDef Methods	371

HFMwQueryDef Properties	372
Chapter 13. HFMwUtilities Type Library	381
HFMwStringUtility Component	381
CalcCRC32	381
ConvertFromUTF8	382
IsValidValue	382
ReverseUnicodeByteOrder	383
HFMwFileReader Component	383
Close	383
IsEOF	383
OpenAsBinary	384
OpenAsText	384
Read	385
ReadLine	385
Size	385
HFMwEncodedFileWriter Component	386
Close	386
Open	386
Write	387
WriteBinary	387
Chapter 14. HFMwSharesCalc Type Library	389
HFMwSharesCalc Component Methods	389
SetWebSession	389
SharesCalculation	390
Chapter 15. HsvResourceManager Type Library	391
Error Handling with Financial Management	391
HsvResourceManager Object Methods	392
Handling Errors with the HsvResourceManager Object	392
System Message Detail Strings	393
GetAvailableLanguages	393
GetCurrentVersion	394
GetFormattedDateTime	394
GetFormattedError	395
GetFormattedErrorWithLineFeed	396
GetFormattedResourceString	397
GetHelpDirectoryForLanguageID	397
GetHFMLanguageIdFromUserLanguages	398

GetLanguageCountryCodeFromLanguageId	398
GetResourceString	398
GetResourceStringFromHR	398
GetUserDisplayDataTimeFormats	399
GetUserLanguageFromHFMLanguageId	399
GetWindowsDateFormat	400
Initialize	400
Error Message Lookup Utility	400
Appendix A. Constants: The HFMConstants Type Library	403
Metadata-Related Constants	403
Account Dimension Constants	404
Custom Dimension Constants	407
Consolidation Method Constants	409
Currency Attribute Constant	411
Entity Dimension Constants	411
View Dimension Constants	412
Supported Language Constants	413
Dimension-Related Constants	413
Intercompany Partner Constants	415
Period Dimension Constants	416
Scenario Attribute Constants	417
Member List Constants	418
Value Dimension Constants	418
Application Setting Attribute ID Constants	420
Calculation, Transaction, and Metadata Status Constants	421
Cell Calculation Status Constants	421
Cell Status Constants	422
Cell Metadata Status Constants	422
Subcube Period Calculation Status Constants	423
Transaction Dimension Constants	424
Cell Transaction Status Constants	425
Calculation Status Statistic Constants	426
Journal-Related Constants	427
Journal Action Constants	427
Balance Type Constants	428
Journal Column Display Constants	428
Debit/Credit Constants	429
Period Status Constants	429

Journal Report Display Option Constants	429
Journal Report Sort Option Constants	429
Journal Report Total Flag Constants	430
Journal Status Constants	430
Journal Type Constants	430
Journal Dimension Members Display Option Constants	431
Journal Web Session Parameter Constants	431
Template Column Display Constants	431
Template Type Constants	432
Balance Type Filter Constants	432
Journal Status Filter Constants	432
Journal Type Filter Constants	433
Process Management Constants	433
Process Management Action Constants	433
Process Management Review Level Constants	434
Process Management Filters	435
Process Management Validation Constants	435
Process Management Sorting Constants	436
Report Column Constants	436
Consolidation Type Constants	437
Tier Constants	437
Security Constants	438
Access Rights Constants	438
Role Constants	438
Task Constants	440
User Groups - User Type Flag Constants	445
Identity Type Constants	445
Search Filters	445
Web Constants	446
Data Explorer Task Constants	446
Data Explorer Process Management Constants	447
Data Grid Definition Constants	448
Data Grid Member Expansion Mode Constants	448
Data Grid Dimension Expansion Mode Constants	449
Data Grid Transaction Information Constants	449
Data Information Display Constants	449
Data Display Page Constant	450
Member Display Constants	450
Metadata Information Constants	450

Document Type Constants	451
Document File Type Constants	452
Extracted File Encoding Constants	452
Task Status Constants	452
User Activity Constants	453
Entity Transaction Detail Report Constants	456
Entity Transaction Detail Header Options	456
Entity Transaction Detail Display Options	456
Entity Transaction Detail Information	457
Entity Transaction Detail Row Types	458
Member Selection Constants	459
Dimension Value Type Constants	459
Point of View Attribute Constants	459
Processing Directive ID Constants	461
Validation Error Flags	462
Task ID Constants	462
Miscellaneous Constants	464
Showing Public and Private Documents	464
Date and Time Format Constants	465
Member ID Range	466
Number Defaults Constants	466
Share Calculation Ownership Constants	466
Share Calculation Types Constants	466
Validation Type Constants	467
Default Security Class Constant	468
Module ID Constants	468
Appendix B. XML String DTD Reference	473
Member List Enumerations DTD	473
Dimension Member Enumerations DTD	474
Data Grid Definition DTD	474
Data Grid Data and Headers DTD	477
Process Management Information DTD	480
Line Item Detail DTD	481
Cell Text DTD	481
Point of View DTD	482
Transaction Information DTD	483
Cell Adjustment DTD	486
Cell Information DTD	486

Cell Attachments DTD	487
Appendix C. Change History	489
Changes for Release 11.1.1.3	489
HFMwApplications	489
HFMwData	490
HFMwJournals	490
HFMwSecurity	490
Changes for Release 11.1.1	490
HFMwDataGrid Type Library	490
HFMwMetadata Type Library	490
HFMwSecurity Type Library	491
HFMwSession Type Library	491
Changes for Release 9.3.1	491
HFMwSession Type Library	491
HFMwData Type Library	491
HFMwSecurity Type Library	492
HFMwDataGrid Type Library	492
Changes for Release 9.3.0.1	492
Classic Applications	492
HFMwApplications Type Library	492
HFMwMetadata Type Library	493
HFMwData Type Library	493
HFMwDataGrid Type Library	493
HFMwSecurity Type Library	493
Changes for Release 4.1	494
HFMwApplications Type Library	494
HFMwSession Type Library	494
HFMwMetadata Type Library	494
HFMwSecurity Type Library	495
HFMwDocuments Type Library	495
HFMwSystemInfo Type Library	496
HFMwDataGrid Type Library	496
Changes for Release 4.0	496
HFMwPOV Type Library - Removed	497
HFMwMbrSel Type Library	497
HFMwData Type Library	497
HFMwMetadata Type Library	497
HFMwDataGrid Type Library	498

HFMwDocuments Type Library	498
HFMwSecurity Type Library	498
HFMwSession Type Library	499
HFMwSystemInfo Type Library	499
Changes for Release 3.5.1	499
HsvResourceManager Type Library	499
Documentation Errors	500
Changes for Release 3.5	500
HFMwApplications Type Library	500
HFMwData Type Library	500
HFMwDocuments Type Library	501
HFMwSecurity Type Library	501
HFMwSystemInfo Type Library	501
HFMwJournals Type Library	501
HFMwDataGrid Type Library	502
Changes for Release 3.4	502
HFMwApplications Type Library	502
HFMwSession Type Library	502
HFMwData Type Library	503
HFMwDocuments Type Library	503
HFMwMetadata Type Library	503
HFMwPOV Type Library	504
HFMwSecurity Type Library	505
HFMwJournals Type Library	505
HFMwSystemInfo Type Library	505
HFMwDataGrid Type Library	505
HFMwUtilities Type Library	506
Changes for Release 3.0.4	506
HFMwApplications Type Library	506
HFMwSession Type Library	506
HFMwMetadata Type Library	507
HFMwData Type Library	507
HFMwDataGrid Type Library	507
HFMwUtilities Type Library	508
Miscellaneous Changes	508
Index	509

In This Chapter

Features Exposed by the Object Model	19
Overview of the Object Model.....	20
About Subcubes	23
About Process Units	24
About Cell Statuses.....	25
Activity User IDs.....	25
Error Handling	25
Constant Files that You Should Include.....	26
Input/Output Arguments	26
Files Containing Supplemental Methods	26

ASP and scripting language programmers can use the object model described in this guide to programmatically extend Oracle Hyperion Financial Management, Fusion Edition on the Web. This chapter describes various concepts that apply to the object model.

Features Exposed by the Object Model

The Web object model exposes many Financial Management features, including the following:

- Logging in
- Opening Financial Management applications
- Registering and unregistering clusters and application servers
- Getting attributes of dimension members
- Getting and setting line item details
- Performing process management actions
- Obtaining the history of a process unit's process management actions
- Verifying users' security rights
- Loading and extracting documents
- Storing Points of View in memory
- Persisting Points of View past a session

- Storing grids of data in memory
- Calculating ownership
- Getting task and data audit information
- Enabling and disabling connections
- Logging users off
- Getting information on the logged-on users
- Getting usernames and IDs of an application's users
- Creating and deleting journals
- Opening and closing periods
- Performing journal actions such as submitting and posting

Overview of the Object Model

The components documented in the *Oracle Hyperion Financial Management, Fusion Edition Web Developer's Guide* are designed for use by ASP programmers and programmers of scripting languages such as VBScript, JavaScript, Visual Basic, and so on. The following table summarizes the available type libraries and components, and the diagrams in [“Sequence Diagrams of Object Model” on page 23](#) illustrate the dependencies of the object model's components.

Table 1 Summary of Components

Type Library	Description	Chapter
HFMwApplications	<p>The HFMwManageApplications component works with applications and user information. Following are examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Set and get user logon information. ● Get names of applications and registered clusters. ● Work with system errors. 	Chapter 3, “HFMwApplications Type Library”
HFMwSession	<p>The HFMwSession component represents a user's connection to an application. Following are examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Get and set user preferences. ● Get the name of the current application. ● Obtain HFMwMetadata and HsvResourceManager object references. 	Chapter 4, “HFMwSession Type Library”
HFMwMetadata	<p>Includes the HFMwMetadata and HFMwDimension components, as well as components for each of the Financial Management dimensions and for currencies. Use these components to obtain</p>	Chapter 5, “HFMwMetadata Type Library”

Type Library	Description	Chapter
	metadata information for the various dimensions.	
HFMwData	<p>Includes the HFMwData, HFMwManageProcess, and HFMwCalculate components.</p> <p>The HFMwData component provides access to application data. Following are examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Get and set line item details. ● Return input periods for a scenario. <p>The HFMwManageProcess components exposes Financial Management's process management functionality.</p> <p>The HFMwCalculate component provides methods for executing consolidations, translations, calculations, and allocations.</p>	Chapter 6, "HFMwData Type Library"
HFMwMbrSel	<p>Includes the HFMwMbrSel and HFMwMbrSelDim components. Following are examples of the functionality these components expose:</p> <ul style="list-style-type: none"> ● Store Points of View in memory. ● Persist Points of View past a Financial Management session. 	Chapter 7, "HFMwMbrSel Type Library"
HFMwDataGrid	<p>The HFMwDataGrid component provides a data grid interface. Following are a few examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Define and save grids using XML strings. ● Get and set data in grid cells. ● Save cells in the grid to the database. ● Return column and row header information. ● Return a cell's process management information. 	Chapter 8, "HFMwDataGrid Type Library"
HFMwDocuments	<p>Includes the HFMwManageDocuments and HFMwWorkspace components.</p> <p>The HFMwManageDocuments component exposes Financial Management's document repository. Following are examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Works with documents such as data forms and reports. ● Add and delete documents. ● Return information on documents. 	Chapter 9, "HFMwDocuments Type Library"

Type Library	Description	Chapter
	<p>The HFMWorkspace component exposes task lists. Following are examples of the component's functionality:</p> <ul style="list-style-type: none"> ● Add documents to task lists. ● Delete documents from task lists. ● Get documents in task lists. 	
HFMwSecurity	<p>The HFMwSecurity component provides security-related functionality such as the following:</p> <ul style="list-style-type: none"> ● Enumerate an application's security classes. ● Test user rights to a given task. ● Return all users' access rights to a security class. ● Return a user's access rights to all security classes. 	Chapter 10, "HFMwSecurity Type Library"
HFMwJournals	<p>The HFMwJournals component provides journal-related functionality such as the following:</p> <ul style="list-style-type: none"> ● Create and delete journals and templates. ● Manage periods. ● Perform journal actions such as submitting and posting. ● Define journal and template column filtering and sorting criteria. ● Create journal reports based upon specified criteria. 	Chapter 12, "HFMwJournals Type Library"
HFMwSystemInfo	<p>The HFMwSystemInfo component exposes various system-related features such as the following:</p> <ul style="list-style-type: none"> ● Get task audit information. ● Clear the task audit history. ● Enable and disable new connections. ● Log off users. 	Chapter 11, "HFMwSystemInfo Type Library"
HFMwSharesCalc	<p>The HFMwSharesCalc component enables you to calculate ownership for an application.</p>	Chapter 14, "HFMwSharesCalc Type Library"
HFMwUtilities	<p>Includes the following components, which operate upon strings and files:</p> <ul style="list-style-type: none"> ● HFMwStringUtility - Provides methods that work with strings. ● HFMwFileReader - Reads files. 	Chapter 13, "HFMwUtilities Type Library"

Type Library	Description	Chapter
	<ul style="list-style-type: none"> ● HFMwEncodedFileWriter - Writes to files. 	
HsvResourceManager	The HsvResourceManager object enables you to handle Financial Management errors, and also provides localized resource string handling and formatting.	Chapter 15, “HsvResourceManager Type Library”
HFMConstants	The HFMConstants type library includes constants used by many of the methods and properties.	Appendix A, “Constants: The HFMConstants Type Library”

Tip: [Chapter 2, “Getting Started”](#) further introduces the components and provides links to snippet examples of basic functionality.

Sequence Diagrams of Object Model

The following diagrams illustrate the sequential dependencies of the object model’s components. [Figure 1](#) illustrates most of the components in the object model. The HFMwMetadata type library contains several components and is diagrammed in [Figure 2 on page 23](#).

Figure 1 Web Object Model

The following sequence diagram illustrates the components in the HFMwMetadata type library. The HFMwMetadata component provides references to the objects that represent the various dimensions, and the HFMwDimension component is implemented as an interface in the dimension-related objects.

Figure 2 HFMwMetadata Type Library Components

About Subcubes

Several methods work with *subcubes*. A subcube consists of all the cells that share the same members of the following dimensions:

- Year
- Scenario
- Entity
- Value

There are two types of subcubes—*currency subcubes* and *node subcubes*. These types of subcubes differ in how they use Entity and Value dimension members:

- A currency subcube contains cells that share applicable non-node Value dimension members. For currency subcubes, the parent of the Entity member is irrelevant. The applicable non-node Value dimension members are listed as follows:

- Members for user-defined currencies. There is one triplet of Value dimension members for each user-defined currency. For example, if an application contains a currency named USD, the currency's triplet of Value dimension members will be USD, USD Adjs, and USD Total.
- The triplet that points to the entity's default currency. This triplet consists of the <Entity Currency>, <Entity Curr Adjs>, and <Entity Curr Total> Value members.
- [None] Value member.

Note: The non-node Value dimension members that point to parent entities' default currencies—<Parent Currency>, <Parent Curr Adjs>, and <Parent Curr Total>—are irrelevant to currency subcubes.

- A node subcube contains cells that share a common node Value dimension member. For node subcubes, both parent and child Entity members must be specified. The node Value dimension members are listed as follows:
 - [Contribution Total]
 - [Contribution Adjs]
 - [Contribution]
 - [Elimination]
 - [Proportion]
 - [Parent Total]
 - [Parent Adjs]
 - [Parent]

Examples of Subcubes

The following list provides examples of the two types of subcubes:

- Currency subcube - A currency subcube stores the cells for the year 2002, Actual scenario, Connecticut entity, and USD currency. The USD currency is represented by the [USD], [USD Adjs], and [USD Total] Value dimension triplet.
- Node subcube - A node subcube stores the cells for the year 2002, Actual scenario, Connecticut child entity, UnitedStates parent entity, and [Contribution] Value dimension member.

About Process Units

Several methods work with *process units*. Process units are subcubes with periods. For example, if an application has a 12-period calendar, each subcube contains 12 periods.

Tip: For details on subcubes, see [“About Subcubes” on page 23](#).

About Cell Statuses

Some methods return cell statuses. Cell statuses are numeric codes that are represented by constants in the `HFMConstants` type library and that provide the following types of status information for a cell:

- Transaction status, which indicates the source of a cell's data. Constants that represent the transaction statuses are listed in [“Cell Status Constants” on page 422](#).
- Metadata status, which provides information derived from the attributes of a cell's dimension members. Constants that represent the metadata statuses are listed in [“Cell Metadata Status Constants” on page 422](#).
- Calculation status, which provides information such as whether a cell is locked and whether it requires calculation, consolidation, or translation. Constants that represent the calculation statuses are listed in [“Cell Calculation Status Constants” on page 421](#).

The applicable methods return all three types of statuses within one return value. The return value is a bit-field containing 32 bits, in which the least significant 8 bits store the cell's transaction status codes, the most significant 10 bits store the cell's calculation status codes, and the middle bits store the cell's metadata status codes.

To access a cell's status codes in VBScript, perform bitwise comparisons with operators such as `AND`. For example, the calculation status indicating that a cell contains no data is represented by the constant `CELLSTATUS_NODATAINTABLE`. The following `If` statement tests whether a cell status includes the “no data” calculation status. In this example, assume that the cell status has been assigned to the `lStat` variable:

```
If (CELLSTATUS_NODATAINTABLE And lStat) Then
```

Activity User IDs

Some methods take or return *activity user IDs*. An activity user ID is a token that tracks user activities for audit purposes. Financial Management creates an activity user ID the first time a user performs an activity in an application.

Error Handling

Financial Management enables you to obtain both simple descriptions of errors that you can display to users and technical error information that can help debug issues. If Financial Management has been localized into a given language, you can also obtain error messages translated into that language. To take advantage of these features, use the `HsvResourceManager` type library. For details, see [Chapter 15, “HsvResourceManager Type Library.”](#)

Note that Financial Management reserves the hexadecimal range of `0x80040200` through `0x8004BFFF` for error numbers. Financial Management also reserves the hexadecimal range of `0x8004C000` through `0x8004FFFF` for third party components.

Constant Files that You Should Include

When you work with the Financial Management Web object model, Hyperion recommends that you include the following files. These files contain constants that are used across the object model:

- `HfmConstants.dll`
- `HsvStringIDLib.dll`

Input/Output Arguments

In Financial Management's Web object model, many method arguments are defined as "input/output." However, most such arguments ignore any input value that is specified. In other words, most arguments defined as "input/output" actually operate as output arguments; they are defined as "input/output" only for internal technical reasons.

How do you know whether a given "input/output" argument uses or ignores an input value? If the argument uses an input value, the argument description explicitly mentions the input usage; otherwise, the description mentions only the output usage.

Files Containing Supplemental Methods

To supplement the documented Web object model, Financial Management provides files containing methods that you may find useful. These methods are undocumented, though some may be commented. The files containing these methods are listed as follows:

- `Core.asp`: contains functions used throughout Financial Management's Web pages. `Core.asp` provides various types of methods; as one example, there are methods for array handling.
- `MessageDisplayFunctions.asp`: contains functions for displaying messages.

These files are installed on the Web server in the following directory:

`<install_directory>\FinancialManagement\Web\HFM\Common`

Caution! The methods in these files are considered unsupported, and are subject to change without notice.

2

Getting Started

In This Chapter

Logging a User on to an Application	27
Changing User Settings.....	28
Working with Metadata	28
Inserting, Updating, and Deleting Line Items.....	29
Process Management.....	29
Using the Document Repository	29
Working with Journals.....	30
Getting an Application's Security Classes	30
Defining a Data Grid	30
Error Handling	30

This chapter introduces you to the components documented in the *Oracle Hyperion Financial Management, Fusion Edition Web Developer's Guide*. The chapter topics summarize functionality provided by the components and link to code snippets.

Note: All examples are coded in Visual Basic 6. Using code examples in other scripting languages may yield errors.

Logging a User on to an Application

To log a user on to an application, use the following HFMwManageApplications component methods:

- Pass the user's domain name, username, and password to [SetLoginInfoSSO](#).
- Pass the application server cluster name and the application name to [OpenApplication](#).

The example for `OpenApplication` shows how to log a user on.

Note: `OpenApplication` returns an object reference to the `HFMwSession` object. You will need the `HFMwSession` object reference to use most of the other components in the Web object model. These components have a `SetWebSession` method that associates the component with the `HFMwSession` object that represents a user connection to an application.

Other Useful Methods for Logging On

- [EnumClusterNames](#) returns the names of the clusters that have been registered for the Web server.
- [EnumApplicationNames](#) returns the names of the Financial Management applications on a given application server cluster.

Changing User Settings

The `HFMwSession` component includes properties and methods that apply to user preferences. The following table lists the user preferences and the corresponding `HFMwSession` properties:

Table 2 User Preferences and `HFMwSession` Properties

User Preference	<code>HFMwSession</code> property
Decimal character	decimalSeparator
Thousands separator character	thousandsSeparator
Default language for descriptions	descriptionLanguage
Uses applets	usesApplets
Save files in Unicode	extractEncoding

By default, changes to these properties do not persist past the current session. However, you can save changes to these user settings with the `HFMwSession` method `ApplyUserSettings`.

For an example that shows how to change the decimal and thousands separator characters, see the example for [ApplyUserSettings](#).

Working with Metadata

The `HFMwMetadata` type library consists of components that provide access to metadata information such as the names and aliases of the dimensions, the dimension's members and member lists, a member's name and member ID, and so on. These components are introduced in [Chapter 5, "HFMwMetadata Type Library."](#)

The following examples introduce you to the `HFMwMetadata` type library's components:

- `HFMwMetadata.EnumDimensions` returns information on the Financial Management dimensions in an application. The example for [EnumDimensions](#) shows how to loop through the dimensions.
- `HFMwDimension.EnumChildren` returns information on a dimension member's children. This is useful when working with hierarchical dimensions such as the Entity dimension. The example for [EnumChildren](#) shows how to get the names of an Entity dimension member's children.

- `HFMwDimension.EnumMembers` returns information on the dimension members in a member list. The example for [EnumMembers](#) shows how to get the names of the entities in a member list.

Inserting, Updating, and Deleting Line Items

The `HFMwData` component enables you to work with line items, as shown by the following examples:

- The example for [InsertLineItemDetails](#) shows how to insert line item descriptions for a range of cells that intersect specified dimension members.
- The example for [SetTextCellLineItems](#) shows how to update line items for a cell.
- The example for [DeleteLineItemDetails](#) shows how to delete line item descriptions for a range of cells. The example also shows how to get line items for a cell with [GetTextCellLineItems](#).

Process Management

The `HFMwManageProcess` component of the `HFMwData` type library exposes Financial Management's process management functionality. This component provides methods for the various process management actions, and the method descriptions include examples. For a list of `HFMwManageProcess` methods, see [“HFMwManageProcess Component” on page 135](#).

The `GetHistory` method returns arrays of information that represent the history of a process unit's process management actions. The example for [GetHistory](#) prints a process unit's process management history to the browser as an HTML table.

Using the Document Repository

Financial Management's document repository enables you to store documents in the database. Document repository functionality is exposed by the `HFMwManageDocuments` component, and task lists are exposed by the `HFMwWorkspace` component. For more information, see [Chapter 9, “HFMwDocuments Type Library.”](#)

Document repository functionality is demonstrated by the following examples:

- The example for `HFMwManageDocuments.CreateFolder` shows how to create a folder in which to categorize documents.
- The example for `HFMwManageDocuments.LoadDocuments` shows how to load files from disk into a document repository folder.
- The example for `HFMwManageDocuments.DeleteDocuments` shows how to delete all the data forms in a folder. The example also shows how to enumerate all the documents in a folder with [EnumDocuments](#).

Tip: The `HFMwManageDocuments.SaveDocument` method creates a document in the document repository from a string.

Working with Journals

The `HFMwJournals` type library exposes Financial Management's journal features. For more information, see [Chapter 12, "HFMwJournals Type Library."](#) The following examples demonstrate a few of the exposed features:

- The example for `HFMwManageJournals.CreateJournalFromTemplate` shows how to create a journal from a template.
- The example for `HFMwManageJournals.GetJournalsDisplayData` shows how to obtain journal data for journals that meet the specified filtering criteria.

Getting an Application's Security Classes

The `HFMwSecurity` component exposes Financial Management security functionality. For more information, see [Chapter 10, "HFMwSecurity Type Library."](#)

The example for `EnumSecurityClasses` shows how to loop through an application's security classes.

Defining a Data Grid

The `HFMwDataGrid` component stores grids of data in memory. You can use a `HFMwDataGrid` grid to populate a graphical user interface (GUI). For details, see ["HFMwDataGrid Component" on page 201.](#)

The `HFMwDataGrid` component includes numerous properties that represent a data grid's settings, as well as methods that operate on a grid. The example for the `maxColsPerPage` property shows how to define a grid.

Error Handling

To handle Financial Management errors, use the `HsvResourceManager` object, which is introduced in ["HsvResourceManager Object Methods" on page 392.](#) The `HsvResourceManager` object enables you to return error messages that correspond to Financial Management error numbers. Messages can be returned in any language into which Financial Management has been localized.

The example for `GetFormattedError` shows how to get an error message localized into a given language.

Tip: Financial Management also provides a utility with which you can look up descriptions of error numbers. For more information, see ["Error Message Lookup Utility" on page 400.](#)



HFMwApplications Type Library

In This Chapter

HFMwManageApplications Component: Methods	31
HFMwManageApplications Component: Properties.....	51

The HFMwApplications type library contains the HFMwManageApplications component. The HFMwManageApplications component provides methods and properties to set and get user logon information, get names of applications and registered clusters, and work with system errors.

After creating an HFMwManageApplications object, you must log on a user with [SetLogonInfoSSO](#) before using any other HFMwManageApplications methods or properties.

Note: The example for [OpenApplication](#) shows how to log a user on to an application.

HFMwManageApplications Component: Methods

The HFMwManageApplications component contains the methods described in the following topics.

CreateApplication

Deprecated. Use [CreateApplicationCAS](#).

CreateApplicationCAS

Creates a Classic application.

Syntax

```
<HFMwManageApplications>.CreateApplicationCAS bstrClusterName,  
bstrApplicationName, bstrApplicationDesc, varabyCalendarStream,  
bstrProject, bstrAppWebServerUrl
```

Argument	Description
<i>bstrClusterName</i>	The name of the application server cluster. Input argument. String subtype.
<i>bstrApplicationName</i>	The name of the application. Input argument. String subtype.
<i>bstrApplicationDesc</i>	A description of the application. Input argument. String subtype.
<i>varabyCalendarStream</i>	A binary array containing the contents of the application profile file for the application. Tip: An application profile file is defined with Financial Management and has an extension of <code>.per</code> . Input argument.
<i>bstrProject</i>	String (ByVal). The name of the Oracle's Hyperion® Shared Services provisioning project. Tip: To get the names of the provisioning projects associated with an application server cluster, use EnumProvisioningProjectNames . Input argument. String subtype.
<i>bstrAppWebServerUrl</i>	The URL of the virtual directory for Financial Management. The URL should include the protocol, Web server name and port, and virtual directory name. Input argument. String subtype.

DeleteApplication

Deletes a Classic application.

Caution! The method will fail if used upon a Oracle Hyperion EPM Architect, Fusion Edition application.

Syntax

```
<HFMwManageApplications>.DeleteApplication bstrClusterName,  
bstrApplicationName
```

Argument	Description
<i>bstrClusterName</i>	The name of the server cluster that contains the application. Input argument. String subtype.
<i>bstrApplicationName</i>	The name of the application. Input argument. String subtype.

DeleteXMLErrorsFromDatabase

Deletes records of system errors for an application server cluster.

Syntax

```
<HFMwManageApplications>.DeleteXMLErrorsFromDatabase bstrClusterName,  
vbDeleteAll, varastrErrorReference
```

Argument	Description
<i>bstrClusterName</i>	The name of the application server cluster. Input argument. String subtype.
<i>vbDeleteAll</i>	Specifies whether all system errors should be deleted. Pass TRUE to delete all system errors, FALSE to delete only those errors specified by the <i>varastrErrorReference</i> argument. Input argument.
<i>varastrErrorReference</i>	An array of error reference numbers that identify the system errors to be deleted. Tip: Error reference numbers are enclosed within braces ({ }), and so the array items must include these braces. Input argument.

DoesUserHaveCreateApplicationRights

Indicates whether the current user belongs to the Financial Management Creator Group.

Syntax

```
<HFMwManageApplications>.DoesUserHaveCreateApplicationRights (bstrCluster)
```

Argument	Description
----------	-------------

<i>bstrCluster</i>	The name of the application server cluster on which to authenticate the current user's logon credentials. Input argument. String subtype.
--------------------	--

Return Value

Returns TRUE if the current user belongs to the Creator Group, FALSE otherwise. Boolean subtype.

DoesUserHaveSystemAdminRights

Indicates whether the current user belongs to the Financial Management Administrator Group.

Syntax

```
<HFMwManageApplications>.DoesUserHaveSystemAdminRights (bstrCluster)
```

Argument	Description
----------	-------------

<i>bstrCluster</i>	The name of the application server cluster on which to authenticate the current user's logon credentials. Input argument. String subtype.
--------------------	--

Return Value

Returns TRUE if the current user belongs to the Administrator Group, FALSE otherwise.
Boolean subtype.

EnableNewConnections

Enables or disables new Financial Management connections on a given application server cluster. You can enable or disable new connections for the entire server cluster, or for specific applications and application servers.

Syntax

```
<HFMwManageApplications>.EnableNewConnections vbEnable, bstrCluster,  
vbAllApplications, bstrAppName, vbAllServers, bstrServer
```

Argument	Description
----------	-------------

<i>vbEnable</i>	A flag that determines whether new connections are enabled or disabled. Pass TRUE to enable new connections, FALSE to disable. Input argument. Boolean subtype.
<i>bstrCluster</i>	The name of the application server cluster. Input argument. String subtype.
<i>vbAllApplications</i>	Indicates whether new connections to all applications are enabled or disabled. Pass TRUE to impact all applications, FALSE to impact a specific application. If you pass FALSE, you must pass an application name to the <i>bstrAppName</i> argument. Input argument. Boolean subtype.
<i>bstrAppName</i>	The name of the application for which to enable or disable new connections. This is used only if the <i>vbAllApplications</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllServers</i>	Indicates whether new connections to all of the server cluster's application servers are enabled or disabled. Pass TRUE to impact all servers, FALSE to impact a specific server. If you pass FALSE, you must pass an application server name to the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to enable or disable new connections. This is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.

EnumApplicationNames

Returns the names of the applications that exist on an application server cluster.

Syntax

```
<HFMwManageApplications>.EnumApplicationNames (bstrClusterName,  
pvarabstrApplicationDescs)
```

Argument	Description
<i>bstrClusterName</i>	The name of the application server cluster. Input argument. String subtype.
<i>pvarabstrApplicationDescs</i>	Returns an array of application descriptions for the returned application names. This array has a one-to-one correspondence with the return value's array. Output argument.

Return Value

An array containing the application names.

EnumClusterNames

Returns the names of the registered clusters on the Web server.

Syntax

```
<HFMwManageApplications>.EnumClusterNames ( )
```

Return Value

An array containing the names of the registered clusters.

EnumProhibitConnections

Returns information on the applications, application servers, and users for which connections have been disabled on a given application server cluster. The information is returned in arrays that have a one-to-one correspondence.

Tip: To return this information for all application servers regardless of the server cluster, use `HFMwSystemInfo.EnumProhibitConnections`.

Syntax

```
<HFMwManageApplications>.EnumProhibitConnections bstrCluster,  
pvaravarvbAllApps, pvaravarbstrAppNames, pvaravarvbAllServers,  
pvaravarbstrServerNames, pvaravarvbAllUsers, pvaravarlActivityUserIDs,  
pvaravarbstrActivityUserNames
```

Argument	Description
<i>bstrCluster</i>	<p>The name of the application server cluster.</p> <p>Input argument. String subtype.</p>
<i>pvaravrbAllApps</i>	<p>Returns an array that indicates whether connections are disabled for all applications. Array items can contain 0 or -1:</p> <ul style="list-style-type: none"> ● -1 indicates connections to all applications are disabled. ● 0 indicates that only connections to the application returned by the corresponding <i>pvaravrbstrAppNames</i> argument's array item are disabled. <p>Input/output argument.</p>
<i>pvaravrbstrAppNames</i>	<p>Returns an array containing the names of the applications for which connections are disabled. Application names are returned only when the corresponding <i>pvaravrbAllApps</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravrbAllApps</i> argument's array item contains -1, this array item contains the string "AllApps".</p> <p>Input/output argument.</p>
<i>pvaravrbAllServers</i>	<p>Returns an array that indicates whether connections are disabled for all application servers. Array items can contain 0 or -1:</p> <ul style="list-style-type: none"> ● -1 indicates connections to all application servers are disabled. ● 0 indicates that only connections to the application server returned by the corresponding <i>pvaravrbstrServerNames</i> argument's array item are disabled. <p>Input/output argument.</p>
<i>pvaravrbstrServerNames</i>	<p>Returns an array containing the names of the application servers for which connections are disabled. Application server names are returned only when the corresponding <i>pvaravrbAllServers</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravrbAllServers</i> argument's array item contains -1, this array item contains the string "AllServers".</p> <p>Input/output argument.</p>
<i>pvaravrbAllUsers</i>	<p>Returns an array that indicates whether connections are disabled for all users. Array items can contain 0 or -1:</p> <ul style="list-style-type: none"> ● -1 indicates connections for all users are disabled. ● 0 indicates that only connections for the user represented by the corresponding <i>pvaravarlActivityUserIDs</i> and <i>pvaravrbstrActivityUserNames</i> arguments' array items are disabled. <p>Input/output argument.</p>
<i>pvaravarlActivityUserIDs</i>	<p>Returns an array containing the activity user IDs of the users for whom connections are disabled. Valid IDs are returned only when the corresponding <i>pvaravrbAllUsers</i> argument's array item contains 0.</p> <p>The corresponding item in the array returned by the <i>pvaravrbstrActivityUserNames</i> argument contains the username associated with an activity user ID.</p>

Argument	Description
	<p>Note: If the corresponding <i>pvaravarvbAllUsers</i> argument's array item contains -1, this array item contains -1.</p> <p>Input/output argument.</p>
<i>pvaravarbstrActivityUserNames</i>	<p>Returns an array containing the usernames of the users for whom connections are disabled. Usernames are returned only when the corresponding <i>pvaravarvbAllUsers</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravarvbAllUsers</i> argument's array item contains -1, this array item contains an empty string.</p> <p>Input/output argument.</p>

EnumProvisioningProjectNames

Returns the names of the Shared Services provisioning projects associated with a given cluster.

Syntax

```
<HFMwManageApplications>.EnumProvisioningProjectNames (bstrClusterName)
```

Argument	Description
<i>bstrClusterName</i>	<p>The name of the application server cluster.</p> <p>Input argument. String subtype.</p>

Return Value

Returns an array containing the names of the provisioning projects.

EnumUserAppPreferences

For internal use.

EnumUsersOnSystem

Returns the usernames of and other information applicable to the users logged on to a given cluster. You can return information for users on all application servers, or filter by application, application server, and user.

The information is returned in arrays that have a one-to-one correspondence.

Note: To return the names of the user's active modules in a given language, use [EnumUsersOnSystemEx](#).

Syntax

```
<HFMwManageApplications>.EnumUsersOnSystem bstrCluster, vbAllApplications,  
bstrAppName, vbAllServers, bstrServer, vbAllUsers, lActivityUserID,  
pvaralActivitySessionIDs, pvaravarServerNames, pvaravarApplicationNames,  
pvaralActivityUserIDs, pvaravarUserNames, pvaralCurrentActivity,  
pvaravarModuleNames, pvaradTimeStarted, pvaravarUserDescs
```

Argument	Description
<i>bstrCluster</i>	The name of the application server cluster. Input argument. String subtype.
<i>vbAllApplications</i>	Indicates whether users of all applications are returned. Pass TRUE to return all users, FALSE to filter by application. If you pass FALSE, you must pass an application name to the <i>bstrAppName</i> argument. Input argument. Boolean subtype.
<i>bstrAppName</i>	The name of the application for which to return users. This is used only if the <i>vbAllApplications</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllServers</i>	Indicates whether users of all application servers in the application server cluster are returned. Pass TRUE to return users for all servers, FALSE to filter by application server. If you pass FALSE, you must pass an application server name to the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to return users. This is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllUsers</i>	Indicates whether all users who meet the other arguments' selection criteria are returned. Pass TRUE to return all users, FALSE otherwise. If you pass FALSE, you must pass a value for the <i>lActivityUserID</i> argument. Input argument. Boolean subtype.
<i>lActivityUserID</i>	The activity user ID for which to return users. This is used only if the <i>vbAllUsers</i> argument is set to FALSE. Input argument. Long subtype.
<i>pvaralActivitySessionIDs</i>	Returns an array of activity session IDs for the users. Note: Since a user can be logged on to multiple sessions of Financial Management, these activity session IDs are generated by the system to uniquely identify each user session. Input/output argument.
<i>pvaravarServerNames</i>	Returns an array containing the names of the servers to which the users are logged on. Note: This is a 1-based array. Input/output argument.
<i>pvaravarApplicationNames</i>	Returns an array containing the names of the applications to which the users are logged on.

Argument	Description
	<p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaralActivityUserIDs</i>	<p>Returns an array containing the activity user IDs of the logged on users.</p> <p>The corresponding item in the array returned by the <i>pvaravarUserNames</i> argument contains the username associated with an activity user ID.</p> <p>Input/output argument.</p>
<i>pvaravarUserNames</i>	<p>Returns an array containing the usernames of the logged on users.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaralCurrentActivity</i>	<p>Returns an array of activity IDs that identify the users' current activities. Valid values are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 453.</p> <p>Input/output argument.</p>
<i>pvaravarModuleNames</i>	<p>Returns an array containing the names of the modules in which the users are active.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaradTimeStarted</i>	<p>Returns an array of timestamps indicating when the users logged on.</p> <p>Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time.</p> <p>Input/output argument.</p>
<i>pvaravarUserDescs</i>	<p><i>Future use.</i> An array is returned, but you can ignore it.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>

Example

The following function returns an HTML table listing the usernames, application names, and application servers applicable to the currently logged-on users. The function takes an HFMwManageApplications object reference and the server cluster name.

```
Function GetTableOfUsers(cManageApp, sCluster)
Dim vaActivitySessionIDs, vaServerNames, vaApplicationNames
Dim vaActivityUserIDs, vaUserNames, vaCurrentActivity
Dim vaModules, vaTimeStarted, vaUserDescs, sReturn
cManageApp.EnumUsersOnSystem sCluster, TRUE, "", TRUE, "", _
    FALSE, 0, vaActivitySessionIDs, vaServerNames, _
    vaApplicationNames, vaActivityUserIDs, vaUserNames, _
    vaCurrentActivity, vaModules, vaTimeStarted, vaUserDescs
sReturn = "<table border=2><tr><td><b>User</b></td><td>" & _
    "<b>Application</b></td><td><b>Application Server</b>" & _
    "</td></tr>"
For i = lBound(vaUserNames) to uBound(vaUserNames)
    sReturn = sReturn & "<tr><td>" & vaUserNames(i) & "</td>"
```

```

        sReturn = sReturn & "<td>" & vaApplicationNames(i) & "</td>"
        sReturn = sReturn & "<td>" & vaServerNames(i) & "</td></tr>"
    Next
    sReturn = sReturn & "</table>"
    GetTableOfUsers = sReturn
End Function

```

EnumUsersOnSystemEx

Returns the usernames of and other information applicable to users logged on to a given cluster; you can specify the language in which the names of the user's active modules are returned. You can return information for users on all application servers, or filter by application, application server, and user.

The information is returned in arrays that have a one-to-one correspondence.

Note: To return module names in the system's default language, use [EnumUsersOnSystem](#).

Syntax

```

<HFMwManageApplications>.EnumUsersOnSystemEx bstrCluster,
vbAllApplications, bstrAppName, vbAllServers, bstrServer, vbAllUsers,
lActivityUserID, lLanguageID, pvaralActivitySessionIDs,
pvaravarServerNames, pvaravarApplicationNames, pvaralActivityUserIDs,
pvaravarUserNames, pvaralCurrentActivity, pvaravarModuleNames,
pvaradTimeStarted, pvaravarUserDescs

```

Argument	Description
<i>bstrCluster</i>	The name of the application server cluster. Input argument. String subtype.
<i>vbAllApplications</i>	Indicates whether users of all applications are returned. Pass TRUE to return all users, FALSE to filter by application. If you pass FALSE, you must pass an application name to the <i>bstrAppName</i> argument. Input argument. Boolean subtype.
<i>bstrAppName</i>	The name of the application for which to return users. This is used only if the <i>vbAllApplications</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllServers</i>	Indicates whether users of all application servers in the application server cluster are returned. Pass TRUE to return users for all servers, FALSE to filter by application server. If you pass FALSE, you must pass an application server name to the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to return users. This is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.

Argument	Description
<i>vbAllUsers</i>	<p>Indicates whether all users who meet the other arguments' selection criteria are returned. Pass TRUE to return all users, FALSE otherwise.</p> <p>If you pass FALSE, you must pass a value for the <i>IActivityUserID</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>IActivityUserID</i>	<p>The activity user ID for which to return users. This is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. Long subtype.</p>
<i>ILanguageID</i>	<p>The ID of the language in which to return module names.</p> <p>Tip: You can use the HsvResourceManager method GetAvailableLanguages to obtain the IDs of the languages valid for a given release.</p> <p>Input argument. Long subtype.</p>
<i>pvaralActivitySessionIDs</i>	<p>Returns an array of activity session IDs for the users.</p> <p>Note: Since a user can be logged on to multiple sessions of Financial Management, these activity session IDs are generated by the system to uniquely identify each user session.</p> <p>Input/output argument.</p>
<i>pvaravarServerNames</i>	<p>Returns an array containing the names of the servers to which the users are logged on.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaravarApplicationNames</i>	<p>Returns an array containing the names of the applications to which the users are logged on.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaralActivityUserIDs</i>	<p>Returns an array containing the activity user IDs of the logged on users.</p> <p>The corresponding item in the array returned by the <i>pvaravarUserNames</i> argument contains the username associated with an activity user ID.</p> <p>Input/output argument.</p>
<i>pvaravarUserNames</i>	<p>Returns an array containing the usernames of the logged on users.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaralCurrentActivity</i>	<p>Returns an array of activity IDs that identify the users' current activities. Valid values are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 453.</p> <p>Input/output argument.</p>
<i>pvaravarModuleNames</i>	<p>Returns an array containing the names of the modules in which the users are active.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaradTimeStarted</i>	<p>Returns an array of timestamps indicating when the users logged on.</p>

Argument	Description
	<p>Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time.</p> <p>Input/output argument.</p>
<i>pvaravarUserDescs</i>	<p><i>Future use.</i> An array is returned, but you can ignore it.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>

EnumUsersOnSystemEx2

Returns the usernames and other information applicable to all users logged on to or logged out of a given cluster. You can specify the language in which the names of the user's active modules are returned. You can return information for users on all application servers, or filter by application, application server, and user. The information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<HFMwManageApplications>.EnumUsersOnSystemEx2 bstrCluster,
vbAllApplications, bstrAppName, vbAllServers, bstrServer, vbAllUsers,
lActivityUserID, lLanguageID, pvaralActivitySessionIDs,
pvaravarServerNames, pvaravarApplicationNames, pvaralActivityUserIDs,
pvaravarUserNames, pvaralCurrentActivity, pvaravarModuleNames,
pvaradTimeStarted, pvaravarUserDescs, pvaralSessionStatus
```

Argument	Description
<i>bstrCluster</i>	<p>The name of the application server cluster.</p> <p>Input argument. String subtype.</p>
<i>vbAllApplications</i>	<p>Indicates whether users of all applications are returned. Pass TRUE to return all users, FALSE to filter by application.</p> <p>If you pass FALSE, you must pass an application name to the <i>bstrAppName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrAppName</i>	<p>The name of the application for which to return users. This is used only if the <i>vbAllApplications</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllServers</i>	<p>Indicates whether users of all application servers in the application server cluster are returned. Pass TRUE to return users for all servers, FALSE to filter by application server.</p> <p>If you pass FALSE, you must pass an application server name to the <i>bstrServer</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServer</i>	<p>The name of the application server for which to return users. This is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>

Argument	Description
<i>vbAllUsers</i>	<p>Indicates whether all users who meet the other arguments' selection criteria are returned. Pass TRUE to return all users, FALSE otherwise.</p> <p>If you pass FALSE, you must pass a value for the <i>IActivityUserID</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>IActivityUserID</i>	<p>The activity user ID for which to return users. This is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. Long subtype.</p>
<i>ILanguageID</i>	<p>The ID of the language in which to return module names.</p> <p>Tip: You can use the HsvResourceManager method GetAvailableLanguages to obtain the IDs of the languages valid for a given release.</p> <p>Input argument. Long subtype.</p>
<i>pvarIActivitySessionIDs</i>	<p>Returns an array of activity session IDs for the users.</p> <p>Note: Because a user can be logged on to multiple sessions of Financial Management, these activity session IDs are generated by the system to uniquely identify each user session.</p> <p>Input/output argument.</p>
<i>pvaravarServerNames</i>	<p>Returns an array containing the names of the servers to which the users are logged on.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaravarApplicationNames</i>	<p>Returns an array containing the names of the applications to which the users are logged on.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvarIActivityUserIDs</i>	<p>Returns an array containing the activity user IDs of the logged on users.</p> <p>The corresponding item in the array returned by the <i>pvaravarUserNames</i> argument contains the username associated with an activity user ID.</p> <p>Input/output argument.</p>
<i>pvaravarUserNames</i>	<p>Returns an array containing the usernames of the logged on users.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvarIActivityCurrentActivity</i>	<p>Returns an array of activity IDs that identify the users' current activities. Valid values are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 453.</p> <p>Input/output argument.</p>
<i>pvaravarModuleNames</i>	<p>Returns an array containing the names of the modules in which the users are active.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaradTimeStarted</i>	<p>Returns an array of timestamps indicating when the users logged on.</p>

Argument	Description
	<p>Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time.</p> <p>Input/output argument.</p>
<code>pvaravarUserDescs</code>	<p><i>Future use.</i> An array is returned, but you can ignore it.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<code>pvaralSessionStatus</code>	<p>Returns an array of status IDs that identify the user's task status. Valid values are represented by the <code>HFMConstants</code> type library constants listed in "Task Status Constants" on page 452.</p> <p>Input/output argument.</p>

GetDisableAspTechnicalErrorMessage

For internal use.

GetDMEListenerConnectionInfo

For internal use.

GetErrorListCount

Future use.

GetErrorListInRange

Future use.

GetHFMErrLogRecordSet

Future use.

GetSSOTokenUsingWebSecurityAgentCredentials

Returns an external authentication token generated from logon credentials that have been validated by a security agent such as SiteMinder. You use an ASP Request object to pass the logon credentials from the security agent.

Tip: You can log on a user by passing the returned token to [SetLogonInfoSSO](#). You can check whether Web security agent support is enabled with [IsWebSecurityAgentEnabled](#).

Syntax

```
<HFMwManageApplications>.GetSSOTokenUsingWebSecurityAgent  
Credentials bstrCluster, varRequest, pvarToken
```

Argument	Description
----------	-------------

<i>bstrCluster</i>	The name of the server cluster on which the user is being validated.
--------------------	--

Input argument. String subtype.

<i>varRequest</i>	An ASP Response object that includes the HTTP header from the security agent. The HTTP header must contain logon credentials validated by the security agent.
-------------------	---

Input argument.

<i>pvarToken</i>	Returns the external authentication token.
------------------	--

If the Response object passed to *varRequestHTTP* does not contain a username that the security agent has validated, no token is returned.

Output argument.

GetXMLErrorFromDatabase

Returns the message for a system error on an application server cluster, given the system error's reference number.

Syntax

```
<HFMwManageApplications>.GetXMLErrorFromDatabase (bstrClusterName,  
bstrErrorReference)
```

Argument	Description
----------	-------------

<i>bstrClusterName</i>	The name of the application server cluster.
------------------------	---

Input argument. String subtype.

<i>bstrErrorReference</i>	The system error's reference number.
---------------------------	--------------------------------------

Tip: Error reference numbers are enclosed within braces ({ }), and so the value passed must include these braces.

Input argument. String subtype.

Return Value

The error message. String subtype.

GetXMLErrorsListFromDatabase

Returns the reference numbers, log types, timestamps, application server names, and application names of system errors that apply to an application server cluster. You can filter the errors to be returned by date range, application server name, and application name.

These error details are returned in a set of arrays that have a one-to-one correspondence to each other.

Syntax

```
<HFMwManageApplications>.GetXMLErrorsListFromDatabase bstrClusterName,  
dStartTimeStamp, dEndTimeStamp, bstrServerName, bstrApplicationName,  
pvarabstrReference, pvarabstrLogType, pvarabstrTimeStamp,  
pvarabstrServerName, pvarabstrApplicationName
```

Argument	Description
<i>bstrClusterName</i>	The name of the application server cluster. Input argument. String subtype.
<i>dStartTimeStamp</i>	The start of the date range to filter by, or 0 to omit filtering by timestamp. Input argument. Double subtype.
<i>dEndTimeStamp</i>	The end of the date range to filter by, or 0 to omit filtering by timestamp. Input argument. Double subtype.
<i>bstrServerName</i>	The name of the application server to filter by, or an empty string to omit filtering by server. Input argument. String subtype.
<i>bstrApplicationName</i>	The name of the application to filter by, or an empty string to omit filtering by application. Input argument. String subtype.
<i>pvarabstrReference</i>	Returns an array of the reference numbers for the system errors that match the search criteria. Note: This is a 1-based array. Output argument.
<i>pvarabstrLogType</i>	Returns an array of the errors' log types. The valid return values are represented by the constants in "Access Rights Constants" on page 438 . Note: This is a 1-based array. Output argument.
<i>pvarabstrTimeStamp</i>	Returns an array of the errors' timestamps. Note: This is a 1-based array. Output argument.
<i>pvarabstrServerName</i>	Returns an array of the errors' application server names. Note: This is a 1-based array. Output argument.
<i>pvarabstrApplicationName</i>	Returns an array of the errors' application names. Note: This is a 1-based array. Output argument.

IsValidApplication

Indicates whether the specified application exists on the specified cluster.

Syntax

```
<HFMwManageApplications>.IsValidApplication(bstrCluster, bstrAppName)
```

Argument	Description
----------	-------------

<i>bstrCluster</i>	The cluster name. Input argument. String subtype.
--------------------	--

<i>bstrAppName</i>	The application name. Input argument. String subtype.
--------------------	--

Return Value

Boolean. Returns TRUE if the application exists on the cluster.

IsWebSecurityAgentEnabled

Indicates whether Web security agent support is enabled for the application server cluster.

Note: Security agent support is enabled using the Oracle's Hyperion Enterprise Performance Management System Configurator. For more information, see the Oracle Hyperion Enterprise Performance Management System Installer, Fusion Edition documentation.

Syntax

```
<HFMwManageApplications>.IsWebSecurityAgentEnabled()
```

Return Value

Returns TRUE if security agent support is enabled, FALSE otherwise. Boolean subtype.

KillUsers

Logs off users on a given application server cluster. You can log off all users on the server cluster, or users logged on to a given application and application server.

Tip: To log off a given user or session, use `HFMwSystemInfo.KillUsers`.

Syntax

```
<HFMwManageApplications>.KillUsers bstrCluster, vbAllApplications,  
bstrAppName, vbAllServers, bstrServer
```

Argument	Description
<i>bstrCluster</i>	The name of the application server cluster. Input argument. String subtype.
<i>vbAllApplications</i>	Indicates whether to log off users from all applications. Pass TRUE to log off users from all applications, FALSE to log off users from a given application. If you pass FALSE, you must pass an application name to the <i>bstrAppName</i> argument. Input argument. Boolean subtype.
<i>bstrAppName</i>	The name of the application for which to log off users. This is used only if the <i>vbAllApplications</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllServers</i>	Indicates whether to log off users from all application servers in the application server cluster. Pass TRUE to log off users from all servers, FALSE to log off users from a given application server. If you pass FALSE, you must pass an application server name to the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to log off users. This is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.

OpenApplication

Opens an application, and returns an HFMwSession object reference.

Tip: Before calling `OpenApplication`, specify the user's logon information with `SetLogonInfoSSO`.

Syntax

```
<HFMwManageApplications>.OpenApplication (bstrClusterName,  
bstrApplicationName)
```

Argument	Description
<i>bstrClusterName</i>	The name of the application server cluster on which the application is located. Input argument. String subtype.
<i>bstrApplicationName</i>	The name of the application to open. Input argument. String subtype.

Return Value

An HFMwSession object that represents the connection with the application.

You will use the returned HFMwSession object reference with most of the components documented in the *Oracle Hyperion Financial Management, Fusion Edition Web Developer's*

Guide. Most of the components have a `SetWebSession` method that associates the component with the `HFMwSession` object for an application.

Example

The following subroutine logs a user on to an application with NTLM user information and sets an `HFMwSession` object reference. The subroutine takes the server cluster name, user information, and application name. The user information is passed to `SetLogonInfoSSO`, and then `OpenApplication` is called.

```
Sub SetAppSession(sDomain, sUser, sPassword, sCluster, _
    sApp)
    Dim cHFMMangeApps, sToken
    Set cHFMMangeApps = Server.CreateObject _
        ("Hyperion.HFMwManageApplications")
    cHFMMangeApps.SetLogonInfoSSO sDomain, sUser, sToken, sPassword
    ' Set the HFMwSession object reference
    Set g_cHFMSession = cHFMMangeApps.OpenApplication (sCluster, _
        sApp)
End Sub
```

RegisterApplicationCAS

Registers a Classic application with Shared Services.

Caution! The method will fail if used upon a Performance Management Architect application.

Syntax

```
<HFMMwManageApplications>.RegisterApplicationCAS bstrClusterName,
bstrProjectName, bstrAppName, bstrHFMMWebServerURL
```

Argument	Description
<i>bstrClusterName</i>	The name of the application's server cluster. Input argument. String subtype.
<i>bstrProjectName</i>	The name of the Shared Services provisioning project for the application. Input argument. String subtype.
<i>bstrAppName</i>	The name of the application. Input argument. String subtype.
<i>bstrHFMMWebServerURL</i>	The URL of the virtual directory for Financial Management. The URL should include the protocol, Web server name and port, and virtual directory name. Input argument. String subtype.

SetLogonInfo

Deprecated - use `SetLogonInfoSSO`.

SetLogonInfoSSO

Sets a user's logon information. To log on, you can specify any of the following security items:

- A Windows NT LAN Manager (NTLM) domain, username, and password
- A Lightweight Directory Access Protocol (LDAP) username and password
- A Microsoft Active Directory (MSAD) username and password
- An external authentication token

Note: To get a token from logon credentials validated by a security agent, use [GetSSOTokenUsingWebSecurityAgentCredentials](#).

The logon information is authenticated when methods that open applications are called.

Syntax

```
<HFMwManageApplications>.SetLogonInfoSSO bstrDomain, bstrUser, bstrToken, bstrPassword
```

Argument	Description
----------	-------------

<i>bstrDomain</i>	Pass one of the following values: <ul style="list-style-type: none">● If NTLM is being used, and if a token is not being used, pass the name of the domain for the username.● If a token is being used, pass an empty string. Input argument. String subtype.
<i>bstrUser</i>	Pass one of the following values: <ul style="list-style-type: none">● If a username and password are being used, pass the username.● If a token is being used, pass an empty string. Input argument. String subtype.
<i>bstrToken</i>	Pass one of the following values: <ul style="list-style-type: none">● If a username and password are being used, pass an empty string.● If a token is being used, pass the token. Input argument. String subtype.
<i>bstrPassword</i>	Pass one of the following values: <ul style="list-style-type: none">● If a username and password are being used, pass the password.● If a token is being used, pass an empty string. Input argument. String subtype.

Example

SetLogonInfoSSO is used in the example for [OpenApplication](#).

UpdateUserAppPreferences

For internal use.

ValidateTokenOnClusterSSO

Indicates whether an external authentication token is valid, and returns the validated token. You can also use `ValidateTokenOnClusterSSO` to obtain a token from a logged-on user's password.

Tip: You can use the token returned by `ValidateTokenOnClusterSSO` to replace an expired token.

Caution! The user's logon information must have been set with `SetLogonInfoSSO` before `ValidateTokenOnClusterSSO` is called.

Syntax

```
<HFMwManageApplications>.ValidateTokenOnClusterSSO (bstrCluster,
bstrSessionPassword, pvbstrToken)
```

Argument	Description
<i>bstrCluster</i>	The name of the application server cluster to which the user is logged on. Input argument. String subtype.
<i>bstrSessionPassword</i>	To obtain a token from a user's password, pass the password; otherwise, pass a blank string. Input argument. String subtype.
<i>pvbstrToken</i>	Pass the token to be validated, or a variable if you want to obtain the token from the password. If the token or password is valid, an updated token is returned. Input/output argument.

Return Value

Indicates whether the token is valid. Returns TRUE if the token is valid, FALSE if it is invalid.

WarnUsersForShutDown

For internal use.

HFMwManageApplications Component: Properties

The HFMwManageApplications component contains the properties described in the following topics.

domain

Returns the name of the current user's domain.

Read-only.

userName

Returns the username of the current user.

Read-only.

4

HFMwSession Type Library

In This Chapter

HFMwSession Component: Methods	53
HFMwSession Component: Properties	58

The HFMwSession type library contains the HFMwSession component. Instances of this component represent a user's connection to a Financial Management application.

In addition, the HFMwSession component provides methods and properties that work with user preferences. For example, the HFMwSession component enables you to change the user languages, thousands separator, and decimal separator. In addition, the component provides methods that get, set and remove Point of View settings identified by uniquely named tags. The component also contains properties that return information about the current Financial Management session, such as the username of the currently connected user.

Obtaining an HFMwSession Object Reference

HFMwSession object references are returned by [OpenApplication](#). The HFMwSession methods and properties apply to the application opened with [OpenApplication](#).

HFMwSession Component: Methods

The HFMwSession component contains the methods described in the following topics.

ApplyUserSettings

Saves user preferences that were changed in the current session. The method includes a Boolean argument that specifies whether the language setting should also be saved.

Syntax

```
<HFMwSession>.ApplyUserSettings vbSaveLanguage
```

Argument	Description
----------	-------------

<i>vbSaveLanguage</i>	Specifies whether the current language setting should be saved. Pass TRUE to save the current language setting, FALSE otherwise.
-----------------------	--

Tip: The [language](#) property gets and sets the current language setting.

Argument	Description
	Input argument.

Example

This function takes a decimal separator of either a comma or a period, then applies the opposite character as the thousands separator.

```
Function setSeparators(sDec)
' This example takes only periods and commas as valid values
' for the parameter. The If structure tests the parameter;
' if some other character is passed, the separators are not
' changed, and FALSE is returned.
If sDec <> "." and sDec <> "," Then
    setSeparators = FALSE
Else
    'g_cHFMSession is an HFMwSession object reference
    g_cHFMSession.decimalSeparator = sDec
    If sDec = "." Then
        g_cHFMSession.thousandsSeparator = ","
    Else
        g_cHFMSession.thousandsSeparator = "."
    End If
    ' If the separators are changed, TRUE is returned.
    g_cHFMSession.applyUserSettings TRUE
    setSeparators = TRUE
End If
End Function
```

GetLicenseExpirationStatus

Deprecated.

GetPOVByTag

Returns an XML representation of the Point of View identified by the specified tag. You can then set the Point of View represented by an HFMwMbrSel object by passing the return value to the HFMwMbrSel component's [XmlString](#) property.

Syntax

```
<HFMwSession>.GetPOVByTag (bstrTag, pvbIsTagTemporary)
```

Argument	Description
<i>bstrTag</i>	The tag that identifies the Point of View. Input argument. String subtype.
<i>pvbIsTagTemporary</i>	Indicates whether the tag has been set to persist past the current session. Returns FALSE if the tag persists, TRUE if it does not persist.

Argument	Description
	Output argument.

Return Value

An XML string that represents the Point of View. If the tag is not found, an empty string is returned.

Example

The following example sets an HFMwMbrSel object to the Point of View stored in a tag named “myTag.”

```
Dim cMbrSel, bIsTemp
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
cMbrSel.xmlString = g_cSession.GetPOVByTag("myTag", bIsTemp)
```

GetUserSetting

Returns the value of the setting identified by a given tag. GetUserSetting returns the value of a custom setting that has been set with [SetUserSetting](#).

Syntax

```
<HFMwSession>.GetUserSetting (bstrTag)
```

Argument Description

<i>bstrTag</i>	The tag that identifies the setting. Input argument. String subtype.
----------------	---

Return Value

Returns the value of the specified setting.

HasSystemChanged

Indicates whether an application’s system information has changed in a way that might require a consolidation, calculation, or translation to be run. For example, HasSystemChanged returns TRUE if a metadata file has been loaded since the last consolidation.

Syntax

```
<HFMwSession>.HasSystemChanged pvbMajorChange, pvbMinorChange
```

Argument	Description
<i>pvbMajorChange</i>	Returns TRUE if the application's system information has changed, FALSE otherwise. Output argument.
<i>pvbMinorChange</i>	<i>For internal use.</i> Output argument.

IsRunningTasks

For internal use.

RemoveFile

Deletes the specified file from the Web server. If the file does not exist, an error is thrown.

Syntax

```
<HFMwSession>.RemoveFile bstrFilename
```

Argument	Description
<i>bstrFilename</i>	The name and path of the file to be deleted. Input argument. String subtype.

RemovePOVByTag

Removes the Point of View identified by the specified tag.

Note: If the tag is invalid for the application, RemovePOVByTag does not throw an error.

Syntax

```
<HFMwSession>.RemovePOVByTag bstrTag
```

Argument	Description
<i>bstrTag</i>	The tag that identifies the Point of View. Input argument. String subtype.

RemoveUserSetting

Removes the setting identified by the specified tag. RemoveUserSetting removes a custom setting that has been defined with [SetUserSetting](#).

Note: If the tag does not exist, `RemoveUserSetting` still succeeds.

Syntax

```
<HFMwSession>.RemoveUserSetting bstrTag
```

Argument	Description
----------	-------------

<i>bstrTag</i>	The tag that identifies the setting to remove. Input argument. String subtype.
----------------	---

SetPOVByTag

Stores a specified Point of View with a tag that identifies it. To persist the tag beyond the current session, you must pass `FALSE` to the *vbIsTagTemporary* argument, then call [ApplyUserSettings](#) before the current session closes.

Syntax

```
<HFMwSession>.SetPOVByTag bstrPOVTag, vbIsTagTemporary, bstrPOV
```

Argument	Description
----------	-------------

<i>bstrPOVTag</i>	A unique tag that identifies the Point of View. Input argument. String subtype.
<i>vbIsTagTemporary</i>	Specifies whether the tag will persist past the current session. Pass <code>FALSE</code> to persist, <code>TRUE</code> if it will not persist. Caution! If you pass <code>FALSE</code> , you must also call <code>ApplyUserSettings</code> to save the tag to the database. Input argument.
<i>bstrPOV</i>	An XML representation of the Point of View to be saved. To obtain an XML representation of a Point of View, use <code>HFMwMbrSel.XmlString</code> . Note: <code>SetPOVByTag</code> does not validate the XML. Input argument. String subtype.

Example

`SetPOVByTag` is used in the example for `HFMwMbrSel.XmlString`.

SetUserSetting

Stores a specified setting with a tag that identifies it. Use `SetUserSetting` to add custom settings to an application.

To persist a setting past the current session, you must call [ApplyUserSettings](#).

Note: To get a setting, use [GetUserSetting](#). To remove a setting, use [RemoveUserSetting](#).

Syntax

```
<HFMwSession>.SetUserSetting bstrPOVTag, bstrPOV
```

Argument	Description
<i>bstrPOVTag</i>	A unique tag that identifies the setting. Input argument. String subtype.
<i>bstrPOV</i>	The setting's value. Input argument. String subtype.

Example

The following subroutine sets and persists the specified setting.

```
Sub SetCustomProperty(sTag, sValue)
' cHFMwSession is a previously set HFMwSession object reference
cHFMwSession.SetUserSetting sTag, sValue
cHFMwSession.applyUserSettings FALSE
End Sub
```

HFMwSession Component: Properties

The HFMwSession component contains the properties described in the following topics.

applicationName

Returns the name of the currently open Financial Management application.

Read-only.

Example

applicationName is used in the example for [EnumDimensions](#).

clusterName

Returns the name of the application server cluster for the current Financial Management session.

Read-only.

decimalSeparator

Gets or sets the decimal separator character for the currently open Financial Management application.

Tip: If you set this property, the decimal separator that you specify persists past the current session only if you pass TRUE to [ApplyUserSettings](#).

Read-write.

Example

For an example, see [ApplyUserSettings](#).

defaultDataGridDef

Gets or sets the default data grid definition for the current user. The data grid definition is represented by an XML string.

Tip: The XML string's structure is the same as that of the string returned by [GetPOVByTag](#) and the HFMwMbrSel component's [XmlString](#) property.

Read-write.

defaultProcessControlGridDef

Gets or sets the default process control grid definition for the current user. The process control grid definition is represented by an XML string.

Read-write.

descriptionLanguage

Gets or sets the language of metadata descriptions for the currently open Financial Management application.

Read-write.

extractEncoding

Gets or sets the user preference that determines the encoding type of files extracted from the application. Valid values are represented by the HFMConstants type library constants listed in [“Extracted File Encoding Constants”](#) on page 452.

Read-write.

Example

The following example sets the application to extract files in the UTF-16 format. Note how [ApplyUserSettings](#) is called to persist the setting.

```
' cHFMSession represents a previously set HFMwSession object.  
cHFMSession.extractEncoding = EXTRACT_FILE_ENCODING_UTF16  
cHFMSession.applyUserSettings FALSE
```

IsBusy

For internal use.

IsDebug

For internal use.

language

Gets or sets the user language for the currently open Financial Management application.

Note: If you set this property, the language that you specify persists past the current session only if you pass TRUE to [ApplyUserSettings](#).

Read-write.

metadata

Returns an HFMwMetadata object reference.

Read-only.

Example

metadata is used in the example for [EnumDimensions](#) (HFMwMetadata component).

resourceManager

Returns an HsvResourceManager object reference.

Read-only.

Note: For information on the HsvResourceManager object, see “[HsvResourceManager Object Methods](#)” on page 392

session

Unsupported – do not use.

showsDescriptions

Gets or sets whether member descriptions are displayed on pages in which the user sets the Point of View. TRUE means the descriptions are displayed, FALSE means they are not displayed.

Read-write.

Example

The following subroutine sets whether member descriptions are displayed and persists the setting. The subroutine's argument is passed to `showsDescriptions`, and then `ApplyUserSettings` persists the setting.

```
Sub persistShowDesc(bShow)
  cHFMSession.showsDescriptions = bShow
  cHFMSession.ApplyUserSettings TRUE
End Sub
```

thousandsSeparator

Gets or sets the thousands separator character for the currently open Financial Management application.

Tip: If you set this property, the separator character that you specify persists past the current session only if you pass TRUE to `ApplyUserSettings`.

Read-write.

Example

For an example, see `ApplyUserSettings`.

userName

Returns the username of the connected user for the current Financial Management session.

Read-only.

usesApplets

Gets or sets the user preference that determines whether applets are used.

Note: If you set this property, the setting does not persist past the current session.

Read-write.

In This Chapter

HFMwMetadata Component	64
HFMwAccounts Component	72
HFMwScenarios Component	73
HFMwYears Component	74
HFMwPeriods Component	75
HFMwEntities Component	78
HFMwValues Component	79
HFMwICPs Component	80
HFMwViews Component	80
HFMwCustom1 Component	81
HFMwCustom2 Component	81
HFMwCustom3 Component	81
HFMwCustom4 Component	82
HFMwDimension Component	82
HFMwCurrencies Component	101

The HFMwMetadata type library contains several components that enable you to work with metadata. The type library contains one component for each Financial Management dimension. The library also contains the HFMwMetadata, HFMwDimension, and HFMwCurrencies components:

- The HFMwMetadata component includes properties that provide object references to the components for the Financial Management dimensions.
- The HFMwDimension component contains methods and properties that are implemented for all of the dimensions.
- The HFMwCurrencies component exposes functionality for Financial Management currencies.

To operate upon a dimension with this type library, you can take the following steps:

- To set object references for the type library's components:

- 1 Set an object reference to the HFMwMetadata component with the `metadata` property of the HFMwSession component:

```
Set cHFMwMetadata = cHFMwSession.metadata
```

- 2 Obtain an object reference to a dimension-specific component with the HFMwMetadata component's properties. For example, the following line sets an object reference for the HFMwAccounts component:

```
Set cHFMAccounts = cHFMMetadata.accounts
```
- 3 Initialize an object reference to the HFMwDimension component by using the `dimension` property of the dimension-specific component. For example, the following line initializes the HFMwDimension component to the Account dimension:

```
Set cDimension = cHFMAccounts.dimension
```

Note: As an alternative, you can create HFMwMetadata objects with `Server.CreateObject` and `SetWebSession`.

HFMMwMetadata Component

The HFMMwMetadata component contains methods that apply to metadata. In addition, the component contains various properties, including properties that provide object references to the components for the Financial Management dimensions.

Obtaining an HFMMwMetadata Object Reference

HFMMwMetadata object references are returned by the `metadata` property of the HFMMwSession component.

HFMMwMetadata Methods

The HFMMwMetadata component contains the methods described in the following topics.

EnableOrDisableUseSecurityAsPartner

Enables or disables metadata security by Entity dimension members' SecurityAsPartner attributes.

By default, metadata security is determined by the security classes assigned to dimension members. However, for the Entity dimension you can use this method to enable and disable an alternate form of security that uses entities' SecurityAsPartner attributes, and not their SecurityClass attributes, to determine access rights.

If you enable SecurityAsPartner attribute security, it is in effect for the session until you disable it.

Syntax

```
<HFMMwMetadata>.EnableOrDisableUseSecurityAsPartner vbEnabled
```

Argument	Description
----------	-------------

<code>vbEnabled</code>	A flag that specifies whether SecurityAsPartner security is enabled. Pass TRUE to enable, FALSE to disable.
------------------------	---

Argument	Description
----------	-------------

	Input argument. Boolean subtype.
--	----------------------------------

EnumDescriptionLanguages

Returns arrays containing the labels and internal IDs of the application's metadata description languages.

Syntax

```
<HFMwMetadata>.EnumDescriptionLanguages (pvaravarLanguageIDs)
```

Argument	Description
----------	-------------

<i>pvaravarLanguageIDs</i>	Returns an array of the metadata description languages' internal IDs. Output argument.
----------------------------	---

Return Value

Returns an array containing the labels of the metadata description languages.

This array has a one-to-one correspondence with the array returned by the *pvaravarLanguageIDs* argument.

EnumDimensions

Enumerate the IDs, names, and aliases of an application's dimensions. These items are returned in arrays, the items of which have a one-to-one correspondence with each other.

Syntax

```
<HFMwMetadata>.EnumDimensions pvaravarlIDs, pvaravarbstrNames,  
pvaravarbstrAliases
```

Argument	Description
----------	-------------

<i>pvaravarlIDs</i>	Returns an array of IDs for the application's dimensions. Valid values are represented by the Dimension ID constants in the HFMConstants type library. For details, see "Dimension ID Constants" on page 414 . Output argument.
---------------------	--

<i>pvaravarbstrNames</i>	Returns an array of names for the application's dimensions. Output argument.
--------------------------	---

<i>pvaravarbstrAliases</i>	Returns an array of aliases for the application's dimensions. If a dimension does not have an alias, the corresponding array item will contain the dimension name. Output argument.
----------------------------	---

Example

This example prints the names and aliases of an application's dimensions to the browser. Note that the application name is also printed, using the HFMwSession property [applicationName](#).

```
Dim cHFMMetadata, vaIDs, vaNames, vaAliases
'g_CHFMwSession is an HFMwSession object reference
Set cHFMMetadata = g_CHFMwSession.metadata
cHFMMetadata.EnumDimensions vaIDs, vaNames, vaAliases
Response.Write "<h1>" & g_CHFMwSession.applicationName & _
    "Application Dimensions and Aliases</h1>"
For i = lBound(vaIDs) to uBound(vaIDs)
    Response.Write "<p><b>" & vaNames(i) & " dimension alias: " & _
        "</b>" & " " & vaAliases(i) & "</p>"
Next
```

IsMemberAChildOf

Indicates whether a specified Member ID is a child of the specified Parent ID.

Syntax

```
<HFMwDimension>.IsMemberAChildOf(lMemberID, lParentID)
```

Argument	Description
<i>lMemberID</i>	The member ID of the Member. Input argument. Long subtype.
<i>lParentID</i>	The parent ID of the Parent. Input argument. Long subtype.

Return Value

Returns TRUE if the Member is a child of the Parent, FALSE otherwise. Boolean subtype.

SetWebSession

Associates an HFMwSession object with an HFMwMetadata object.

However, instead of using `SetWebSession`, Hyperion recommends that you obtain HFMwMetadata object references with the HFMwSession component's [metadata](#) property.

Syntax

```
<HFMwMetadata>.SetWebSession varpIDispHFMwSession
```

Argument	Description
<i>varpIDispHFMwSession</i>	The HFMwSession object to be associated with the HFMwMetadata object. Input argument.

Example

SetWebSession is used in the example for [FindMemberOccurrencesInHierarchy](#).

HFmwMetadata Properties

The HFmwMetadata component contains the properties described in the following topics.

accounts

Returns an object reference for the HFmwAccounts component.

Read-only.

Example

accounts is used in the example for [EnumChildren](#).

ApplicationAttribute

Gets the raw value of a given application setting attribute.

Read-only.

Syntax

```
<HFmwMetadata>.ApplicationAttribute(iAttribute)
```

Argument	Description
<i>iAttribute</i>	The ID of the attribute. Valid values are represented by the HFmConstants enumeration tagAPPSETTING_ATTRIBS, which is described in “Application Setting Attribute ID Constants” on page 420 . Input argument. Integer subtype.

ApplicationCurrency

Returns the label of the application's default currency.

Read-only.

ApplicationSettingsTimeStamp

Returns a timestamp that indicates when the application settings were last updated. Returned as type Double.

Read-only

ApplicationVersion

Returns the application's metadata schema version.

Read-only

ConsolidationMethodsTimeStamp

Returns a timestamp that indicates when the application's consolidation methods were last updated. Returned as type Double.

Read-only

currencies

Returns an object reference for the HFMwCurrencies component.

Read-only.

Example

currencies is used in the example for [GetCurrencyTranslationOperator](#).

CurrencyTimeStamp

Returns a timestamp that indicates when the application's currencies were last updated. Returned as type Double.

Read-only

custom1

Returns an object reference for the HFMwCustom1 component.

Read-only.

Example

custom1 is used in the example for [GetMemberLabel](#).

custom2

Returns an object reference for the HFMwCustom2 component.

Read-only.

custom3

Returns an object reference for the HFMwCustom3 component.

Read-only.

custom4

Returns an object reference for the HFMwCustom4 component.

Read-only.

dimension

Returns an object reference for a component that represents a dimension.

Read-only.

Syntax

```
<HFMwMetadata>.dimension(lDimensionID)
```

Argument	Description
<i>lDimensionID</i>	The ID of the dimension. Pass one of the HFMConstants type library constants listed in "Dimension ID Constants" on page 414 . Input argument. Long subtype.

Example

dimension is used in the example for the HFMwDataGrid component's [maxColsPerPage](#) property.

entities

Returns an object reference for the HFMwEntities component.

Read-only.

Example

entities is used in the example for [EnumMemberLists](#).

ICPs

Returns an object reference for the HFMwICPs component.

Read-only.

IsInputCell

For Oracle Hyperion Financial Data Quality Management, Fusion Edition. Checks if the specified Entity, Account, ICP, C1, C2, C3, C4 are all base members. If all members are base members, then data can be input into the cell.

Read-only

Syntax

```
<HFMwMetadata>.IsInputCell lScenarioId, lYearId, lPeriodId, varalDimIds,  
varalMemberIds
```

Argument	Description
<i>lScenarioId</i>	The member ID of the Scenario dimension member. Input argument. Long subtype.
<i>lYearId</i>	The member ID of the Year dimension member. Input argument. Long subtype.
<i>lPeriodId</i>	The member ID of the Period dimension member. Input argument. Long subtype.
<i>varalDimIds</i>	Array containing the Dimension IDs used for the intersection. Input argument. Long subtype.
<i>varalMemberIds</i>	Array containing the Members IDs used for the intersection which correspond to the array of dimensions. For example, if the dimension IDs represent Entity, Account, ICP, C1, C2, C3, C4, then the member IDs belong to each of those dimensions, respectively. Input argument. Long subtype.

Return Value

Returns True if all specified members are base members; False otherwise.

isOrgByPeriod

Indicates whether the OrgByPeriodApplication setting for the currently open application is on or off. Returns TRUE if the setting is on, FALSE otherwise.

Read-only.

periods

Returns an object reference for the HFMwPeriods component.

Read-only.

scenarios

Returns an object reference for the HFMwScenarios component.

Read-only.

Example

`scenarios` is used in the example for [GetMemberID](#).

submissionPhaseFlag

Indicates whether phased submissions are enabled for the application. Returns TRUE if enabled.

Read-only

timestamp

Returns the timestamp of when the dimension was last updated.

Read-only

ValidationAccount

Returns the member ID of the application's validation account.

Read-only.

Example

The following function returns the label of the validation account.

```
Function GetValAcct()  
Dim cHFMMetadata, cHFMAccts, cHFMDim, lId  
' cHFMSession is a previously set HFMwSession object  
Set cHFMMetadata = cHFMSession.metadata  
Set cHFMAccts = cHFMMetadata.accounts  
Set cHFMDim = cHFMAccts.dimension  
lId = cHFMMetadata.ValidationAccount  
GetValAcct = cHFMDim.GetMemberLabel(lId)  
End Function
```

ValidationAccountByPhase

Returns the member ID of the validation account for a given submission phase.

Read-only

Syntax

```
<HFMwMetadata>.ValidationAccountByPhase lPhaseID
```

Argument	Description
----------	-------------

<i>lPhaseID</i>	The phase ID of the submission phase.
-----------------	---------------------------------------

Input argument. Long subtype.

values

Returns an object reference for the HFMwValues component.

Read-only.

views

Returns an object reference for the HFMwViews component.

Read-only.

years

Returns an object reference for the HFMwYears component.

Read-only.

HFMwAccounts Component

The HFMwAccounts component contains the [GetTopMemberOfCustomsForAccount](#) method and the [dimension](#) property. This component enables you to work with the Account dimension.

Obtaining an HFMwAccounts Object Reference

HFMwAccounts object references are returned by the [accounts](#) property of the HFMwMetadata component. After obtaining an HFMwAccounts object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Account dimension.

GetTopMemberOfCustomsForAccount

Returns the member IDs of the top members of an account's Custom dimension hierarchies. In other words, [GetTopMemberOfCustomsForAccount](#) returns the member IDs of the Custom dimension members that have been defined as an account's Custom1TopMember, Custom2TopMember, Custom3TopMember, and Custom4TopMember attributes.

Tip: To get the label of the Custom dimension member IDs returned, use `HFMwDimension.GetMemberLabel`.

Syntax

```
<HFMwAccounts>.GetTopMemberOfCustomsForAccount varabstrAccount,  
pvarlCusotm1Top, pvarlCusotm2Top, pvarlCusotm3Top, pvarlCusotm4Top
```

Argument	Description
<i>varabstrAccount</i>	The label of the Account dimension member. Input argument.
<i>pvarlCusotm1Top</i>	Returns the member ID of the account's Custom1TopMember attribute. Input/output argument.

Argument	Description
<i>pvarlCusotm2Top</i>	Returns the member ID of the account's Custom2TopMember attribute. Input/output argument.
<i>pvarlCusotm3Top</i>	Returns the member ID of the account's Custom3TopMember attribute. Input/output argument.
<i>pvarlCusotm4Top</i>	Returns the member ID of the account's Custom4TopMember attribute. Input/output argument.

Example

The following function takes an Account member's label and returns the label of the account's Custom1TopMember attribute.

```
Function GetAcctCust1Label(sAcct)
Dim cHFMMetadata, cHFMAccts, cHFMDim, cHFMCust1
Dim lCust1, lCust2, lCust3, lCust4
' cHFMSession is a previously set HFMwSession object
Set cHFMMetadata = cHFMSession.metadata
Set cHFMAccts = cHFMMetadata.accounts
Set cHFMCust1 = cHFMMetadata.custom1
Set cHFMDim = cHFMCust1.dimension
cHFMAccts.GetTopMemberOfCustomsForAccount sAcct, _
    lCust1, lCust2, lCust3, lCust4
GetAcctCust1Label = cHFMDim.GetMemberLabel(lCust1)
End Function
```

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Account dimension.

Read-only.

HFMwScenarios Component

The HFMwScenarios component contains the properties described in the following topics.

This component enables you to work with the Scenario dimension.

Obtaining an HFMwScenarios Object Reference

HFMwScenarios object references are returned by the [scenarios](#) property of the HFMwMetadata component. After obtaining an HFMwScenarios object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Scenario dimension.

defaultFrequency

Returns the ID of the default frequency for a given Scenario dimension member.

Read-only.

Tip: You can get the member IDs of a frequency's Period dimension members with `HFMwPeriods.GetPeriodsInFrequency`.

Syntax

```
<HFMwScenarios>.defaultFrequency varabstrScenario
```

Argument	Description
----------	-------------

<code>varabstrScenario</code>	The label of the Scenario dimension member.
-------------------------------	---

Input argument.

Example

The `defaultFrequency` property is used in the example for `HFMwPeriods.GetPeriodsInFrequency`.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Scenario dimension.

Read-only.

HFMwYears Component

The HFMwYears component contains the `dimension` property. This component enables you to work with the Year dimension.

Obtaining an HFMwYears Object Reference

HFMwYears object references are returned by the `years` property of the HFMwMetadata component. After obtaining an HFMwYears object reference, you can then use the `dimension` property to initialize the HFMwDimension component to the Year dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Year dimension.

Read-only.

HFMwPeriods Component

The HFMwPeriods component enables you to work with the Period dimension. For information on the component's members, see [“HFMwPeriods Component: Methods” on page 75](#) and [“HFMwPeriods Component: Properties” on page 78](#).

Obtaining an HFMwPeriods Object Reference

HFMwPeriods object references are returned by the [periods](#) property of the HFMwMetadata component. After obtaining an HFMwPeriods object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Period dimension.

HFMwPeriods Component: Methods

The HFMwPeriods component contains the methods described in the following topics.

GetAncestorAtFrequency

Returns the label of given Period dimension member's ancestor at the specified frequency of the application's frequency hierarchy.

Syntax

```
<HFMwPeriods>.GetAncestorAtFrequency (bstrPeriod, lFreq)
```

Argument	Description
<i>bstrPeriod</i>	The label of the Period dimension member. The specified period must be at a lower level in the frequency hierarchy than the frequency specified in the <i>lFreq</i> argument. Input argument. String subtype.
<i>lFreq</i>	The internal ID of the frequency that contains the ancestor to be returned. Tip: You can get internal ID of a period's frequency with GetFrequency , and the internal ID of a scenario's default frequency with HFMwScenarios.defaultFrequency . Input argument. Long subtype.

Return Value

Returns the label of the ancestor Period dimension member.

Example

The following function returns the label of a given period's ancestor period at the default frequency of a given scenario. If the specified period is not at a lower level in the frequency hierarchy than the scenario's default frequency, an empty string is returned.

```
Function getAncestorDefScenFreq(sPeriod, sScenario)
Dim cHFMMetadata, cHFMPPeriods, cHFMScenarios
Dim lFrequency, sRet, lDefaultFreq
Set cHFMMetadata = cHFMSession.metadata
```

```

Set cHFMPeriods = cHFMMetadata.periods
Set cHFMScenarios = cHFMMetadata.scenarios
lDefaultFreq = cHFMScenarios.defaultFrequency(sScenario)
lFrequency = cHFMPeriods.GetFrequency(sPeriod)
If lDefaultFreq < lFrequency Then
    sRet = cHFMPeriods.GetAncestorAtFrequency(sPeriod, _
        lDefaultFreq)
Else
    sRet = " "
End If
getAncestorDefScenFreq = sRet
End Function

```

GetDescendantsAtFrequency

Returns an array containing the labels of given Period dimension member's descendants at the specified frequency of the application's frequency hierarchy.

Syntax

```
<HFmwPeriods>.GetDescendantsAtFrequency (bstrPeriod, lFreq)
```

Argument Description

bstrPeriod The label of the Period dimension member. The specified period must be at a higher level in the frequency hierarchy than the frequency specified in the *lFreq* argument.

Input argument. String subtype.

lFreq The internal ID of the frequency that contains the descendants to be returned.

Tip: You can get internal ID of a period's frequency with [GetFrequency](#), and the internal ID of a scenario's default frequency with `HFmwScenarios.defaultFrequency`.

Input argument. Long subtype.

Return Value

Returns an array containing the labels of the descendants.

Example

The following function returns the labels of a given period's descendant periods at the default frequency of a given scenario. If the specified period is not at a higher level in the frequency hierarchy than the scenario's default frequency, an empty string is returned.

```

Function getDescendantsDefScenFreq(sPeriod, sScenario)
Dim cHFMMetadata, cHFMPeriods, cHFMScenarios
Dim lFrequency, sRet, lDefaultFreq
Set cHFMMetadata = cHFMSession.metadata
Set cHFMPeriods = cHFMMetadata.periods
Set cHFMScenarios = cHFMMetadata.scenarios
lDefaultFreq = cHFMScenarios.defaultFrequency(sScenario)
lFrequency = cHFMPeriods.GetFrequency(sPeriod)
If lDefaultFreq > lFrequency Then
    sRet = cHFMPeriods.GetDescendantsAtFrequency(sPeriod, _
        lDefaultFreq)

```

```

Else
    sRet = ""
End If
getDescendantsDefScenFreq = sRet
End Function

```

GetFrequency

Returns the internal ID of the frequency to which a given Period dimension member belongs.

Syntax

```
<HFMwPeriods>.GetFrequency (bstrPeriod)
```

Argument Description

bstrPeriod The label of the Period dimension member.
Input argument. String subtype.

Return Value

Returns the internal ID of the period's frequency.

GetPeriodsInFrequency

Returns an array containing the member IDs of a given frequency's Period dimension members.

Syntax

```
<HFMwPeriods>.GetPeriodsInFrequency varlFrequencyID,  
varlFlagsRequestedInfo, pVal
```

Argument

Description

varlFrequencyID

The ID of the frequency.

Tip: You can get internal ID of a period's frequency with [GetFrequency](#), and the internal ID of a scenario's default frequency with `HFMwScenarios.defaultFrequency`.

Input argument.

varlFlagsRequestedInfo

Future use. You must specify a value for this argument, but the value currently has no effect.

Input argument.

pVal

Returns an array containing the member IDs of the frequency's Period dimension members.

Input/output argument.

Example

The following function takes a Scenario dimension member's label and returns the labels of the Period dimension members for the scenario's default frequency. The example gets the scenario's default frequency with `HFMwScenarios.defaultFrequency`.

```

Function GetScenDefFreq(sScen)
Dim cHFMMetadata, cHFMScen, cHFMPer, cHFMDim
Dim lId, laPers, saPers()
' cHFMSession is a previously set HFMwSession object
Set cHFMMetadata = cHFMSession.metadata
Set cHFMScen = cHFMMetadata.scenarios
Set cHFMPer = cHFMMetadata.periods
Set cHFMDim = cHFMPer.dimension
lId = cHFMScen.defaultFrequency(sScen)
cHFMPer.GetPeriodsInFrequency lId, 0, laPers
ReDim saPers(uBound(laPers))
For i = lBound(laPers) to uBound(laPers)
    saPers(i) = cHFMDim.GetMemberLabel(laPers(i))
Next
GetScenDefFreq = saPers
End Function

```

HF MwPeriods Component: Properties

The HFMwPeriods component contains the [dimension](#) property.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Period dimension.

Read-only.

HF MwEntities Component

The HFMwEntities component contains the [dimension](#) property and the [GetDefaultCurrenciesForEntities](#) method. This component enables you to work with the Entity dimension.

Obtaining an HFMwEntities Object Reference

HF MwEntities object references are returned by the [entities](#) property of the HFMwMetadata component. After obtaining an HFMwEntities object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Entity dimension.

GetDefaultCurrenciesForEntities

Returns the labels of the default currencies for the specified Entity dimension members.

Syntax

```
<HF MwEntities>.GetDefaultCurrenciesForEntities (avarlEntityIDs)
```

Argument	Description
----------	-------------

<i>avarEntityIDs</i>	An array containing the member IDs of the Entity dimension members. Input argument.
----------------------	--

Return Value

Returns the labels of the entities' default currencies. This array has a one-to-one correspondence with the array passed to the argument.

Example

The following function returns an array of default currencies for the specified Entity dimension members. The function takes the labels of the Entity members, using HFMwDimension.[GetMemberID](#) to obtain the member IDs passed to [GetDefaultCurrenciesForEntities](#).

```
Function getEntCurrs(vaLabels)
Dim cMetadata, cEntities, cDimension, vaIDs()
'g_cSession is an HFMwSession object reference
Set cMetadata = g_cSession.metadata
Set cEntities = cMetadata.entities
Set cDimension = cEntities.dimension
Redim vaIDs(UBound(vaLabels))
For i = LBound(vaLabels) to UBound(vaLabels)
    vaIDs(i) = cDimension.GetMemberID(vaLabels(i))
Next
getEntCurrs = cEntities.GetDefaultCurrenciesForEntities(vaIDs)
End Function
```

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Entity dimension.

Read-only.

HFMwValues Component

The HFMwValues component contains the [dimension](#) property. This component enables you to work with the Value dimension.

Obtaining an HFMwValues Object Reference

HFMwValues object references are returned by the [values](#) property of the HFMwMetadata component. After obtaining an HFMwValues object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Value dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Value dimension.

Read-only.

HFMwICPs Component

The HFMwICPs component contains the [dimension](#) property. This component enables you to work with the Intercompany Partner dimension.

Obtaining an HFMwICPs Object Reference

HFMwICPs object references are returned by the [ICPs](#) property of the HFMwMetadata component. After obtaining an HFMwICPs object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Intercompany Partner dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Intercompany Partner dimension.

Read-only.

HFMwViews Component

The HFMwViews component contains the [dimension](#) property. This component enables you to work with the View dimension.

Obtaining an HFMwViews Object Reference

HFMwViews object references are returned by the [views](#) property of the HFMwMetadata component. After obtaining an HFMwViews object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the View dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the View dimension.

Read-only.

HF MwCustom1 Component

The HF MwCustom1 component contains the `dimension` property. This component enables you to work with the Custom 1 dimension.

Obtaining an HF MwCustom1 Object Reference

HF MwCustom1 object references are returned by the `custom1` property of the HF MwMetadata component. After obtaining an HF MwCustom1 object reference, you can then use the `dimension` property to initialize the HF MwDimension component to the Custom 1 dimension.

`dimension`

Returns an object reference to the HF MwDimension component, initializing the object so that it operates upon the Custom 1 dimension.

Read-only.

HF MwCustom2 Component

The HF MwCustom2 component contains the `dimension` property. This component enables you to work with the Custom 2 dimension.

Obtaining an HF MwCustom2 Object Reference

HF MwCustom2 object references are returned by the `custom2` property of the HF MwMetadata component. After obtaining an HF MwCustom2 object reference, you can then use the `dimension` property to initialize the HF MwDimension component to the Custom 2 dimension.

`dimension`

Returns an object reference to the HF MwDimension component, initializing the object so that it operates upon the Custom 2 dimension.

Read-only.

HF MwCustom3 Component

The HF MwCustom3 component contains the `dimension` property. This component enables you to work with the Custom 3 dimension.

Obtaining an HF MwCustom3 Object Reference

HF MwCustom3 object references are returned by the `custom3` property of the HF MwMetadata component. After obtaining an HF MwCustom3 object reference, you can then use the `dimension` property to initialize the HF MwDimension component to the Custom 3 dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Custom 3 dimension.

Read-only.

HFMwCustom4 Component

The HFMwCustom4 component contains the [dimension](#) property. This component enables you to work with the Custom 4 dimension.

Obtaining an HFMwCustom4 Object Reference

HFMwCustom4 object references are returned by the [custom4](#) property of the HFMwMetadata component. After obtaining an HFMwCustom4 object reference, you can then use the [dimension](#) property to initialize the HFMwDimension component to the Custom 4 dimension.

dimension

Returns an object reference to the HFMwDimension component, initializing the object so that it operates upon the Custom 4 dimension.

Read-only.

HFMwDimension Component

The HFMwDimension component contains methods and properties that you can apply to all of the Financial Management dimensions. For example, the component includes methods that return all of a dimension member's children, all the member lists for a dimension, member IDs, and member names.

Obtaining an HFMwDimension Object Reference

HFMwDimension object references are returned by the [dimension](#) properties of the components that represent the various Financial Management dimensions. The component from which you obtain the HFMwDimension object reference determines the dimension upon which the HFMwDimension members will operate. For example, to operate upon the Account dimension, you would obtain the HFMwDimension object reference from the HFMwAccounts component's [dimension](#) property

HFMwDimension Methods

The HFMwDimension component contains the methods described in the following topics.

EnumChildren

Returns an array of the member IDs, labels, descriptions, or number of children of the specified dimension member's children. A flag specifies which of these types of information will be returned.

Syntax

```
<HFMwDimension>.EnumChildren (varOBPScenario, varOBPYear, varOBPPeriod,  
varParentMember, lStartingIndex, lMaxMembers, lFlagsRequestedInfo,  
pvaravarlMemberIDs, pvaravarbstrMemberLabels, pvaravarbstrDescriptions,  
pvaravarlNumChildren, pvarlTotalMembersInEnum)
```

Argument	Description
<i>varOBPScenario</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Scenario dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPYear</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Year dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPPeriod</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Period dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varParentMember</i>	<p>Specifies the member for which you want to return children. You can pass either a member ID or a member label.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>lStartingIndex</i>	<p>Specifies the starting index (base 0) of the child members to return from the total set of the parent's children. You can use this and the <i>lMaxMembers</i> argument to return subsets of children with iterative calls.</p> <p>Caution! On the first call to <code>EnumChildren</code> for a parent, this argument defaults to 0, meaning that any non-zero value will be ignored by the first call.</p> <p>Input argument. Long subtype.</p>
<i>lMaxMembers</i>	<p>Specifies the maximum number of members to return. To return all child members, pass 0.</p>

Argument	Description
	To return a subset of the parent's children, use this argument with the <i>lStartingIndex</i> argument. For example, suppose a parent entity has 1000 children, and you want to return 100 at a time. You could use a loop that makes iterative calls to <i>EnumChildren</i> , passing 100 to <i>lMaxMembers</i> and incrementing <i>lStartingIndex</i> by 100 in each call.
	Input argument. Long subtype.
<i>lFlagsRequestedInfo</i>	Specifies the type of information to return. Valid values are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450 . You can pass more than one constant by using <i>Or</i> .
	Input argument. Long subtype.
<i>pvaravarlMemberIDs</i>	If <i>WEBOM_METADATA_INFO_ID</i> or <i>WEBOM_METADATA_INFO_ALL</i> is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array of the member IDs of the children.
	Output argument.
<i>pvaravarbstrMemberLabels</i>	If <i>WEBOM_METADATA_INFO_LABEL</i> or <i>WEBOM_METADATA_INFO_ALL</i> is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array of the labels of the children.
	Output argument.
<i>pvaravarbstrDescriptions</i>	If <i>WEBOM_METADATA_INFO_DESCRIPTION</i> or <i>WEBOM_METADATA_INFO_ALL</i> is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array of the descriptions of the children.
	Output argument.
<i>pvaravarlNumChildren</i>	If <i>WEBOM_METADATA_INFO_NUMCHILDREN</i> or <i>WEBOM_METADATA_INFO_ALL</i> is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array indicating the number of child members for each enumerated child.
	Output argument.
<i>pvarlTotalMembersInEnum</i>	Returns the total number of the parent's children.
	Output argument.

Return Value

Returns a count of the members that are returned by this method.

Example

The following subroutine takes the name of a parent entity and the number that gets passed to the *lMaxMembers* argument, and prints out the labels of the child entities returned in each call to *EnumChildren*. The example shows how to use the *lStartingIndex* and *lMaxMembers* arguments.

```
Sub getChildEntityLabels(sPar, lNum)
Dim lIndex, lNumReturned, vaIDs, vaLabels, vaDescs
Dim vaNumChildren, lTotalEnum, cHFMMetadata, cHFMEntities
Dim cHFMDimension, lCounter
'g_cHFMSession is an HFMwSession object reference
Set cHFMMetadata = g_cHFMSession.metadata
Set cHFMEntities = cHFMMetadata.entities
Set cHFMDimension = cHFMEntities.dimension
lCounter = 0
```

```

lIndex = 0
Do
    lNumReturned = cHFMDimension.EnumChildren(CLng(-1), _
        CLng(-1), CLng(-1), sPar, lIndex, lNum, _
        WEBOM_METADATA_INFO_LABEL, vaIDs, vaLabels, vaDescs, _
        vaNumChildren, lTotalEnum)
    If (lNumReturned > 0) Then
        ' Print an <h1> that shows the ordinal #s of the entities
        ' within the set of child entities
        Response.Write "<h1>Child Entities # " & lCounter + 1 & _
            " through # " & lCounter + lNumReturned & "</h1>"
        For i = 0 to uBound(vaLabels)
            Response.Write "<p>" & vaLabels(i) & "</p>"
        Next
        ' Increment the index for the lStartingIndex argument
        lIndex = lIndex + lNum
    End If
    ' lCounter counts the number of items that have been
    ' returned by the calls to EnumChildren.
    lCounter = lCounter + lNumReturned
    ' If the # of child entities returned is less than the number
    ' passed to the lMaxMembers argument, then all the children
    ' have been enumerated, so exit the loop
    If lNum = 0 Or lNumReturned < lNum then Exit Do
Loop
End Sub

```

EnumChildrenAsXML

Returns an XML string of the member IDs, labels, descriptions, or number of children of the specified dimension member's children. A flag specifies which of these types of information will be returned.

Syntax

```

<HFMwDimension>.EnumChildrenAsXML (varOBPSscenario, varOBPYear,
varOBPPeriod, varParentMember, lStartingIndex, lMaxMembers,
lFlagsRequestedInfo, pvarbstrXMLDoc, pvarlTotalMembersInEnum)

```

Argument	Description
<i>varOBPSscenario</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> For Organization by Period enumerations, pass the member ID or label of the Scenario dimension member. For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPYear</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> For Organization by Period enumerations, pass the member ID or label of the Year dimension member. For non-Organization by Period enumerations, pass -1.

Argument	Description
	<p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPPeriod</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● For Organization by Period enumerations, pass the member ID or label of the Period dimension member. ● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varParentMember</i>	<p>Specifies the member for which you want to return children. You can pass either a member ID or a member label.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>lStartingIndex</i>	<p>Specifies the starting index (base 0) within the enumeration for retrieving member information.</p> <p>Input argument. Long subtype.</p>
<i>lMaxMembers</i>	<p>Specifies the maximum number of members to return. To return all child members, pass 0.</p> <p>Input argument. Long subtype.</p>
<i>lFlagsRequestedInfo</i>	<p>Specifies the type of information to return. Valid values are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450. You can pass more than one constant by using <code>Or</code>.</p> <p>Input argument. Long subtype.</p>
<i>pvarbstrXMLDoc</i>	<p>Returns an XML string containing the requested information. The XML string is described in “Dimension Member Enumerations DTD” on page 474.</p> <p>Output argument.</p>
<i>pvarlTotalMembersInEnum</i>	<p>Returns the total number of the parent’s child members.</p> <p>Output argument.</p>

Return Value

Returns the number of child members in the XML string that is returned by the *pvarbstrXMLDoc* argument.

EnumMemberLists

Returns an array that enumerates the member lists available for a dimension. A flag specifies whether names or IDs of the member lists are returned.

Syntax

```
<HFMwDimension>.EnumMemberLists (lFlagsRequestedInfo, pvaravarlIDs,
pvaravarbstrListNames)
```

Argument	Description
<i>lFlagsRequestedInfo</i>	<p>Specifies whether names or IDs of member lists are returned. Pass any of the following HFMConstants type library constants (for descriptions, see “Metadata Information Constants” on page 450):</p> <ul style="list-style-type: none"> ● WEBOM_METADATA_INFO_ID ● WEBOM_METADATA_INFO_LABEL ● WEBOM_METADATA_INFO_ALL <p>Tip: You can pass more than one constant by using <code>Or</code>.</p> <p>Input argument. Long subtype.</p>
<i>pvaravrlIDs</i>	<p>If WEBOM_METADATA_INFO_ID or WEBOM_METADATA_INFO_ALL is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array of the member list IDs.</p> <p>Output argument.</p>
<i>pvaravrbstrListNames</i>	<p>If WEBOM_METADATA_INFO_LABEL or WEBOM_METADATA_INFO_ALL is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array of the member list names.</p> <p>Output argument.</p>

Return Value

Returns a count of the member lists for the dimension.

Example

The following function returns the names of the member lists for an application’s Entity dimension.

```
Function getEntityMemberLists()
Dim cHFMMetadata, cHFMEntities, cDimension, lCount
Dim vaIDs, vaListNames
Set cHFMMetadata = cHFMSession.metadata
Set cHFMEntities = cHFMMetadata.entities
Set cDimension = cHFMEntities.dimension
lCount = cDimension.EnumMemberLists(WEBOM_METADATA_INFO_LABEL, _
vaIDs, vaListNames)
getEntityMemberLists = vaListNames
End Function
```

EnumMemberListsAsXML

Returns an XML string that enumerates the member lists available for a dimension. A flag specifies whether the names or the IDs of the member lists are returned.

Syntax

```
<HFMwDimension>.EnumMemberListsAsXML (lFlagsRequestedInfo, pvarbstrXMLDoc)
```

Argument	Description
<i>lFlagsRequestedInfo</i>	<p>Specifies whether names or IDs of member lists are returned. Pass one of the following HFMConstants type library constants (for descriptions, see “Metadata Information Constants” on page 450):</p> <ul style="list-style-type: none"> ● WEBOM_METADATA_INFO_ID ● WEBOM_METADATA_INFO_LABEL ● WEBOM_METADATA_INFO_ALL <p>Tip: You can pass more than one constant by using <code>Or</code>.</p> <p>Input argument. Long subtype.</p>
<i>pvarbstrXMLDoc</i>	<p>Returns an XML string that contains the requested information. The XML string is described in “Member List Enumerations DTD” on page 473.</p> <p>Output argument.</p>

Return Value

Returns a count of the member lists for the dimension.

EnumMembers

Enumerates the member IDs, labels, descriptions, and number of children of the members in a static member list. A flag specifies which of these types of information will be returned.

Note: You can return members from a dynamic member list with [EnumMembers2](#).

Syntax

```
<HFMwDimension>.EnumMembers (varOBPScenario, varOBPYear, varOBPPeriod,
varMemberlist, varTopMember, lStartingIndex, lMaxMembers,
lFlagsRequestedInfo, pvaravarlMemberIDs, pvaravarlParentIDs,
pvaravarbstrMemberLabels, pvaravarbstrDescriptions, pvaravarlNumChildren,
pvarlTotalMembersInEnum)
```

Argument	Description
<i>varOBPScenario</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● For Organization by Period enumerations, pass the member ID or label of the Scenario dimension member. An ID must be cast as a Long subtype. ● For non-Organization by Period enumerations, pass the HFMConstants type library constant <code>MEMBERNOTUSED</code>. <p>Input argument.</p>
<i>varOBPYear</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● For Organization by Period enumerations, pass the member ID or label of the Year dimension member. An ID must be cast as a Long subtype. ● For non-Organization by Period enumerations, pass the HFMConstants type library constant <code>MEMBERNOTUSED</code>.

Argument	Description
	Input argument.
<i>varOBPPeriod</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● For Organization by Period enumerations, pass the member ID or label of the Period dimension member. An ID must be cast as a Long subtype. ● For non-Organization by Period enumerations, pass the HFMConstants type library constant <code>MEMBERNOTUSED</code>. <p>Input argument.</p>
<i>varMemberlist</i>	<p>The ID or label of the member list.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varTopMember</i>	<p>The member ID or label of the top member in the hierarchy with which to begin enumerating. To start enumerating at the top of the hierarchy, pass either -1 or an empty string.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>lStartingIndex</i>	<p>The starting index (base 0) within the enumeration for retrieving member information.</p> <p>Input argument. Long subtype.</p>
<i>lMaxMembers</i>	<p>The maximum number of members to return. To return all child members, pass 0.</p> <p>Input argument. Long subtype.</p>
<i>lFlagsRequestedInfo</i>	<p>Specifies the type of information to return. Valid values for the bits are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450.</p> <p>Input argument. Long subtype.</p>
<i>pvaravarlMemberIDs</i>	<p>If <code>WEBOM_METADATA_INFO_ID</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument’s bitmask, this returns an array of member IDs.</p> <p>Output argument.</p>
<i>pvaravarlParentIDs</i>	<p>If <code>EnumMembers</code> is called for the entity dimension, and <code>WEBOM_METADATA_INFO_ID</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument’s bitmask, this returns an array of member IDs for the parents of the list’s members.</p> <p>Output argument.</p>
<i>pvaravarbstrMemberLabels</i>	<p>If <code>WEBOM_METADATA_INFO_LABEL</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument’s bitmask, this returns an array of the labels of the list’s members.</p> <p>Output argument.</p>
<i>pvaravarbstrDescriptions</i>	<p>If <code>WEBOM_METADATA_INFO_DESCRIPTION</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument’s bitmask, this returns an array of the descriptions of the list’s members.</p> <p>Output argument.</p>

Argument	Description
<i>pvaravarlNumChildren</i>	If <code>WEBOM_METADATA_INFO_NUMCHILDREN</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array indicating the number of child members for each enumerated member. Output argument.
<i>pvarlTotalMembersInEnum</i>	Returns the total number of members in the list. Output argument.

Return Value

Returns a count of the members returned by this method.

Example

The following function returns the names of the entities that belong to a member list.

```
Function getEntitiesInMemList(sList)
Dim cMetadata, cEntities, cDimension, lCount
Dim vaIDs, vaParIDs, vaLabels, vaDescs, vaKids, lTotal
'g_cSession is an HFMwSession object reference
Set cMetadata = g_cSession.metadata
Set cEntities = cMetadata.entities
Set cDimension = cEntities.dimension
lCount = cDimension.EnumMembers(MEMBERNOTUSED, MEMBERNOTUSED, _
    MEMBERNOTUSED, sList, "", 0, 0, WEBOM_METADATA_INFO_LABEL, _
    vaIDs, vaParIDs, vaLabels, vaDescs, vaKids, lTotal)
getEntitiesInMemList = vaLabels
End Function
```

EnumMembers2

Enumerates the member IDs, labels, descriptions, and number of children of the members in a static or dynamic member list. A flag specifies which of these types of information will be returned.

Note: You can also return members from a static member list with [EnumMembers](#), which does not require you to pass labels or member IDs of dimension members.

Syntax

```
<HFMwDimension>.EnumMembers2 (varPOVScenario, varPOVYear, varPOVPeriod,
varPOVEntity, varMemberlist, varTopMember, lStartingIndex, lMaxMembers,
lFlagsRequestedInfo, pvaravarlMemberIDs, pvaravarlParentIDs,
pvaravarbstrMemberLabels, pvaravarbstrDescriptions, pvaravarlNumChildren,
pvarlTotalMembersInEnum)
```

Argument	Description
<i>varPOVScenario</i>	The member ID or label of the Scenario dimension member.

Argument	Description
	<p>Tip: If you are returning members of a static member list, you can pass the HFMConstants type library constant <code>MEMBERNOTUSED</code> to this argument and to the <code>varPOVYear</code>, <code>varPOVPeriod</code>, and <code>varPOVEntity</code> arguments.</p> <p>Input argument.</p>
<code>varPOVYear</code>	<p>The member ID or label of the Year dimension member.</p> <p>Input argument.</p>
<code>varPOVPeriod</code>	<p>The member ID or label of the Period dimension member.</p> <p>Input argument.</p>
<code>varPOVEntity</code>	<p>The member ID or label of the Entity dimension member.</p> <p>Input argument.</p>
<code>varMemberlist</code>	<p>The ID or label of the member list.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<code>varTopMember</code>	<p>The member ID or label of the top member in the hierarchy with which to begin enumerating. To start enumerating at the top of the hierarchy, pass either -1 or an empty string.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<code>lStartingIndex</code>	<p>The starting index (base 0) within the enumeration for retrieving member information.</p> <p>Input argument. Long subtype.</p>
<code>lMaxMembers</code>	<p>The maximum number of members to return. To return all child members, pass 0.</p> <p>Input argument. Long subtype.</p>
<code>lFlagsRequestedInfo</code>	<p>A bitmask that specifies the type of information to return. Valid values for the bits are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450.</p> <p>Input argument. Long subtype.</p>
<code>pvaravarlMemberIDs</code>	<p>If <code>WEBOM_METADATA_INFO_ID</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <code>lFlagsRequestedInfo</code> argument’s bitmask, this returns an array of member IDs.</p> <p>Input/output argument.</p>
<code>pvaravarlParentIDs</code>	<p>If <code>EnumMembers2</code> is called for the entity dimension, and <code>WEBOM_METADATA_INFO_ID</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <code>lFlagsRequestedInfo</code> argument’s bitmask, this returns an array of member IDs for the parents of the list’s members.</p> <p>Input/output argument.</p>
<code>pvaravarbstrMemberLabels</code>	<p>If <code>WEBOM_METADATA_INFO_LABEL</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <code>lFlagsRequestedInfo</code> argument’s bitmask, this returns an array of the labels of the list’s members.</p> <p>Input/output argument.</p>

Argument	Description
<i>pvaravarbstrDescriptions</i>	If <code>WEBOM_METADATA_INFO_DESCRIPTION</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array of the descriptions of the list's members. Input/output argument.
<i>pvaravarlNumChildren</i>	If <code>WEBOM_METADATA_INFO_NUMCHILDREN</code> or <code>WEBOM_METADATA_INFO_ALL</code> is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array indicating the number of child members for each enumerated member. Input/output argument.
<i>pvarlTotalMembersInEnum</i>	Returns the total number of members in the list. Input/output argument.

Return Value

Returns a count of the members returned by this method.

EnumMembersAsXML

Returns an XML string containing the member IDs, labels, descriptions, or number of children of the members in a specified member list. A flag specifies which of these types of information will be returned.

Syntax

```
<HFMwDimension>.EnumMembersAsXML (varOBPScenario, varOBPYear,
varOBPPeriod, varMemberlist, varTopMember, lStartingIndex, lMaxMembers,
lFlagsRequestedInfo, pvarbstrXMLDoc, pvarlTotalMembersInEnum)
```

Argument	Description
<i>varOBPScenario</i>	Pass one of the following: <ul style="list-style-type: none"> For Organization by Period enumerations, pass the member ID or label of the Scenario dimension member. For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPYear</i>	Pass one of the following: <ul style="list-style-type: none"> For Organization by Period enumerations, pass the member ID or label of the Year dimension member. For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPPeriod</i>	Pass one of the following:

Argument	Description
	<ul style="list-style-type: none"> For Organization by Period enumerations, pass the member ID or label of the Period dimension member. For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varMemberlist</i>	<p>Specifies the member list for which members will be enumerated. You can pass either the ID or the name of the member list.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varTopMember</i>	<p>Specifies the top member in the hierarchy with which to begin enumerating. You can pass either the member ID or name of the member. To start enumerating at the top of the hierarchy, pass either -1 or an empty string.</p> <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>lStartingIndex</i>	<p>Specifies the starting index (base 0) within the enumeration for retrieving member information.</p> <p>Input argument. Long subtype.</p>
<i>lMaxMembers</i>	<p>Specifies the maximum number of members to return. To return all child members, pass 0.</p> <p>Input argument. Long subtype.</p>
<i>lFlagsRequestedInfo</i>	<p>Specifies the type of information to return. Valid values are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450. You can pass more than one constant by using <code>Or</code>.</p> <p>Input argument. Long subtype.</p>
<i>pvarbstrXMLDoc</i>	<p>Returns an XML String containing the requested information. The XML string is described in “Dimension Member Enumerations DTD” on page 474.</p> <p>Output argument.</p>
<i>pvarlTotalMembersInEnum</i>	<p>Returns the total number of members in the list.</p> <p>Output argument.</p>

Return Value

Returns the number of members in the XML string that is returned by the *pvarbstrXMLDoc* argument.

EnumParents

Enumerates the member IDs, labels, descriptions, and number of children of the parent members of a given child member. A flag specifies which of these types of information will be returned.

Syntax

```
<HFMwDimension>.EnumParents (varOBPSscenario, varOBPYear, varOBPPeriod,  
varChildMember, lStartingIndex, lMaxMembers, lFlagsRequestedInfo,  
pvaravarlMemberIDs, pvaravarbstrMemberLabels, pvaravarbstrDescriptions,  
pvaravarlNumParents, pvarlTotalMembersInEnum)
```

Argument	Description
<i>varOBPSscenario</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Scenario dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPYear</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Year dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varOBPPeriod</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none">● For Organization by Period enumerations, pass the member ID or label of the Period dimension member.● For non-Organization by Period enumerations, pass -1. <p>Note: An ID must be cast as a Long subtype.</p> <p>Input argument.</p>
<i>varChildMember</i>	<p>The label or member ID of the dimension member for which the parent labels are being returned.</p> <p>Input argument.</p>
<i>lStartingIndex</i>	<p>Specifies the starting index (base 0) within the enumeration for retrieving parent information.</p> <p>Input argument. Long subtype.</p>
<i>lMaxMembers</i>	<p>Specifies the maximum number of members to return. To return all parent members, pass 0.</p> <p>Input argument. Long subtype.</p>
<i>lFlagsRequestedInfo</i>	<p>Specifies the type of information to return. Valid values are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450. You can pass more than one constant by using <code>OR</code>.</p> <p>Input argument. Long subtype.</p>
<i>pvaravarlMemberIDs</i>	<p>If <code>WEBOM_METADATA_INFO_ID</code> or <code>WEBOM_METADATA_INFO_ALL</code> is passed to the <i>lFlagsRequestedInfo</i> argument, this returns an array containing the parent entities' member IDs.</p>

Argument	Description
	Input/output argument.
<i>pvaravbstrMemberLabels</i>	If <code>WEBOM_METADATA_INFO_LABEL</code> or <code>WEBOM_METADATA_INFO_ALL</code> is passed to the <i>IFlagsRequestedInfo</i> argument, this returns an array containing the labels of the parent entities. Input/output argument.
<i>pvaravbstrDescriptions</i>	If <code>WEBOM_METADATA_INFO_DESCRIPTION</code> or <code>WEBOM_METADATA_INFO_ALL</code> is passed to the <i>IFlagsRequestedInfo</i> argument, this returns an array containing the descriptions of the parent entities. Input/output argument.
<i>pvaravlNumParents</i>	If <code>WEBOM_METADATA_INFO_NUMCHILDREN</code> or <code>WEBOM_METADATA_INFO_ALL</code> is passed to the <i>IFlagsRequestedInfo</i> argument, this returns an array indicating the number of child members for each enumerated member. Input/output argument.
<i>pvarlTotalMembersInEnum</i>	Returns the total number of members in the enumeration. Input/output argument.

Return Value

Returns a count of the members that are enumerated by this method.

Example

The following function returns the labels of a given Entity dimension member's parents.

```
Function getEntityParents(sChild)
Dim cHFMMetadata, cHFMEntities, cDimension, lRet
Dim vaIDs, vaLabels, vaDescs, lNumParents, vaTotal
Set cHFMMetadata = cHFMSession.metadata
Set cHFMEntities = cHFMMetadata.entities
Set cDimension = cHFMEntities.dimension
lRet = cDimension.EnumParents (cLng(-1), cLng(-1), cLng(-1), _
    sChild, 0, 0, WEBOM_METADATA_INFO_LABEL, vaIDs, vaLabels, _
    vaDescs, lNumParents, vaTotal)
getEntityParents = vaLabels
End Function
```

FindMemberOccurrencesInHierarchy

Returns an array containing the fully qualified names of all members in a dimension hierarchy whose names match the specified search string.

Tip: To return the names of all members that match a search string and that are descendants of a specified member, use [FindMemberOccurrencesInHierarchyEx](#).

Syntax

```
<HFMDimension>.FindMemberOccurrencesInHierarchy (bstrSearchText)
```

Argument	Description
----------	-------------

<i>bstrSearchText</i>	The search string. Specify either a member name or a string containing a wildcard to search for a partial match. For example, "C*" searches for any member that begins with the letter C. Input argument. String subtype.
-----------------------	--

Return Value

Returns an array containing the fully qualified names of the members that match the search criteria. In the array items, the member names in the hierarchy are delimited by backslashes. For example, suppose that an entity named Italy appears twice.

`FindMemberOccurrencesInHierarchy` would return an array containing two items similar to the following:

- \Regional\Europe\Italy
- \Management\Imbler\Italy

Example

The following function returns the names of all Entity dimension members that match the passed search string.

```
Function getEntityInHierarchy(sList)
Dim cHFMMetadata, cHFMEntities, cDimension, sEntity
set cHFMMetadata = Server.CreateObject("Hyperion.HFMwMetadata")
cHFMMetadata.SetWebSession cHFMSession
Set cHFMEntities = cHFMMetadata.entities
Set cDimension = cHFMEntities.dimension
sEntity = cDimension.FindMemberOccurrencesInHierarchy(sList)
getEntityInHierarchy = sEntity
End Function
```

FindMemberOccurrencesInHierarchyEx

Returns an array containing the fully qualified names of all members whose names match the specified search string and that are descendants of the specified member.

Tip: To return the names of all members in a dimension that match a search string, use [FindMemberOccurrencesInHierarchy](#).

Syntax

```
<HFMwDimension>.FindMemberOccurrencesInHierarchyEx (bstrSearchText,
varTopMember)
```

Argument	Description
----------	-------------

<i>bstrSearchText</i>	The search string. Specify either a member name or a string containing a wildcard to search for a partial match. For example, "C*" searches for any member that begins with the letter C. Input argument. String subtype.
-----------------------	--

Argument	Description
<i>varTopMember</i>	<p>The ancestor of the descendants to be searched. Identify this member by passing one of the following items:</p> <ul style="list-style-type: none"> ● A string containing the member's label ● A long containing the member's ID <p>Input argument.</p>

Return Value

Returns an array containing the fully qualified names of the members that match the search criteria; the path lists all members beneath the member specified in the *varTopMember* argument. In the array items, the member names in the hierarchy are delimited by backslashes. For example, if you search for all entities that begin with the letter "C", `FindMemberOccurrencesInHierarchyEx` would return array items similar to the following, and *varTopMember* would be the parent of the UnitedStates and Canada entities:

- \UnitedStates\California
- \UnitedStates\Connecticut
- \Canada

Example

The following function returns the names of all Entity dimension members that match the passed search string and that are descendants of the passed member.

```
Function getEntitiesUnderAncestor(sList, sTopMem)
Dim cHFMMetadata, cHFMEntities, cDimension, sarEntity
set cHFMMetadata = Server.CreateObject("Hyperion.HFMwMetadata")
cHFMMetadata.SetWebSession cHFMSession
Set cHFMEntities = cHFMMetadata.entities
Set cDimension = cHFMEntities.dimension
sarEntity = cDimension.FindMemberOccurrencesInHierarchyEx _
(sList, sTopMem)
getEntitiesUnderAncestor = sarEntity
End Function
```

GetAttributeValue

Returns the value that has been assigned to a given metadata attribute of a dimension member.

Tip: To return the value assigned to a member's Description attribute, use [GetMemberDescription](#).

Syntax

```
<HFMDimension>.GetAttributeValue (lMemberID, iAttribute)
```

Argument	Description
----------	-------------

<i>IMemberID</i>	The member ID of the dimension member.
------------------	--

Input argument. Long subtype.

<i>iAttribute</i>	The ID of the metadata attribute. Valid values are represented by the HFMConstants type library constants listed in the following topics:
-------------------	---

- [“Account Attribute Constants” on page 404](#)
- [“Entity Attribute Constants” on page 411](#)
- [“Scenario Attribute Constants” on page 417](#)
- [“Intercompany Partner Member Attribute Constants” on page 416](#)
- [“Custom Dimension Attributes” on page 408](#)

Input argument. Integer subtype.

Return Value

Returns the value of the specified metadata attribute.

Example

The following function takes the label of an Entity dimension member and returns the label of the entity’s DefCurrency attribute. GetAttributeValue returns the ID of the Value dimension member set as the DefCurrency attribute, and this ID is then passed to [GetMemberLabel](#).

```
Function GetEntDefaultValue(sEnt)
Dim cHFMMetadata, cHFMENT, cHFMDimScen, cHFMMVal
Dim cHFMDimVal, lScenId, lValId
' cHFMSession is a previously set HFMwSession object
Set cHFMMetadata = cHFMSession.metadata
Set cHFMENT = cHFMMetadata.entities
Set cHFMMVal = cHFMMetadata.values
Set cHFMDimScen = cHFMENT.dimension
lId = cHFMDimScen.GetMemberID(sEnt)
lValId = cHFMDimScen.GetAttributeValue(lId, _
    ATTRIB_ENTITY_DEFAULT_VALUE_ID)
Set cHFMDimVal = cHFMMVal.dimension
GetEntDefaultValue = cHFMDimVal.GetMemberLabel(lValId)
End Function
```

GetDefaultParent

Returns the member ID of a given member’s default parent.

Syntax

```
<HFMwDimension>.GetDefaultParent (lMemberID)
```

Argument	Description
----------	-------------

<i>IMemberID</i>	The member ID of the member.
------------------	------------------------------

Input argument. Long subtype.

Return Value

Returns the member ID of the default parent.

Example

The following function returns the label of a member's default parent. The function takes the member's label and the ID of the member's dimension.

```
Function getDefaultParentLabel(lDim, sMem)
Dim cMetadata, cDimension, lMem, lPar, sPar
'g_CHFMSession is an HFMwSession object reference
Set cMetadata = g_CHFMSession.metadata
'Set the HFMwDimension object reference for the specified dimension
Set cDimension = cMetadata.dimension(lDim)
lMem = cDimension.GetMemberID(sMem)
lPar = cDimension.GetDefaultParent(lMem)
sPar = cDimension.GetMemberLabel(lPar)
getDefaultParentLabel = sPar
End Function
```

GetMemberDescription

Returns the description set as the Description attribute of the specified dimension member.

Syntax

```
<HFMwDimension>.GetMemberDescription (lMemberID)
```

Argument	Description
----------	-------------

<i>lMemberID</i>	The member ID of the dimension member.
------------------	--

Input argument. Long subtype.

Return Value

Returns the description of the member.

GetMemberID

Returns a member ID, given a dimension member's name.

Syntax

```
<HFMwDimension>.GetMemberID (bstrLabel)
```

Argument	Description
----------	-------------

<i>bstrLabel</i>	The name of the member.
------------------	-------------------------

Input argument. String subtype.

Return Value

Returns the member ID for the passed member name. If no member of the passed name exists, -1 is returned.

Example

The following function returns the member ID of a Scenario dimension member.

```
Function getScenarioID(sLabel)
Dim cHFMMetadata, cHFMScenarios, cDimension, lId
Set cHFMMetadata = cHFMSession.metadata
Set cHFMScenarios = cHFMMetadata.scenarios
Set cDimension = cHFMScenarios.dimension
lId = cDimension.GetMemberID(sLabel)
getScenarioID = lId
End Function
```

GetMemberLabel

Returns a dimension member's name, given a member ID.

Syntax

```
<HFMDimension>.GetMemberLabel (lMemberID)
```

Argument	Description
----------	-------------

<i>lMemberID</i>	The member ID.
------------------	----------------

Input argument. Long subtype.

Return Value

Returns the name of the member. If no member for the passed member ID exists, a blank string is returned.

Example

The following function returns the name of the default member of the Custom 1 dimension member. The [defaultMemberID](#) property returns the member ID of the default Custom 1 dimension member. This ID is then passed to `GetMemberLabel`.

```
Function getDefaultCustom1MemLabel()
Dim cHFMMetadata, cHFMCustom1, cDimension, lID, sLabel
Set cHFMMetadata = cHFMSession.metadata
Set cHFMCustom1 = cHFMMetadata.custom1
Set cDimension = cHFMCustom1.dimension
lID = cDimension.defaultMemberID
sLabel = cDimension.GetMemberLabel(lID)
getDefaultCustom1MemLabel = sLabel
End Function
```

HF MwDimension Properties

The HF MwDimension component contains the properties described in the following topics.

alias

Returns a string containing a dimension's alias. If the dimension does not have an alias, then the dimension's [name](#) property is returned.

Read-only.

defaultMemberID

Returns the member ID of a dimension's default member. If there are no members, -1 is returned.

Note: If the dimension has a [None] member, then [None] is the default member. Otherwise, the default member is the first member returned by [EnumMembers](#).

Read-only.

Example

`defaultMemberID` is used in the example for [GetMemberLabel](#).

name

Returns a string containing a dimension's name.

Read-only.

numMembers

Returns the number of members in a dimension.

Read-only.

HF MwCurrencies Component

The HF MwCurrencies component exposes functionality for Financial Management currencies. HF MwCurrencies contains the methods described in the following topics.

Obtaining an HF MwCurrencies Object Reference

HF MwCurrencies object references are returned by the HF MwMetadata component's [currencies](#) property

EnumCurrencies

Returns arrays containing the IDs, labels, and descriptions of the application's currencies, with the option to return only those currencies for which the DisplayInICT attribute is enabled. The arrays have a one-to-one correspondence.

Tip: You determine which arrays are returned with the *lFlagsRequestedInfo* argument.

Syntax

```
<HFMwCurrencies>.EnumCurrencies (vbICTOnly, lFlagsRequestedInfo,  
pvaravar1CurrencyIDs, pvaravarbstrCurrencyLabels,  
pvaravarbstrCurrencyDescriptions)
```

Argument	Description
<i>vbICTOnly</i>	<p>A flag that specifies whether to filter currencies for which the DisplayInICT attribute is enabled. Pass TRUE to filter currencies, FALSE to return all currencies.</p> <p>Input argument. Boolean subtype.</p>
<i>lFlagsRequestedInfo</i>	<p>A bitmask that specifies the type of information to return. Valid values for the bits are represented by the HFMConstants type library constants listed in “Metadata Information Constants” on page 450.</p> <p>Input argument. Long subtype.</p>
<i>pvaravar1CurrencyIDs</i>	<p>If WEBOM_METADATA_INFO_ID or WEBOM_METADATA_INFO_ALL is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array containing the currencies' IDs.</p> <p>Input/output argument.</p>
<i>pvaravarbstrCurrencyLabels</i>	<p>If WEBOM_METADATA_INFO_LABEL or WEBOM_METADATA_INFO_ALL is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array containing the currencies' labels.</p> <p>Input/output argument.</p>
<i>pvaravarbstrCurrencyDescriptions</i>	<p>If WEBOM_METADATA_INFO_DESCRIPTION or WEBOM_METADATA_INFO_ALL is included in the <i>lFlagsRequestedInfo</i> argument's bitmask, this returns an array containing the currencies' descriptions.</p> <p>Input/output argument.</p>

Return Value

Returns a count of the currencies that are returned by this method.

Example

The following function returns the labels of all the currencies in an application.

```
Function getCurrencyLabels()  
Dim cMetadata, cCurrencies, lCnt, vaIDs, vaLabels, vaDescs  
'g_cSession is an HFMwSession object reference  
Set cMetadata = g_cSession.metadata
```

```

Set cCurrencies = cMetadata.currencies
cCurrencies.EnumCurrencies true, WEBOM_METADATA_INFO_LABEL, vaIDs, _
    vaLabels, vaDescs
getCurrencyLabels = vaLabels
End Function

```

GetCurrencyID

Returns the currency ID of a given currency.

Syntax

```
<HFMwCurrencies>.GetCurrencyID (bstrLabel)
```

Argument Description

bstrLabel The label of the currency.
Input argument. String subtype.

Return Value

Returns the currency ID.

Example

GetCurrencyID is used in the example for [GetCurrencyTranslationOperator](#).

GetCurrencyTranslationOperator

Returns the conversion operator for a given currency.

Tip: The operator is specified with the currency's TranslationOperator attribute.

Syntax

```
<HFMwCurrencies>.GetCurrencyTranslationOperator (lCurrencyID)
```

Argument Description

lCurrencyID The currency ID of the currency.
Tip: You can obtain a currency ID with [GetCurrencyID](#).
Input argument. Long subtype.

Return Value

Returns a string containing the currency's conversion operator.

Example

The following function returns a currency's conversion operator. The function takes a currency label, and [GetCurrencyID](#) gets the currency ID for [GetCurrencyTranslationOperator](#).

```
Function getTransOp(sCurr)
Dim cMetadata, cCurrencies, lCurrID
'g_cSession is an HFMwSession object reference
Set cMetadata = g_cSession.metadata
Set cCurrencies = cMetadata.currencies
lCurrID = cCurrencies.GetCurrencyID(sCurr)
getTransOp = cCurrencies.GetCurrencyTranslationOperator(lCurrID)
End Function
```


In This Chapter

HF MwData Component	105
HF MwManageProcess Component	135
HF MwCalculate Component	184

The HF MwData type library contains the following components:

- HF MwData component, which provides methods that work with data. For example, this component enables you to get and set line item details for cells, insert and remove line item details for all cells that intersect specified dimension members, and return input periods for a scenario.
- HF MwManageProcess component, which exposes Financial Management's process management functionality.
- HF MwCalculate component, which provides methods for executing consolidations, translations, calculations, and allocations.

HF MwData Component

The HF MwData component contains methods for working with Financial Management data. These methods are described in the following topics.

Obtaining an HF MwData Object Reference

Create an HF MwData object with `Server.CreateObject`. After creating the object, call [SetWebSession](#) to associate the object with the HF MwSession object for the application to which you want to get or set data:

```
Set cHFMDData = Server.CreateObject("Hyperion.HF MwData")
' cHFMSession is a previously set HF MwSession object
cHFMDData.SetWebSession cHFMSession
```

ClearData

Deletes data from all cells that intersect the specified Scenario, Year, Period, Entity, and Account dimension members.

Syntax

```
<HFMwData>.ClearData bstrScenario, bstrYear, bstrPeriods, bstrEntities,  
bstrAccounts, vbClearInputValueID, vbClearNoneValueID,  
vbEnableDetailedLogging, bstrLogFileName
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The ID label the Year dimension member. Input argument. String subtype.
<i>bstrPeriods</i>	The label of the Period dimension member. Input argument. String subtype.
<i>bstrEntities</i>	The label of the Entity dimension member. Input argument. String subtype.
<i>bstrAccounts</i>	The label of the Account dimension member. Input argument. String subtype.
<i>vbClearInputValueID</i>	Specifies whether intersecting cells for currency-related Value dimension members are to be cleared. Pass TRUE to clear these cells, FALSE otherwise. Input argument. Boolean subtype.
<i>vbClearNoneValueID</i>	Specifies whether intersecting cells for [None] Value dimension member are to be cleared. Pass TRUE to clear these cells, FALSE otherwise. Note: [None] Value dimension members are used with system accounts and currency rate accounts. Input argument. Boolean subtype.
<i>vbEnableDetailedLogging</i>	Specifies whether to include details in the log file. Pass TRUE to include details, FALSE otherwise. Input argument. Boolean subtype.
<i>bstrLogFileName</i>	The name of the log file. Input argument. String subtype.

ClearDataAuditItems

Deletes the audit history for all data changes that occurred before a given date and time.

Syntax

```
<HFMwData>.ClearDataAuditItems dTimeToClearBefore
```

Argument	Description
<i>dTimeToClearBefore</i>	The timestamp that represents the date and time. This must be a Double that can be cast to a Date format. Input argument. Double subtype.

Example

The following example deletes the data audit history through the present moment.

```
Dim cHFMDData
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.ClearDataAuditItems CDbl(Now)
```

ClearInvalidData

Scans for or deletes invalid records.

Note: Invalid records can occur when an application's metadata is changed.

Syntax

```
<HFMwData>.ClearInvalidData bScanOnly, lEnumOptions, bstrLogFileName
```

Argument	Description
<i>bScanOnly</i>	A flag indicating whether to scan for or delete invalid records. Pass TRUE to scan, FALSE to delete. Input argument. Boolean subtype.
<i>lEnumOptions</i>	This argument's value is ignored in the current release. You must pass a valid Long. Input argument. Long subtype.
<i>bstrLogFileName</i>	The name and path of the log file. Input argument. String subtype.

CopyData

Copies data from one set of cells to another. All cells that intersect the specified source dimension members are copied to the cells that intersect the specified target members.

Syntax

```
<HFMwData>.CopyData bstrSourceScenario, bstrDestScenario, bstrSourceYear,
bstrDestYear, bstrSourcePeriods, bstrDestPeriods, bstrView, bstrEntities,
bstrAccounts, vbClearInputValueID, vbClearNoneValueID, vbCopyDerived,
vbCopyCellText, vbEnableDetailedLogging, dFactor, lEnumUpdateMode,
bstrLogFileName
```

Argument	Description
<i>bstrSourceScenario</i>	The label of the source Scenario dimension member. Input argument. String subtype.
<i>bstrDestScenario</i>	The label of the destination Scenario dimension member. Input argument. String subtype.
<i>bstrSourceYear</i>	The label of the source Year dimension member. Input argument. String subtype.
<i>bstrDestYear</i>	The label of the destination Year dimension member. Input argument. String subtype.
<i>bstrSourcePeriods</i>	The label of the source Period dimension member. Input argument. String subtype.
<i>bstrDestPeriods</i>	The label of the destination Period dimension member. Input argument. String subtype.
<i>bstrView</i>	The label of the source and destination View dimension member. Input argument. String subtype.
<i>bstrEntities</i>	The label of the source and destination Entity dimension member. Input argument. String subtype.
<i>bstrAccounts</i>	The label of the source and destination Account dimension member. Input argument. String subtype.
<i>vbClearInputValueID</i>	Specifies whether cells for the <Entity Currency> Value dimension member are to be included in the source and target cells. Pass TRUE to include these cells, FALSE otherwise. Input argument. Boolean subtype.
<i>vbClearNoneValueID</i>	Specifies whether cells for the [None] Value dimension member are to be included in the source and target cells. Pass TRUE to include these cells, FALSE otherwise. Input argument. Boolean subtype.
<i>vbCopyDerived</i>	Specifies whether derived source data is to be copied to the destination cells as stored input. Pass TRUE to copy derived source data, FALSE otherwise. Input argument. Boolean subtype.
<i>vbCopyCellText</i>	Specifies whether the source cells' cell text descriptions are to be copied to the target cells. Pass TRUE to include the cell text descriptions, FALSE otherwise. Input argument. Boolean subtype.
<i>vbEnableDetailedLogging</i>	Specifies whether to include details in the log file. Pass TRUE to include details, FALSE otherwise. Input argument. Boolean subtype.
<i>dFactor</i>	The factor by which the copied data is to be multiplied before it is placed in the target cells. To copy the data without changing it, pass 1.

Argument	Description
	Input argument. Double subtype.
<i>IEnumUpdateMode</i>	<p>A flag that specifies the update mode. Pass one of the following values:</p> <ul style="list-style-type: none"> ● 0 = Replace mode: existing data in all specified cells is cleared, then the data is copied. However, if the connected user does not have full access rights to all specified cells, none of the data is copied. ● 1 = Merge mode: if a cell contains data, and the corresponding cell to be copied also contains data, the existing data is replaced. However, if a cell to be copied does not contain data, then the existing data is preserved. ● 2 = Accumulate mode: if a cell contains data, the data in the cell to be copied is added to the existing data. ● 3 = Replace with Security mode: existing data is cleared in all specified cells to which the user has full access rights, then the data is copied. Cells to which the user lacks full access are ignored. <p>Input argument. Long subtype.</p>
<i>bstrLogFileName</i>	<p>The name of the log file.</p> <p>Input argument. String subtype.</p>

DeleteLineItemDetails

Deletes line item details for the cells that intersect the specified dimension members. Line item descriptions are used to identify the line items to be deleted.

Syntax

```
<HFMwData>.DeleteLineItemDetails bstrScenario, bstrYear, bstrEntity,
bstrAccount, bstrICP, bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4,
varabstrDetail
```

Argument	Description
<i>bstrScenario</i>	<p>The name of the Scenario dimension member for the intersecting cells.</p> <p>Input argument. String subtype.</p>
<i>bstrYear</i>	<p>The name of the Year dimension member for the intersecting cells.</p> <p>Input argument. String subtype.</p>
<i>bstrEntity</i>	<p>The name of the Entity dimension member for the intersecting cells.</p> <p>Input argument. String subtype.</p>
<i>bstrAccount</i>	<p>The name of the Account dimension member for the intersecting cells.</p> <p>Input argument. String subtype.</p>
<i>bstrICP</i>	<p>The name of the Intercompany Partner dimension member for the intersecting cells.</p> <p>Input argument. String subtype.</p>
<i>bstrCustom1</i>	<p>The name of the Custom 1 dimension member for the intersecting cells.</p>

Argument	Description
	Input argument. String subtype.
<i>bstrCustom2</i>	The name of the Custom 2 dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom3</i>	The name of the Custom 3 dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom4</i>	The name of the Custom 1 dimension member for the intersecting cells. Input argument. String subtype.
<i>varabstrDetail</i>	An array of strings containing the line item descriptions to be deleted. For each array item, <code>DeleteLineItemDetails</code> deletes the line item with the matching description. Note: If an array item does not match a corresponding line item, the item is ignored and no error occurs. Input argument.

Example

The following subroutine deletes the line items for the cells that intersect the Scenario, Year, Entity, Account, Intercompany Partner, and Custom dimension members passed to it. The first input period returned by [GetInputPeriods](#) is passed along with the above members to [GetTextCellLineItems](#), and then the line item descriptions returned are passed to `DeleteLineItemDetails`.

```
Sub RemoveLineItems(sScen, sYr, sEnt, sAcct, sIcp, _
    sCust1, sCust2, sCust3, sCust4)
Dim CHFMDData, vaPers, vaDescriptions
Set CHFMDData = Server.CreateObject("Hyperion.HFMwData")
'g_CHFMSession is a previously set HFMwSession object
CHFMDData.SetWebSession g_CHFMSession
CHFMDData.GetInputPeriods sScen, vaPers
CHFMDData.GetTextCellLineItems sScen, sYr, vaPers(0), sEnt, _
    sAcct, sIcp, sCust1, sCust2, sCust3, sCust4, vaData, _
    vaDescriptions
CHFMDData.DeleteLineItemDetails sScen, sYr, sEnt, sAcct, _
    sIcp, sCust1, sCust2, sCust3, sCust4, vaDescriptions
End Sub
```

EnumDataAuditItems

Returns data audit information from a given range of audit records that meet the specified filtering criteria; the filtering criteria include date range, application server, and username. Audit information is returned in several arrays that have a one-to-one correspondence, with data values of audited cells returned as doubles.

Note: To return data values of audited cells as formatted strings instead of doubles, or to filter by cells' dimension members, use [EnumDataAuditItems2](#). To get the history of a given cell, use [GetCellHistory](#).

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *pvarlTotalRecords* argument returns the total number of records that match the filtering criteria. To iterate through all the matching records, in the first call to `EnumDataAuditItems` pass 0 to *lStartRecord*, then use the count returned by *pvarlTotalRecords* to loop through the remaining records.

Caution! The number of matching records can change after you call `EnumDataAuditItems`. For example, a user might delete some or all of the audit records.

An application stores audit histories only if the audit-related application settings have been turned on. For more information, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HFMwData>.EnumDataAuditItems dStartTime, dEndTime, vbAllServers,
bstrServer, vbAllUsers, bstrUserName, lStartRecord, lEndRecord,
pvaravarlScenarios, pvaravarlYear, pvaravarlPeriod, pvaravarlEntity,
pvaravarlParent, pvaravarlValue, pvaravarlICP, pvaravarlAccount,
pvaravarlCustom1, pvaravarlCustom2, pvaravarlCustom3, pvaravarlCustom4,
pvaravarbstrServers, pvaravarbstrUserNames, pvaravardTimeModified,
pvaravarlActivityCode, pvaravardValues, pvaravarlNoData, pvarlTotalRecords
```

Argument	Description
<i>dStartTime</i>	<p>The timestamp of the date range's starting time and date. The timestamp must be expressed as a Double that can be cast into a valid date.</p> <p>Input argument. Double subtype.</p>
<i>dEndTime</i>	<p>The timestamp of the date range's closing time and date. The timestamp must be expressed as a Double that can be cast into a valid date.</p> <p>Input argument. Double subtype.</p>
<i>vbAllServers</i>	<p>A flag that determines whether to return data audit information for all application servers. Pass TRUE for all servers, FALSE to filter by a specific application server.</p> <p>If you pass FALSE, use the <i>bstrServer</i> argument to specify the application server by which to filter.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServer</i>	<p>The name of the application server by which to filter. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllUsers</i>	<p>A flag that determines whether to return data audit information for all users. Pass TRUE for all users, FALSE to filter by a specific username.</p> <p>If you pass FALSE, use the <i>bstrUserName</i> argument to specify the username by which to filter.</p> <p>Input argument. Boolean subtype.</p>

Argument	Description
<i>bstrUserName</i>	The username by which to filter. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. Input argument. String subtype.
<i>lStartRecord</i>	The index of the first record in the range of records to retrieve. This is a zero-based index. Input argument. Long subtype.
<i>lEndRecord</i>	The index of the last record in the range of records to retrieve. This is a zero-based index. Input argument. Long subtype.
<i>pvaravarlScenarios</i>	Returns an array containing the member IDs of the Scenario dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlYear</i>	Returns an array containing the member IDs of the Year dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlPeriod</i>	Returns an array containing the member IDs of the Period dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlEntity</i>	Returns an array containing the member IDs of the Entity dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlParent</i>	<i>Future use.</i> Since only data for base entities is tracked for audit purposes in this release, this argument returns an array of items containing the value -1. Input/output argument.
<i>pvaravarlValue</i>	Returns an array containing the member IDs of the Value dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlICP</i>	Returns an array containing the member IDs of the Intercompany Partner dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlAccount</i>	Returns an array containing the member IDs of the Account dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom1</i>	Returns an array containing the member IDs of the Custom 1 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom2</i>	Returns an array containing the member IDs of the Custom 2 dimension members for the data changes' cells. Input/output argument.

Argument	Description
<i>pvaravarCustom3</i>	Returns an array containing the member IDs of the Custom 3 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarCustom4</i>	Returns an array containing the member IDs of the Custom 4 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarbstrServers</i>	Returns an array containing the names of the application servers on which the data changes were made. Note: This is a 1-based array. Input/output argument.
<i>pvaravarbstrUserNames</i>	Returns an array containing the usernames of the users who made the data changes. Note: This is a 1-based array. Input/output argument.
<i>pvaravardTimeModified</i>	Returns an array containing the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. Input/output argument.
<i>pvaravarActivityCode</i>	Returns an array containing the IDs of the user activities that caused the data changes. Valid values are represented by the HFMCconstants type library constants listed in "User Activity Constants" on page 453 . Input/output argument.
<i>pvaravardValues</i>	Returns an array containing the cell values that the data changes resulted in. The values are returned as doubles. Input/output argument.
<i>pvaravarNoData</i>	Returns an array containing that indicates whether cells contain data or no data. Valid values are as follows: <ul style="list-style-type: none"> ● 0 - The cell contains data. ● 1 - The cell contains no data. Input/output argument.
<i>pvarITotalRecords</i>	Returns the total number of audit records in the database that meet the filtering criteria. Input/output argument.

EnumDataAuditItems2

Returns data audit information from a given range of audit records that meet the specified filtering criteria; the filtering criteria include dimension members, date range, application server, and username. Audit information is returned in several arrays that have a one-to-one correspondence, with data values of audited cells returned as formatted strings.

Note: To return data values of audited cells as doubles instead of strings, or to return data audit information without filtering by dimension members, use [EnumDataAuditItems](#). To get the history of a given cell, use [GetCellHistory](#).

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *pvarlTotalRecords* argument returns the total number of records that match the filtering criteria. To iterate through all the matching records, in the first call to `EnumDataAuditItems` pass 0 to *lStartRecord*, then use the count returned by *pvarlTotalRecords* to loop through the remaining records.

Caution! The number of matching records can change after you call `EnumDataAuditItems2`. For example, a user might delete some or all of the audit records.

An application stores audit histories only if the audit-related application settings have been turned on. For more information, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HFMwData>.EnumDataAuditItems2 dStartTime, dEndTime, vbAllServers,
bstrServer, vbAllUsers, bstrUserName, varabstrMemberFilter, lStartRecord,
lEndRecord, pvaravarlScenarios, pvaravarlYear, pvaravarlPeriod,
pvaravarlEntity, pvaravarlParent, pvaravarlValue, pvaravarlICP,
pvaravarlAccount, pvaravarlCustom1, pvaravarlCustom2, pvaravarlCustom3,
pvaravarlCustom4, pvaravarbstrServers, pvaravarbstrUserNames,
pvaravardTimeModified, pvaravarlActivityCode, pvarabstrValues,
pvaravarlNoData, pvarlTotalRecords
```

Argument	Description
<i>dStartTime</i>	The timestamp of the date range's starting time and date. The timestamp must be expressed as a Double that can be cast into a valid date. Input argument. Double subtype.
<i>dEndTime</i>	The timestamp of the date range's closing time and date. The timestamp must be expressed as a Double that can be cast into a valid date. Input argument. Double subtype.
<i>vbAllServers</i>	A flag that determines whether to return data audit information for all application servers. Pass TRUE for all servers, FALSE to filter by a specific application server. If you pass FALSE, use the <i>bstrServer</i> argument to specify the application server by which to filter. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server by which to filter. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.

Argument	Description
<i>vbAllUsers</i>	<p>A flag that determines whether to return data audit information for all users. Pass TRUE for all users, FALSE to filter by a specific username.</p> <p>If you pass FALSE, use the <i>bstrUserName</i> argument to specify the username by which to filter.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrUserName</i>	<p>The username by which to filter. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>varabstrMemberFilter</i>	<p>An array that contains the dimension member names by which to filter cells. The array contains 12 items—one item for each dimension—and is indexed by the HFMConstants type library constants listed in “Dimension ID Constants” on page 414; however, do not use the constant that represents parent Entity members, because parents cannot be used as filters.</p> <p>If you do not want to filter by a given dimension, pass an empty string as the corresponding array item. For example, the following array filters by only the Scenario and Account dimensions:</p> <pre> vaFilter(DIMENSIONACCOUNT) = "Sales" vaFilter(DIMENSIONCUSTOM1) = " " vaFilter(DIMENSIONCUSTOM2) = " " vaFilter(DIMENSIONCUSTOM3) = " " vaFilter(DIMENSIONCUSTOM4) = " " vaFilter(DIMENSIONENTITY) = " " vaFilter(DIMENSIONICP) = " " vaFilter(DIMENSIONPERIOD) = " " vaFilter(DIMENSIONSCENARIO) = "Actual" vaFilter(DIMENSIONVALUE) = " " vaFilter(DIMENSIONVIEW) = " " vaFilter(DIMENSIONYEAR) = " " </pre> <p>Input argument.</p>
<i>lStartRecord</i>	<p>The index of the first record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>lEndRecord</i>	<p>The index of the last record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>pvaravarlScenarios</i>	<p>Returns an array containing the member IDs of the Scenario dimension members for the data changes' cells.</p> <p>Input/output argument.</p>
<i>pvaravarlYear</i>	<p>Returns an array containing the member IDs of the Year dimension members for the data changes' cells.</p>

Argument	Description
	Input/output argument.
<i>pvaravarlPeriod</i>	Returns an array containing the member IDs of the Period dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlEntity</i>	Returns an array containing the member IDs of the Entity dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlParent</i>	<i>Future use.</i> Since only data for base entities is tracked for audit purposes in this release, this argument returns an array of items containing the value -1. Input/output argument.
<i>pvaravarlValue</i>	Returns an array containing the member IDs of the Value dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlICP</i>	Returns an array containing the member IDs of the Intercompany Partner dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlAccount</i>	Returns an array containing the member IDs of the Account dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom1</i>	Returns an array containing the member IDs of the Custom 1 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom2</i>	Returns an array containing the member IDs of the Custom 2 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom3</i>	Returns an array containing the member IDs of the Custom 3 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarlCustom4</i>	Returns an array containing the member IDs of the Custom 4 dimension members for the data changes' cells. Input/output argument.
<i>pvaravarbstrServers</i>	Returns an array containing the names of the application servers on which the data changes were made. Note: This is a 1-based array. Input/output argument.
<i>pvaravarbstrUserNames</i>	Returns an array containing the usernames of the users who made the data changes. Note: This is a 1-based array. Input/output argument.

Argument	Description
<i>pvaravardTimeModified</i>	Returns an array containing the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. Input/output argument.
<i>pvaravarlActivityCode</i>	Returns an array containing the IDs of the user activities that caused the data changes. Valid values are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 453 . Input/output argument.
<i>pvarabstrValues</i>	Returns an array containing the cell values that the data changes resulted in. The values are returned as formatted strings. Note: This is a 1-based array. Input/output argument.
<i>pvaravarlNoData</i>	Returns an array containing that indicates whether cells contain data or no data. Valid values are as follows: <ul style="list-style-type: none"> ● 0 - The cell contains data. ● 1 - The cell contains no data. Input/output argument.
<i>pvarlTotalRecords</i>	Returns the total number of audit records in the database that meet the filtering criteria. Input/output argument.

Example

The following subroutine displays information about the cells that match the filtering criteria. The subroutine takes an array of dimension member filters and the start and end indexes of the range of audit records to retrieve. Note that the labels of the cells' dimension members are concatenated in a string and obtained by passing the member IDs returned by EnumDataAuditItems2 to HFMwDimension.[GetMemberLabel](#).

```
Sub PrintDataAudit(vaMemFilter, lStart, lEnd)
Dim cData, vaScen, cMeta, vaYear, vaPer, vaEnt, vaPar
Dim vaValue, vaICP, vaAccount, vaCustom1, vaCustom2
Dim vaCustom3, vaCustom4, vaServers, vaUsers, vaTimeModified
Dim vaActivities, vaValues, vaNoData, vaRecs
Set cData = Server.CreateObject("Hyperion.HFMwData")
Set cMeta = Server.CreateObject("Hyperion.HFMwMetadata")
' cHFMSession is a previously set HFMwSession object
cData.SetWebSession cHFMSession
cMeta.SetWebSession cHFMSession
cData.EnumDataAuditItems2 0, CDBl(Now), True, "", True, "", _
    vaMemFilter, lStart, lEnd, vaScen, vaYear, vaPer, vaEnt, _
    vaPar, vaValue, vaICP, vaAccount, vaCustom1, vaCustom2, _
    vaCustom3, vaCustom4, vaServers, vaUsers, vaTimeModified, _
    vaActivities, vaValues, vaNoData, vaRecs
For i = LBound(vaScen) To UBound(vaScen)
    ' the following lines concatenate the labels of the
    ' cell's members
    strPOV = cMeta.dimension(DIMENSIONSCENARIO).GetMemberLabel _
        (vaScen(i)) + ", " + cMeta.dimension(DIMENSIONYEAR) _
```

```

        .GetMemberLabel(vaYear(i)) + ", " + cMeta.dimension _
        (DIMENSIONPERIOD).GetMemberLabel(vaPer(i)) + ", " + _
        cMeta.dimension(DIMENSIONENTITY).GetMemberLabel(vaPar(i)) _
        + "." + cMeta.dimension(DIMENSIONENTITY).GetMemberLabel _
        (vaEnt(i)) + ", " + cMeta.dimension(DIMENSIONVALUE). _
        GetMemberLabel(vaValue(i)) + ", " + cMeta.dimension _
        (DIMENSIONICP).GetMemberLabel(vaICP(i)) + ", " + _
        cMeta.dimension(DIMENSIONACCOUNT).GetMemberLabel _
        (vaAccount(i)) + ", " + cMeta.dimension(DIMENSIONCUSTOM1). _
        GetMemberLabel(vaCustom1(i)) + ", " + cMeta.dimension _
        (DIMENSIONCUSTOM2).GetMemberLabel(vaCustom2(i)) + ", " + _
        cMeta.dimension(DIMENSIONCUSTOM3).GetMemberLabel _
        (vaCustom3(i)) + ", " + cMeta.dimension(DIMENSIONCUSTOM4). _
        GetMemberLabel(vaCustom4(i))
    Response.Write "<p>POV: " & strPOV & " --User: " & _
        vaUsers(i + 1) & " -- Modified: " & CDate(vaTimeModified(i)) _
        & " Amount: " & vaValues(i + 1) & "</p>"
Next
End Sub

```

GetCellHistory

Returns the audit history of a cell's data changes, with cell values returned as Doubles. Audit information is returned in arrays that have a one-to-one correspondence.

To enumerate the history of multiple cells, use [EnumDataAuditItems](#) or [EnumDataAuditItems2](#). To return a cell's data changes with cell values returned as Strings, use [GetCellHistory2](#).

Note: An application stores audit histories only if the audit-related application settings have been turned on. For more information, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```

<HFMwData>.GetCellHistory lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lICP, lAccount, lCustom1, lCustom2, lCustom3, lCustom4,
pvaravarbstrServers, pvaravarbstrUserNames, pvaravardTimeModified,
pvaravarlActivityCode, pvaravardValues, pvaravarlNoData

```

Argument	Description
<i>lScenario</i>	The member ID of the cell's Scenario dimension member. Input argument. Long subtype.
<i>lYear</i>	The member ID of the cell's Year dimension member. Input argument. Long subtype.
<i>lPeriod</i>	The member ID of the cell's Period dimension member. Input argument. Long subtype.

Argument	Description
<i>IEntity</i>	The member ID of the cell's Entity dimension member. Input argument. Long subtype.
<i>IParent</i>	You must pass the <code>MEMBERNOTUSED</code> constant to this argument. For information on this constant, see "Dimension Member Constants" on page 414 . Input argument. Long subtype.
<i>IValue</i>	The member ID of the cell's Value dimension member. Input argument. Long subtype.
<i>IICP</i>	The member ID of the cell's Intercompany Partner dimension member. Input argument. Long subtype.
<i>IAccount</i>	The member ID of the cell's Account dimension member. Input argument. Long subtype.
<i>ICustom1</i>	The member ID of the cell's Custom 1 dimension member. Input argument. Long subtype.
<i>ICustom2</i>	The member ID of the cell's Custom 2 dimension member. Input argument. Long subtype.
<i>ICustom3</i>	The member ID of the cell's Custom 3 dimension member. Input argument. Long subtype.
<i>ICustom4</i>	The member ID of the cell's Custom 4 dimension member. Input argument. Long subtype.
<i>pvaravabstrServers</i>	Returns an array containing the names of the application servers on which the data changes were made. Note: This is a 1-based array. Input/output argument.
<i>pvaravabstrUserNames</i>	Returns an array containing the usernames of the users who made the data changes. Note: This is a 1-based array. Input/output argument.
<i>pvaravardTimeModified</i>	Returns an array containing the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. Input/output argument.
<i>pvaravarlActivityCode</i>	Returns an array containing the IDs of the user activities that caused the data changes. Valid values are represented by the <code>HFMConstants</code> type library constants listed in "User Activity Constants" on page 453 . Input/output argument.
<i>pvaravardValues</i>	Returns an array containing the cell values that the data changes resulted in. Cell values are returned as Doubles.

Argument	Description
	Input/output argument.
<i>pvaravar1/NoData</i>	Returns an array containing that indicates whether cells contain data or no data. Valid values are as follows: <ul style="list-style-type: none"> ● 0 - The cell contains data. ● 1 - The cell contains no data. Input/output argument.

Example

The following function returns a three-dimensional array containing the timestamps, users, and values of changes to a given cell. Note that if no changes have been made to the cell, the function returns a blank string instead of an array.

```
Function GetCellDatesUsersValues(lScen, lYear, _
    lPer, lEnt, lPar, lVal, lIcp, lAcct, lCust1, lCust2, _
    lCust3, lCust4)
Dim cHFMDData, vaServers, vaUsers, vaTime, vaActivity
Dim vaVal, vaNoData, vaRet(), lItems
'cHFMSession is an HFMwSession object reference.
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.GetCellHistory lScen, lYear, lPer, lEnt, lPar, lVal, _
    lIcp, lAcct, lCust1, lCust2, lCust3, lCust4, vaServers, _
    vaUsers, vaTime, vaActivity, vaVal, vaNoData
lItems = UBound(vaTime)
'If the array contains an item, the cell has a change history
If lItems >= 0 Then
    ReDim vaRet(lItems, 2)
    For i = 0 To lItems
        vaRet(i, 0) = CDate(vaTime(i))
        vaRet(i, 1) = vaUsers(i + 1)
        vaRet(i, 2) = vaVal(i)
    Next
    GetCellDatesUsersValues = vaRet
Else
    GetCellDatesUsersValues = ""
End If
End Function
```

GetCellHistory2

Returns the audit history of a cell's data changes, with cell values returned as formatted strings. Audit information is returned in arrays that have a one-to-one correspondence.

To enumerate the history of multiple cells, use [EnumDataAuditItems](#) or [EnumDataAuditItems2](#). To return a cell's data changes with cell values returned as Doubles, use [GetCellHistory](#).

Note: An application stores audit histories only if the audit-related application settings have been turned on. For more information, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HFMwData>.GetCellHistory2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lICP, lAccount, lCustom1, lCustom2, lCustom3, lCustom4,
pvaravarbstrServers, pvaravarbstrUserNames, pvaravardTimeModified,
pvaravarlActivityCode, pvarabstrValues, pvaravarlNoData
```

Argument	Description
<i>lScenario</i>	The member ID of the cell's Scenario dimension member. Input argument. Long subtype.
<i>lYear</i>	The member ID of the cell's Year dimension member. Input argument. Long subtype.
<i>lPeriod</i>	The member ID of the cell's Period dimension member. Input argument. Long subtype.
<i>lEntity</i>	The member ID of the cell's Entity dimension member. Input argument. Long subtype.
<i>lParent</i>	You must pass the <code>MEMBERNOTUSED</code> constant to this argument. For information on this constant, see "Dimension Member Constants" on page 414 . Input argument. Long subtype.
<i>lValue</i>	The member ID of the cell's Value dimension member. Input argument. Long subtype.
<i>lICP</i>	The member ID of the cell's Intercompany Partner dimension member. Input argument. Long subtype.
<i>lAccount</i>	The member ID of the cell's Account dimension member. Input argument. Long subtype.
<i>lCustom1</i>	The member ID of the cell's Custom 1 dimension member. Input argument. Long subtype.
<i>lCustom2</i>	The member ID of the cell's Custom 2 dimension member. Input argument. Long subtype.
<i>lCustom3</i>	The member ID of the cell's Custom 3 dimension member. Input argument. Long subtype.
<i>lCustom4</i>	The member ID of the cell's Custom 4 dimension member. Input argument. Long subtype.

Argument	Description
<i>pvaravabstrServers</i>	Returns an array containing the names of the application servers on which the data changes were made. Note: This is a 1-based array. Input/output argument.
<i>pvaravabstrUserNames</i>	Returns an array containing the usernames of the users who made the data changes. Note: This is a 1-based array. Input/output argument.
<i>pvaravardTimeModified</i>	Returns an array containing the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. Input/output argument.
<i>pvaravarlActivityCode</i>	Returns an array containing the IDs of the user activities that caused the data changes. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453 . Input/output argument.
<i>pvarabstrValues</i>	Returns an array containing the cell values that the data changes resulted in. Cell values are returned as formatted strings. Note: This is a 1-based array. The values are formatted according to the user’s preferences for decimal separator and thousands separator characters. If you want to programmatically get and set these preferences, use the HFMwSession properties decimalSeparator and thousandsSeparator . Input/output argument.
<i>pvaravarlNoData</i>	Returns an array containing that indicates whether cells contain data or no data. Valid values are as follows: <ul style="list-style-type: none"> ● 0 - The cell contains data. ● 1 - The cell contains no data. Input/output argument.

Example

The following function returns a three-dimensional array containing the timestamps, users, and values of changes to a given cell. Note that if no changes have been made to the cell, the function returns a blank string instead of an array.

```
Function GetCellDatesUsersValuesStr(lScen, lYear, _
    lPer, lEnt, lPar, lVal, lIcp, lAcct, lCust1, lCust2, _
    lCust3, lCust4)
Dim cHFMDData, vaServers, vaUsers, vaTime, vaActivity
Dim vaVal, vaNoData, vaRet(), lItems
' cHFMSession is an HFMwSession object reference.
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.GetCellHistory2 lScen, lYear, lPer, lEnt, lPar, lVal, _
```

```

    lIcp, lAcct, lCust1, lCust2, lCust3, lCust4, vaServers, _
    vaUsers, vaTime, vaActivity, vaVal, vaNoData
lItems = UBound(vaTime)
'If the array contains an item, the cell has a change history
If lItems >= 0 Then
    ReDim vaRet(lItems, 2)
    For i = 0 To lItems
        vaRet(i, 0) = CDate(vaTime(i))
        vaRet(i, 1) = vaUsers(i + 1)
        vaRet(i, 2) = vaVal(i + 1)
    Next
    GetCellDatesUsersValuesStr = vaRet
Else
    GetCellDatesUsersValuesStr = ""
End If
End Function

```

GetDoubleFromText

Converts a numeric string to a Double subtype.

Syntax

```

<HFMwData>.GetDoubleFromText bstrText, bstrDecimalChar, bstrThousandsChar,
pbIsValidNumber, pbIsNumberNoData, pvardNumber

```

Argument	Description
<i>bstrText</i>	The string to be converted. Input argument. String subtype.
<i>bstrDecimalChar</i>	Specifies the decimal separator character. Input argument. String subtype.
<i>bstrThousandsChar</i>	Specifies the thousands separator character. Input argument. String subtype.
<i>pbIsValidNumber</i>	Indicates whether the <i>bstrText</i> argument's string can be evaluated as a Double. Returns TRUE if he string can be evaluated as a Double, FALSE otherwise. Output argument.
<i>pbIsNumberNoData</i>	Indicates whether the <i>bstrText</i> argument's string evaluates to no data. Returns TRUE if the string evaluates to no data, FALSE otherwise. For example, a blank string passed to the <i>bstrText</i> argument would evaluate to no data. Output argument.
<i>pvardNumber</i>	Returns the Double equivalent of the <i>bstrText</i> argument. Note: If the <i>pbIsValidNumber</i> argument returns FALSE, or the <i>pbIsNumberNoData</i> argument returns TRUE, then the <i>bstrText</i> argument's string returns 0. Output argument.

Example

The following function converts a string to a Double, using the application's default decimal and thousands separator characters. If the passed string evaluates to no data, the function returns null.

```
Function GetDoubleDefaultSeparators(sNum)
Dim cHFMDData, bIsValid, bIsNoData, sRetVal
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.GetDoubleFromText sNum, cHFMSession.decimalSeparator, _
cHFMSession.thousandsSeparator, bIsValid, bIsNoData, dRetVal
If bIsNoData = TRUE Then
    GetDoubleDefaultSeparators = null
Else
    GetDoubleDefaultSeparators = dRetVal
End If
End Function
```

GetEntityDetails

Returns entity transaction details for the specified cell.

The information returned by `GetEntityDetails` corresponds to the data displayed by Entity Transaction Detail reports. For more information on the data returned, see the topics for Entity Transaction Detail reports in the *Oracle Hyperion Financial Management, Fusion Edition User's Guide*.

Tip: To return header information for the cell's entity transaction details, use `GetEntityDetailsHeader`.

Syntax

```
<HFMwData>.GetEntityDetails lEntityDetailsOptions, bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrView, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4,
pvaravar2DRowColumnDetails
```

Argument	Description
<i>lEntityDetailsOptions</i>	Determines which entity transaction detail data is returned. Valid values are represented by the HFMConstants type library constants listed in "Entity Transaction Detail Display Options" on page 456 . You can use the addition operator (+) with these constants to return multiple types of information. Input argument. Long subtype.
<i>bstrScenario</i>	The label of the cell's Scenario dimension member. Input argument. String subtype.

Argument	Description
<i>bstrYear</i>	The label of the cell's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the cell's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the cell's child Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the cell's parent Entity dimension member. Input argument. String subtype.
<i>bstrView</i>	The label of the cell's View dimension member. Input argument. String subtype.
<i>bstrAccount</i>	The label of the cell's Account dimension member. Input argument. String subtype.
<i>bstrICP</i>	The label of the cell's Intercompany Partner dimension member. Input argument. String subtype.
<i>bstrCustom1</i>	The label of the cell's Custom 1 dimension member. Input argument. String subtype.
<i>bstrCustom2</i>	The label of the cell's Custom 2 dimension member. Input argument. String subtype.
<i>bstrCustom3</i>	The label of the cell's Custom 3 dimension member. Input argument. String subtype.
<i>bstrCustom4</i>	The label of the cell's Custom 4 dimension member. Input argument. String subtype.
<i>pvaravar2DRowColumnDetails</i>	Returns a two-dimensional array that contains the entity transaction details. The first dimension represents the rows, the second dimension represents the columns. The second dimension is indexed by the HFMConstants type library constants listed in "Entity Transaction Detail Information" on page 457 . Input/output argument.

Example

The following function takes the labels of a cell's dimension members, and returns an HTML string that contains some header information returned by [GetEntityDetailsHeader](#), as well as the Value dimension member, Entity dimension member, amount, debit, and credit entity transaction detail data.

Note: Because some system-generated Value dimension member names contain greater-than and less-than characters, the example calls the VBScript Replace function to replace these characters with their corresponding HTML symbols.

```
Function GetHtmlEntDetails(sScen, sYear, sPer, sEnt, sPar, sView, _
    sAcct, sICP, sCust1, sCust2, sCust3, sCust4)
Dim cHFMDData, vaHead, lOptions, vaRowCols, sRet
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.GetEntityDetailsHeader sScen, sEnt, sPar, vaHead
sRet = "<p><b>Time: </b>" & vaHead(ENTITYDETAILS_HEADER_TIME) & "</p>"
sRet = sRet & "<p><b>Entity Currency: </b>" & _
    vaHead(ENTITYDETAILS_HEADER_ENTITY_CURRENCY) & "</p>"
sRet = sRet & "<p><b>Parent Currency: </b>" & _
    vaHead(ENTITYDETAILS_HEADER_PARENT_CURRENCY) & "</p>"
lOptions = ENTITYDETAILS_CREDIT + ENTITYDETAILS_DEBIT + _
    ENTITYDETAILS_ENTITY + ENTITYDETAILS_LID + ENTITYDETAILS_BASE_RECORDS
cHFMDData.GetEntityDetails lOptions, sScen, sYear, sPer, sEnt, sPar, _
    sView, sAcct, sICP, sCust1, sCust2, sCust3, sCust4, vaRowCols
sRet = sRet & "<table cellpadding=2>"
sRet = sRet & "<tr><td><b>Value</b></td><td><b>Amount</b></td>" & _
    "<td><b>Debit</b></td><td><b>Credit</b></td></tr>"
For i = lBound(vaRowCols) to uBound(vaRowCols)
    sVal = Replace(vaRowCols(i, ENTITYDETAILS_DIMENSION_VALUE), "<", "&lt;")
    sVal = Replace(sVal, ">", "&gt;")
    sRet = sRet + "<tr><td>" & sVal & "</td>"
    sRet = sRet & "<td>" & vaRowCols(i, ENTITYDETAILS_DIMENSION_AMOUNT) & _
        "</td>"
    sRet = sRet & "<td>" & vaRowCols(i, ENTITYDETAILS_DIMENSION_DEBIT) & _
        "</td>"
    sRet = sRet & "<td>" & vaRowCols(i, ENTITYDETAILS_DIMENSION_CREDIT) & _
        "</td></tr>"
Next
sRet = sRet + "</table>"
GetHtmlEntDetails = sRet
End Function
```

GetEntityDetailsHeader

Returns the header information for a cell's Entity Transaction Detail report.

Tip: To return Entity Transaction Detail report data, use [GetEntityDetails](#).

Syntax

```
<HFMwData>.GetEntityDetailsHeader bstrScenario, bstrEntity, bstrParent,
pvaravarHeader
```

Argument	Description
<i>bstrScenario</i>	The label of the cell's Scenario dimension member.

Argument	Description
	Input argument. String subtype.
<i>bstrEntity</i>	The label of the cell's child Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the cell's parent Entity dimension member. Input argument. String subtype.
<i>pvaravarHeader</i>	A variant array of variants that contain the header information. The indexes of the array elements are represented by the HFMConstants type library constants listed in “Entity Transaction Detail Header Options” on page 456 . Input/output argument.

Example

GetEntityDetailsHeader is used in the example for [GetEntityDetails](#).

GetFormattedNumber

Formats a number by applying characteristics such as scaling and delimiters.

Syntax

```
<HFMwData>.GetFormattedNumber dNumber, lNumDecimals, bstrDecimalCharacter,
bstrThousandsCharacter, sScale, bRemoveTrailingZeroes, pbstrFormattedNumber
```

Argument	Description
<i>dNumber</i>	The number to be formatted. Input argument. Double subtype.
<i>lNumDecimals</i>	Specifies the number of decimal places to be included in the formatted number. Input argument. Long subtype.
<i>bstrDecimalCharacter</i>	Specifies the decimal separator character to be used in the formatted number. Tip: To apply the application's default decimal separator, use the HFMwSession component's decimalSeparator property. Input argument. String subtype.
<i>bstrThousandsCharacter</i>	Specifies the thousands separator character to be used in the formatted number. Tip: To apply the application's default thousands separator, use the HFMwSession component's thousandsSeparator property. Input argument. String subtype.
<i>sScale</i>	Specifies the degree of scaling to apply to the formatted number. Scaling works as described in the following list: <ul style="list-style-type: none"> ● To leave the return value unscaled, pass 0. ● To scale the return value, each whole-number increment over 0 scales by a tenth, and each whole-number increment less than 0 scales by ten. In other words, passing 1 scales

Argument	Description
	by a tenth, passing 2 scales by a hundredth, passing -1 scales by ten, passing -2 scales by one hundred, and so on. Input argument. Integer subtype.
<i>bRemoveTrailingZeroes</i>	Determines whether trailing zeroes will be stripped from the decimal segment of the formatted number. Pass TRUE to strip trailing zeroes, FALSE otherwise. Input argument.
<i>pbstFormattedNumber</i>	Returns the formatted number. Output argument.

Example

The following function scales a number by a thousandth and applies the application's default decimal and thousands separators to the formatted number.

```
Function ScaleToThousandths(dNum)
Dim cHFMDData, sNum
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
' cHFMSession is a previously set HFMwSession object
cHFMDData.SetWebSession cHFMSession
cHFMDData.GetFormattedNumber dNum, 0, _
    cHFMSession.decimalSeparator, _
    cHFMSession.thousandsSeparator, 3, TRUE, sRetVal
ScaleToThousandths = sRetVal
End Function
```

GetInputPeriods

Returns the labels of the input periods for a given scenario. You optionally can filter periods that for a given Point of View are read-only or do not support line item input . Passing a Scenario member label returns labels of all the scenario's input periods; passing an array containing a Point of View's labels filters the return value.

Note: A scenario's input periods are determined by the scenario's DefaultFreq attribute.

Syntax

```
<HFMwData>.GetInputPeriods varbstrScenarioStrOrPovArray, pvarbstrPeriods
```

Argument	Description
<i>varbstrScenarioStrOrPovArray</i>	Specifies the scenario and whether to filter the return value. Pass one of the following values: <ul style="list-style-type: none"> To return labels of all the scenario's input periods, pass the scenario's label. To filter the return value, pass an array of member labels that identify a Point of View. The array must contain 12 elements, and is indexed by the HFMConstants type library constants listed in "Dimension ID Constants" on page 414.

Argument	Description
	<p>For the View dimension, always specify the <Scenario View> member. The Period dimension member must be specified but is ignored. If you specify a parent for the Entity dimension member, the parent is ignored.</p> <p>Following is an example that shows how to define a Point of View array:</p> <pre>Dim vaPov(11) vaPov(DIMENSIONSCENARIO) = "Actual" vaPov(DIMENSIONYEAR) = "2004" vaPov(DIMENSIONPERIOD) = "June" vaPov(DIMENSIONVIEW) = "<Scenario View>" vaPov(DIMENSIONENTITY) = "Virginia" vaPov(DIMENSIONVALUE) = "USD" vaPov(DIMENSIONACCOUNT) = "Salaries" vaPov(DIMENSIONICP) = "[ICP None]" vaPov(DIMENSIONCUSTOM1) = "Golfballs" vaPov(DIMENSIONCUSTOM2) = "Customer2" vaPov(DIMENSIONCUSTOM3) = "[None]" vaPov(DIMENSIONCUSTOM4) = "Increases"</pre> <p>Input argument.</p>
<i>pvarbstrPeriods</i>	<p>Returns an array of strings containing the labels of the input periods.</p> <p>If you pass in a Point of View array, the return value is filtered in the following ways:</p> <ul style="list-style-type: none"> Input periods that are invalid for line item details are omitted. If all periods are invalid, an empty array is returned. A read-only period is flagged with a pipe () character prefixed to its label. The caller will need to parse for pipe characters. <p>Note: Locked periods that do not contain line item details are considered as invalid input periods and thus are omitted.</p> <p>Output argument.</p>

Example

GetInputPeriods is used in the example for [DeleteLineItemDetails](#).

GetRegionsForCell

Returns the display names and corresponding drill-through URLs for the region defined by a Scenario, Year, Period, Entity, and Account.

Syntax

```
<HFMwData>.GetRegionsForCell bstrLanguage, bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrAccount, pvarabstrDisplayNames, pvarabstrUrls
```

Argument	Description
<i>bstrLanguage</i>	<p>String (ByVal). The two-letter language code representing the language in which to return the string. To obtain language codes, use GetLanguageCountryCodeFromLanguageId</p>

Argument	Description
	Input argument. String subtype.
<i>bstrScenario</i>	The name of the cell's scenario. Input argument. String subtype.
<i>bstrYear</i>	The name of the cell's year. Input argument. String subtype.
<i>bstrPeriod</i>	The name of the cell's period. Input argument. String subtype.
<i>bstrEntity</i>	The name of the cell's entity. Input argument. String subtype.
<i>bstrAccount</i>	The name of the cell's account. Input argument. String subtype.
<i>pvarabstrDisplayNames</i>	Returns an array of strings that contains the display names of the drillable regions. Input/output argument.
<i>pvarabstrUrls</i>	Returns an array of strings that contains the drill-through URLs of the drillable regions. Input/output argument.

GetTextCellLineItems

Returns arrays of the data and descriptions for the specified cell's line items.

Syntax

```
<HFMwData>.GetTextCellLineItems bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrAccount, bstrICP, bstrCustom1, bstrCustom2, bstrCustom3,
bstrCustom4, pvarabstrData, pvarabstrDetails
```

Argument	Description
<i>bstrScenario</i>	The name of the cell's scenario. Input argument. String subtype.
<i>bstrYear</i>	The name of the cell's year. Input argument. String subtype.
<i>bstrPeriod</i>	The name of the cell's period. Input argument. String subtype.
<i>bstrEntity</i>	The name of the cell's entity. Input argument. String subtype.
<i>bstrAccount</i>	The name of the cell's account.

Argument	Description
	Input argument. String subtype.
<i>bstrICP</i>	The name of the cell's Intercompany Partner dimension member. Input argument. String subtype.
<i>bstrCustom1</i>	The name of the cell's Custom 1 dimension member. Input argument. String subtype.
<i>bstrCustom2</i>	The name of the cell's Custom 2 dimension member. Input argument. String subtype.
<i>bstrCustom3</i>	The name of the cell's Custom 3 dimension member. Input argument. String subtype.
<i>bstrCustom4</i>	The name of the cell's Custom 4 dimension member. Input argument. String subtype.
<i>pvarabstrData</i>	Returns an array of strings that contain the cell's line item data. Output argument.
<i>pvarabstrDetails</i>	Returns an array of strings that contain the cell's line item descriptions. This array has a one-to-one correspondence with the array returned by the <i>pvarabstrData</i> argument. Output argument.

Example

GetTextCellLineItems is used in the example for [DeleteLineItemDetails](#).

InsertLineItemDetails

Inserts line item descriptions in the cells that intersect the specified dimension members.

Syntax

```
<HFMwData>.InsertLineItemDetails bstrScenario, bstrYear, bstrEntity,
bstrAccount, bstrICP, bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4,
varabstrDetail, varalInsertPos
```

Argument	Description
<i>bstrScenario</i>	The name of the Scenario dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrYear</i>	The name of the Year dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrEntity</i>	The name of the Entity dimension member for the intersecting cells. Input argument. String subtype.

Argument	Description
<i>bstrAccount</i>	The name of the Account dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrICP</i>	The name of the Intercompany Partner dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom1</i>	The name of the Custom 1 dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom2</i>	The name of the Custom 2 dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom3</i>	The name of the Custom 3 dimension member for the intersecting cells. Input argument. String subtype.
<i>bstrCustom4</i>	The name of the Custom 4 dimension member for the intersecting cells. Input argument. String subtype.
<i>varabstrDetail</i>	An array of strings that contain the line item descriptions to be added. Input argument.
<i>varallInsertPos</i>	An array of Longs that indicate the positions in the intersecting cells in which the descriptions are to be added. The line item position identifiers are 0-based, and the array has a one-to-one correspondence with the <i>varabstrDetail</i> argument's array. Caution! The array items must be cast as Long subtypes. Input argument.

Example

The following subroutine creates the first line item for the cells that intersect the dimension members passed. The subroutine also takes the description to be applied to the line items.

```
Sub InsertFirstLineItem(sScen, sYr, sEnt, sAcct, sIcp, _
    sCust1, sCust2, sCust3, sCust4, sLineDesc)
Dim cHFMDData, vaPos(0), vaDesc(0)
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
'g_chFMSSession is an HFMwSession object reference
cHFMDData.SetWebSession g_chFMSSession
' Assign the passed description to an array
vaDesc(0) = sLineDesc
' The line item position must be cast as a Long
vaPos(0) = CLng(0)
cHFMDData.InsertLineItemDetails sScen, sYr, sEnt, sAcct, _
    sIcp, sCust1, sCust2, sCust3, sCust4, vaDesc, vaPos
End Sub
```

SetTextCellLineItems

Appends or updates line items for the specified cell.

Syntax

```
<HFMwData>.SetTextCellLineItems bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrAccount, bstrICP, bstrCustom1, bstrCustom2, bstrCustom3,  
bstrCustom4, varabstrData, varabstrDetails
```

Argument	Description
<i>bstrScenario</i>	The name of the cell's scenario. Input argument. String subtype.
<i>bstrYear</i>	The name of the cell's year. Input argument. String subtype.
<i>bstrPeriod</i>	The name of the cell's period. Input argument. String subtype.
<i>bstrEntity</i>	The name of the cell's entity. Input argument. String subtype.
<i>bstrAccount</i>	The name of the cell's account. Input argument. String subtype.
<i>bstrICP</i>	The name of the cell's Intercompany Partner dimension member. Input argument. String subtype.
<i>bstrCustom1</i>	The name of the cell's Custom 1 dimension member. Input argument. String subtype.
<i>bstrCustom2</i>	The name of the cell's Custom 2 dimension member. Input argument. String subtype.
<i>bstrCustom3</i>	The name of the cell's Custom 3 dimension member. Input argument. String subtype.
<i>bstrCustom4</i>	The name of the cell's Custom 4 dimension member. Input argument. String subtype.
<i>varabstrData</i>	An array of strings containing the line item data to be added, formatted according to the connected user's preferences. Tip: The HFMwSession component provides access to user preferences. For details, see Chapter 4, "HFMwSession Type Library." Input argument.
<i>varabstrDetails</i>	An array of strings containing the line item descriptions to be added. This array has a one-to-one correspondence with the <i>varabstrData</i> argument's array. Input argument.

Example

The following example inserts two line items for the specified cell.

```

Dim cHFMDData, vaData(1), vaDetails(1)
Set cHFMDData = Server.CreateObject("Hyperion.HFMwData")
'g_cHFMSession is an HFMwSession object reference
cHFMDData.SetWebSession g_cHFMSession
vaData(0) = "61"
vaData(1) = "56"
vaDetails(0) = "Roger"
vaDetails(1) = "Mickey"
cHFMDData.SetTextCellLineItems "Actual", "2000", "July", _
    "NewYork", "Salaries", "[ICP None]", "GolfBalls", _
    "Customer2", "[None]", "Increases", vaData, vaDetails

```

SetTextCellsLineItems

Appends or updates line items for the specified cells.

The cells' dimension members are passed in arrays that have a one-to-one correspondence with each other.

Syntax

```

<HFMwData>.SetTextCellsLineItems varabstrScenario, varabstrYear,
varabstrPeriod, varabstrEntity, varabstrAccount, varabstrICP,
varabstrCustom1, varabstrCustom2, varabstrCustom3, varabstrCustom4,
varabstrData, varabstrDetail

```

Argument	Description
<i>varabstrScenario</i>	An array of strings containing the names of the cells' scenarios. Input argument.
<i>varabstrYear</i>	An array of strings containing the names of the cells' years. Input argument.
<i>varabstrPeriod</i>	An array of strings containing the names of the cells' periods. Input argument.
<i>varabstrEntity</i>	An array of strings containing the names of the cells' entities. Input argument.
<i>varabstrAccount</i>	An array of strings containing the names of the cells' accounts. Input argument.
<i>varabstrICP</i>	An array of strings containing the names of the cells' Intercompany Partner dimension members. Input argument.
<i>varabstrCustom1</i>	An array of strings containing the names of the cells' Custom 1 dimension members. Input argument.
<i>varabstrCustom2</i>	An array of strings containing the names of the cells' Custom 2 dimension members. Input argument.

Argument	Description
<i>varabstrCustom3</i>	An array of strings containing the names of the cells' Custom 3 dimension members. Input argument.
<i>varabstrCustom4</i>	An array of strings containing the names of the cells' Custom 4 dimension members. Input argument.
<i>varabstrData</i>	An array of strings containing the line item data to be added to the cells, formatted according to the connected user's preferences. Tip: The HFMwSession component provides access to user preferences. For details, see Chapter 4, "HFMwSession Type Library." Input argument.
<i>varabstrDetail</i>	An array of strings containing the line item descriptions to be added to the cells. Input argument.

SetWebSession

Associates an HFMwData object with the HFMwSession object for an application.

Syntax

```
<HFMwData>.SetWebSession varpIUnkHFMwSession
```

Argument	Description
<i>varpIUnkHFMwSession</i>	The HFMwSession object for the application. Input argument.

Example

SetWebSession is used in the example for [InsertLineItemDetails](#).

HFMwManageProcess Component

The HFMwManageProcess component exposes Financial Management's process management functionality, and contains the methods described in the following topics.

Obtaining an HFMwManageProcess Object Reference

Create an HFMwManageProcess object with `Server.CreateObject`. After creating the object, call [SetWebSession](#) to associate the object with the HFMwSession object for the application:

```
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
```

Approve

Performs a approve action on the specified process units. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

To attach documents while approving process units, use [Approve2](#).

Syntax

```
<HFMwManageProcess>.Approve (bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units. Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Return Value

Returns a number that represents the process units' review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example approves the specified process unit and prints the resulting review level.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Approve("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", FALSE, _
"")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

Approve2

Performs a approve action on the specified process units and optionally attaches documents. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

Syntax

```
<HFMwManageProcess>.Approve2 (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment,
varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units. Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument. Boolean subtype.

Argument	Description
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the process units' review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example approves a process unit and attaches two documents.

```
Dim iState, saPaths(1), saNames(1)
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'CHFMPProcMan is an HFMwManageProcess object reference
iState = CHFMPProcMan.Approve2("Budget", "2004", "August", "NewYork", _
    "UnitedStates", "<EntityCurrency>", False, "see attachments", _
    saPaths, saNames)
```

ChangeProcessManagementStateForMultipleEntities

Performs a given process management action for the specified process units.

To attach documents while performing an action, use [ChangeProcessManagementStateForMultipleEntities2](#).

Syntax

```
<HFMwManageProcess>.ChangeProcessManagementStateForMultipleEntities
(bstrScenario, bstrYear, bstrPeriod, varabstrEntity, varabstrParent,
```

bstrValue, *bstrComment*, *lAction*, *vbUseAllValueMembers*,
vbApplyToAllPeriods, *sProcessStateToPromoteTo*)

Argument	Description
<i>bstrScenario</i>	The label containing the process units' Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label containing the process units' Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label containing the process units' Period dimension member. Input argument. String subtype.
<i>varabstrEntity</i>	An array containing the labels of the process units' child Entity dimension members. Input argument.
<i>varabstrParent</i>	An array containing the labels of the process units' parent Entity dimension members. Note: The <i>varabstrEntity</i> and <i>varabstrParent</i> arguments' arrays have a one-to-one correspondence. Input argument.
<i>bstrValue</i>	The label containing the process units' Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action. Input argument. String subtype.
<i>lAction</i>	The action to perform for the process units. Valid values are represented by the HFMConstants type library constants listed in "Process Management Action Constants" on page 433 . Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	A flag that determines whether the action applies to process units for all Value dimension members related to the <i>bstrValue</i> argument's member. Pass TRUE to apply the action to the related process units, FALSE to apply the action to only the process unit for the <i>bstrValue</i> member Input argument. Boolean subtype.
<i>vbApplyToAllPeriods</i>	A flag that determines whether the action applies to process units for all Period dimension members that intersect the other specified dimension members. Pass TRUE to apply the action to intersecting process units, FALSE to apply the action to only the process unit for the period specified in the <i>bstrPeriod</i> argument. Input argument. Boolean subtype.
<i>sProcessStateToPromoteTo</i>	The review level to which to promote the process unit. Valid values are represented by the HFMConstants type library constants listed in "Process Management Review Level Constants" on page 434 . Input argument. Integer subtype.

Return Value

Returns the resulting review level for the specified process units. Valid values are represented by the HFMConstants type library constants listed in [“Process Management Review Level Constants” on page 434](#).

ChangeProcessManagementStateForMultipleEntities2

Performs a given process management action for the specified process units and optionally attaches documents.

Syntax

```
<HFMwManageProcess>.ChangeProcessManagementStateForMultipleEntities2  
(bstrScenario, bstrYear, bstrPeriod, varabstrEntity, varabstrParent,  
bstrValue, bstrComment, lAction, vbUseAllValueMembers,  
vbApplyToAllPeriods, sProcessStateToPromoteTo, varabstrPaths,  
varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label containing the process units' Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label containing the process units' Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label containing the process units' Period dimension member. Input argument. String subtype.
<i>varabstrEntity</i>	An array containing the labels of the process units' child Entity dimension members. Input argument.
<i>varabstrParent</i>	An array containing the labels of the process units' parent Entity dimension members. Note: The <i>varabstrEntity</i> and <i>varabstrParent</i> arguments' arrays have a one-to-one correspondence. Input argument.
<i>bstrValue</i>	The label containing the process units' Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>lAction</i>	The action to perform for the process units. Valid values are represented by the HFMConstants type library constants listed in “Process Management Action Constants” on page 433 . Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	A flag that determines whether the action applies to process units for all Value dimension members related to the <i>bstrValue</i> argument's member. Pass TRUE to apply the action

Argument	Description
	to the related process units, FALSE to apply the action to only the process unit for the <i>bstrValue</i> member Input argument. Boolean subtype.
<i>vbApplyToAllPeriods</i>	A flag that determines whether the action applies to process units for all Period dimension members that intersect the other specified dimension members. Pass TRUE to apply the action to intersecting process units, FALSE to apply the action to only the process unit for the period specified in the <i>bstrPeriod</i> argument. Input argument. Boolean subtype.
<i>sProcessStateToPromoteTo</i>	The review level to which to promote the process unit. Valid values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434 . Input argument. Integer subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument’s array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument’s array, otherwise an error will be thrown. Input argument.

Return Value

Returns the resulting review level for the specified process units. Valid values are represented by the HFMConstants type library constants listed in [“Process Management Review Level Constants” on page 434](#).

GetHistory

Returns arrays that represent the timestamps, usernames, action types, review levels, and comments for the process management actions of the specified process unit.

To get a process unit history that includes document attachments, use [GetHistory2](#).

Syntax

```
<HFMwManageProcess>.GetHistory bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, pvaradTime, pvarabstrUser, pvarasAction,
pvarasNewState, pvarabstrComment
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.

Argument	Description
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>pvaradTime</i>	Returns an array of the timestamps for the process management actions. Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time. The input value of the argument is ignored. Input/output argument.
<i>pvarabstrUser</i>	Returns an array containing the usernames for the process management actions. The input value of the argument is ignored. Input/output argument.
<i>pvarasAction</i>	Returns an array of the action types of the process management actions. The return values are represented by the <code>HFMConstants</code> type library constants listed in “Process Management Action Constants” on page 433 . The input value of the argument is ignored. Input/output argument.
<i>pvarasNewState</i>	Returns an array of the review levels resulting from the process management actions. The return values are represented by the <code>HFMConstants</code> type library constants listed in “Process Management Review Level Constants” on page 434 . Tip: To return the string equivalent of an array item, pass the item to MapReviewLevelToString . The input value of the argument is ignored. Input/output argument.
<i>pvarabstrComment</i>	Returns an array of the comments specified for the process management actions. The input value of the argument is ignored. Input/output argument.

Example

The following example prints the specified process unit's process management history to the browser as an HTML table. Note that if [GetReviewLevel](#) indicates that the process unit has not been started, a message is printed to the browser instead of a history.

```
Sub printUnitHistory(sScen, sYr, sPer, sEnt, sPar, sVal)
Dim cHFMPProcMan, vaTime, vaUser, vaAction, vaLevel, vaComment
Set cHFMPProcMan = Server.CreateObject("Hyperion.HFMwManageProcess")
'g_cHFMSession is an HFMwSession object reference
cHFMPProcMan.SetWebSession g_cHFMSession
If cHFMPProcMan.GetReviewLevel(sScen, sYr, sPer, sEnt, sPar, _
    sVal) <> PROCESS_FLOW_STATE_NOT_STARTED Then
    cHFMPProcMan.GetHistory sScen, sYr, sPer, sEnt, sPar, sVal, _
        vaTime, vaUser, vaAction, vaLevel, vaComment
    Response.Write "<table cellpadding=" & chr(34) & "7em" & _
        chr(34) & ">"
    Response.Write "<tr>" & "<td><b>Date/Time</b></td>"
    Response.Write "<td><b>User</b></td>" & "<td><b>Action</b></td>"
    Response.Write "<td><b>Review Level</b></td>"
    Response.Write "<td><b>Comment</b></td>" & "</tr>"
    For i = lBound(vaTime) to uBound(vaTime)
        Response.Write "<tr>"
        Response.Write "<td>" & CDate(vaTime(i)) & "</td>"
        Response.Write "<td>" & vaUser(i) & "</td>"
        'getActionString() is not part of the API, is described below
        Response.Write "<td>" & getActionString(vaAction(i)) & "</td>"
        Response.Write "<td>" & _
            cHFMPProcMan.MapReviewLevelToString(vaLevel(i)) & "</td>"
        Response.Write "<td>" & vaComment(i) & "</td>"
        Response.Write "</tr>"
    Next
    Response.Write "</table>"
Else
    Response.Write "<p>No actions taken for the process unit"
End If
End Sub
```

Note: In the example, the action descriptions printed to the browser are obtained by passing the action array items returned by `GetHistory` to the following custom function (`getActionString`).

```
Function getActionString(lLevelId)
Select Case lLevelId
Case PROCESS_FLOW_ACTION_APPROVE
    getActionString = "Approve"
Case PROCESS_FLOW_ACTION_PROMOTE
    getActionString = "Promote"
Case PROCESS_FLOW_ACTION_PUBLISH
    getActionString = "Publish"
Case PROCESS_FLOW_ACTION_REJECT
    getActionString = "Reject"
Case PROCESS_FLOW_ACTION_SIGN_OFF
    getActionString = "Sign off"
Case PROCESS_FLOW_ACTION_START
```

```

        getActionString = "Start"
    Case PROCESS_FLOW_ACTION_SUBMIT
        getActionString = "Submit"
    End Select
End Function

```

GetHistory2

Returns arrays that represent the timestamps, usernames, action types, review levels, comments, and names and paths of document attachments for the process management actions of the specified process unit.

Syntax

```

<HFMwManageProcess>.GetHistory2 bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, pvaradTime, pvarabstrUser, pvarasAction,
pvarasNewState, pvarabstrComment, pvaravarabstrPaths, pvaravarabstrFiles

```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>pvaradTime</i>	Returns an array of the timestamps for the process management actions. Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time. The input value of the argument is ignored. Input/output argument.
<i>pvarabstrUser</i>	Returns an array containing the usernames for the process management actions. The input value of the argument is ignored. Input/output argument.

Argument	Description
<i>pvarasAction</i>	<p>Returns an array of the action types of the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Action Constants” on page 433.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvarasNewState</i>	<p>Returns an array of the review levels resulting from the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434.</p> <p>Tip: To return the string equivalent of an array item, pass the item to MapReviewLevelToString.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvarabstrComment</i>	<p>Returns an array of the comments specified for the process management actions.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvaravarabstrPaths</i>	<p>Returns an array of arrays that contain the paths of attached documents for each action.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvaravarabstrFiles</i>	<p>Returns an array of arrays that contain the names of attached documents for each action.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>

GetHistory2UsingPhaseId

Returns arrays containing the history of a submission phase in a process unit. The arrays have a one-to-one correspondence.

Syntax

```
<HFMwManageProcess>.GetHistory2UsingPhaseId bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, lPhaseID, pvaradTime,
pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrComment,
pvaravarabstrPaths, pvaravarabstrFiles
```

Argument	Description
<i>bstrScenario</i>	<p>The label of the process unit's Scenario dimension member.</p> <p>Input argument. String subtype.</p>
<i>bstrYear</i>	<p>The label of the process unit's Year dimension member.</p> <p>Input argument. String subtype.</p>

Argument	Description
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>lPhaseID</i>	The phase ID of the submission phase. Input argument. Long subtype.
<i>pvaradTime</i>	Returns an array of the timestamps for the process management actions. Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time. The input value of the argument is ignored. Input/output argument.
<i>pvarabstrUser</i>	Returns an array containing the usernames for the process management actions. The input value of the argument is ignored. Input/output argument.
<i>pvarasAction</i>	Returns an array of the action types of the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Action Constants” on page 433 . The input value of the argument is ignored. Input/output argument.
<i>pvarasNewState</i>	Returns an array of the review levels resulting from the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434 . Tip: To return the string equivalent of an array item, pass the item to MapReviewLevelToString . The input value of the argument is ignored. Input/output argument.
<i>pvarabstrComment</i>	Returns an array of the comments specified for the process management actions. The input value of the argument is ignored. Input/output argument.
<i>pvaravarabstrPaths</i>	Returns an array of arrays that contain the paths of attached documents for each action. The input value of the argument is ignored. Input/output argument.

Argument	Description
<i>pvaravarabstrFiles</i>	Returns an array of arrays that contain the names of attached documents for each action. The input value of the argument is ignored. Input/output argument.

GetPhaseSubmissionGroupAndPhaseForCell

Returns the submission group and phase ID for the specified cell.

Syntax

```
<HFMwManageProcess>.GetPhaseSubmissionGroupAndPhaseForCell bstrScenario,
bstrYear, bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount,
bstrICP, bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, pvarbstrGroup,
pvarbstrPhase
```

Argument	Description
<i>bstrScenario</i>	The label of the cell's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the cell's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the cell's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the cell's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the cell's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	The label of the cell's Account dimension member. Input argument. String subtype.
<i>bstrICP</i>	The label of the cell's Intercompany Partner dimension member. Input argument. String subtype.
<i>bstrCustom1</i>	The label of the cell's Custom1 dimension member. Input argument. String subtype.
<i>bstrCustom2</i>	The label of the cell's Custom2 dimension member. Input argument. String subtype.

Argument	Description
<i>bstrCustom3</i>	The label of the cell's Custom3 dimension member. Input argument. String subtype.
<i>bstrCustom4</i>	The label of the cell's Custom4 dimension member. Input argument. String subtype.
<i>pvarbstrGroup</i>	Returns the cell's submission group. Input/output argument.
<i>pvarbstrPhase</i>	Returns the cell's phase ID. Input/output argument.

GetPhaseSubmissionReviewLevel

Returns the review level of a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.GetPhaseSubmissionReviewLevel(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.

Argument	Description
----------	-------------

<i>bstrAccount</i>	Pass one of the following:
--------------------	----------------------------

- The label of the cell's Account dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>bstrICP</i>	Pass one of the following:
----------------	----------------------------

- The label of the cell's Intercompany Partner dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>bstrCustom1</i>	Pass one of the following:
--------------------	----------------------------

- The label of the cell's Custom1 dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>bstrCustom2</i>	Pass one of the following:
--------------------	----------------------------

- The label of the cell's Custom2 dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>bstrCustom3</i>	Pass one of the following:
--------------------	----------------------------

- The label of the cell's Custom3 dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>bstrCustom4</i>	Pass one of the following:
--------------------	----------------------------

- The label of the cell's Custom4 dimension member.
- MEMBERNOTUSED

Input argument. String subtype.

<i>IPhaseID</i>	Pass one of the following:
-----------------	----------------------------

- The phase ID.
- MEMBERNOTUSED

Input argument. Long subtype.

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

GetProcessManagementRoleAccess

Returns an array listing the process management access rights for the connected user.

Syntax

```
<HFMwManageProcess>.GetProcessManagementRoleAccess  
pvaravbAccessRightsToProcessManagementRoles
```

Argument	Description
<i>pvaravbAccessRightsToProcessManagementRoles</i>	Returns an array that represents the Submitter, Supervisor, and Reviewer 1–10 roles. Interpret the array as follows: <ul style="list-style-type: none">● An array item containing a non-zero value indicates that the user has rights to the role identified by the value. Valid values are represented by the HFMConstants type library constants listed in “Role Constants” on page 438 that represent roles.● An array item with a value of zero indicates that the user does not have access to a role. Input/output argument.

GetReviewLevel

Returns the review level for the specified process unit.

Syntax

```
<HFMwManageProcess>.GetReviewLevel (bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrParent, bstrValue)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.

Return Value

Returns a number that represents the process unit's review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example prints the process unit's review level to the browser.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.GetReviewLevel ("Budget", "2003", _
"April", "Connecticut", "UnitedStates", "<Entity Currency>")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

GetReviewLevelUsingPhaseID

Returns the review level for a submission phase in a process unit.

Syntax

```
<HFMwManageProcess>.GetReviewLevelUsingPhaseID(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, lPhaseID)
```

Argument	Description
----------	-------------

<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
---------------------	---

<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
-----------------	---

<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
-------------------	---

<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
-------------------	---

<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
-------------------	---

<i>bstrValue</i>	The label of the process unit's Value dimension member.
------------------	---

Argument	Description
	Input argument. String subtype.
<i>lPhaseID</i>	The phase ID of the submission phase. Input argument. Long subtype.

Return Value

Returns a number that represents the review level of the submission phase. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

MapReviewLevelToString

Returns a string that represents a process management review level, given a valid review level ID.

Many of the HFMwManageProcess component’s methods return numerical IDs of review levels. Use [MapReviewLevelToString](#) to obtain the string equivalents of these methods’ return values.

Note: The strings returned are non-localized. In other words, the strings are in English.

Syntax

```
<HFMwManageProcess>.MapReviewLevelToString (lReviewLevel)
```

Argument	Description
----------	-------------

<i>lReviewLevel</i>	A valid review level ID. Input argument. Long subtype.
---------------------	---

Return Value

Returns a string that represents the passed review level ID.

Example

[MapReviewLevelToString](#) is used in the example for [GetReviewLevel](#).

PhasedSubmissionProcessManagementChangeStateForMultipleEntities

Applies a given process management action and attaches documents to submission phases for multiple Entity dimension members, given the member IDs of cells in the phases. The member IDs of the cells' Entity, Account, Intercompany Partner, and Custom dimension members are passed in arrays that have a one-to-one correspondence.

Syntax

```
<HFMwManageProcess>.PhasedSubmissionProcessManagementChangeStateForMultipleEntities(bstrScenario, bstrYear, bstrPeriod, varabstrEntity, varabstrParent, bstrValue, varabstrAccount, varabstrICP, varabstrCustom1, varabstrCustom2, varabstrCustom3, varabstrCustom4, bstrComment, lAction, vbUseAllValueMembers, vbApplyToAllPeriods, sProcessStateToPromoteTo, varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>varabstrEntity</i>	An array containing the labels of the process units' child Entity dimension members. Input argument.
<i>varabstrParent</i>	An array containing the labels of the process units' parent Entity dimension members. Note: The <i>varabstrEntity</i> and <i>varabstrParent</i> arguments' arrays have a one-to-one correspondence. Input argument.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>varabstrAccount</i>	The member IDs of the cells' Account dimension members. Input argument.
<i>varabstrICP</i>	The member IDs of the cells' Intercompany Partner dimension members. Input argument.
<i>varabstrCustom1</i>	The member IDs of the cells' Custom1 dimension members. Input argument.
<i>varabstrCustom2</i>	The member IDs of the cells' Custom2 dimension members. Input argument.

Argument	Description
<i>varabstrCustom3</i>	The member IDs of the cells' Custom3 dimension members. Input argument.
<i>varabstrCustom4</i>	The member IDs of the cells' Custom4 dimension members. Input argument.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>lAction</i>	The action to apply to the submission phases. Valid values are represented by the HFMConstants type library constants listed in “Process Management Action Constants” on page 433 . Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases. Input argument. Boolean subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether to apply the action to submission phases containing base periods that intersect with the specified dimension members. Pass TRUE to apply the action to all such submission phases. Input argument. Boolean subtype.
<i>sProcessStateToPromoteTo</i>	The process management level to apply to the process units. Valid values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434 . Input argument. Integer subtype.
<i>varabstrPaths</i>	The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	The file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phases' review level. See [“Process Management Review Level Constants” on page 434](#).

PhaseSubmissionApprove

Approves a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionApprove(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,
vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths,
varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Account dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Intercompany Partner dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Custom1 dimension member.

Argument	Description
	<ul style="list-style-type: none"> ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom2</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom3</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom4</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>lPhaseID</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED <p>Input argument. Long subtype.</p>
<i>vbUseAllValueMembers</i>	<p>Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases.</p> <p>Input argument. Boolean subtype.</p>
<i>vbApplyToAllPeriods</i>	<p>Specifies whether to apply the action to submission phases containing base periods that intersect with the specified dimension members. Pass TRUE to apply the action to all such submission phases.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrAnnotation</i>	<p>A user comment for the action, or an empty string if there is no comment.</p> <p>Input argument. String subtype.</p>
<i>varabstrPaths</i>	<p>The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\).</p> <p>This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames.</p> <p>Input argument.</p>
<i>varabstrFiles</i>	<p>The file names of the documents to attach.</p> <p>Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown.</p> <p>Input argument.</p>

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

PhaseSubmissionGetHistory

Returns the history of a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionGetHistory bstrScenario, bstrYear,  
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,  
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID, pvaradTime,  
pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation,  
pvaravarabstrPaths, pvaravarabstrFiles
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Account dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following:

Argument	Description
	<ul style="list-style-type: none"> ● The label of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom1</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom1 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom2</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom3</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom4</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>lPhaseID</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED <p>Input argument. Long subtype.</p>
<i>pvaradTime</i>	<p>Returns an array of the timestamps for the process management actions.</p> <p>Note: Array items are in a Double format. To render user-readable dates and times, you must perform conversions. For example, in VBScript, passing an array item to <code>CDate</code> returns a user-readable date and time.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvarabstrUser</i>	<p>Returns an array containing the usernames for the process management actions.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvarasAction</i>	<p>Returns an array of the action types of the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Action Constants” on page 433.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>

Argument	Description
<i>pvarasNewState</i>	<p>Returns an array of the review levels resulting from the process management actions. The return values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434.</p> <p>Tip: To return the string equivalent of an array item, pass the item to MapReviewLevelToString.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvarabstrAnnotation</i>	<p>Returns an array of the comments specified for the process management actions.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvaravarabstrPaths</i>	<p>Returns an array of arrays that contain the paths of attached documents for each action.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>
<i>pvaravarabstrFiles</i>	<p>Returns an array of arrays that contain the names of attached documents for each action.</p> <p>The input value of the argument is ignored.</p> <p>Input/output argument.</p>

PhaseSubmissionPromote

Promotes a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionPromote(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,
vbUseAllValueMembers, bstrAnnotation, lProcessStateToPromoteTo,
varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	<p>The label of the process unit's Scenario dimension member.</p> <p>Input argument. String subtype.</p>
<i>bstrYear</i>	<p>The label of the process unit's Year dimension member.</p> <p>Input argument. String subtype.</p>

Argument	Description
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Account dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom1 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom2</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom3</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom4</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>IPhaseID</i>	Pass one of the following: <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED

Argument	Description
	Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases. Input argument. Boolean subtype.
<i>bstrAnnotation</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>lProcessStateToPromoteTo</i>	The review level to apply to the submission phases. Valid values are represented by the constants listed in “Process Management Review Level Constants” on page 434 . Input argument. Long subtype.
<i>varabstrPaths</i>	The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument’s array of filenames. Input argument.
<i>varabstrFiles</i>	The file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument’s array, otherwise an error will be thrown. Input argument.

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

PhaseSubmissionReject

Rejects a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionReject(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,
vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Account dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom1 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom2</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom3</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom4</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED

Argument	Description
	Input argument. String subtype.
<i>lPhaseID</i>	Pass one of the following: <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases. Input argument. Boolean subtype.
<i>bstrAnnotation</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	The file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

PhaseSubmissionSignOff

Signs off on a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionSignOff bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,
vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Account dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom1 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom2</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom3</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom4</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED

Argument	Description
	Input argument. String subtype.
<i>lPhaseID</i>	Pass one of the following: <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED Input argument. Long subtype.
<i>vbUseAllValueMembers</i>	Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases. Input argument. Boolean subtype.
<i>bstrAnnotation</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	The file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

PhaseSubmissionStart

Starts a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionStart(bstrScenario, bstrYear,
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,
vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths,
varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.

Argument	Description
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Account dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom1 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom2</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom2 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom3</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom3 dimension member. ● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom4</i>	Pass one of the following: <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED Input argument. String subtype.

Argument	Description
<i>IPhaseID</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> • The phase ID. • MEMBERNOTUSED <p>Input argument. Long subtype.</p>
<i>vbUseAllValueMembers</i>	<p>Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases.</p> <p>Input argument. Boolean subtype.</p>
<i>vbApplyToAllPeriods</i>	<p>Specifies whether to apply the action to submission phases containing base periods that intersect with the specified dimension members. Pass TRUE to apply the action to all such submission phases.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrAnnotation</i>	<p>A user comment for the action, or an empty string if there is no comment.</p> <p>Input argument. String subtype.</p>
<i>varabstrPaths</i>	<p>The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\).</p> <p>This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames.</p> <p>Input argument.</p>
<i>varabstrFiles</i>	<p>The file names of the documents to attach.</p> <p>Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown.</p> <p>Input argument.</p>

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

PhaseSubmissionSubmit

Submits a submission phase, given either the member labels of a cell in the phase or the phase ID. You must pass one of the following:

- If you use a phase ID, pass the HFMConstant MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you pass member labels to the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HFMwManageProcess>.PhaseSubmissionSubmit(bstrScenario, bstrYear,  
bstrPeriod, bstrEntity, bstrParent, bstrValue, bstrAccount, bstrICP,  
bstrCustom1, bstrCustom2, bstrCustom3, bstrCustom4, lPhaseID,  
vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrAccount</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Account dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrICP</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Intercompany Partner dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom1</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Custom1 dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom2</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Custom2 dimension member.● MEMBERNOTUSED Input argument. String subtype.
<i>bstrCustom3</i>	Pass one of the following: <ul style="list-style-type: none">● The label of the cell's Custom3 dimension member.

Argument	Description
	<ul style="list-style-type: none"> ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>bstrCustom4</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The label of the cell's Custom4 dimension member. ● MEMBERNOTUSED <p>Input argument. String subtype.</p>
<i>lPhaseID</i>	<p>Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID. ● MEMBERNOTUSED <p>Input argument. Long subtype.</p>
<i>vbUseAllValueMembers</i>	<p>Specifies whether to apply the action to submission phases for Value dimension members related to the <i>bstrValue</i> member. Pass TRUE to apply the action to related submission phases.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrAnnotation</i>	<p>A user comment for the action, or an empty string if there is no comment.</p> <p>Input argument. String subtype.</p>
<i>varabstrPaths</i>	<p>The paths in which the documents to be attached have been loaded. Folders in the path are delimited by backslashes (\).</p> <p>This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames.</p> <p>Input argument.</p>
<i>varabstrFiles</i>	<p>The file names of the documents to attach.</p> <p>Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown.</p> <p>Input argument.</p>

Return Value

Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See [“Process Management Review Level Constants” on page 434](#). Long subtype.

Promote

Performs a promote action on the specified process unit.

To attach documents when promoting, use [Promote2](#).

Syntax

```
<HFMwManageProcess>.Promote (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, bstrComment, lProcessStateToPromoteTo)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>lProcessStateToPromoteTo</i>	The review level to which the process unit is being promoted. Valid values are represented by the constants listed in “Process Management Review Level Constants” on page 434 . Input argument. Long subtype.

Return Value

Returns a number that represents the review level to which the process unit was promoted. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example promotes the specified process unit to review level 8.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Promote ("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", _
" ", PROCESS_FLOW_STATE_REVIEW8)
```

Promote2

Performs a promote action on the specified process unit and optionally attaches documents.

Syntax

```
<HFMwManageProcess>.Promote2 (bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrParent, bstrValue, bstrComment, lProcessStateToPromoteTo,  
varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>lProcessStateToPromoteTo</i>	The review level to which the process unit is being promoted. Valid values are represented by the constants listed in “Process Management Review Level Constants” on page 434 . Input argument. Long subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the review level to which the process unit was promoted. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example promotes the specified process unit to review level 7 and attaches two documents.

```
Dim iState, saPaths(1), saNames(1)
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'CHFMProcMan is an HFMwManageProcess object reference
iState = CHFMProcMan.Promote2("Budget", "2004", "August", "NewYork", _
    "UnitedStates", "<EntityCurrency>", "see attachments", _
    PROCESS_FLOW_STATE_REVIEW7, saPaths, saNames)
```

Publish

Performs a publish action on the specified process unit. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

To attach documents while publishing an action, use [Publish2](#).

Syntax

```
<HFMwManageProcess>.Publish (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument.

Argument	Description
	Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units. Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Return Value

Returns a number that represents the process unit's review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example performs a publish action for the specified process unit and prints the resulting review level to the browser.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Publish("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", FALSE, _
"")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

Publish2

Performs a publish action on the specified process unit and optionally attaches documents. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

Syntax

```
<HFMwManageProcess>.Publish2 (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment,
varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units. Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument. Boolean subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the process unit's review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Reject

Performs a reject action on the specified process unit.

To attach documents while performing a rejection, use [Reject2](#).

Syntax

```
<HFMwManageProcess>.Reject (bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrParent, bstrValue, bstrComment)
```

Argument	Description
----------	-------------

<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Return Value

Returns a number that represents the review level in which the rejection results. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example rejects the specified process unit and prints the resulting review level to the browser.

```
Dim cHFMProcMan, lLevel  
Set cHFMProcMan = _  
Server.CreateObject("Hyperion.HFMwManageProcess")
```

```
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Reject("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", _
"Exceeds new budget.")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

Reject2

Performs a reject action on the specified process unit and optionally attaches documents.

Syntax

```
<HFMwManageProcess>.Reject2 (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, bstrComment, varabstrPaths,
varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the review level in which the rejection results. Valid values are represented by the constants listed in [“Process Management Review Level Constants”](#) on page 434. Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

SetWebSession

Associates an HFMwManageProcess object with the HFMwSession object for an application.

Syntax

```
<HFMwManageProcess>.SetWebSession varpIUnkHFMwSession
```

Argument	Description
<i>varpIUnkHFMwSession</i>	The HFMwSession object for the application. Input argument.

Example

SetWebSession is used in the example for [GetReviewLevel](#).

SignOff

Performs a sign off action on the specified process unit.

To attach documents while signing off on an action, use [SignOff2](#).

Syntax

```
<HFMwManageProcess>.SignOff bstrScenario, bstrYear, bstrPeriod,  
bstrEntity, bstrParent, bstrValue, bstrComment
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member.

Argument	Description
	Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Example

The following example performs a sign off action for the specified process unit.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
cHFMProcMan.SignOff "Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", _
"OK"
```

SignOff2

Performs a sign off action on the specified process unit and optionally attaches documents.

Syntax

```
<HFMwManageProcess>.SignOff2 bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, bstrComment, varabstrPaths,
varabstrFiles
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument.

Argument	Description
	Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Start

Performs a start action for the specified process units. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

To attach documents when starting, use [Start2](#).

Syntax

```
<HFMwManageProcess>.Start (bstrScenario, bstrYear, bstrPeriod, bstrEntity,
bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units.

Argument	Description
	Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Return Value

Returns a number that represents the review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example starts the specified process unit and prints the resulting review level to the browser.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Start("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", FALSE, _
"Annotation")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

Start2

Performs a start action for the specified process units and optionally attaches documents. The *vbApplyToAllPeriods* argument determines whether the action applies to one or to multiple process units.

Syntax

```
<HFMwManageProcess>.Start2 (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, vbApplyToAllPeriods, bstrComment,
varabstrPaths, varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member for the process units.

Argument	Description
	Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member for the process units. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the Period dimension member for the process units. Input argument. String subtype.
<i>bstrEntity</i>	The label of the Entity dimension member for the process units. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the process units. Input argument. String subtype.
<i>vbApplyToAllPeriods</i>	Specifies whether the action should be applied to process units for all input periods that intersect the other specified dimension members. Pass TRUE to apply the action to all these process units, FALSE to apply the action to only the specified process unit. Input argument. Boolean subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames. Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example starts the specified process unit and attaches two files.

```
Dim iState, saPaths(1), saNames(1)
```

```

saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'cHFMProcMan is an HFMwManageProcess object reference
iState = cHFMProcMan.Start2("Budget", "2004", "August", "NewYork", _
    "UnitedStates", "<EntityCurrency>", False, "see attachments", _
    saPaths, saNames)

```

Submit

Performs a submit action on the specified process unit.

To attach documents while submitting a process unit, use [Submit2](#).

Syntax

```

<HFMwManageProcess>.Submit (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, bstrComment)

```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.

Return Value

Returns a number that represents the process unit's review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example submits the specified process unit and prints the resulting review level to the browser.

```
Dim cHFMProcMan, lLevel
Set cHFMProcMan = _
Server.CreateObject("Hyperion.HFMwManageProcess")
' cHFMSession is a previously set HFMwSession object
cHFMProcMan.SetWebSession cHFMSession
lLevel = cHFMProcMan.Submit("Budget", "2003", "April", _
"Connecticut", "UnitedStates", "<Entity Currency>", _
"")
Response.Write "<p>Review level: " & _
cHFMProcMan.MapReviewLevelToString(lLevel) & "</p>"
```

Submit2

Performs a submit action on the specified process unit and optionally attaches documents.

Syntax

```
<HFMwManageProcess>.Submit2 (bstrScenario, bstrYear, bstrPeriod,
bstrEntity, bstrParent, bstrValue, bstrComment, varabstrPaths,
varabstrFiles)
```

Argument	Description
<i>bstrScenario</i>	The label of the process unit's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the process unit's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the process unit's Period dimension member. Input argument. String subtype.
<i>bstrEntity</i>	The label of the process unit's Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the parent of the entity specified in the <i>bstrEntity</i> argument. Input argument. String subtype.
<i>bstrValue</i>	The label of the process unit's Value dimension member. Input argument. String subtype.
<i>bstrComment</i>	A user comment for the action, or an empty string if there is no comment. Input argument. String subtype.
<i>varabstrPaths</i>	An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\). This array has a one-to-one correspondence with the <i>varabstrFiles</i> argument's array of filenames.

Argument	Description
	Input argument.
<i>varabstrFiles</i>	An array of strings containing the file names of the documents to attach. Note: Files must be loaded in the folder specified by the corresponding item in the <i>varabstrPaths</i> argument's array, otherwise an error will be thrown. Input argument.

Return Value

Returns a number that represents the process unit's review level. Valid values are represented by the constants listed in [“Process Management Review Level Constants” on page 434](#). Long subtype.

Tip: You can obtain a string that represents the review level by passing the return value to [MapReviewLevelToString](#).

Example

The following example submits a process unit and attaches two documents.

```
Dim iState, saPaths(1), saNames(1)
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'CHFMProcMan is an HFMwManageProcess object reference
iState = CHFMProcMan.Submit2("Budget", "2004", "August", "NewYork", _
    "UnitedStates", "<EntityCurrency>", "see attachments", _
    saPaths, saNames)
```

HFMwCalculate Component

The HFMwCalculate component enables you to execute consolidations, translations, calculations, and allocations. This component contains the methods described in the following topics.

Obtaining an HFMwCalculate Object Reference

Create an HFMwCalculate object with `Server.CreateObject`. After creating the object, call [SetWebSession](#) to associate the object with the HFMwSession object for the application:

```
Dim cHFMCalc
Set cHFMCalc = Server.CreateObject("Hyperion.HFMwCalculate")
' cHFMSession is a previously set HFMwSession object
cHFMCalc.SetWebSession cHFMSession
```


Allocate

Allocates an entity's data for the specified Scenario, Year, Period, and Value dimension members.

Syntax

```
<HFMwCalculate>.Allocate bstrScenario, bstrYear, bstrStartPeriod,  
bstrEndPeriod, bstrEntity, bstrParent, bstrValue
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>bstrStartPeriod</i>	The label of the first Period dimension member in the range of periods for which the allocation is being executed. Input argument. String subtype.
<i>bstrEndPeriod</i>	The label of the last Period dimension member in the range of periods for which the allocation is being executed. Input argument. String subtype.
<i>bstrEntity</i>	The label of the child Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the <i>bstrEntity</i> argument's parent. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member. Input argument. String subtype.

ChartLogic

Calculates an entity's data for the specified Scenario, Year, Period, and Value dimension members.

Syntax

```
<HFMwCalculate>.ChartLogic bstrScenario, bstrYear, bstrStartPeriod,  
bstrEndPeriod, bstrEntity, bstrParent, bstrValue, varbForceChartLogic
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member.

Argument	Description
	Input argument. String subtype.
<i>bstrStartPeriod</i>	The label of the first Period dimension member in the range of periods for which the calculation is being executed. Input argument. String subtype.
<i>bstrEndPeriod</i>	The label of the last Period dimension member in the range of periods for which the calculation is being executed. Input argument. String subtype.
<i>bstrEntity</i>	The label of the child Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the <i>bstrEntity</i> argument's parent. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member. Input argument. String subtype.
<i>varbForceChartLogic</i>	Specifies whether to force the calculation if the entity does not require calculation. Specify TRUE to force the calculation, FALSE otherwise. Input argument. Boolean subtype.

Consolidate

Consolidates an entity's data for the specified Scenario, Year, and Period dimension members.

Syntax

```
<HFMwCalculate>.Consolidate bstrScenario, bstrYear, bstrStartPeriod,
bstrEndPeriod, bstrEntity, bstrParent, lConsolidationType
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>bstrStartPeriod</i>	The label of the first Period dimension member in the range of periods for which the consolidation is being executed. Input argument. String subtype.
<i>bstrEndPeriod</i>	The label of the last Period dimension member in the range of periods for which the consolidation is being executed. Input argument. String subtype.
<i>bstrEntity</i>	The label of the child Entity dimension member.

Argument	Description
	Input argument. String subtype.
<i>bstrParent</i>	The label of the <i>bstrEntity</i> argument's parent. Input argument. String subtype.
<i>IConsolidationType</i>	Specifies the type of consolidation to be performed. Valid values are represented by the HFMConstants type library constants listed in “Consolidation Type Constants” on page 437 . Input argument. Long subtype.

Example

The following function takes dimension member labels that are passed to Consolidate and performs a consolidation for only those entities that contain data.

```
Function ConsolidateAllData(sScen, sYear, sStartPer, sEndPer, _
sChild, sPar)
Dim cHFMCalc
Set cHFMCalc = Server.CreateObject("Hyperion.HFMwCalculate")
' cHFMSession is a previously set HFMwSession object
cHFMCalc.SetWebSession cHFMSession
cHFMCalc.Consolidate sScen, sYear, sStartPer, sEndPer, sChild, sPar,
CONSOLIDATE_ALLWITHDATA
End Function
```

SetWebSession

Associates an HFMwCalculate object with the HFMwSession object for an application.

Syntax

```
<HFMwCalculate>.SetWebSession varpIUnkHFMwSession
```

Argument	Description
<i>varpIUnkHFMwSession</i>	The HFMwSession object for the application. Input argument.

Example

SetWebSession is used in the example for [Consolidate](#).

Translate

Translates an entity's data from one currency to another for the specified Scenario, Year, Period, and Value dimension members.

Note: Translate uses the default exchange rates defined for an application. If an application has a translation rule, Translate executes the rule instead of applying the default exchange rates.

Syntax

```
<HFMwCalculate>.Translate bstrScenario, bstrYear, bstrStartPeriod,  
bstrEndPeriod, bstrEntity, bstrParent, bstrValue, varbForceTranslate
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>bstrStartPeriod</i>	The label of the first Period dimension member in the range of periods for which the translation is being executed. Input argument. String subtype.
<i>bstrEndPeriod</i>	The label of the last Period dimension member in the range of periods for which the translation is being executed. Input argument. String subtype.
<i>bstrEntity</i>	The label of the child Entity dimension member. Input argument. String subtype.
<i>bstrParent</i>	The label of the <i>bstrEntity</i> argument's parent. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member. Input argument. String subtype.
<i>varbForceTranslate</i>	Determines whether the translation will be forced. Specify TRUE to force the translation, FALSE otherwise. Input argument. Boolean subtype.



HFMwMbrSel Type Library

In This Chapter

HFMwMbrSel Component	189
HFMwMbrSelDim Component.....	199

The HFMwMbrSel type library enables you to get and set Points of View. For example, you can use this type library to persist Point of View settings beyond a session or to create a form in which users can specify Points of View.

The HFMwMbrSel type library contains the HFMwMbrSel and HFMwMbrSelDim components. An HFMwMbrSel instance represents a Point of View, and an HFMwMbrSelDim instance represents a dimension in an HFMwMbrSel instance's collection of dimensions.

Note: You can get or set a dimension's attributes with either the HFMwMbrSel or HFMwMbrSelDim component.

HFMwMbrSel Component

The HFMwMbrSel component provides properties and methods for working with Points of View. For example, HFMwMbrSel enables you to get and set Point of View attributes, specify filters, reset all filtering properties to their defaults, and build and parse valid dimension values. For details, see [“HFMwMbrSel Methods” on page 190](#) and [“HFMwMbrSel Properties” on page 192](#).

The properties `AttrAll` and `AttrDim` get and set various Point of View attributes, such as the dimension members in a Point of View. These attributes and the constants that represent them are described in [“Point of View Attribute Constants” on page 459](#).

Tip: You can persist a Point of View by passing the string returned by `XmlString` to `HFMwSession.SetPOVByTag`. You can set an HFMwMbrSel instance to a persisted Point of View by setting `xmlString` to the string returned by `HFMwSession.GetPOVByTag`.

Obtaining an HFMwMbrSel Object Reference

Create an HFMwMbrSel object with `Server.CreateObject`. After creating the object, call `setWebSession` to associate the object with the HFMwSession object for the application:

```
Dim cMbrSel
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
```

HF Mw Mbr Sel Methods

The HF Mw Mbr Sel component contains the methods described in the following topics.

buildValue

Returns a syntactically valid representation of a member list, dimension member, or function.

When specifying dimension values, you must follow the same syntax rules for specifying member lists, members, and functions that the user interface requires. Using `buildValue` enables you to build valid dimension values without having to perform tasks such as concatenating delimiters and required special characters.

For example, if you pass a member list name and a top member, `buildValue` returns a dimension value that includes the braces ({ }) required to specify member lists and the period (.) that delimits the labels of the top member and the member list.

Tip: You can use the return value to specify dimension values with `AttrAll`, `AttrDim`, and `HF Mw Mbr SelDim.Attr`.

Syntax

```
<HF Mw Mbr Sel>.buildValue (lEnumType, bstrMajor, bstrMinor)
```

Argument Description

lEnumType A flag that indicates the type of dimension value to return. Pass one of the HF M Constants type library constants listed in [“Dimension Value Type Constants” on page 459](#).

Input argument. Long subtype.

bstrMajor The label of the member list, dimension member, or function.

Input argument. String subtype.

bstrMinor Pass one of the following values:

- For dimension members, the label of the parent member, or a blank string if no parent is needed.
- For member lists, the label of the top member, or a blank string if you are not specifying a top member.
- For functions, the value for the parameter, or a blank string if there is no parameter.

Input argument. String subtype.

Return Value

Returns a string that contains the dimension value.

Example

The following example returns the string "{Quarter1.[Hierarchy]}".

```
Dim cMbrSel, vRet
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
vRet = cMbrSel.buildValue(WEBOM_MBRSEL_VALTYPE_LIST, "[Hierarchy]", _
    "Quarter1")
```

clearFilters

Sets the HFMwMbrSel instance's filtering and sorting properties to their default values of FALSE.

Note: The filtering and sorting properties are [EnabledOnly](#), [SortedOnly](#), [ValuedOnly](#), and [VisibleOnly](#).

Syntax

```
<HFMwMbrSel>.clearFilters
```

parseValue

Parses a dimension value into its label and type. The method also returns the dimension value's top member, parent, or parameter, depending upon whether the dimension value is a member list, member label, or function.

parseValue also indicates whether the syntax of the specified dimension value is valid. You should always check the return value. If a dimension value does not consist of valid syntax, the information returned in the output parameters may not be valid, since valid syntax is required for accurate parsing.

Syntax

```
<HFMwMbrSel>.parseValue (bstrValue, pvarbstrMajor, pvarbstrMinor,
    pvarlEnumType)
```

Argument	Description
<i>bstrValue</i>	The dimension value. Input argument. String subtype.
<i>pvarbstrMajor</i>	Returns the label of the member, member list, or function. Input/output argument.
<i>pvarbstrMinor</i>	Returns one of the following values: <ul style="list-style-type: none">For dimension members, the label of the parent member, or a blank string if the dimension value does not contain a parent.For member lists, the label of the top member, or a blank string if the dimension value does not contain a top member.

Argument	Description
	<ul style="list-style-type: none"> For functions, the value for the parameter, or a blank string if the dimension value does not contain a parameter. <p>Input/output argument.</p>
<i>pvarlEnumType</i>	<p>Indicates the type of dimension value passed to the <i>bstrValue</i> parameter. Valid values are represented by the HFMConstants type library constants listed in “Dimension Value Type Constants” on page 459.</p> <p>Input/output argument.</p>

Return Value

Indicates whether the *bstrValue* parameter contains a valid value. Returns TRUE if the value is valid, FALSE otherwise.

Example

The following snippet returns the member list label [Hierarchy], the top member Quarter1, and the dimension value type of member list.

```
Dim cMbrSel, vRet, sMajor, sMinor, lType
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
vRet = cMbrSel.parseValue("{Quarter1.[Hierarchy]}", sMajor, sMinor, _
    lType)
```

setWebSession

Associates an HFMwMbrSel object with the HFMwSession object for an application.

Syntax

```
<HFMwMbrSel>.setWebSession varidispSession
```

Argument	Description
----------	-------------

varidispSession The HFMwSession object for the application.

Input argument.

Example

setWebSession is used in the example for [buildValue](#).

HFMwMbrSel Properties

The HFMwMbrSel component contains the properties described in the following topics.

AttrAll

Gets values of a given attribute for the dimensions represented by the HFMwMbrSel instance and sets values of a given attribute for one or more dimensions. The value for this property varies depending upon whether you are getting or setting:

- If you are setting, you can either set all the dimensions represented by the HFMwMbrSel instance to one value, or set a value for each dimension by passing a dynamic array of attribute values.

Caution! The array must be a dynamic array, not a fixed array.

- If you are getting, `AttrAll` returns a dynamic array of values for the specified attribute. The array contains an item for each dimension in the HFMwMbrSel instance.

In the array that the property takes or returns, array items are indexed by the HFMConstants type library constants listed in [“Dimension ID Constants” on page 414](#).

Read-write.

Tip: To get or set an attribute value for a specific dimension, you can also use [AttrDim](#).

Read-write.

Syntax

```
<HFMwMbrSel>.AttrAll(lidAttr)
```

Argument	Description
<i>lidAttr</i>	The ID of the attribute. Valid values are represented by the HFMConstants type library constants listed in “Point of View Attribute Constants” on page 459 . Input argument. Long subtype.

Example

`AttrAll` is used in the example for [XmlString](#).

AttrDim

Gets or sets an attribute for a given dimension.

Tip: To get or set an attribute for all dimensions, use [AttrAll](#).

Read-write.

Syntax

```
<HFMwMbrSel>.AttrDim(lndxDim, lidAttr)
```

Argument Description

<i>IndxDim</i>	The ID of the dimension. Valid values are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 414 . Input argument. Long subtype.
<i>lidAttr</i>	The ID of the attribute. Attribute IDs are represented by the HFMConstants type library constants listed in “Point of View Attribute Constants” on page 459 . Input argument. Long subtype.

Example

The following example changes the value attribute of the Entity dimension in the HFMwMbrSel instance to UnitedStates.Connecticut. [buildValue](#) builds the dimension value to which AttrDim is set.

```
Dim cMbrSel, sVal
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
sVal = cMbrSel.buildValue(WEBOM_MBRSEL_VALTYPE_MEMBER, "NewYork", _
    "UnitedStates")
cMbrSel.AttrDim(DIMENSIONENTITY, WEBOM_MBRSEL_ID_VALUE) = sVal
```

Count

Returns the number of dimensions contained by the HFMwMbrSel instance.

Note: The number of dimensions contained by an HFMwMbrSel instance is determined by filtering properties such as [EnabledOnly](#), [SortedOnly](#), [ValuedOnly](#), and [VisibleOnly](#).

Read-only.

Custom

Creates, updates, or returns a string containing a user-defined named directive for the HFMwMbrSel instance.

Tip: To get or set standard processing directives provided by HFMwMbrSel, use [Directive](#). To get or set a custom named directive for a dimension, use HFMwMbrSelDim.[Custom](#).

Read-write.

Syntax

```
<HFMwMbrSel>.Custom(bstrName)
```

Argument	Description
----------	-------------

<i>bstrName</i>	The name of the directive. Input argument. String subtype.
-----------------	---

Dim

Returns an `HFMwMbrSelDim` object reference for a given dimension. For information on this object, see [“HFMwMbrSelDim Component” on page 199](#).

Read-only.

Syntax

```
<HFMwMbrSel>.Dim(lndxDim)
```

Argument	Description
----------	-------------

<i>lndxDim</i>	The ID of the dimension. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Dimension ID Constants” on page 414 . Input argument. Long subtype.
----------------	---

Example

`Dim` is used in the example for `HFMwMbrSelDim`. [Attr](#).

Directive

Gets or sets a string containing the value of one of the standard processing directives provided by `HFMwMbrSel`.

Tip: To get or set a user-defined named directive, use [Custom](#).

Read-write.

Syntax

```
<HFMwMbrSel>.Directive(lidDirective)
```

Argument	Description
----------	-------------

<i>lidDirective</i>	The ID of the directive. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Processing Directive ID Constants” on page 461 . Input argument. Long subtype.
---------------------	--

Dump

Returns a text representation of the `HFMwMbrSel` instance. This can be useful when debugging.

Read-only.

EnabledOnly

Gets or sets whether to filter for enabled dimensions. TRUE turns the filter on, FALSE turns it off.

Tip: To enable or disable a dimension, set the enabled attribute with [AttrAll](#), [AttrDim](#), or [HFMwMbrSelDim.Attr](#).

Filters are applied if you attempt to get all dimensions from an HFMwMbrSel instance. Only dimensions that meet all the filters' criteria are returned. For example, if both [enabledOnly](#) and [ValuedOnly](#) are set to TRUE, then only enabled dimensions containing a value are returned.

Read-write.

NextSortOrder

Gets the next available sequential number in the HFMwMbrSel instance's sort order. This number is always one greater than the highest value that has been set in the dimensions' order attributes. Use this property when setting a dimension's order attribute, as [nextSortOrder](#) provides a convenient alternative to manually keeping track of dimensions' order values.

The order attribute applies only when the [SortedOnly](#) property is set to TRUE.

Note: The default value of the order attribute is -1, which indicates the dimension is not included in the sort order.

Read-only.

Example

The following subroutine sets the order attribute for a given dimension. [AttrDim](#) gets the dimension's current order attribute. If the attribute is set to the default of -1, then [nextSortOrder](#) sets it to the next number in the sort order.

```
Sub SetDimensionSortOrder(cMbrSel, nDim)
Dim nSortOrder
nSortOrder = cMbrSel.AttrDim(nDim, WEBOM_MBRSEL_ID_SORTORDER)
' If it's not yet added, put it at the end.
If nSortOrder = -1 Then
    cMbrSel.AttrDim(nDim, WEBOM_MBRSEL_ID_SORTORDER) = _
        cMbrSel.nextSortOrder
End If
End Sub
```

ScriptString

Gets or sets the HFMwMbrSel instance's dimension members with a string that conforms to the syntax used in Financial Management's rules scripts and data forms. The string returned by `scriptString` includes only dimensions for which the `value` attribute has been set.

If the object's filtering properties are set to `TRUE`, then `scriptString` generates syntax only for dimensions that match the filtering criteria.

Note: For details on the syntax used in rules scripts and data forms, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Example

The following snippet sets the members of all dimensions.

Read-write.

```
'cMbrSel is an HFMwMbrSel object reference
cMbrSel.ScriptString = "S#Actual.Y#2000.P#July.W#<Scenario View>." & _
    "E#{[Hierarchy]}.V#<Entity Currency>.A#{[Base]}.I#[ICP Top]." & _
    "C1#[None].C2#[None].C3#[None].C4#[None]"
```

SortedOnly

Gets or sets whether the HFMwMbrSel instance sorts dimensions using the system's native sort order or the dimensions' `order` attributes. `TRUE` indicates that dimensions are sorted by their `order` attributes, `FALSE` indicates that dimensions are sorted using the system's native sorting order.

Note: If `sortedOnly` is `TRUE`, dimensions are sorted from the lowest to the highest `order` value.

By default, a dimension's `order` attribute is set to -1. If `sortedOnly` is set to `TRUE`, then dimensions in which the `order` attribute is set to -1 are not returned.

Tip: To specify a dimension's `order` attribute, use [AttrDim](#), [AttrAll](#), or `HFMwMbrSelDim.Attr`. When setting this attribute, you can use the [NextSortOrder](#) property to get the next available sequential number in the sort order.

Read-write.

ValuedOnly

Gets or sets whether to filter out dimensions in which a dimension value has not been set. `TRUE` turns this filter on, `FALSE` turns it off.

Tip: Dimension values are stored in the `val` attribute. To get or set the `val` attribute, use `AttrDim`, `AttrAll`, or `HFMwMbrSelDim.Attr`.

Filters are applied if you attempt to get all dimensions from an `HFMwMbrSel` instance. Only dimensions that meet all the filters' criteria are returned. For example, if both `ValuedOnly` and `EnabledOnly` are set to `TRUE`, then only enabled dimensions containing a value are returned.

Read-write.

VisibleOnly

Gets or sets whether to filter for dimensions for which the `visible` attribute has been set. `TRUE` turns this filter on, `FALSE` turns it off.

Tip: To set a dimension's `visible` attribute, use `AttrDim`, `AttrAll`, or `HFMwMbrSelDim.Attr`.

Filters are applied if you attempt to get all dimensions from an `HFMwMbrSel` instance. Only dimensions that meet all the filters' criteria are returned. For example, if both `VisibleOnly` and `ValuedOnly` are set to `TRUE`, then only visible dimensions containing a value are returned.

Read-write.

XmlDom

Gets or sets the state of the `HFMwMbrSel` instance using the Point of View settings in a Variant that contains an XML Document Object Model (DOM) document.

Read-write.

XmlString

Gets or sets the state of the `HFMwMbrSel` instance using the Point of View settings in an XML string.

You can persist a Point of View representing the current state of the `HFMwMbrSel` instance by passing the XML string to `HFMwSession.SetPOVByTag`. You can set an `HFMwMbrSel` instance to a persisted Point of View by setting `xmlString` to the string returned by `HFMwSession.GetPOVByTag`.

Read-write.

Example

The following subroutine persists a Point of View. The subroutine takes the tag that identifies the Point of View and the labels of the dimension members, which are set with the `AttrAll` property. The Point of View is persisted with the `HFMwSession` methods `SetPOVByTag` and `ApplyUserSettings`.

```

Sub CreatePOVTag(sTag, aDimVals)
Dim cMbrSel
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object
cMbrSel.SetWebSession g_cSession
cMbrSel.AttrAll(WEBOM_MBRSEL_ID_VALUE) = aDimVals
g_cSession.SetPOVByTag sTag, FALSE, cMbrSel.XmlString
g_cSession.ApplyUserSettings TRUE
End Sub

```

HFmwMbrSelDim Component

The HFmwMbrSelDim component represents dimensions in the HFmwMbrSel component's collection of dimensions; an HFmwMbrSelDim instance is a view of a dimension in an HFmwMbrSel instance. You can use the HFmwMbrSelDim component to get and set a dimension's attributes.

The HFmwMbrSelDim component can be used with any Financial Management dimension. You specify the dimension when you obtain an HFmwMbrSelDim object reference.

Obtaining an HFmwMbrSelDim Object Reference

Create an HFmwMbrSelDim object with the HFmwMbrSel component's [Dim](#) property. The following example sets an HFmwMbrSelDim instance for the Account dimension:

```

Dim cMbrSel, cMbrSelDim
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
Set cMbrSelDim = cMbrSel.Dim(DIMENSIONACCOUNT)

```

The HFmwMbrSelDim component contains the properties described in the following topics.

Attr

Gets or sets a given attribute's value.

[Attr](#) is the HFmwMbrSelDim component's default property. For example, in VBScript `<HFmwMbrSelDim>(lidAttr)` is the same as `<HFmwMbrSelDim>.attribute(lidAttr)`.

Read-write.

Tip: You can also set dimension attributes with the HFmwMbrSel component's [AttrAll](#) and [AttrDim](#) properties.

Read-write.

Syntax

```
<HFmwMbrSelDim>.Attr(lidAttr)
```

Argument Description

<i>lidAttr</i>	The ID of the attribute. Valid values are represented by the HFMConstants type library constants listed in “Point of View Attribute Constants” on page 459 . Input argument. Long subtype.
----------------	---

Example

The following example sets the Account dimension's value to the member list [Hierarchy].

```
Dim cMbrSel, cMbrSelDim, sVal
Set cMbrSel = Server.CreateObject("Hyperion.HFMwMbrSel")
'g_cSession is an HFMwSession object reference
cMbrSel.SetWebSession g_cSession
Set cMbrSelDim = cMbrSel.dim(DIMENSIONACCOUNT)
sVal = cMbrSel.buildValue (WEBOM_MBRSEL_VALTYPE_LIST, "[Hierarchy]", "")
cMbrSelDim.Attr(WEBOM_MBRSEL_ID_VALUE) = sVal
```

Custom

Creates, updates, or returns a string containing a user-defined named directive for the HFMwMbrSelDim instance.

Read-write.

Syntax

```
<HFMwMbrSelDim>.Custom(bstrAttr)
```

Argument Description

<i>bstrAttr</i>	The name of the directive. Input argument. String subtype.
-----------------	---

Dump

Returns a text representation of the HFMwMbrSelDim instance. This can be useful when debugging.

Read-only.



HFMwDataGrid Type Library

In This Chapter

HFMwDataGrid Component	201
------------------------------	-----

The HFMwDataGrid type library provides a data grid interface that you can use to populate a user interface. The type library contains the HFMwDataGrid component.

Obtaining an HFMwDataGrid Object Reference

Create an HFMwDataGrid object with `Server.CreateObject`. After creating the object, call `SetWebSession` to associate the object with the HFMwSession object for the application to which you want to apply the HFMwDataGrid object:

```
Set cHFMDDataGrid = Server.CreateObject("Hyperion.HFMwDataGrid")
' cHFMSession is a previously set HFMwSession object
cHFMDDataGrid.SetWebSession cHFMSession
```

HFMwDataGrid Component

The HFMwDataGrid component contains methods that provide the following functionality:

- Save cells in the grid to the database.
- Define and save grids using XML strings.
- Execute data and process management tasks.
- Return data for the grid's cells.
- Collapse and expand rows and columns in the grid.
- Return column and row header information.
- Return a cell's process management information.
- Return a cell's cell text and line items.
- Return information on a cell's transactions.
- For Organization by Period applications, specify whether active entities are filtered, as well as filtering criteria.

The HFMwDataGrid component also contains properties that return or set the grid's properties. For example, these properties get or set the members of the grid's dimensions.

The methods are described in “[HFMwDataGrid Methods](#)” on page 203, and the properties in “[HFMwDataGrid Properties](#)” on page 254.

The HFMwDataGrid component breaks up grids into pages for performance purposes; for details, see “[Pages on Grids](#)” on page 202. For information on how to define grids, see “[Creating a Grid](#)” on page 202.

Pages on Grids

When you define a grid, the HFMwDataGrid object calculates the total number of rows and columns. A *page* is a subset of these cells. Pages mean that instead of simultaneously sending all the grid’s cells over a Web connection, you can send only one page at a time.

Tip: The [maxColsPerPage](#) and [maxRowsPerPage](#) properties define the maximum size of a grid’s pages.

A grid will have one or more pages; in other words, a grid can be viewed as a two-dimensional array of pages. Methods that work with pages use indexes to access the pages. For example, suppose a grid contains 10,000 rows and 100 columns, and the maximum number of rows and columns per page has been set to 100. Indexes for the rows will range from 0 to 99, and the only index for columns will be 0.

Tip: For an example of how to loop through the pages in a grid, see the example for [hPageCount](#).

Creating a Grid

There are two ways to create a grid:

- From an XML string with [DefineGridUsingXML](#).
- By setting the HFMwDataGrid properties that define a grid. You must specify the following information to define a grid:
 - The maximum columns and rows for a grid page. Set the maximum page size with [maxColsPerPage](#) and [maxRowsPerPage](#).
 - The Financial Management dimensions for the rows and columns. Set these dimensions with [columnDimensions](#) and [rowDimensions](#).
 - The dimension members for the rows and columns. A row or column dimension’s members are specified with the following properties:
 - [useList](#) specifies whether a member list is used to define the members of a row or column dimension.
 - If [useList](#) is set to TRUE, specify the member list with [memberListID](#).
 - If [useList](#) is set to FALSE, specify the dimension members with the [topMemberID](#), [selectedMemberIDs](#), and [selectedParentIDs](#) properties.

- The dimension members for the grid's page dimensions. A page dimension's members are specified with the `memberID` and `parentID` properties.

Tip: The example for `maxColsPerPage` defines a grid with HFMwDataGrid's properties.

HFMwDataGrid Methods

The HFMwDataGrid component contains the methods described in the following topics.

CollapseColumnDimension

Collapses a column dimension (for nested dimensions).

Syntax

```
<HFMwDataGrid>.CollapseColumnDimension lColumnNum, lDimIndex,  
pvarlOldColumnNumOfLastColumnAffected,  
pvarlNewColumnNumOfLastColumnAffected, pvarlTotalNumCollapsedColumns
```

Argument	Description
<i>lColumnNum</i>	The index (0-based) that identifies the column in which the collapse is to occur. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension to collapse. Input argument. Long subtype.
<i>pvarlOldColumnNumOfLastColumnAffected</i>	<i>Future use.</i> Output argument.
<i>pvarlNewColumnNumOfLastColumnAffected</i>	Returns the index that identifies the last of the new columns. Output argument.
<i>pvarlTotalNumCollapsedColumns</i>	Returns a count of the updated total of columns. Output argument.

CollapseColumnMemberInDimension

Collapses a dimension member in a column. If the member has children, the children will become hidden.

Note: Collapsing members is valid only for dimensions using the [Hierarchy] member list.

Syntax

```
<HFMwDataGrid>.CollapseColumnMemberInDimension lColNum, lDimIndex,  
pvarlDeletePos, pvarlNumToDelete, pvarlTotalNumCollapsedCols
```

Argument	Description
<i>lColNum</i>	The index (0-based) that identifies the column to be collapsed. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension to which the member belongs. Each dimension in a column is indexed. Input argument. Long subtype.
<i>pvarlDeletePos</i>	Returns the index number of the column in which the deletion occurred. Output argument.
<i>pvarlNumToDelete</i>	Returns the number of columns that were deleted. Output argument.
<i>pvarlTotalNumCollapsedCols</i>	Returns a count of the new total of columns for the grid. Output argument.

CollapseRowDimension

Collapses a row dimension (for nested dimensions).

Syntax

```
<HFMwDataGrid>.CollapseRowDimension lRowNum, lDimIndex,  
pvarlOldRowNumOfLastRowAffected, pvarlNewRowNumOfLastRowAffected,  
pvarlTotalNumCollapsedRows
```

Argument	Description
<i>lRowNum</i>	The index (0-based) that identifies the row in which the collapse is to occur. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension to collapse. Input argument. Long subtype.
<i>pvarlOldRowNumOfLastRowAffected</i>	<i>Future use.</i> Output argument.
<i>pvarlNewRowNumOfLastRowAffected</i>	Returns the index that identifies the last of the new rows. Output argument.
<i>pvarlTotalNumCollapsedRows</i>	Returns a count of the updated total of rows. Output argument.

CollapseRowMemberInDimension

Collapses a dimension member in a row. If the member has children, the children are hidden.

Note: Collapsing members is valid only for dimensions using the [Hierarchy] member list.

Syntax

```
<HFMwDataGrid>.CollapseRowMemberInDimension lRowNum, lDimIndex,  
pvarlDeletePos, pvarlNumToDelete, pvarlTotalNumCollapsedRows
```

Argument	Description
<i>lRowNum</i>	The index (0-based) that identifies the row to be collapsed. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension to which the member belongs. Each dimension in a row is indexed. Input argument. Long subtype.
<i>pvarlDeletePos</i>	Returns the index number of the row in which the deletion occurred. Output argument.
<i>pvarlNumToDelete</i>	Returns the number of rows that were deleted. Output argument.
<i>pvarlTotalNumCollapsedRows</i>	Returns a count of the new total of rows for the grid. Output argument.

DefineGridUsingXML

Defines a grid using the settings specified in an XML string.

Tip: To define a grid by passing an XML string in an ASP Request object, use [DefineGridUsingXMLFromRequest](#).

Syntax

```
<HFMwDataGrid>.DefineGridUsingXML bstrXML
```

Argument	Description
----------	-------------

<i>bstrXML</i>	An XML string that represents the grid to be defined. The XML string is described in “Data Grid Definition DTD” on page 474 .
----------------	---

Note: The <RDIMS> and <CDIMS> tags are optional; however, if one is specified then the other must also be specified. The <POV> tag is optional; if specified, it must contain at least one <POVDIM> tag. Expansion information will be reset under the following conditions: <RDIMS> and <CDIMS> are present, or a <POVDIM> tag is present for a row or column dimension.

Argument	Description
----------	-------------

	Input argument. String subtype.
--	---------------------------------

DefineGridUsingXMLFromRequest

Defines a grid using the settings specified in an XML string that is contained by an ASP Request object.

Syntax

```
<HFMwDataGrid>.DefineGridUsingXMLFromRequest varpIRequest
```

Argument	Description
----------	-------------

<i>varpIRequest</i>	An ASP Request object containing an XML string that represents the grid to be defined. The XML string is described in “Data Grid Definition DTD” on page 474 .
---------------------	--

Note: The <RDIMS> and <CDIMS> tags are optional; however, if one is specified then the other must also be specified. The <POV> tag is optional; if specified, it must contain at least one <POVDIM> tag. Expansion information will be reset under the following conditions: <RDIMS> and <CDIMS> are present, or a <POVDIM> tag is present for a row or column dimension.

Input argument.

DefinePOVUsingXML

Defines the Point of View of an HFMwDataGrid object, given an XML string.

Tip: [GetPOVAsXML](#) returns a grid’s Point of View settings as an XML string. The returned XML string has the same DTD as that passed to [DefinePOVUsingXML](#).

Syntax

```
<HFMwDataGrid>.DefinePOVUsingXML bstrXML
```

Argument	Description
----------	-------------

<i>bstrXML</i>	The XML string that defines the Point of View. The XML string is described in “Point of View DTD” on page 482 .
----------------	---

Input argument. String subtype.

EnumColumnHeaderInfo

Future use.

Tip: To return column header information, use [GetGridPageWithExpansionInfo](#).

EnumRowHeaderInfo

Future use.

Tip: To return row header information, use [GetGridPageWithExpansionInfo](#).

ExecuteAction

Executes a Data Explorer task for the specified rows and columns.

Syntax

```
<HFMwDataGrid>.ExecuteAction (lTaskID, varavarRows, varavarCols)
```

Argument	Description
----------	-------------

<i>lTaskID</i>	The ID of the task to be executed. These IDs are represented by the HFMConstants type library constants listed in “Data Explorer Task Constants” on page 446 .
----------------	--

Input argument. Long subtype.

<i>varavarRows</i>	An array of the row numbers that identify the rows for the action.
--------------------	--

Input argument.

<i>varavarCols</i>	An array of the column numbers that identify the columns for the action.
--------------------	--

Input argument.

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the HsvResourceManager object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

ExecuteDataExplorerAction

Executes a Data Explorer task for the specified range of cells.

Syntax

```
<HFMwDataGrid>.ExecuteDataExplorerAction (lTaskID, lRowStart, lColStart, lRowEnd, lColEnd)
```

Argument	Description
----------	-------------

<i>lTaskID</i>	The ID of the task to be executed. These IDs are represented by the HFMConstants type library constants listed in “Data Explorer Task Constants” on page 446 .
----------------	--

Input argument. Long subtype.

Argument	Description
----------	-------------

<i>lRowStart</i>	The index (0-based) that identifies the first row in the range of cells. Input argument. Long subtype.
------------------	---

<i>lColStart</i>	The index (0-based) that identifies the first column in the range of cells. Input argument. Long subtype.
------------------	--

<i>lRowEnd</i>	The index that identifies the last row in the range of cells. Input argument. Long subtype.
----------------	--

<i>lColEnd</i>	The index that identifies the last column in the range of cells. Input argument. Long subtype.
----------------	---

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the HsvResourceManager object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

ExecuteDataExplorerProcessManagementTask

Executes a process management task for the specified range of cells.

To attach documents while executing a process management task, use [ExecuteDataExplorerProcessManagementTask2](#). To attach documents and return an XML string that can be passed to [GenerateAlerts](#), use [ExecuteDataExplorerProcessManagementTask3](#).

Syntax

```
<HFMwDataGrid>.ExecuteDataExplorerProcessManagementTask  
(lProcessMgmtActionID, lRowStart, lColStart, lRowEnd, lColEnd,  
sPromotionLevel, bstrComments, pvarsNewState)
```

Argument	Description
----------	-------------

<i>lProcessMgmtActionID</i>	The ID of the task to be executed. These IDs are represented by the HFMConstants type library constants listed in “Data Explorer Process Management Constants” on page 447 . Input argument. Long subtype.
-----------------------------	---

<i>lRowStart</i>	The index (0-based) that identifies the first row in the range of cells. Input argument. Long subtype.
------------------	---

<i>lColStart</i>	The index (0-based) that identifies the first column in the range of cells. Input argument. Long subtype.
------------------	--

<i>lRowEnd</i>	The index that identifies the last row in the range of cells.
----------------	---

Argument	Description
	Input argument. Long subtype.
<i>IColEnd</i>	The index that identifies the last column in the range of cells. Input argument. Long subtype.
<i>sPromotionLevel</i>	Indicates the level to which to promote. This argument applies only if the method is promoting. Promotion levels are represented by the numbers in the following list: <ul style="list-style-type: none"> ● 1 = Not Started ● 2 = First Pass ● 3 = Review Level 1 ● 4 = Review Level 2 ● 5 = Review Level 3 ● 6 = Review Level 4 ● 7 = Review Level 5 ● 8 = Review Level 6 ● 9 = Review Level 7 ● 10 = Review Level 8 ● 11 = Review Level 9 ● 12 = Review Level 10 ● 13 = Submitted ● 14 = Approved ● 15 = Published <p>Note: For tasks other than promoting, you must pass an integer, but the value passed will have no effect.</p> Input argument. Integer subtype.
<i>bstrComments</i>	The comment for the task. Input argument. String subtype.
<i>pvarsNewState</i>	Returns the number that identifies the level to which the process unit has been promoted. If the promotion succeeds, this should be equal to the number passed in the <i>sPromotionLevel</i> argument. Output argument.

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the `HsvResourceManager` object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

ExecuteDataExplorerProcessManagementTask2

Executes a process management task for the specified range of cells and optionally attaches documents.

To attach documents and return an XML string that can be passed to [GenerateAlerts](#), use [ExecuteDataExplorerProcessManagementTask3](#).

Syntax

```
<HFMwDataGrid>.ExecuteDataExplorerProcessManagementTask2  
(lProcessMgmtActionID, lRowStart, lColStart, lRowEnd, lColEnd,  
sPromotionLevel, bstrComments, varavarAttPaths, varavarAttFiles,  
pvarsNewState)
```

Argument	Description
<i>lProcessMgmtActionID</i>	The ID of the task to be executed. These IDs are represented by the HFMConstants type library constants listed in “Data Explorer Process Management Constants” on page 447 . Input argument. Long subtype.
<i>lRowStart</i>	The index (0-based) that identifies the first row in the range of cells. Input argument. Long subtype.
<i>lColStart</i>	The index (0-based) that identifies the first column in the range of cells. Input argument. Long subtype.
<i>lRowEnd</i>	The index that identifies the last row in the range of cells. Input argument. Long subtype.
<i>lColEnd</i>	The index that identifies the last column in the range of cells. Input argument. Long subtype.
<i>sPromotionLevel</i>	Indicates the level to which to promote. This argument applies only if the method is promoting. Promotion levels are represented by the numbers in the following list: <ul style="list-style-type: none">● 1 = Not Started● 2 = First Pass● 3 = Review Level 1● 4 = Review Level 2● 5 = Review Level 3● 6 = Review Level 4● 7 = Review Level 5● 8 = Review Level 6● 9 = Review Level 7● 10 = Review Level 8● 11 = Review Level 9● 12 = Review Level 10● 13 = Submitted● 14 = Approved● 15 = Published

Argument	Description
	<p>Note: For tasks other than promoting, you must pass an integer, but the value passed will have no effect.</p> <p>Input argument. Integer subtype.</p>
<i>bstrComments</i>	<p>The comment for the task.</p> <p>Input argument. String subtype.</p>
<i>varavarAttPaths</i>	<p>An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\).</p> <p>This array has a one-to-one correspondence with the <i>varavarAttFiles</i> argument's array of filenames.</p> <p>Input argument.</p>
<i>varavarAttFiles</i>	<p>An array of strings containing the file names of the documents to attach.</p> <p>Note: Files must be loaded in the folder specified by the corresponding item in the <i>varavarAttPaths</i> argument's array, otherwise an error will be thrown.</p> <p>Input argument.</p>
<i>pvarsNewState</i>	<p>Returns the number that identifies the level to which the process unit has been promoted. If the promotion succeeds, this should be equal to the number passed in the <i>sPromotionLevel</i> argument.</p> <p>Input/output argument.</p>

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the *HsvResourceManager* object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

ExecuteDataExplorerProcessManagementTask3

Executes a process management task for the specified range of cells, optionally attaches documents, and, if e-mail alerting is enabled, returns a list of the process units that may require alerts.

Note: To get information on users who should receive e-mail alerts for the process units, pass the string returned by the *pvarbstrXML* argument to [GenerateAlerts](#).

Syntax

```
<HFMwDataGrid>.ExecuteDataExplorerProcessManagementTask3
(lProcessMgmtActionID, lRowStart, lColStart, lRowEnd, lColEnd,
sPromotionLevel, bstrComments, varavarAttPaths, varavarAttFiles,
pvarsNewState, pvarbstrXML)
```

Argument	Description
<i>IProcessMgmtActionID</i>	<p>The ID of the task to be executed. These IDs are represented by the HFMConstants enumeration <code>tagWEBOM_DATAGRID_PROCESSFLOWACTION_ENUM</code>, which is described in “Data Explorer Process Management Constants” on page 447.</p> <p>Input argument. Long subtype.</p>
<i>IRowStart</i>	<p>The index (0-based) that identifies the first row in the range of cells.</p> <p>Input argument. Long subtype.</p>
<i>IColStart</i>	<p>The index (0-based) that identifies the first column in the range of cells.</p> <p>Input argument. Long subtype.</p>
<i>IRowEnd</i>	<p>The index that identifies the last row in the range of cells.</p> <p>Input argument. Long subtype.</p>
<i>IColEnd</i>	<p>The index that identifies the last column in the range of cells.</p> <p>Input argument. Long subtype.</p>
<i>sPromotionLevel</i>	<p>Indicates the level to which to promote. This argument applies only if the method is promoting. Promotion levels are represented by the numbers in the following list:</p> <ul style="list-style-type: none"> ● 3 = Review Level 1 ● 4 = Review Level 2 ● 5 = Review Level 3 ● 6 = Review Level 4 ● 7 = Review Level 5 ● 8 = Review Level 6 ● 9 = Review Level 7 ● 10 = Review Level 8 ● 11 = Review Level 9 ● 12 = Review Level 10 <p>Note: For tasks other than promoting, you must pass an integer, but the value passed will have no effect.</p> <p>Input argument. Integer subtype.</p>
<i>bstrComments</i>	<p>The comment for the task.</p> <p>Input argument. String subtype.</p>
<i>varavarAttPaths</i>	<p>An array of strings containing the paths in which the documents to be attached have been loaded. Folders in paths are delimited by backslashes (\).</p> <p>This array has a one-to-one correspondence with the <i>varavarAttFiles</i> argument’s array of filenames.</p> <p>Input argument.</p>
<i>varavarAttFiles</i>	<p>An array of strings containing the file names of the documents to attach.</p> <p>Note: Files must be loaded in the folder specified by the corresponding item in the <i>varavarAttPaths</i> argument’s array, otherwise an error will be thrown.</p> <p>Input argument.</p>

Argument	Description
<i>pvarsNewState</i>	Returns the number that identifies the level to which the process unit has been promoted. If the promotion succeeds, this should be equal to the number passed in the <i>sPromotionLevel</i> argument. Input/output argument.
<i>pvarbstrXML</i>	Returns an XML string representing the process units that were successfully changed. The format of the XML string is described in Table 7 . Note: This argument returns an XML representation of process units only if e-mail alerting is enabled. Input/output argument.

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the *HsvResourceManager* object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

ExpandColumnDimension

Expands a column dimension (for nested dimensions).

Note: When dimensions are nested, only the first level dimension has valid members showing after the grid is initialized. In order to generate valid intersections, you must expand each nested dimension.

Syntax

```
<HFMwDataGrid>.ExpandColumnDimension lColumnNum, lDimIndex,  
pvarlOldColumnNumOfLastColumnAffected,  
pvarlNewColumnNumOfLastColumnAffected, pvarlTotalNumExpandedColumns
```

Argument	Description
<i>lColumnNum</i>	The index (0-based) that identifies the column in which the expansion is to occur. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension for which the expansion is to occur. Each dimension in a column is indexed. Input argument. Long subtype.
<i>pvarlOldColumnNumOfLastColumnAffected</i>	<i>Future use.</i> Output argument.
<i>pvarlNewColumnNumOfLastColumnAffected</i>	Returns the index that identifies the last of the new columns.

Argument	Description
	Output argument.
<i>pvarlTotalNumExpandedColumns</i>	Returns a count of the updated total of columns.
	Output argument.

ExpandColumnMemberInDimension

Expands a dimension member in a column. If the member has children, the children will be displayed in new columns.

Note: Expanding members is only valid for dimensions using the [Hierarchy] member list.

Syntax

```
<HFMwDataGrid>.ExpandColumnMemberInDimension lColumnNum, lDimIndex,
pvarlInsertPos, pvarlNumToInsert, pvarlTotalNumExpandedColumns
```

Argument	Description
<i>lColumnNum</i>	The index (0-based) that identifies the column in which the expansion is to occur. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension for which the expansion is to occur. Each dimension in a column is indexed. Input argument. Long subtype.
<i>pvarlInsertPos</i>	Returns a number that identifies the column where the insertion took place. Output argument.
<i>pvarlNumToInsert</i>	Returns a count of the new columns that result from the expansion. Output argument.
<i>pvarlTotalNumExpandedColumns</i>	Returns the new total of columns for the grid. Output argument.

ExpandGridUsingXML

Expands the grid by using the expansion information stored in an XML string.

Syntax

```
<HFMwDataGrid>.ExpandGridUsingXML bstrXML
```

Argument	Description
<i>bstrXML</i>	The XML string, which must contain row or column expansion information. The expansion information is represented by the <REXPS> and <CEXPS> elements described in “Data Grid Definition DTD” on page 474 .

Argument	Description
----------	-------------

	Input argument. String subtype.
--	---------------------------------

ExpandRowDimension

Expands a row dimension (for nested dimensions).

Note: When dimensions are nested, only the first level dimension has valid members showing after the grid is initialized. In order to generate valid intersections, you must expand each nested dimension.

Syntax

```
<HFMwDataGrid>.ExpandRowDimension lRowNum, lDimIndex,  
pvarlOldRowNumOfLastRowAffected, pvarlNewRowNumOfLastRowAffected,  
pvarlTotalNumExpandedRows
```

Argument	Description
<i>lRowNum</i>	The index (0-based) that identifies the row in which the expansion is to occur. Input argument. Long subtype.
<i>lDimIndex</i>	The index (0-based) that identifies the dimension for which the expansion is to occur. Each dimension in a row is indexed. Input argument. Long subtype.
<i>pvarlOldRowNumOfLastRowAffected</i>	Future use. Output argument.
<i>pvarlNewRowNumOfLastRowAffected</i>	Returns a number that identifies the column where the insertion took place. Output argument.
<i>pvarlTotalNumExpandedRows</i>	Returns the new total of rows for the grid. Output argument.

ExpandRowMemberInDimension

Expands a dimension member in a row. If the member has children, the children will be displayed in new rows.

Note: Expanding members is valid only for dimensions using the [Hierarchy] member list.

Syntax

```
<HFMwDataGrid>.ExpandRowMemberInDimension lRowNum, lDimIndex,  
pvarlInsertPos, pvarlNumToInsert, pvarlTotalNumExpandedRows
```

Argument	Description
<i>IRowNum</i>	The index (0-based) that identifies the row in which the expansion is to occur. Input argument. Long subtype.
<i>IDimIndex</i>	The index (0-based) that identifies the dimension for which the expansion is to occur. Each dimension in a row is indexed. Input argument. Long subtype.
<i>pvarIInsertPos</i>	Returns the number of the row in which the expansion took place. Output argument.
<i>pvarINumToInsert</i>	Returns a count of the new rows that result from the expansion. Output argument.
<i>pvarITotalNumExpandedRows</i>	Returns a count of the new total of rows for the grid. Output argument.

GenerateAlerts

Returns an XML string listing the users who should receive e-mail alerts for the specified process units. The string includes the users' security identifiers and usernames.

Syntax

```
<HFMwDataGrid>.GenerateAlerts (varpIProcessUnits)
```

Argument	Description
<i>varpIProcessUnits</i>	An XML string that specifies the process units. Pass the XML string described in Table 7 . This string is returned by methods such as SubmitProcessStateChangeFromXML and ExecuteDataExplorerProcessManagementTask3 . Input argument.

Return Value

Returns an XML string representing the users. The following table describes the XML string.

Table 3 GenerateAlerts Return Value XML String

Element	Description
events	Root element. <events> contains the following attributes: <ul style="list-style-type: none"> xmlns = The namespace, which will be a URL similar to the following example: http://hyperion.com/HFM/4.1/Alerts/1.0 type = The string "ProcMgmt". display = The string "Process Control". <events> also contains an <event> tag.

Element	Description
event	<p>Represents the process management actions and the users to be notified. <event> contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>sid</code> = The user's security identifier. ● <code>user</code> = The user's username. ● <code>time</code> = The date and time of the process management action. ● <code>action</code> = A string describing the new review level. ● <code>newState</code> = The ID of the new review level for the action's first process unit. Review level IDs are represented by the <code>HFMConstants</code> enumeration <code>CEnumProcessFlowStates</code>, which is described in "Process Management Review Level Constants" on page 434. ● <code>actionId</code> = The ID of the process management action. Valid values are represented by the <code>HFMConstants</code> enumeration <code>CEnumProcessFlowActions</code>, which is described in "Process Management Action Constants" on page 433. ● <code>display</code> = The string used in the subject of the e-mail sent to the user who changed the process unit. ● <code>scenarioLabel</code> = The label of the Scenario dimension member for the action's first process unit. ● <code>yearLabel</code> = The label of the Year dimension member for the action's first process unit. ● <code>periodLabel</code> = The label of the Period dimension member for the action's first process unit. ● <code>entityLabel</code> = The label of the Entity dimension member for the action's first process unit. <p><event> also contains <comment> and <target> tags, as well as one <item> tag per process unit.</p> <p>Note: The user who performed the action is represented by the <target> tag that is a child of <event>, and the users to be notified are represented by the <target> tags that are children to the <item> tags.</p>
comment	Contains the comment for the process management action.
target	<p>Represents the user who performed the process management action. and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>type</code> = <i>Future use</i>. ● <code>sid</code> = The user's security identifier. ● <code>user</code> = The user's username. ● <code>skip</code> = <i>Future use</i>.
item	<p>Represents a process unit for which a notification is being generated, as well as the users to be e-mailed. <item> contains the following attributes:</p> <p>Represents the process unit and contains the following attributes.</p> <ul style="list-style-type: none"> ● <code>scenarioLabel</code> = The label of the process unit's Scenario dimension member. ● <code>yearLabel</code> = The label of the process unit's Year dimension member. ● <code>periodLabel</code> = The label of the process unit's Period dimension member. ● <code>entityLabel</code> = The label of the process unit's Entity dimension member. ● <code>entityDesc</code> = The description of the process unit's Entity dimension member. ● <code>action</code> = A period-delimited string describing the process unit. Following is the string's format: <div style="text-align: center;"><i>Scenario.Year.Period.Entity.OrigLevel</i></div> <p>The string's first four items list the labels of the Scenario, Year, Period, and Entity dimensions. The last item contains the label of the process unit's previous review level.</p> ● <code>origState</code> = The ID of the process unit's previous review level.

Element	Description
	<ul style="list-style-type: none"> ● <code>newState</code> = The ID of the process unit's new review level. ● <code>display</code> = The string used in the subject of the e-mail sent to the user who changed the process unit. <p><code><item></code> also contains one <code><target></code> tag for each user to be notified.</p>
target	<p>Represents a user to be notified, and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>sid</code> = The user's security identifier. ● <code>user</code> = The user's username.

GetCalcStatusSummaryAsXML

Returns an XML string summarizing the calculation statuses of the grid's entities that intersect the grid's scenario, year, and periods. For each period, the XML string consists of one element per calculation status; each element contains the number of entities that have that calculation status.

The counts for each status are based upon the grid's top Entity dimension member and the Entity dimension's member list. If no member list is specified for the grid, then the [Hierarchy] member list is used.

If Organization by Period is enabled for the grid, only information for active entities is returned.

Tip: To specify a grid's Organization by Period setting, use [SetOrgByPeriodFilteringInfo](#).

Syntax

```
<HFMwDataGrid>.GetCalcStatusSummaryAsXML pvarbstrXML
```

Argument Description

pvarbstrXML Returns an XML string that summarizes the entities' calculation statuses. The XML string is described in [Table 4](#).

Output argument.

The following table describes the XML string returned by `GetCalcStatusSummaryAsXML`.

Table 4 GetCalcStatusSummaryAsXML XML String

Element	Description
CALCSUMMARY	Root element.
PERIOD	Specifies the period for which status information is returned; the period label is included in the <code>name</code> attribute. Each <code><PERIOD></code> tag contains the calculation tags described in the following rows.
OK	The number of entities that have the OK status.

Element	Description
OKSC	The number of entities that have the OKSC status.
NODATA	The number of entities that have the NODATA status.
CH	The number of entities that have the CH status.
CN	The number of entities that have the CN status.
TR	The number of entities that have the TR status.
OKND	The number of entities that have the OKND status.
CHND	The number of entities that have the CHND status.
CNND	The number of entities that have the CNND status.
TRND	The number of entities that have the TRND status.
LOCKED	The number of entities that have the LOCKED status.

GetCellAdjustments

Returns an array that describes the specified cell's adjustments.

Tip: To return a cell's adjustments as an XML string, use [GetCellAdjustmentsAsXML](#).

Syntax

```
<HFMDDataGrid>.GetCellAdjustments (lRow, lColumn)
```

Argument Description

<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lColumn</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

An array of strings describing the cell adjustments. The array contains one item per adjustment. Each item includes the adjustment's amount and journal label.

Example

The following example prints the adjustments for the specified cell to the browser.

```
Dim sXML
'CHFMDDataGrid is a previously set HFMDDataGrid object
sXML = CHFMDDataGrid.GetCellAdjustments (0, 0)
```

```

For i = 0 to uBound(sXML)
    Response.Write "<p>" & sXML(i) & "</p>"
Next

```

GetCellAttachmentsAsXML

Returns an XML string containing the names and paths of a cell's document attachments.

Syntax

```
<HFMwDataGrid>.GetCellAttachmentsAsXML (lRow, lCol)
```

Argument Description

<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

Returns an XML string that contains the attachments' names and paths. The XML string is described in [“Cell Attachments DTD” on page 487](#).

GetCellInfo

Returns various types of information for the specified cell, such as the cell's dimension members and calculation status.

Tip: To return cell information as an XML string, use [GetCellInfoAsXML](#).

Syntax

```

<HFMwDataGrid>.GetCellInfo lRow, lColumn, pvarbstrProcessUnit,
pvarbstrPOVDetail, pvarbstrView, pvarbstrCalcStatus, pvarbstrProcessLevel,
pvarbstrCellStatus, pvarbstrCellSecurityAccess, pvarbstrDisplayedData,
pvarbstrFullResolutionData, pvarbstrStoredData

```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lColumn</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.
<i>pvarbstrProcessUnit</i>	Returns the labels of the dimension members that the cell's process unit consists of. The labels are delimited by commas. Tip: Process units are described in “About Process Units” on page 24 .

Argument	Description
	Output argument.
<i>pvarbstrPOVDetail</i>	Returns the labels of the cell's Account, Intercompany Partner, and Custom 1-4 dimension members. The labels are delimited by commas. Output argument.
<i>pvarbstrView</i>	Returns the label of the cell's View dimension member. Output argument.
<i>pvarbstrCalcStatus</i>	Returns a string description of the cell's calculation status. Output argument.
<i>pvarbstrProcessLevel</i>	Returns a string description of the cell's process management review level. Output argument.
<i>pvarbstrCellStatus</i>	Returns a string description of the cell's status. Output argument.
<i>pvarbstrCellSecurityAccess</i>	Returns a string description of the cell's security access rights. Output argument.
<i>pvarbstrDisplayedData</i>	Returns the displayed data. Note: The data returned is rounded to the number of decimal places specified for the cell's account, and includes the user's thousands separator preference. Output argument.
<i>pvarbstrFullResolutionData</i>	Returns the full resolution data. Note: The data returned includes the user's thousands separator preference. Output argument.
<i>pvarbstrStoredData</i>	Returns the stored data. Output argument.

Example

The following function takes an HFMwDataGrid object and the row and column indexes that identify a cell in the object and returns the string that describes the cell's calculation status.

```
Function getCellCalcStatus(cDataGrid, lRow, lCol)
Dim sProcessUnit, sPOVDetail, sView, sCalcStatus, sProcessLevel
Dim sCellStatus, sSecurityAccess, sDisplayedData
Dim sFullResolutionData, sStoredData
cDataGrid.GetCellInfo lRow, lCol, sProcessUnit, sPOVDetail, _
sView, sCalcStatus, sProcessLevel, sCellStatus, _
sSecurityAccess, sDisplayedData, sFullResolutionData, _
sStoredData
getCellCalcStatus = sCalcStatus
End Function
```

GetCellInfoAsXML

Returns an XML string containing various types of information for the specified cell, such as the cell's dimension members and calculation status.

Tip: To return cell information in a non-XML format, use [GetCellInfo](#).

Syntax

```
<HFMwDataGrid>.GetCellInfoAsXML (lRow, lColumn)
```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lColumn</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

Returns an XML string of the cell information. The XML string is described in “[Cell Information DTD](#)” on page 486.

GetCellAdjustmentsAsXML

Returns an XML string that describes the specified cell's adjustments.

Tip: To return a cell's adjustments as an array, use [GetCellAdjustments](#).

Syntax

```
<HFMwDataGrid>.GetCellAdjustmentsAsXML (lRow, lColumn)
```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lColumn</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

An XML string that describes the cell's adjustments. The XML string is described in “[Cell Adjustment DTD](#)” on page 486.

GetCellLineItemDetailsAsXML

Returns an XML representation of the specified cell's line item details.

Syntax

```
<HFMwDataGrid>.GetCellLineItemDetailsAsXML (lRow, lCol)
```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

Returns an XML string of the cell's line item details. The XML string is described in [“Line Item Detail DTD” on page 481](#).

GetCellText

Returns the cell text description for the specified cell.

Tip: To return a cell text description as an XML string, use [GetCellTextAsXML](#).

Syntax

```
<HFMwDataGrid>.GetCellText (lRow, lCol)
```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

The cell text for the cell.

GetCellTextAsXML

Returns an XML representation of the cell text description for the specified cell.

Tip: To return a cell text description as a string, use [GetCellText](#).

Syntax

```
<HFMwDataGrid>.GetCellTextAsXML (lRow, lCol)
```

Argument	Description
----------	-------------

<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

Returns an XML string of the cell text for the cell. The XML string is described in [“Cell Text DTD” on page 481](#).

GetConsolidationProgress

Future use.

Syntax

```
<HFMwDataGrid>.GetConsolidationProgress pvarlCurrentTaskID, pvarlProgress,  
pvarlStatus, pvardLastUpdateTime, pvarbstrDescription
```

Argument	Description
<i>pvarlCurrentTaskID</i>	Input/output argument.
<i>pvarlProgress</i>	Input/output argument.
<i>pvarlStatus</i>	Input/output argument.
<i>pvardLastUpdateTime</i>	Input/output argument.
<i>pvarbstrDescription</i>	Input/output argument.

GetDestinationTransactions

Returns information on destination transactions for a cell in a statutory application. The following information is returned:

- Header information, such as the dimension members that define the cell, the current username, the cell's data, and so on. This information is returned in two arrays that have a one-to-one correspondence; one array contains labels for the header information, the other contains the values that correspond to the labels.
- Detailed information on the transactions, such as the dimension members and data for the transactions. This information is contained in several arrays; the arrays contain one item per transaction and have a one-to-one correspondence.
- The sums of the cell's source and destination transaction amounts.

Tip: To return destination transaction information as an XML string, use [GetTransactionInfoAsXML](#).

Syntax

```
<HFMwDataGrid>.GetDestinationTransactions lRow, lColumn,  
pvaravarbstrHeaderLabels, pvaravarbstrHeaderValues,  
pvaravarbstrCurrentEntity, pvaravarbstrSourceScenario,  
pvaravarbstrSourceYear, pvaravarbstrSourcePeriod,  
pvaravarbstrSourceParent, pvaravarbstrSourceEntity,  
pvaravarbstrSourceValue, pvaravarbstrSourceAccount, pvaravarbstrSourceICP,  
pvaravarbstrSourceView, pvaravarbstrSourceCustom1,  
pvaravarbstrSourceCustom2, pvaravarbstrSourceCustom3,  
pvaravarbstrSourceCustom4, pvaravarbstrDestData, pvaravarbstrSourceData,  
pvaravarbstrFactor, pvaravarbstrNature, pvarbstrTotalDestData,  
pvarbstrTotalSourceData
```

Argument	Description
<i>lRow</i>	The index of the row that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>lColumn</i>	The index of the column that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>pvaravarbstrHeaderLabels</i>	Returns an array of labels for the header information. Output argument.
<i>pvaravarbstrHeaderValues</i>	Returns an array of values for the header information. Output argument.
<i>pvaravarbstrCurrentEntity</i>	Returns an array of labels for the transactions' current Entity dimension members. Output argument.
<i>pvaravarbstrSourceScenario</i>	Returns an array of labels for the transactions' source Scenario dimension members. Output argument.
<i>pvaravarbstrSourceYear</i>	Returns an array of labels for the transactions' source Year dimension members. Output argument.
<i>pvaravarbstrSourcePeriod</i>	Returns an array of labels for the transactions' source Period dimension members. Output argument.
<i>pvaravarbstrSourceParent</i>	Returns an array of labels for the transactions' source parent Entity dimension members. Output argument.
<i>pvaravarbstrSourceEntity</i>	Returns an array of labels for the transactions' source child Entity dimension members. Output argument.
<i>pvaravarbstrSourceValue</i>	Returns an array of labels for the transactions' source Value dimension members. Output argument.

Argument	Description
<i>pvaravabstrSourceAccount</i>	Returns an array of labels for the transactions' source Account dimension members. Output argument.
<i>pvaravabstrSourceICP</i>	Returns an array of labels for the transactions' source Intercompany Partner dimension members. Output argument.
<i>pvaravabstrSourceView</i>	Returns an array of labels for the transactions' source View dimension members. Output argument.
<i>pvaravabstrSourceCustom1</i>	Returns an array of labels for the transactions' source Custom 1 dimension members. Output argument.
<i>pvaravabstrSourceCustom2</i>	Returns an array of labels for the transactions' source Custom 2 dimension members. Output argument.
<i>pvaravabstrSourceCustom3</i>	Returns an array of labels for the transactions' source Custom 3 dimension members. Output argument.
<i>pvaravabstrSourceCustom4</i>	Returns an array of labels for the transactions' source Custom 4 dimension members. Output argument.
<i>pvaravabstrDestData</i>	Returns an array of the transactions' destination data amounts. Output argument.
<i>pvaravabstrSourceData</i>	Returns an array of the transactions' source data amounts. Output argument.
<i>pvaravabstrFactor</i>	Returns an array of the factors used in the transactions. Output argument.
<i>pvaravabstrNature</i>	Returns an array of the audit strings specified for the transactions. Output argument.
<i>pvarbstrTotalDestData</i>	Returns the sum of the destination transaction amounts. Output argument.
<i>pvarbstrTotalSourceData</i>	Returns the sum of the source transaction amounts. Output argument.

GetGridDefinitionAsXML

Returns an XML representation of the grid.

Tip: To return only specified elements of the grid instead of the entire grid, use [GetGridDefinitionAsXMLEx](#).

Syntax

```
<HFMwDataGrid>.GetGridDefinitionAsXML()
```

Return Value

A string containing the XML representation of the grid. The XML string is described in [“Data Grid Definition DTD” on page 474](#).

GetGridDefinitionAsXMLeX

Returns an XML representation of the specified grid elements.

Syntax

```
<HFMwDataGrid>.GetGridDefinitionAsXMLeX (lFlagsRequestedInfo)
```

Argument	Description
----------	-------------

<i>lFlagsRequestedInfo</i>	Specifies the grid information to be returned. Pass one or more of the HFMConstants type library constants listed in “Data Grid Definition Constants” on page 448 .
----------------------------	---

You can use the addition operator (+) with these constants to return multiple types of information.

Note: To return information on the grid's expanded dimensions, you must have expanded rows or columns with a method such as [ExpandColumnDimension](#).

Note: To return information on the grid's user interface state, you must have previously called [SaveUIStateUsingXML](#).

Input argument. Long subtype.

Return Value

A string containing the XML representation of the specified grid elements. The XML string is described in [“Data Grid Definition DTD” on page 474](#).

Example

The following example prints information on the grid's row and column dimensions to a file. Note how the call to `GetGridDefinitionAsXMLeX` uses the addition operator to return information on both the row and column dimensions.

```
Dim sXml, oFs, oTs
' cHFMDDataGrid is a previously set HFMwDataGrid object
sXml = cHFMDDataGrid.GetGridDefinitionAsXMLeX( _
WEBOM_DATAGRID_DEFINITION_INFO_COLDIMS + _
WEBOM_DATAGRID_DEFINITION_INFO_ROWDIRMS)
' Write the string to disk as a text file
set oFs = CreateObject("Scripting.FileSystemObject")
' sFile was previously set to a file name and path
set oTs = oFs.CreateTextFile(sFile)
oTs.Write(sXml)
oTs.Close
```

GetGridPage

Returns data for the specified page of the grid, as well as the corresponding row and column headers.

Tip: For information on pages, see [“Pages on Grids” on page 202](#).

Syntax

```
<HFMwDataGrid>.GetGridPage lhPage, lvPage, sDisplayData,  
pvarabstrRowHeaders, pvarabstrColHeaders, pvarabstrValues, pvaralStatus,  
pvaralColors
```

Argument	Description
<i>lhPage</i>	The horizontal index (0-based) of the desired page. Input argument. Long subtype.
<i>lvPage</i>	The vertical index (0-based) of the desired page. Input argument. Long subtype.
<i>sDisplayData</i>	The data display mode. Pass one of the HFMConstants type library constants listed in “Data Information Display Constants” on page 449 . Input argument. Integer subtype.
<i>pvarabstrRowHeaders</i>	Returns a nested array of the row headers. The argument returns an array consisting of one element for each row. Each element consists of an array that contains one string element for each nested dimension label. Output argument.
<i>pvarabstrColHeaders</i>	Returns a nested array of the column headers. The argument returns an array consisting of one element for each column. Each element consists of an array that contains one string element for each nested dimension label. Output argument.
<i>pvarabstrValues</i>	Returns an array of strings that contain data for the grid page’s cells. The data is formatted according to the user’s scaling and decimal separator preferences, and is organized in row major order. Output argument.
<i>pvaralStatus</i>	Returns an array of numbers representing the cells’ statuses. The data is organized in row major order. For details on cell statuses, see “About Cell Statuses” on page 25 . Output argument.
<i>pvaralColors</i>	Returns an array of numbers representing the colors that indicate the cell statuses. The data is organized in row major order. The numbers correspond to colors used in the Data Explorer. These numbers are mapped to styles in the <code>DataCellStyles.css</code> style sheet, which is part of the Financial Management installation for the Web server. This mapping works as follows:

Argument	Description
	<ul style="list-style-type: none"> The styles are named according to the following convention, with <i>n</i> corresponding to the numbers returned in this array: <pre>.datagrid TD INPUT.COLOR<i>n</i></pre> The colors of the cell and font are defined by the applicable style's <code>BACKGROUND-COLOR</code> and <code>COLOR</code> attributes, respectively. For example, if 2 is returned as an array item, the color would be the <code>BACKGROUND-COLOR</code> attribute of the <code>.datagrid TD INPUT.COLOR2</code> style. <p>Note: If the style does not contain a <code>BACKGROUND-COLOR</code> attribute, the cell color defaults to white. If the style does not contain a <code>COLOR</code> attribute, the font color defaults to black.</p>
	Output argument.

GetGridPageWithExpansionInfo

Returns data from the specified page of the grid, as well as the corresponding row headers, column headers, and member expansion information. (For information on pages, see [“Pages on Grids” on page 202.](#))

Tip: To return grid page information in an XML or JavaScript format, use [GetGridPageWithExpansionInfoAsXML](#) or [GetGridPageWithExpansionInfoAsJavaScript](#).

Syntax

```
<HFMwDataGrid>.GetGridPageWithExpansionInfo lhPage, lvPage,
sDisplayDataMode, pvaravarabstrRowHeaders, pvaravarabstrColHeaders,
pvaravarabyteRowMemberExpansionModes, pvaravarasRowMemberGenerationLevels,
pvaravarabyteColumnMemberExpansionModes,
pvaravarasColumnMemberGenerationLevels, pvarabstrData, pvaralStatus,
pvaralColors
```

Argument	Description
<i>lhPage</i>	<p>The horizontal index (0-based) of the desired page.</p> <p>Input argument. Long subtype.</p>
<i>lvPage</i>	<p>The vertical index (0-based) of the desired page.</p> <p>Input argument. Long subtype.</p>
<i>sDisplayDataMode</i>	<p>Specifies the data display mode. Pass one of the HFMConstants type library constants listed in “Data Information Display Constants” on page 449.</p> <p>Input argument. Integer subtype.</p>
<i>pvaravarabstrRow Headers</i>	<p>Returns a nested array of the row headers. The argument returns an array consisting of one element for each row. Each element consists of an array that contains one string element for each nested dimension label.</p>

Argument	Description
	Output argument.
<i>pvaravarabstrColHeaders</i>	<p>Returns a nested array of the column headers. The argument returns an array consisting of one element for each column. Each element consists of an array that contains one string element for each nested dimension label.</p> <p>Output argument.</p>
<i>pvaravarabyteRowMemberExpansionModes</i>	<p>Returns an array with one element for each row. Each element contains an array that consists of a flag for each nested dimension member. The flag indicates whether the member is expanded in the hierarchy.</p> <p>The HFMConstants type library constants that represent the valid return values are listed in “Data Grid Member Expansion Mode Constants” on page 448.</p> <p>Output argument.</p>
<i>pvaravarasRowMemberGenerationLevels</i>	<p>Returns an array of arrays that indicate the position of the dimension members in the hierarchy. 0 indicates the top level of the hierarchy, 1 the next level, and so on.</p> <p>Note: This argument applies only in a hierarchical view of a dimension</p> <p>Output argument.</p>
<i>pvaravarabyteColumnMemberExpansionModes</i>	<p>Returns an array with one element for each row. Each element contains an array that consists of a flag for each nested dimension member. The flag indicates whether the member is expanded in the hierarchy.</p> <p>The HFMConstants type library constants that represent the valid return values are listed in “Data Grid Member Expansion Mode Constants” on page 448.</p> <p>Output argument.</p>
<i>pvaravarasColumnMemberGenerationLevels</i>	<p>Returns an array of arrays that indicate the position of the dimension members in the hierarchy. 0 indicates the top level of the hierarchy, 1 the next level, and so on.</p> <p>Note: This argument applies only in a hierarchical view of a dimension</p> <p>Output argument.</p>
<i>pvarabstrData</i>	<p>Returns an array of strings that represent the page’s data, organized in row major order. Data is formatted according to the user’s scaling and decimal separator preferences.</p> <p>Output argument.</p>
<i>pvaralStatus</i>	<p>Returns an array of numbers representing the cells’ statuses. The data is organized in row major order.</p> <p>For details on cell statuses, see “About Cell Statuses” on page 25.</p> <p>Output argument.</p>
<i>pvaralColors</i>	<p>Returns an array of numbers representing the colors that indicate the cell statuses. The data is organized in row major order.</p> <p>The numbers correspond to colors used in the Data Explorer. These numbers are mapped to styles in the <code>DataCellStyle.css</code> style</p>

Argument

Description

sheet, which is part of the Financial Management installation for the Web server. This mapping works as follows:

- The styles are named according to the following convention, with *n* corresponding to the numbers returned in this array:

`.datagrid TD INPUT.COLORn`

- The colors of the cell and font are defined by the applicable style's BACKGROUND-COLOR and COLOR attributes, respectively. For example, if 2 is returned as an array item, the color would be the BACKGROUND-COLOR attribute of the `.datagrid TD INPUT.COLOR2` style.

Note: If the style does not contain a BACKGROUND-COLOR attribute, the cell color defaults to white. If the style does not contain a COLOR attribute, the font color defaults to black.

Output argument.

GetGridPageWithExpansionInfoAsJavaScript

Returns a JavaScript representation of the specified page. (For information on pages, see [“Pages on Grids” on page 202.](#))

Tip: To return grid page information in other formats, use

[GetGridPageWithExpansionInfo](#) or [GetGridPageWithExpansionInfoAsXML](#).

Syntax

```
<HFMwDataGrid>.GetGridPageWithExpansionInfoAsJavaScript (lhPage, lvPage,  
sDisplayDataMode, varbForceGetPage)
```

Argument

Description

lhPage

The horizontal index of the desired page.
Input argument. Long subtype.

lvPage

The vertical index of the desired page.
Input argument. Long subtype.

sDisplayDataMode

Specifies the data display mode. Pass one of the HFMConstants type library constants listed in [“Data Information Display Constants” on page 449.](#)
Input argument. Integer subtype.

varbForceGetPage

Specifies whether to force a reload of the page. By default, if the horizontal and vertical page numbers are the same as the last request, the grid page is retrieved from a cache instead of the database. Passing TRUE will override this default by forcing a load from the database, passing FALSE will enact the default behavior.
Input argument. Boolean subtype.

Return Value

Returns a string containing a JavaScript representation of the grid.

GetGridPageWithExpansionInfoAsXML

Returns an XML representation of the specified page. The XML string includes the page's data, row and column headers, and member expansion information. (For information on pages, see [“Pages on Grids” on page 202.](#))

Tip: To return grid page information in other formats, use [GetGridPageWithExpansionInfo](#) or [GetGridPageWithExpansionInfoAsJavaScript](#).

Syntax

```
<HFMwDataGrid>.GetGridPageWithExpansionInfoAsXML (lhPage, lvPage, sDisplayDataMode, varbForceGetPage)
```

Argument	Description
<i>lhPage</i>	The horizontal index of the desired page. Input argument. Long subtype.
<i>lvPage</i>	The vertical index of the desired page. Input argument. Long subtype.
<i>sDisplayDataMode</i>	Specifies the data display mode. Pass one of the HFMConstants type library constants listed in “Data Information Display Constants” on page 449. Input argument. Integer subtype.
<i>varbForceGetPage</i>	Specifies whether to force a reload of the page. By default, if the horizontal and vertical page numbers are the same as the last request, the grid page is retrieved from a cache instead of the database. Passing TRUE will override this default by forcing a load from the database, passing FALSE will enact the default behavior. Input argument.

Return Value

Returns a string containing an XML representation of the grid. The XML string is described in [“Data Grid Data and Headers DTD” on page 477.](#)

GetOverlappingConsolidationInfoAsXML

Indicates whether any consolidations are currently running or queued for the specified cells.

Tip: You can avoid executing redundant consolidations by calling [GetOverlappingConsolidationInfoAsXML](#) before launching a consolidation.

Syntax

```
<HFMwDataGrid>.GetOverlappingConsolidationInfoAsXML (lRowStart, lRowEnd,  
lColStart, lColEnd, varalRows, varalCols, sConsolidationType, pvarbstrXML)
```

Argument	Description
<i>lRowStart</i>	The index (0-based) that identifies the first row of cells to check, or -1 to pass an array of rows to the <i>varalRows</i> argument. Input argument. Long subtype.
<i>lRowEnd</i>	The index (0-based) that identifies the last row of cells to check. Input argument. Long subtype.
<i>lColStart</i>	The index (0-based) that identifies the first column of cells to check, or -1 to pass an array of columns to the <i>varalCols</i> argument. Input argument. Long subtype.
<i>lColEnd</i>	The index (0-based) that identifies the last column of cells to check. Input argument. Long subtype.
<i>varalRows</i>	If -1 is passed to the <i>lRowStart</i> argument, this argument takes an array of indexes that identify the grid rows to check. Input argument.
<i>varalCols</i>	If -1 is passed to the <i>lColStart</i> argument, this argument takes an array of indexes that identify the grid columns to check. Input argument.
<i>sConsolidationType</i>	Identifies the type of consolidation to be run. Pass one of the HFMConstants type library constants listed in “Consolidation Type Constants” on page 437 . Input argument. Integer subtype.
<i>pvarbstrXML</i>	Returns an XML string that represents the dimension members and consolidation types of any consolidations that are found. Output argument.

The following table describes the XML string returned by `GetOverlappingConsolidationInfoAsXML`.

Table 5 GetOverlappingConsolidationInfoAsXML XML String

Element	Description
OVERLAPPEDCONSOLS	Root element; contains one <OVERLAPPEDCONSOL> tag for each consolidation that is found.
OVERLAPPEDCONSOL	Describes the consolidation’s type and dimension members in the tags described in the following rows.
YEAR	The label of the consolidation’s Year dimension member.
SCENARIO	The label of the consolidation’s Scenario dimension member.

Element	Description
ENTITY	The label of the consolidation's Entity dimension member.
PARENT	The label of the consolidation's parent entity.
STARTPERIOD	The label of the first period in the consolidation's range of periods.
ENDPERIOD	The label of the last period in the consolidation's range of periods.
TYPE	The number that represents the consolidation type. Valid values are represented by the HFMConstants type library enumeration tagCONSOLIDATIONTYPE, which is described in "Consolidation Type Constants" on page 437 .

Return Value

Returns an HRESULT that indicates the call's success or failure.

GetPhaseGroupInfoAsJavaScript

Returns a JavaScript representation of the grid that corresponds to the specified submission phases for a given Scenario dimension member.

Syntax

```
<HFMwDataGrid>.GetPhaseGroupInfoAsJavaScript( lScenario, varabstrPhaseIDs )
```

Argument	Description
----------	-------------

<i>lScenario</i>	The member ID of the Scenario dimension member. Input argument. Long subtype.
------------------	--

<i>varabstrPhaseIDs</i>	An array of the phase IDs for the submission phases. You must cast the phase IDs to Longs, as in the following example: <pre>vaPhases(0) = cLng(1)</pre> Input argument.
-------------------------	---

Return Value

Returns the JavaScript representation of the grid.

GetPOVAsXML

Returns an XML string that represents the current Point of View settings.

Syntax

```
<HFMwDataGrid>.GetPOVAsXML( )
```

Return Value

Returns the XML string that represents the Point of View settings. The XML string is described in [“Point of View DTD” on page 482](#).

GetProcessGridPageMultiWithExpansionInfoAsJavaScript

Returns a JavaScript representation of the specified grid page; the return value includes information regarding the grid's current submission phase. (For information on pages, see [“Pages on Grids” on page 202](#).)

Syntax

```
<HFMwDataGrid>.GetProcessGridPageMultiWithExpansionInfoAsJavaScript (lhPage,  
lvPage, sDisplayDataMode, varbForceGetPage)
```

Argument	Description
<i>lhPage</i>	The horizontal index of the desired page. Input argument. Long subtype.
<i>lvPage</i>	The vertical index of the desired page. Input argument. Long subtype.
<i>sDisplayDataMode</i>	Specifies the data display mode. Valid values are represented by the HFMConstants enumeration tagWEBOM_DATAGRID_DATA_DISPLAY_FLAGS, which is described in “Data Information Display Constants” on page 449 . Input argument. Integer subtype.
<i>varbForceGetPage</i>	Specifies whether to force a reload of the page. By default, if the horizontal and vertical page numbers are the same as the last request, the grid page is retrieved from a cache instead of the database. Passing TRUE will override this default by forcing a load from the database, passing FALSE will enact the default behavior. Input argument. Boolean subtype.

Return Value

Returns the JavaScript representation of the page.

GetProcessGridPageWithExpansionInfoAsJavaScript

Returns a JavaScript representation of the specified grid page. (For information on pages, see [“Pages on Grids” on page 202](#).)

Syntax

```
<HFMwDataGrid>.GetProcessGridPageWithExpansionInfoAsJavaScript (lhPage,  
lvPage, sDisplayDataMode, varbForceGetPage)
```

Argument	Description
<i>lhPage</i>	The horizontal index of the desired page. Input argument. Long subtype.
<i>lvPage</i>	The vertical index of the desired page. Input argument. Long subtype.
<i>sDisplayDataMode</i>	Specifies the data display mode. Valid values are represented by the HFMConstants enumeration <code>tagWEBOM_DATAGRID_DATA_DISPLAY_FLAGS</code> , which is described in “Data Information Display Constants” on page 449 . Input argument. Integer subtype.
<i>varbForceGetPage</i>	Specifies whether to force a reload of the page. By default, if the horizontal and vertical page numbers are the same as the last request, the grid page is retrieved from a cache instead of the database. Passing TRUE will override this default by forcing a load from the database, passing FALSE will enact the default behavior. Input argument. Boolean subtype.

Return Value

Returns the JavaScript representation of the page.

GetProcessManagementGridPage

Returns process management-related information for the specified page of the HFMwDataGrid instance.

Syntax

```
<HFMwDataGrid>.GetProcessManagementGridPage lhPage, pvarabstrEntities,
pvarabstrData, pvarabstrCalcStatus, pvarabstrReviewLevels, pvaralPassFail,
pvaralStatus, pvaralColors, pvaralUnfilteredRowNumbers
```

Argument	Description
<i>lhPage</i>	The horizontal index (0-based) of the desired page. Input argument. Long subtype.
<i>pvarabstrEntities</i>	Returns an array containing the labels of the entities for the process units on the page. Input/output argument.
<i>pvarabstrData</i>	Returns an array containing the data in the process units. Input/output argument.
<i>pvarabstrCalcStatus</i>	Returns an array containing the calculation statuses of the process units. Input/output argument.
<i>pvarabstrReviewLevels</i>	Returns an array containing the review levels of the process units. Input/output argument.

Argument	Description
<i>pvaralPassFail</i>	<p>Returns an array indicating whether the process units passed or failed validation for promotion. Valid values are represented by the HFMConstants type library constants listed in “Process Management Validation Constants” on page 435.</p> <p>Input/output argument.</p>
<i>pvaralStatus</i>	<p>Returns an array of numbers representing the cells' statuses.</p> <p>For details on cell statuses, see “About Cell Statuses” on page 25.</p> <p>Input/output argument.</p>
<i>pvaralColors</i>	<p>Returns an array of numbers representing the colors that indicate the cell statuses. The data is organized in row major order.</p> <p>The numbers correspond to colors used in the Data Explorer. These numbers are mapped to styles in the <code>DataCellStyles.css</code> style sheet, which is part of the Financial Management installation for the Web server. This mapping works as follows:</p> <ul style="list-style-type: none"> • The styles are named according to the following convention, with <i>n</i> corresponding to the numbers returned in this array: <pre>.datagrid TD INPUT.COLOR<i>n</i></pre> • The colors of the cell and font are defined by the applicable style's <code>BACKGROUND-COLOR</code> and <code>COLOR</code> attributes, respectively. For example, if 2 is returned as an array item, the color would be the <code>BACKGROUND-COLOR</code> attribute of the <code>.datagrid TD INPUT.COLOR2</code> style. <p>Note: If the style does not contain a <code>BACKGROUND-COLOR</code> attribute, the cell color defaults to white. If the style does not contain a <code>COLOR</code> attribute, the font color defaults to black.</p> <p>Input/output argument.</p>
<i>pvaralUnfilteredRowNumbers</i>	<p>Returns an array of the unfiltered row numbers in the data grid.</p> <p>Input/output argument.</p>

GetProcessManagementSummary

Returns a summary of process management information.

Syntax

```
<HFMwDataGrid>.GetProcessManagementSummary pvarbstrReviewLevelsSummary,
pvarbstrProcessFlowStatusSummary, pvarbstrValidationStatusSummary
```

Argument	Description
<i>pvarbstrReviewLevelsSummary</i>	<p>Returns a comma-delimited list containing numerical representations of the process management review levels for which summary information is being returned. The numerical representations correspond to the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434.</p> <p>Input/output argument.</p>
<i>pvarbstrProcessFlowStatusSummary</i>	<p>Returns a comma-delimited list containing the number of process units that passed and failed validation for promotion.</p>

Argument	Description
	Input/output argument.
<i>pvarbstrValidationStatusSummary</i>	Returns a comma-delimited list containing the number of process units that passed and failed account validation.
	Input/output argument.

GetSourceTransactions

Returns information on source transactions for a cell in a statutory application. The following information is returned:

- Header information, such as the dimension members that define the cell, the current username, the cell's data, and so on. This information is returned in two arrays that have a one-to-one correspondence; one array contains labels for the header information, the other contains the values that correspond to the labels.
- Detailed information on the transactions, such as the dimension members and data for the transactions. This information is contained in several arrays; the arrays contain one item per transaction, and have a one-to-one correspondence.
- The sums of the cell's source and destination transaction amounts.

Tip: To return source transaction information as an XML string, use [GetTransactionInfoAsXML](#).

Syntax

```
<HFMwDataGrid>.GetSourceTransactions lRow, lColumn,
pvaravarbstrHeaderLabels, pvaravarbstrHeaderValues,
pvaravarbstrCurrentPeriod, pvaravarbstrCurrentParent,
pvaravarbstrCurrentEntity, pvaravarbstrDestEntity, pvaravarbstrDestValue,
pvaravarbstrDestAccount, pvaravarbstrDestICP, pvaravarbstrDestCustom1,
pvaravarbstrDestCustom2, pvaravarbstrDestCustom3, pvaravarbstrDestCustom4,
pvaravarbstrDestData, pvaravarbstrSourceData, pvaravarbstrFactor,
pvaravarbstrNature, pvarbstrTotalDestData, pvarbstrTotalSourceData
```

Argument	Description
<i>lRow</i>	The index of the row that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>lColumn</i>	The index of the column that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>pvaravarbstrHeaderLabels</i>	Returns an array of labels for the header information. Output argument.
<i>pvaravarbstrHeaderValues</i>	Returns an array of values for the header information. Output argument.

Argument	Description
<i>pvaravabstrCurrentPeriod</i>	Returns an array of labels for the transactions' current Period dimension members. Output argument.
<i>pvaravabstrCurrentParent</i>	Returns an array of labels for the transactions' current parent Entity dimension members. Output argument.
<i>pvaravabstrCurrentEntity</i>	Returns an array of labels for the transactions' current child Entity dimension members. Output argument.
<i>pvaravabstrDestEntity</i>	Returns an array of labels for the transactions' destination Entity dimension members. Output argument.
<i>pvaravabstrDestValue</i>	Returns an array of labels for the transactions' destination Value dimension members. Output argument.
<i>pvaravabstrDestAccount</i>	Returns an array of labels for the transactions' destination Account dimension members. Output argument.
<i>pvaravabstrDestICP</i>	Returns an array of labels for the transactions' destination Intercompany Partner dimension members. Output argument.
<i>pvaravabstrDestCustom1</i>	Returns an array of labels for the transactions' destination Custom 1 dimension members. Output argument.
<i>pvaravabstrDestCustom2</i>	Returns an array of labels for the transactions' destination Custom 2 dimension members. Output argument.
<i>pvaravabstrDestCustom3</i>	Returns an array of labels for the transactions' destination Custom 3 dimension members. Output argument.
<i>pvaravabstrDestCustom4</i>	Returns an array of labels for the transactions' destination Custom 4 dimension members. Output argument.
<i>pvaravabstrDestData</i>	Returns an array of the transactions' destination data amounts. Output argument.
<i>pvaravabstrSourceData</i>	Returns an array of the transactions' source data amounts. Output argument.
<i>pvaravabstrFactor</i>	Returns an array of the factors used in the transactions. Output argument.
<i>pvaravabstrNature</i>	Returns an array of the audit strings specified for the transactions. Output argument.
<i>pvarbstrTotalDestData</i>	Returns the sum of the destination transaction amounts. Output argument.

Argument	Description
<i>pvarbstrTotalSourceData</i>	Returns the sum of the source transaction amounts. Output argument.

GetTransactionInfoAsXML

Returns an XML string containing information on transactions for a cell in a statutory application. You can specify whether source transactions, destination transactions, or both are returned.

Tip: To return transaction information in a non-XML format, use [GetDestinationTransactions](#) or [GetSourceTransactions](#).

Syntax

```
<HFMwDataGrid>.GetTransactionInfoAsXML (lRow, lColumn, lFlagsRequestedInfo)
```

Argument	Description
<i>lRow</i>	The index of the row that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>lColumn</i>	The index of the column that contains the cell for which to return transaction information. Input argument. Long subtype.
<i>lFlagsRequestedInfo</i>	Specifies the type of information to be returned in the XML string. Pass one of the HFMConstants type library listed in “Data Grid Transaction Information Constants” on page 449 . Tip: You can use the addition operator (+) with these constants to return multiple types of information. Input argument. Long subtype.

Return Value

Returns an XML string that represents the transaction information. The XML string is described in [“Transaction Information DTD” on page 483](#).

Example

The following example prints the specified cell’s transaction header information and source transactions to a file. Note how the call to `GetTransactionInfoAsXML` uses the addition operator to return both header and source transaction information.

```
Dim sXml, oFs, oTs
' cHFMDDataGrid is a previously set HFMwDataGrid object
sXML = cHFMDDataGrid.GetTransactionInfoAsXML (0, 0, _
WEBOM_DATAGRID_TRANSACTION_INFO_HEADER + _
WEBOM_DATAGRID_TRANSACTION_INFO_SOURCE)
' Write the string to disk as a text file
set oFs = CreateObject("Scripting.FileSystemObject")
```



```
' sFile is a previously set string containing the output file's
' name and path
set oTs = oFs.CreateTextFile(sFile)
oTs.Write(sXml)
oTs.Close
```

GetUnassignedGroups

Returns the names of submission groups assigned to dimension members but not to submission phases, and of submission groups assigned to submission phases but not to members.

Syntax

```
<HFMwDataGrid>.GetUnassignedGroups lScenario, lPeriodID,
pvarbstrMetaNoPhase, pvarbstrPhaseNoMeta
```

Argument	Description
<i>lScenario</i>	The label of the process unit's Entity dimension member. Input argument. Long subtype.
<i>lPeriodID</i>	The label of the process unit's Period dimension member. Input argument. Long subtype.
<i>pvarbstrMetaNoPhase</i>	Returns the names of the submission groups assigned to dimension members but not to submission phases. Input/output argument.
<i>pvarbstrPhaseNoMeta</i>	Returns the names of the submission groups assigned to submission phases but not to members. Input/output argument.

ProcessFlowGetInfo

Returns a variety of process management-related information for the specified cell.

Tip: To return this process management information as an XML string, use [ProcessFlowGetInfoAsXML](#). To return the names and paths of document attachments with this type of information, use [ProcessFlowGetInfo2](#).

Syntax

```
<HFMwDataGrid>.ProcessFlowGetInfo lRow, lCol, pvarbstrScenario,
pvarbstrYear, pvarbstrPeriod, pvarbstrEntity, pvarbstrParent,
pvarbstrValue, pvarbstrUserName, pvaravrbAccessRightsToProcessFlowRoles,
pvarlSecurityClassAccessRightsToSubCube,
pvarlCompleteAccessRightsToSubCube, pvarsProcessState, pvaravardTime,
pvaravrbstrUser, pvaravarsAction, pvaravarsNewState,
pvaravrbstrAnnotation
```

Argument	Description
<i>IRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>ICol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.
<i>pvarbstrScenario</i>	Returns the name of the cell's scenario. Output argument.
<i>pvarbstrYear</i>	Returns the name of the cell's year. Output argument.
<i>pvarbstrPeriod</i>	Returns the name of the cell's period. Output argument.
<i>pvarbstrEntity</i>	Returns the name of the cell's child entity. Output argument.
<i>pvarbstrParent</i>	Returns the name of the cell's parent entity. Output argument.
<i>pvarbstrValue</i>	Returns the name of the cell's Value dimension member. Output argument.
<i>pvarbstrUserName</i>	Returns the username of the currently connected user. Output argument.
<i>pvaravarbAccessRightsToProcessFlowRoles</i>	<p>Returns an array of Booleans indicating the connected user's access rights for the various process management roles.</p> <p>The array contains an item for each process management-related role. The HFMConstants type library constants that represent roles are listed in "Role Constants" on page 438.</p> <p>An array item returns TRUE if the user has access to the role, FALSE otherwise.</p> <p>Output argument.</p>
<i>pvarlSecurityClassAccessRightsToSubCube</i>	<p>Returns the connected user's access rights to the security class that applies to the cell. Access rights are represented by the HFMConstants type library constants listed in "Access Rights Constants" on page 438.</p> <p>Output argument.</p>
<i>pvarlCompleteAccessRightsToSubCube</i>	<p>Returns the connected user's access rights to the cell's process unit. Access rights are represented by the HFMConstants type library constants listed in "Access Rights Constants" on page 438.</p> <p>Output argument.</p>
<i>pvarsProcessState</i>	<p>Returns the current process management state of the cell. The following list shows the numbers that represent process management states:</p> <ul style="list-style-type: none"> ● 1 = Not Started

Argument	Description
	<ul style="list-style-type: none"> ● 2 = First Pass ● 3 = Review Level 1 ● 4 = Review Level 2 ● 5 = Review Level 3 ● 6 = Review Level 4 ● 7 = Review Level 5 ● 8 = Review Level 6 ● 9 = Review Level 7 ● 10 = Review Level 8 ● 11 = Review Level 9 ● 12 = Review Level 10 ● 13 = Submitted ● 14 = Approved ● 15 = Published <p>Output argument.</p>
<i>pvaravardTime</i>	<p>Returns an array of DateTimeStamps for the cell's process management actions.</p> <p>Output argument.</p>
<i>pvaravarbstrUser</i>	<p>Returns an array of the usernames who performed the cell's process management actions.</p> <p>Output argument.</p>
<i>pvaravarsAction</i>	<p>Returns an array of the cell's process management actions. The valid return values are represented by the HFMConstants type library constants listed in "Data Explorer Process Management Constants" on page 447.</p> <p>Output argument.</p>
<i>pvaravarsNewState</i>	<p>Returns an array of the states that resulted from the cell's process management actions.</p> <p>Output argument.</p>
<i>pvaravarbstrAnnotation</i>	<p>Returns an array of the comments for the cell's process management actions.</p> <p>Output argument.</p>

ProcessFlowGetInfo2

Returns a variety of process management-related information for the specified cell, including the names and paths of document attachments.

Note: The method contains several input/output arguments; however, input values are ignored.

Syntax

```
<HFMwDataGrid>.ProcessFlowGetInfo2 lRow, lCol, pvarbstrScenario,  
pvarbstrYear, pvarbstrPeriod, pvarbstrEntity, pvarbstrParent,  
pvarbstrValue, pvarbstrUserName, pvaravarbAccessRightsToProcessFlowRoles,  
pvarlSecurityClassAccessRightsToSubCube,  
pvarlCompleteAccessRightsToSubCube, pvarsProcessState, pvaravardTime,  
pvaravarbstrUser, pvaravarsAction, pvaravarsNewState,  
pvaravarbstrAnnotation, pvaravarabstrAttPaths, pvaravarabstrAttFiles
```

Argument	Description
<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.
<i>pvarbstrScenario</i>	Returns the name of the cell's scenario. Input/output argument.
<i>pvarbstrYear</i>	Returns the name of the cell's year. Input/output argument.
<i>pvarbstrPeriod</i>	Returns the name of the cell's period. Input/output argument.
<i>pvarbstrEntity</i>	Returns the name of the cell's child entity. Input/output argument.
<i>pvarbstrParent</i>	Returns the name of the cell's parent entity. Input/output argument.
<i>pvarbstrValue</i>	Returns the name of the cell's Value dimension member. Input/output argument.
<i>pvarbstrUserName</i>	Returns the username of the currently connected user. Input/output argument.
<i>pvaravarbAccessRightsToProcessFlowRoles</i>	Returns an array of Booleans indicating the connected user's access rights for the various process management roles. The array contains an item for each process management-related role. The HFMConstants type library constants that represent roles are listed in "Role Constants" on page 438 . An array item returns TRUE if the user has access to the role, FALSE otherwise. Input/output argument.
<i>pvarlSecurityClassAccessRightsToSubCube</i>	Returns the connected user's access rights to the security class that applies to the cell. Access rights are represented by the HFMConstants type library constants listed in "Access Rights Constants" on page 438 . Input/output argument.

Argument	Description
<i>pvarlCompleteAccessRightsToSubCube</i>	<p>Returns the connected user's access rights to the cell's process unit. Access rights are represented by the HFMCConstants type library constants listed in "Access Rights Constants" on page 438.</p> <p>Input/output argument.</p>
<i>pvarsProcessState</i>	<p>Returns the current process management state of the cell. The following list shows the numbers that represent process management states:</p> <ul style="list-style-type: none"> ● 1 = Not Started ● 2 = First Pass ● 3 = Review Level 1 ● 4 = Review Level 2 ● 5 = Review Level 3 ● 6 = Review Level 4 ● 7 = Review Level 5 ● 8 = Review Level 6 ● 9 = Review Level 7 ● 10 = Review Level 8 ● 11 = Review Level 9 ● 12 = Review Level 10 ● 13 = Submitted ● 14 = Approved ● 15 = Published <p>Input/output argument.</p>
<i>pvaravardTime</i>	<p>Returns an array of DateTimeStamps for the cell's process management actions.</p> <p>Input/output argument.</p>
<i>pvaravarbstrUser</i>	<p>Returns an array of the usernames who performed the cell's process management actions.</p> <p>Input/output argument.</p>
<i>pvaravarsAction</i>	<p>Returns an array of the cell's process management actions. The valid return values are represented by the HFMCConstants type library constants listed in "Data Explorer Process Management Constants" on page 447.</p> <p>Input/output argument.</p>
<i>pvaravarsNewState</i>	<p>Returns an array of the states that resulted from the cell's process management actions.</p> <p>Input/output argument.</p>
<i>pvaravarbstrAnnotation</i>	<p>Returns an array of the comments for the cell's process management actions.</p> <p>Input/output argument.</p>
<i>pvaravarabstrAttPaths</i>	<p>Returns an array containing the paths of the cell's document attachments.</p>

Argument	Description
	Input/output argument.
<i>pvaravarabstrAttFiles</i>	Returns an array containing the names of the cell's document attachments. This array has a one-to-one correspondence with the array returned by the <i>pvaravarabstrAttPaths</i> argument.
	Input/output argument.

ProcessFlowGetInfoAsXML

Returns an XML representation of process management-related information for the specified cell.

Tip: To return this process management information in a non-XML format, use [ProcessFlowGetInfo](#).

Syntax

```
<HFMwDataGrid>.ProcessFlowGetInfoAsXML (lRow, lCol)
```

Argument Description

<i>lRow</i>	The index (0-based) that identifies the cell's row. Input argument. Long subtype.
<i>lCol</i>	The index (0-based) that identifies the cell's column. Input argument. Long subtype.

Return Value

Returns an XML string that represents the cell's process management information. The XML string is described in “[Process Management Information DTD](#)” on page 480.

SaveDirtyCellsOnLastGridPage

Saves the specified cells in the grid to the database.

Tip: To save cells from an XML string, use [SaveDirtyCellsOnLastGridPageFromXML](#).

Syntax

```
<HFMwDataGrid>.SaveDirtyCellsOnLastGridPage varavarDirtyRows,  
varavarDirtyCols, varavarDirtyCells
```

Argument Description

<i>varavarDirtyRows</i>	An array of the row numbers of the cells to be saved.
-------------------------	---

Argument	Description
	Input argument.
<i>varavarDirtyCols</i>	An array of the column numbers of the cells to be saved. Input argument.
<i>varavarDirtyCells</i>	An array of the data to be saved for the cells. This array has a one-to-one correspondence with the arrays passed to the <i>varavarDirtyRows</i> and <i>varavarDirtyCols</i> arguments' arrays. Input argument.

SaveDirtyCellsOnLastGridPageFromXML

Saves the cells specified in ASP Request object that contains an XML string of a grid's cells.

Tip: To save cells from a non-XML source, use [SaveDirtyCellsOnLastGridPage](#).

Syntax

```
<HFMwDataGrid>.SaveDirtyCellsOnLastGridPageFromXML varpIRequest
```

Argument	Description
<i>varpIRequest</i>	An ASP Request object containing an XML string that represents the grid cells to be saved. The XML string is documented in “Data Grid Data and Headers DTD” on page 477 . However, do not include the following elements of the DTD in the XML string: <ul style="list-style-type: none"> ● <RHDRS> ● <CHDRS> ● <HDR> In addition, for the <CELL> tag, include only the <i>row</i> , <i>col</i> , and <i>value</i> attributes. Input argument. String subtype.

SaveLastGridPage

Saves the last page that was retrieved from the grid.

Tip: For information on pages, see [“Pages on Grids” on page 202](#).

Syntax

```
<HFMwDataGrid>.SaveLastGridPage varavarNewCells
```

Argument	Description
<i>varavarNewCells</i>	An array with strings representing the cell values for the last page that was accessed. The array is organized in a row major order.

Argument	Description
	You can obtain these cell values with GetGridPage .
	Input argument.

SavePhaseGroupInfoUsingXML

For internal use.

SaveUIStateUsingXML

Saves the grid's user interface state information using an XML string.

Tip: To get grid information as an XML string, use [GetGridDefinitionAsXML](#) or [GetGridDefinitionAsXMLEx](#). To save a grid by passing an XML string in an ASP Request object, use [SaveUIStateUsingXMLFromRequest](#).

Syntax

```
<HFMwDataGrid>.SaveUIStateUsingXML bstrXMLStateInfo
```

Argument	Description
<i>bstrXMLStateInfo</i>	An XML string containing user interface state information, which is represented by the <UISTATE> element described in "Data Grid Definition DTD" on page 474
	Input argument. String subtype.

SaveUIStateUsingXMLFromRequest

Saves the grid's user interface state information using an XML string that is contained by an ASP Request object.

Tip: To get grid information as an XML string, use [GetGridDefinitionAsXML](#) or [GetGridDefinitionAsXMLEx](#).

Syntax

```
<HFMwDataGrid>.SaveUIStateUsingXMLFromRequest varpIRequest
```

Argument	Description
<i>varpIRequest</i>	An ASP Request object containing an XML string that defines the user interface state information. The user interface state is represented by the <UISTATE> element described in "Data Grid Definition DTD" on page 474
	Input argument.

SetOrgByPeriodFilteringInfo

For Organization by Period applications, specifies whether active entities are filtered for display on the grid. If filtering is enabled, `SetOrgByPeriodFilteringInfo` also specifies the filtering criteria.

`SetOrgByPeriodFilteringInfo` has no effect for non-Organization by Period applications. Note that the `HFMwMetadata` component's `isOrgByPeriod` property indicates whether an application is an Organization by Period application.

Tip: The filtering criteria arguments take dimension member IDs. To return a member's ID from its label, pass the label to `HFMwDimension.GetMemberID`.

Syntax

```
<HFMwDataGrid>.SetOrgByPeriodFilteringInfo varbEnableFiltering,  
IOBPScenarioMemberID, IOBPYearMemberID, IOBPPeriodMemberID
```

Argument	Description
<code>varbEnableFiltering</code>	Specifies whether filtering is enabled. Pass TRUE to enable filtering, FALSE to disable filtering. Input argument.
<code>IOBPScenarioMemberID</code>	The member ID of the Scenario dimension member to filter by. Input argument. Long subtype.
<code>IOBPYearMemberID</code>	The member ID of the Year dimension member by which to filter. Input argument. Long subtype.
<code>IOBPPeriodMemberID</code>	The member ID of the Period dimension member by which to filter. Input argument. Long subtype.

SetPhaseGroupInfoAsJavaScript

For internal use.

SetProcessManagementFilterAndSortOptions

Sets the process management filtering and sorting criteria for the `HFMwDataGrid` instance.

Tip: The `totalRowCount` and `vFilteredPageCount` properties return the number of rows and pages in the grid after filtering has occurred.

Syntax

```
<HFMwDataGrid>.SetProcessManagementFilterAndSortOptions  
lReviewLevelFilter, lReviewLevelFilterCondition, lPassFailFilter,  
lReviewLevelSortOrder
```

Argument	Description
<i>lReviewLevelFilter</i>	<p>The review level by which to filter. Valid values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 434.</p> <p>The <i>lReviewLevelFilterCondition</i> argument determines whether filtering is for all review levels above or below the specified level, or only for the specified level.</p> <p>Note: The ReviewLevelFilter property returns the value set by this argument.</p> <p>Input argument. Long subtype.</p>
<i>lReviewLevelFilterCondition</i>	<p>The condition by which to filter the review level specified in the <i>lReviewLevelFilter</i> argument. For example, you can use this argument to filter all process units above the review level specified with the <i>lReviewLevelFilter</i> argument. Valid values are represented by the HFMConstants type library constants listed in “Process Management Filters” on page 435 other than those listed for the <i>lPassFailFilter</i> argument.</p> <p>Note: The ReviewLevelFilterCondition property returns the value set by this argument.</p> <p>Input argument. Long subtype.</p>
<i>lPassFailFilter</i>	<p>The validation for promotion state by which to filter. Valid values are represented by the following HFMConstants type library constants, which are described in “Process Management Filters” on page 435.</p> <ul style="list-style-type: none"> ● PROCESS_FLOW_FILTER_PASSFAIL ● PROCESS_FLOW_FILTER_PASSONLY ● PROCESS_FLOW_FILTER_FAILONLY <p>Note: The PassFailFilter property returns the value set by this argument.</p> <p>Input argument. Long subtype.</p>
<i>lReviewLevelSortOrder</i>	<p>The order in which to sort process units in the grid. Valid values are represented by the HFMConstants type library constants listed in “Process Management Sorting Constants” on page 436.</p> <p>Note: The ReviewLevelSortOrder property returns the value set by this argument.</p> <p>Input argument. Long subtype.</p>

SetRowandColumnCount

Sets the number of rows and columns of a selected group of cells in a data grid on which an action is being performed.

Syntax

```
<HFMwDataGrid>.SetRowandColumnCount lRow, lCol
```

Argument Description

<i>lRow</i>	<p>The index (0-based) that identifies the cell's row.</p> <p>Input argument. Long subtype.</p>
<i>lCol</i>	<p>The index (0-based) that identifies the cell's column.</p>

Argument	Description
----------	-------------

	Input argument. Long subtype.
--	-------------------------------

SetWebSession

Associates an HFMwDataGrid object with the HFMwSession object for an application. You must call SetWebSession before calling any other HFMwDataGrid property or method.

Syntax

```
<HFMwDataGrid>.SetWebSession varpIDispHFMwSession
```

Argument	Description
----------	-------------

<i>varpIDispHFMwSession</i>	The HFMwSession object for the application. Input argument.
-----------------------------	--

Example

SetWebSession is used in the example for [maxColsPerPage](#).

StartConsolidation

Starts a consolidation task for the specified range of cells.

Syntax

```
<HFMwDataGrid>.StartConsolidation lRowStart, lColStart, lRowEnd, lColEnd,  
sConsolidationType
```

Argument	Description
----------	-------------

<i>lRowStart</i>	The index (0-based) that identifies the first row in the range of cells. Input argument. Long subtype.
------------------	---

<i>lColStart</i>	The index (0-based) that identifies the first column in the range of cells. Input argument. Long subtype.
------------------	--

<i>lRowEnd</i>	The index that identifies the last row in the range of cells. Input argument. Long subtype.
----------------	--

<i>lColEnd</i>	The index that identifies the last column in the range of cells. Input argument. Long subtype.
----------------	---

<i>sConsolidationType</i>	Identifies the type of consolidation to be run. Pass one of the HFMConstants type library constants listed in "Consolidation Type Constants" on page 437 . Input argument. Integer subtype.
---------------------------	--

StopConsolidation

Future use.

SubmitProcessStateChangeFromXML

Submits the process management changes specified in an XML string that is contained by an ASP Request object, and, if e-mail alerting is enabled, returns a list of the process units that may require alerts.

Note: To get information on users who should receive e-mail alerts for the process units, pass the value returned by the *pvarbstrXML* argument to [GenerateAlerts](#).

Syntax

```
<HFMwDataGrid>.SubmitProcessStateChangeFromXML (varpIRequest)
```

Argument	Description
----------	-------------

<i>varpIRequest</i>	An ASP Request object containing an XML string that represents the process management action to be applied. The XML string is described in Table 6, “Process Management XML String,” on page 252 . Input argument.
---------------------	---

<i>pvarbstrXML</i>	Returns an XML string representing the process units that were successfully changed. The format of the XML string is described in Table 7, “Process Unit XML String,” on page 253 .
--------------------	---

Note: This argument returns an XML representation of process units only if e-mail alerting is enabled.

Return Value

Returns an HRESULT – 0 indicates success, a non-zero value indicates an error. You should always check the return value for a non-zero HRESULT.

Tip: You can use the HsvResourceManager object to obtain more information on non-zero return values. For details, see [“HsvResourceManager Object Methods” on page 392](#).

XML Strings for SubmitProcessStateChangeFromXML

The following table describes the XML string that represents process management actions.

Table 6 Process Management XML String

Element	Description
CHANGEPROCESSSTATE	Root element. This contains the Boolean <code>descendants</code> attribute, which indicates whether to apply the process management action to all descendants of the cells' entities.
CELLS	Represents the grid cells to be processed. A <CELLS> tag contains one <CELL> tag for each cell.
CELL	Represents a cell to be processed. <CELL> contains the following attributes:

Element	Description
	<ul style="list-style-type: none"> row = The row number (0-based). col = The column number (0-based).
ACTION	A number that represents the action to apply. Valid values are represented by the HFMConstants type library enumeration CEnumProcessFlowActions, which is described in “Process Management Action Constants” on page 433 .
REVIEWLEVEL	A number that represents the review level to apply. Valid values are represented by the HFMConstants type library enumeration CEnumProcessFlowStates, which is described in “Process Management Review Level Constants” on page 434 .
COMMENT	The comment to apply to the process units.
ATTACHMENTS	Represents files to be attached to the process units. An <ATTACHMENTS> tag contains one <ATTACHMENT> tag for each file to be attached.
ATTACHMENT	Represents a file attachment, and contains the following attributes: <ul style="list-style-type: none"> name = The attachment’s file name. path = The attachment’s path.

The following table describes the XML string that represents successfully changed process units.

Table 7 Process Unit XML String

Element	Description
events	<p>Root element. <events> contains the following attributes:</p> <ul style="list-style-type: none"> xmlns = The namespace, which will be a URL similar to the following example: http://hyperion.com/HFM/4.1/Alerts/1.0 type = The string “ProcMgmt”. <p><events> also contains an <event> tag.</p>
event	<p>Represents the process management action and the users to be notified. <event> contains the following attributes:</p> <ul style="list-style-type: none"> sid = The security identifier of the user who changed the process unit. user = The username of the user who changed the process unit. time = The date and time of the process management action. action = The label of the process management action. newState = The ID of the review level applied to the action’s process units. Review level IDs are represented by the HFMConstants enumeration CEnumProcessFlowStates, which is described in “Process Management Review Level Constants” on page 434. actionId = The ID of the process management action. Valid values are represented by the HFMConstants enumeration CEnumProcessFlowActions, which is described in “Process Management Action Constants” on page 433. <p><event> also contains a <comment> tag, as well as one <item> tag per process unit.</p>
comment	Contains the comment for the process management action.
item	Represents a process unit to which the process management action has been applied. <item> contains the following attributes:

Element	Description
	<ul style="list-style-type: none"> ● <code>scenarioLabel</code> = The label of the process unit's Scenario dimension member. ● <code>yearLabel</code> = The label of the process unit's Year dimension member. ● <code>periodLabel</code> = The label of the process unit's Period dimension member. ● <code>entityLabel</code> = The label of the process unit's Entity dimension member. ● <code>entityDesc</code> = The description of the process unit's Entity dimension member. ● <code>action</code> = A period-delimited string describing the process unit. Following is the string's format: <i>Scenario.Year.Period.Entity.OrigLevel</i> The string's first four items list the labels of the Scenario, Year, Period, and Entity dimensions. The last item contains the label of the process unit's previous review level. ● <code>origState</code> = The ID of the process unit's previous review level. ● <code>newState</code> = The ID of the process unit's new review level.

HFMDDataGrid Properties

The HFMDDataGrid component contains the properties described in the following topics.

colPhases

Returns or sets the submission phases displayed on the grid. The property takes an array of phase IDs, which can range from 1 through 9, and should be specified in ascending order. You must cast the phase IDs to Longs.

Read-write.

Example

The following example sets 1 and 2 as the grid's submission phases.

```
dim vaColPhases(1)
vaColPhases(0) = CLng(1)
vaColPhases(1) = CLng(2)
'cHFMDDataGrid is an HFMDDataGrid object reference
cHFMDDataGrid.colPhases = vaColPhases
```

columnDimensions

Returns or sets the dimensions for the grid's columns. The property returns or takes an array of dimension IDs. The dimension IDs are represented by the HFMDConstants type library constants in [“Dimension ID Constants” on page 414](#).

Read-write.

Example

`columnDimensions` is used in the example for [maxColsPerPage](#).

consolStatusFilter

Specifies the grid's consolidation status filter. Valid values are represented by the HFMConstants enumeration tagCALCSTATUSSTATISTICS, which is described in [“Calculation Status Statistic Constants” on page 426](#).

Write-only.

dataDisplayMode

Gets or sets the grid's data display mode. Valid values are represented by the HFMConstants enumeration tagWEBOM_DATAGRID_DATA_DISPLAY_FLAGS, which is described in [“Data Information Display Constants” on page 449](#).

Read-write.

gridNumDecimals

Gets or sets the grid's number of decimal places. A grid can either use or override the number of decimal spaces specified in Account dimension members. Valid values are as follows:

- 9999: Use the number of decimal places specified in Account members.
- Integers from 0 to 9: Override Account members and apply the specified number of decimal places.

Read-write.

gridScale

Gets or sets the grid's scaling. A grid can either use or override the scaling specified in the default currencies of Entity dimension members. Valid values are as follows:

- 9999: Use the scaling specified in the default currencies of Entity members.
- Integers from -12 to 12: Override Entity members' scaling and apply the degree of scaling represented by the integer.

Read-write.

hPageCount

Returns the number of pages into which the grid would be divided based upon the grid's total number of columns and the [maxColsPerPage](#) property. Use hPageCount with the [vPageCount](#) property to loop through the pages in the grid.

Tip: For information on pages, see [“Pages on Grids” on page 202](#).

Read-only.

Example

The following example uses the `hPageCount` and `vPageCount` properties to loop through the pages in the grid. Any code placed in the nested `For . . . Next` loop would be executed for each page in the grid.

```
For i = 0 to CHFMDDataGrid.hPageCount - 1
  For j = 0 to CHFMDDataGrid.vPageCount - 1
    ' Any code here would be executed for each page
  Next
Next
```

listStyle

Gets or sets whether the grid represents members as a tree or a flat list. This property applies only to Process Management grids. Valid values are listed below:

- 0 = list
- 1 = tree

Read-write.

maxColsPerPage

Returns or sets the maximum number of columns on a page. If you are setting this property, the value passed must be cast as a `Long`.

Tip: For information on pages, see [“Pages on Grids” on page 202](#).

Read-write.

Example

The following example defines the grid for an `HFMwDataGrid` object. The column dimensions are `Account` and `Period`, and the row dimension is `Entity`. The members of the row and column dimensions are specified with the dimensions’ [Hierarchy] member lists, while the members of the page dimensions are set to the dimensions’ default members.

```
Dim CHFMDDataGrid, CHFMMetadata, CHFMDimension
Dim vaColDims(1), vaRowDims(0), vaListIDs, vaListNames
Set CHFMDDataGrid = Server.CreateObject("Hyperion.HFMwDataGrid")
'g_CHFMSession is a previously set HFMwSession object
CHFMDDataGrid.SetWebSession g_CHFMSession
' Set the maximum columns/rows per page
CHFMDDataGrid.maxColsPerPage = CLng(50)
CHFMDDataGrid.maxRowsPerPage = CLng(50)
'Set the column and row dimensions
vaColDims(0) = DIMENSIONACCOUNT
vaColDims(1) = DIMENSIONPERIOD
CHFMDDataGrid.columnDimensions = vaColDims
vaRowDims(0) = DIMENSIONENTITY
CHFMDDataGrid.rowDimensions = vaRowDims
```



```

Set cHFMMetadata = g_cHFMSession.metadata
cHFMDDataGrid.parentID = MEMBERNOTUSED
'Loop thru the dimensions
For i = DIMENSION_LBOUND To DIMENSION_UBOUND
    Set cHFMDimension = cHFMMetadata.dimension(i)
    ' If a row or column dimension, use [Hierarchy]
    If i = DIMENSIONENTITY Or i = DIMENSIONACCOUNT or _
        i = DIMENSIONPERIOD Then
        'Get the ID for the [Hierarchy] member list
        cHFMDimension.EnumMemberLists WEBOM_METADATA_INFO_ALL, _
            vaListIDs, vaListNames
        For j = 0 to uBound(vaListNames)
            If vaListNames(j) = "[Hierarchy]" Then
                lListID = vaListIDs(j)
                Exit For
            End If
        Next
        cHFMDDataGrid.memberListID(i) = lListID
        cHFMDDataGrid.useList(i) = TRUE
        ' Otherwise it's a page dimension
    Else
        cHFMDDataGrid.memberID(i) = cHFMDimension.defaultMemberID
    End If
Next

```

maxRowsPerPage

Returns or sets the maximum number of rows on a page. If you are setting this property, the value passed must be cast as a Long.

Tip: For information on pages, see [“Pages on Grids” on page 202](#).

Read-write.

Example

maxRowsPerPage is used in the example for [maxColsPerPage](#).

memberID

Returns or sets a single member for the specified page dimension; the member is identified by a member ID. If you are setting this property, the member ID must be cast as a Long.

Tip: You can return a member ID from a member name with [GetMemberID](#) (HFMwDimension component).

Read-write.

Syntax

```
<HFMwDataGrid>.memberID(lDimensionID)
```

Argument	Description
----------	-------------

<i>lDimensionID</i>	Specifies the dimension to which the member belongs. Pass one of the HFMConstants type library constants listed in “Dimension ID Constants” on page 414 .
---------------------	---

Input argument. Long subtype.

Example

memberID is used in the example for [maxColsPerPage](#).

memberListID

Returns or sets the ID of the default member list for the specified row or column dimension. If you are setting this property, the value passed must be cast as a Long.

You can get the ID for a member list with [EnumMemberLists](#) (HFMwDimension component).

Note: The grid uses the specified member list only if the [useList](#) property is set to TRUE.

Read-write.

Syntax

```
<HFMwDataGrid>.memberListID lDimensionID
```

Argument	Description
----------	-------------

<i>lDimensionID</i>	Specifies the dimension for the member list. Pass one of the HFMConstants type library constants listed in “Dimension ID Constants” on page 414 .
---------------------	---

Input argument. Long subtype.

Example

memberListID is used in the example for [maxColsPerPage](#).

metadataDisplayMode

Returns or sets the display mode for the column and row headers. The valid values for this property are represented by the HFMConstants type library constants listed in [“Member Display Constants” on page 450](#).

Read-write.

parentID

Returns or sets the member ID of the parent of the grid's entity, which is represented by the [memberID](#) property. If you are setting this property, the value passed must be cast as a Long.

Note: `parentID` is used when Entity is set as a page dimension.

Read-write.

Example

`parentID` is used in the example for [maxColsPerPage](#).

PassFailFilter

Returns the validation for promotion state by which to filter process units. Valid values are represented by the following HFMConstants type library constants, which are described in [“Process Management Filters” on page 435](#).

- `PROCESS_FLOW_FILTER_PASSFAIL`
- `PROCESS_FLOW_FILTER_PASSONLY`
- `PROCESS_FLOW_FILTER_FAILONLY`

Note: Process management filtering criteria are set with [SetProcessManagementFilterAndSortOptions](#).

Write-only.

phaseID

Returns or sets the grid's submission phase.

Read-write.

processCtrlDispCols

Returns or sets the columns displayed in the grid. The property value is an array; valid values for the array items are described in [“Process Control Grid Constants” on page 463](#).

Read-write.

ReviewLevelFilter

Returns the review level filter by which to filter process units. Valid values are represented by the HFMConstants type library constants listed in [“Process Management Review Level Constants” on page 434](#).

The filtering criteria returned by the [ReviewLevelFilterCondition](#) property determines whether filtering is for all review levels above, all review levels below, or only the specific review level returned by [ReviewLevelFilter](#).

Note: Process management filtering criteria are set with [SetProcessManagementFilterAndSortOptions](#).

Write-only.

ReviewLevelFilterCondition

Returns the condition by which to filter the review level returned by the [ReviewLevelFilter](#) property. For example, you can use [ReviewLevelFilterCondition](#) to filter all process units above the [ReviewLevelFilter](#) property's review level. Valid values are represented by the [HFMCConstants](#) type library constants listed in "[Process Management Filters](#)" on page 435, excepting the constants related to validation for promotion.

Note: Process management filtering criteria are set with [SetProcessManagementFilterAndSortOptions](#).

Write-only.

ReviewLevelSortOrder

Returns the order in which to sort process units in the grid. Valid values are represented by the [HFMCConstants](#) type library constants listed in "[Process Management Sorting Constants](#)" on page 436.

Note: The process unit sort order is set with [SetProcessManagementFilterAndSortOptions](#).

Write-only.

rowDimensions

Returns or sets the dimensions for the grid's rows. The property returns or takes an array of dimension IDs. The dimension IDs are represented by the [HFMCConstants](#) type library constants in "[Dimension ID Constants](#)" on page 414.

Read-write.

Example

`rowDimensions` is used in the example for [maxColsPerPage](#).

selectedMemberIDs

Returns or sets multiple members as the grid's members for the specified dimension. Use this property for row and column dimensions. `selectedMemberIDs` takes or returns an array of member IDs. If you are setting this property, the member IDs must be cast as a Long.

Tip: You can return a member ID from a member name with [GetMemberID](#) (HFMwDimension component).

Read-write.

Syntax

```
<HFMwDataGrid>.selectedMemberIDs lDimensionID
```

Argument	Description
----------	-------------

<i>lDimensionID</i>	Specifies the dimension for the member list. Pass one of the HFMConstants type library constants listed in "Dimension ID Constants" on page 414 .
---------------------	---

Input argument. Long subtype.

selectedParentIDs

Returns or sets multiple members as the parents for the grid's entities. Use this property for row and column dimensions. `selectedParentIDs` takes or returns an array of member IDs. If you are setting this property, the member IDs must be cast as a Long.

You can get or set multiple entities for a grid with the [selectedMemberIDs](#) property.

Tip: You can return a member ID from a member name with [GetMemberID](#) (HFMwDimension component).

Read-write.

suppressNodataColumns

Future use.

suppressNodataRows

Future use.

suppressZeroRows

Future use.

supressZeroColumns

Future use.

topMemberID

Returns or sets the top member to be used in the grid for the specified row or column dimension. The member is identified by its member ID. If you are setting this property, the member ID must be cast as a Long.

To start at the top of the dimension hierarchy, use the HFMConstants type library constant `MEMBERNOTUSED`.

Tip: You can return a member ID from a member name with `GetMemberID` (HFMwDimension component).

Read-write.

Syntax

```
<HFMwDataGrid>.topMemberID lDimensionID
```

Argument	Description
----------	-------------

<i>lDimensionID</i>	Specifies the dimension for the member list. Pass one of the HFMConstants type library constants listed in "Dimension ID Constants" on page 414 .
---------------------	---

Input argument. Long subtype.

totalColumnCount

Returns the total number of columns on the grid.

Read-only.

totalFilteredRowCount

Returns the number of rows on the grid after filtering has occurred.

Note: Process management filtering criteria are set with `SetProcessManagementFilterAndSortOptions`.

Read-only.

totalRowCount

Returns the total number of rows on the grid.

Read-only.

useList

Returns or sets whether the specified row or column dimension should use a member list or specified members to populate the grid. This is a Boolean property: TRUE indicates that a member list will be used for the dimension, FALSE indicates that a member list will not be used.

Tip: A dimension's member list is specified or returned with [memberListID](#).

Read-write.

Syntax

```
<HFMwDataGrid>.useList lDimensionID
```

Argument	Description
<i>lDimensionID</i>	Specifies the dimension for the member list. Pass one of the HFMConstants type library constants listed in "Dimension ID Constants" on page 414 . Input argument. Long subtype.

Example

useList is used in the example for [maxColsPerPage](#).

vFilteredPageCount

Returns the number of pages in the grid after filtering has occurred.

Note: Process management filtering criteria are set with [SetProcessManagementFilterAndSortOptions](#).

Read-only.

vPageCount

Returns the number of pages into which the grid would be divided based upon the grid's total number of rows and the [maxRowsPerPage](#) property. Use vPageCount with the [hPageCount](#) property to loop through the pages in the grid.

Tip: For information on pages, see ["Pages on Grids" on page 202](#).

Read-only.

Example

`vPageCount` is used in the example for `hPageCount`.

In This Chapter

HFMwManageDocuments Component	265
HFMwWorkspace Component	281

The HFMwDocuments type library contains the following components:

- HFMwManageDocuments component, which exposes Financial Management's *document repository*. The document repository works with Financial Management documents such as data forms and reports that are stored in the database.
- HFMwWorkspace component, which exposes Financial Management's task list functionality.

HFMwManageDocuments Component

The HFMwManageDocuments component exposes the following functionality for Financial Management's document repository:

- Add documents to the database.
- Organize documents by folder.
- Return various types of information on documents in the database.
- Delete documents from the database.

Obtaining an HFMwManageDocuments Object Reference

Create an HFMwManageDocuments object with `Server.CreateObject`. After creating the object, call `SetWebSession` to associate the object with the HFMwSession object for the application for which you want to work with documents:

```
Set cHFMDocuments = _  
Server.CreateObject("Hyperion.HFMwManageDocuments")  
' cHFMSession is a previously set HFMwSession object  
cHFMDocuments.SetWebSession cHFMSession
```

HFmwManageDocuments Methods

The HFmwManageDocuments component contains the methods described in the following topics.

CreateFolder

Creates a folder in the document repository.

Tip: `CreateFolderEx` also creates a folder, and provides additional options for flagging the folder as private or public, overwriting existing folders, and specifying the folder's content type.

Syntax

```
<HFmwManageDocuments>.CreateFolder bstrPath, bstrName, bstrDescription,  
bstrSecurityClass
```

Argument	Description
<i>bstrPath</i>	The path of the document repository folder in which to create the new folder. Precede the new folder name with a backslash (\). Caution! If the specified path does not exist, an error occurs. To create the new folder in the root folder, pass an empty string. Input argument. String subtype.
<i>bstrName</i>	The name of the new folder. Input argument. String subtype.
<i>bstrDescription</i>	The description of the new folder. Input argument. String subtype.
<i>bstrSecurityClass</i>	The name of the security class for the folder. Input argument. String subtype.

Example

The following example creates a folder named Joe in the Sales folder.

```
Dim cHFMDocuments  
Set cHFMDocuments = Server.CreateObject("Hyperion.HFmwManageDocuments")  
'g_cHFMSession is an HFmwSession object reference  
cHFMDocuments.SetWebSession g_cHFMSession  
cHFMDocuments.CreateFolder "\Sales", "Joe", "Joe's folder", "[Default]"
```

CreateFolderEx

Creates a folder in the document repository, with options for flagging the folder as public or private, overwriting an existing folder, and specifying the type of documents that the folder can contain.

Tip: If you want to create a folder but do not need to specify the privacy, overwrite, and document type options, you can use [CreateFolder](#).

Syntax

```
<HFMwManageDocuments>.CreateFolderEx bstrPath, bstrName, bstrDescription,  
bstrSecurityClass, vbIsPrivate, lContentType, vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	The path of the document repository folder in which to create the new folder. Precede the new folder name with a backslash (\). Caution! If the specified path does not exist, an error occurs. To create the new folder in the root folder, pass an empty string. Input argument. String subtype.
<i>bstrName</i>	The name of the new folder. Input argument. String subtype.
<i>bstrDescription</i>	The description of the new folder. Input argument. String subtype.
<i>bstrSecurityClass</i>	The name of the security class for the folder. Input argument. String subtype.
<i>vbIsPrivate</i>	Specifies whether the folder is public or private. Pass TRUE for private, FALSE for public. Input argument. Boolean subtype.
<i>lContentType</i>	Specifies the type of documents that the folder can contain. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Tip: To specify that the folder can contain any type of document, pass the WEBOM_DOCTYPE_ALL constant. Input argument. Long subtype.
<i>vbOverwriteExisting</i>	Specifies whether to overwrite any folder of the same name. Pass TRUE to overwrite, FALSE otherwise. Input argument. Boolean subtype.

Example

The following subroutine creates a private task list folder and takes the folder’s path, name, description, and security class.

```
Sub CreatePrivateWorkFolder(sPath, sName, sDesc, sSecClass)
```

```

Dim cHFMDocuments
Set cHFMDocuments = _
    Server.CreateObject("Hyperion.HFMwManageDocuments")
' cHFMSession is a previously set HFMwSession object
cHFMDocuments.SetWebSession cHFMSession
cHFMDocuments.CreateFolderEx sPath, sName, sDesc, sSecClass, _
    TRUE, WEBOM_DOCTYPE_WORKSPACE, TRUE
End Sub

```

DeleteDocuments

Deletes the specified documents from the document repository.

The paths, names, document types, and file types that identify the documents to be deleted are passed in arrays that have a one-to-one correspondence with each other.

Syntax

```

<HFMwManageDocuments>.DeleteDocuments varabstrPaths, varabstrNames,
lDocumentType, lDocumentFileType, vbIncludeSubFolders

```

Argument	Description
<i>varabstrPaths</i>	An array of the document repository folders that contain the documents to be deleted. Each array item should contain the full path to the applicable folder. Input argument.
<i>varabstrNames</i>	An array of the names of the documents to be deleted. Input argument.
<i>lDocumentType</i>	An array of the document types for the documents to be deleted. Document types are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Input argument. Long subtype.
<i>lDocumentFileType</i>	An array of the file types for the documents to be deleted. File types are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 452 . Input argument. Long subtype.
<i>vbIncludeSubFolders</i>	<i>Future use.</i> You must pass a valid Boolean, but the value is ignored by this method. Input argument.

Example

The following subroutine deletes all data forms in the specified document repository folder. [EnumDocuments](#) returns the names of the forms in the folder, and then these names are passed to DeleteDocuments.

```

Sub DeleteFormsFromFolder(sPath)
Dim cHFMDocuments, vaDocNames, vaDescs, vaTime, vaSecClass
Dim vaPaths
Set cHFMDocuments = Server.CreateObject("Hyperion.HFMwManageDocuments")
'g_cHFMSession is an HFMwSession object reference
cHFMDocuments.SetWebSession g_cHFMSession

```

```

vaDocNames = cHFMDocuments.EnumDocuments (sPath, _
    WEBOM_DOCTYPE_WEBFORM, WEBOM_DOCFILETYPE_FORMDEF, _
    FALSE, FALSE, 0, 0, vaDescs, vaTime, vaSecClass)
' Create array of paths to pass to DeleteDocuments
ReDim vaPaths(uBound(vaDocNames))
For i = 0 To uBound(vaDocNames)
    vaPaths(i) = sPath
Next
For i = 0 To uBound(vaDocNames)
    cHFMDocuments.DeleteDocuments vaPaths, vaDocNames, _
        WEBOM_DOCTYPE_WEBFORM, WEBOM_DOCFILETYPE_FORMDEF, FALSE
Next
End Sub

```

DoesDocumentExist

For internal use.

EnumDocuments

Enumerates the names of the documents that match the specified criteria, as well as the documents' descriptions, timestamps, and security classes.

This information is returned in arrays that have a one-to-one correspondence with each other.

Tip: [EnumDocumentsEx](#) is a similar method that returns additional information such as the documents' owners and document types.

Syntax

```

<HFMMwManageDocuments>.EnumDocuments (bstrPath, lDocumentType,
lDocumentFileType, vbFilterByCurrentUserOnly, vbFilterByCreateTime,
dStartTime, dStopTime, pvarabstrDescriptions, pvaradTimestamp,
pvarabstrSecurityClass)

```

Argument	Description
<i>bstrPath</i>	The document repository folder that contains the documents. Input argument. String subtype.
<i>lDocumentType</i>	The document type of the documents to be enumerated. Document types are represented by the HFMMConstants type library constants listed in “Document Type Constants” on page 451 . Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the documents to be enumerated. File types are represented by the HFMMConstants type library constants listed in “Document File Type Constants” on page 452 . Input argument. Long subtype.

Argument	Description
<i>vbFilterByCurrentUserOnly</i>	Specifies whether to enumerate only those documents that were created by the current user. Pass TRUE to enumerate only the current user's documents, FALSE otherwise. Input argument.
<i>vbFilterByCreateTime</i>	Specifies whether information will be returned for only those documents with timestamps in a specified range. Pass TRUE to filter by timestamp, FALSE otherwise. If you pass TRUE, information will be returned for only those documents with timestamps that fall into the range specified by the <i>dStartTime</i> and <i>dStopTime</i> arguments. Note: If you pass FALSE, the <i>dStartTime</i> and <i>dStopTime</i> arguments will be ignored, but will still require a value. Input argument.
<i>dStartTime</i>	The beginning of the timestamp filtering range. Tip: To make the beginning of the range open-ended, pass 0 (zero). Input argument. Double subtype.
<i>dStopTime</i>	The end of the timestamp filtering range. Tip: To make the end of the range open-ended, pass 0 (zero). Input argument. Double subtype.
<i>pvarabstrDescriptions</i>	Returns an array of the documents' descriptions. Output argument.
<i>pvaradTimestamp</i>	Returns an array of the documents' timestamps. Output argument.
<i>pvarabstrSecurityClass</i>	Returns an array of the documents' security classes. Output argument.

Return Value

Returns an array of the documents' names.

Example

`EnumDocuments` is used in the example for [DeleteDocuments](#).

EnumDocumentsEx

Enumerates the names of the documents that match the specified criteria, as well as the documents' descriptions, timestamps, security classes, privacy flags, content types, owners, file types, and document types. You can also filter for public or private documents.

This information is returned in arrays that have a one-to-one correspondence with each other.

Tip: [EnumDocuments](#) is a similar method that enumerates only the documents' names, descriptions, timestamps, and security classes.

Syntax

```
<HFMwManageDocuments>.EnumDocumentsEx (bstrPath, varalDocumentType,  
varalDocumentFileType, vbFilterByCreateTime, dStartTime, dStopTime,  
lShowPrivateDocs, pvarabstrDescriptions, pvaradTimestamp,  
pvarabstrSecurityClass, pvarabIsPrivate, pvaralFolderContentType,  
pvarabstrDocOwner, pvaralFileType, pvaralReportType)
```

Argument	Description
<i>bstrPath</i>	The document repository folder that contains the documents. Input argument. String subtype.
<i>varalDocumentType</i>	The document type of the documents to be enumerated. Document types are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Input argument.
<i>varalDocumentFileType</i>	The file type of the documents to be enumerated. File types are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 452 . Input argument.
<i>vbFilterByCreateTime</i>	Specifies whether information will be returned for only those documents with timestamps in a specified range. Pass TRUE to filter by timestamp, FALSE otherwise. If you pass TRUE, information will be returned for only those documents with timestamps that fall into the range specified by the <i>dStartTime</i> and <i>dStopTime</i> arguments. Note: If you pass FALSE, the <i>dStartTime</i> and <i>dStopTime</i> arguments will be ignored, but will still require a value. Input argument. Boolean subtype.
<i>dStartTime</i>	The beginning of the timestamp filtering range. Tip: To make the beginning of the range open-ended, pass 0 (zero). Input argument. Double subtype.
<i>dStopTime</i>	The end of the timestamp filtering range. Tip: To make the end of the range open-ended, pass 0 (zero). Input argument. Double subtype.
<i>lShowPrivateDocs</i>	Specifies whether to return public documents, private documents, or both. Valid values are represented by the HFMConstants type library constants listed in “Showing Public and Private Documents” on page 464 . Input argument. Long subtype.
<i>pvarabstrDescriptions</i>	Returns an array of the documents’ descriptions. Input/output argument.
<i>pvaradTimestamp</i>	Returns an array of the documents’ timestamps. Input/output argument.
<i>pvarabstrSecurityClass</i>	Returns an array of the documents’ security classes.

Argument	Description
	Input/output argument.
<i>pvarablsPrivate</i>	Returns an array indicating whether the documents are public or private. TRUE indicates private, FALSE indicates public. Input/output argument.
<i>pvalFolderContentType</i>	Returns an array indicating the types of documents that folders can contain. This value has meaning only for folders; ignore array items that correspond to document types other than folders. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Input/output argument.
<i>pvarabstrDocOwner</i>	Returns an array containing the usernames of the documents’ owners. Input/output argument.
<i>pvalFileType</i>	Returns an array of the documents’ file types. File types are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 452 . Input/output argument.
<i>pvalReportType</i>	Returns an array of the documents’ document types. Document types are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Input/output argument.

Return Value

Returns an array of the documents’ names.

Example

The following function takes a folder path and returns an HTML table that lists the folder’s public data forms and the usernames of the forms’ owners.

```
Function GetDocAndFolderTypes(sPath)
Dim cHFMDocuments, vaDocNames, vaDescs, vaTime, vaSecClass, sRet
Dim vaPrivate, vaContent, vaOwners, vaFileTypes, vaReportTypes
Set cHFMDocuments = _
    Server.CreateObject("Hyperion.HFMwManageDocuments")
' cHFMSession is a previously set HFMwSession object
cHFMDocuments.SetWebSession cHFMSession
vaDocNames = cHFMDocuments.EnumDocumentsEx(sPath, _
    WEBOM_DOCTYPE_ALL, WEBOM_DOCFILETYPE_ALL, _
    FALSE, 0, 0, ENUMSHOWPRIVATEDOCS_DONTSHOW, vaDescs, _
    vaTime, vaSecClass, vaPrivate, vaContent, vaOwners, _
    vaFileTypes, vaReportTypes)
sRet = sRet & "<table cellpadding=2>"
sRet = sRet & "<tr><td><b>Form</b></td>" & _
    "<td><b>Doctypes</b></td>" & _
    "<td><b>file types</b></td></tr>"
For i = lBound(vaDocNames) to uBound(vaDocNames)
    sRet = sRet + "<tr><td>" & vaDocNames(i) & "</td>"
    sRet = sRet + "<td>" & vaReportTypes(i) & "</td>"
    sRet = sRet + "<td>" & vaFileTypes(i) & "</td></tr>"
```



```

Next
sRet = sRet + "</table>"
GetDocAndFolderTypes = sRet
End Function

```

GetDocument

Returns a string containing the content of the specified document, as well as strings containing the document's description and security class.

Tip: [SaveDocument](#) and [SaveDocumentEx](#) create a document from a string.

Syntax

```

<HFMwManageDocuments>.GetDocument (bstrPath, bstrName, lDocumentType,
lDocumentFileType, pvarbstrDescription, pvarbstrSecurityClass)

```

Argument	Description
<i>bstrPath</i>	The document repository folder that contains the document. Input argument. String subtype.
<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>lDocumentType</i>	The document type of the document. Document types are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the document. File types are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.
<i>pvarbstrDescription</i>	Returns the description of the document. Output argument.
<i>pvarbstrSecurityClass</i>	Returns the label of the document's security class. Output argument.

Return Value

Returns a string that contains the content of the document.

Example

The following function saves the content of the specified data form to disk.

```

Function saveWebFormToDisk(sDocFolder, sFormName)
Dim cHFMDocuments, sText, sDesc, sSecClass, oFs, oTs
Set cHFMDocuments = _
Server.CreateObject("Hyperion.HFMwManageDocuments")

```

```

cHFMDocuments.SetWebSession cHFMSession
sText = cHFMDocuments.GetDocument(sDocFolder, sFormName, _
WEBOM_DOCTYPE_WEBFORM, WEBOM_DOCFILETYPE_FORMDEF, sDesc, _
sSecClass)
' Write the string to disk as a text file
set oFs = CreateObject("Scripting.FileSystemObject")
set oTs = oFs.CreateTextFile("c:\temp\" & sFormName & ".txt")
oTs.Write(sText)
oTs.Close
End Function

```

GetDocumentEx

Returns the definition of a document, as well as other properties such as the document's type, file type, and security class.

Syntax

```

<HFMMwManageDocuments>.GetDocumentEx (bstrPath, bstrName, lDocumentType,
lDocumentFileType, pvarbstrDescription, pvarbstrSecurityClass,
pvarbPrivate, pvbstrOwner, pvdTimestamp, pvlFolderContentType)

```

Argument	Description
<i>bstrPath</i>	The path to the document. Input argument. String subtype.
<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>lDocumentType</i>	Identifies the type of document to return. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Input argument. Long subtype.
<i>lDocumentFileType</i>	Identifies the file type of the documents to return. Valid values are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.
<i>pvarbstrDescription</i>	Returns the description of the document. Input/output argument.
<i>pvarbstrSecurityClass</i>	Returns the ID of the document's security class. Input/output argument.
<i>pvarbPrivate</i>	Indicates whether the document is public or private. TRUE indicates private, FALSE indicates public. Input/output argument.
<i>pvbstrOwner</i>	Returns the username of the document's owner. Input/output argument.

Argument	Description
<i>pvdTimestamp</i>	Returns the document's timestamp. The timestamp can be converted to a Date format; for example, in VBScript you can convert with <code>CDate</code> . Input/output argument.
<i>pvlFolderContentType</i>	Returns the type of document that a folder document can contain. This value has meaning only if the document is a folder; ignore this value for non-folder documents. Valid values are represented by the <code>HFMConstants</code> type library constants listed in "Document Type Constants" on page 451 . Input/output argument.

Return Value

Returns the document's definition.

LoadDocuments

Loads one or more documents from disk into the specified document repository folder, applying the same security class to all of the documents.

Tip: To apply different security classes to loaded documents or flag them as private, use [LoadDocumentsEx](#).

Syntax

```
<HFMwManageDocuments>.LoadDocuments bstrDestinationPath, lDocumentType,
lDocumentFileType, bstrSecurityClass, varabstrFilePaths, bstrLogFilePath,
vbIncludeSubFolders, vbOverwriteExisting
```

Argument	Description
<i>bstrDestinationPath</i>	The document repository folder that will contain the documents. Precede the folder name with a backslash (\). Input argument. String subtype.
<i>lDocumentType</i>	The document type of the documents. Document types are represented by the <code>HFMConstants</code> type library constants listed in "Document Type Constants" on page 451 . Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the documents. File types are represented by the <code>HFMConstants</code> type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.
<i>bstrSecurityClass</i>	The security class for the documents. Input argument. String subtype.
<i>varabstrFilePaths</i>	An array of strings that contain the names and file system paths of the documents to be loaded. Input argument.

Argument	Description
<i>bstrLogFilePath</i>	The name and path of the log file for the load operation. You must specify a name and path, otherwise an error occurs. Input argument. String subtype.
<i>vbIncludeSubFolders</i>	<i>Future use.</i> You must pass a valid Boolean, but the value is ignored by this method. Input argument.
<i>vbOverwriteExisting</i>	Indicates whether existing documents of the same name should be overwritten. Pass TRUE to overwrite, FALSE otherwise. Input argument.

Example

The following example loads three data forms into a document repository folder named Sales.

```
Dim cHFMDocuments, saDocs(2)
Set cHFMDocuments = _
Server.CreateObject("Hyperion.HFMwManageDocuments")
' cHFMSession is a previously set HFMwSession object
cHFMDocuments.SetWebSession cHFMSession
' Populate an array with the form names and paths
saDocs(0) = "C:\Acme\WebForms\Demol.wdf"
saDocs(1) = "C:\Acme\WebForms\DemoList1.wdf"
saDocs(2) = "C:\Acme\WebForms\DemoList2.wdf"
cHFMDocuments.LoadDocuments "\Sales", WEBOM_DOCTYPE_WEBFORM, _
WEBOM_DOCFILETYPE_FORMDEF, "[Default]", saDocs, _
"C:\Acme\WebForms\webformload.log", FALSE, TRUE
```

LoadDocumentsEx

Loads one or more documents from disk into the specified document repository folder, applying a given security class and a privacy flag to each document. You can also apply labels to documents.

The file names and paths, document labels, privacy flags, and security classes are passed as arrays that have a one-to-one correspondence.

Tip: To apply the same security classes to all documents, or to omit the private document flag, use [LoadDocuments](#).

Syntax

```
<HFMwManageDocuments>.LoadDocumentsEx bstrDestinationPath, lDocumentType,
lDocumentFileType, varabstrNames, varabstrSecurityClass, varabPrivate,
varabstrFilePaths, bstrLogFilePath, vbIncludeSubFolders,
vbOverwriteExisting, varabOverwriteSecurityClass
```

Argument	Description
<i>bstrDestinationPath</i>	<p>The document repository folder that will contain the documents. Precede the folder name with a backslash (\).</p> <p>Input argument. String subtype.</p>
<i>IDocumentType</i>	<p>The document type of the documents. Document types are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451.</p> <p>Input argument. Long subtype.</p>
<i>IDocumentFileType</i>	<p>The file type of the documents. File types are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 452.</p> <p>Input argument. Long subtype.</p>
<i>varabstrNames</i>	<p>The labels to be applied to the documents.</p> <p>Some types of documents, such as reports and data forms, contain default labels. If you pass a label in the array item for the document, that will override the default label; if you pass a blank string, the default label is not overridden.</p> <p>Input argument.</p>
<i>varabstrSecurityClass</i>	<p>An array of the security classes for the documents.</p> <p>Input argument.</p>
<i>varabPrivate</i>	<p>An array of flags indicating whether the corresponding documents are public or private. Pass TRUE for private, FALSE for public.</p> <p>Input argument.</p>
<i>varabstrFilePaths</i>	<p>An array containing the names and paths of the files to be loaded.</p> <p>Input argument.</p>
<i>bstrLogFilePath</i>	<p>The name and path of the log file for the load operation. You must specify a name and path, otherwise an error occurs.</p> <p>Input argument. String subtype.</p>
<i>vbIncludeSubFolders</i>	<p><i>Future use.</i> You must pass a valid Boolean, but the value is ignored by this method.</p> <p>Input argument. Boolean subtype.</p>
<i>vbOverwriteExisting</i>	<p>A flag that specifies whether existing documents of the same name should be overwritten. Pass TRUE to overwrite, FALSE otherwise.</p> <p>Input argument. Boolean subtype.</p>
<i>varabOverwriteSecurityClass</i>	<p>An array of flags that specify whether to overwrite the existing security classes of the corresponding documents. Pass TRUE to overwrite, FALSE otherwise.</p> <p>Input argument.</p>

SaveDocument

Saves a document in the document repository, given a string containing the content of the document to be created.

Tip: [SaveDocumentEx](#) also saves a document, and provides additional options to specify the document's content type and privacy flag. To save documents using an array of bytes, or to save custom binary documents, use [SaveDocument2](#).

Syntax

```
<HFMwManageDocuments>.SaveDocument bstrPath, bstrName, bstrDescription,  
lDocumentType, lDocumentFileType, bstrSecurityClass, bstrDocument,  
vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	The document repository folder that will contain the document. Input argument. String subtype.
<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>bstrDescription</i>	The description of the document. Input argument. String subtype.
<i>lDocumentType</i>	The document type of the document. Document types are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Do not pass WEBOM_DOCTYPE_CUSTOM to this method. To save custom documents, use SaveDocument2 . Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the document. File types are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.
<i>bstrSecurityClass</i>	The name of the security class for the document. Input argument. String subtype.
<i>bstrDocument</i>	The content to be saved in the document. Tip: GetDocument returns a string that contains the content of an existing document. The HFMwQueryDef method GetReportDefinition returns a string that can be used to create a journal report. Input argument. String subtype.
<i>vbOverwriteExisting</i>	Indicates whether an existing document of the same name should be overwritten. Pass TRUE to overwrite, FALSE otherwise. Input argument.

SaveDocument2

Saves a document in the document repository and specifies the document's content type and privacy flag, given an array of bytes that defines the content of the document to be created. This method can also save a custom binary document.

Syntax

```
<HFMwManageDocuments>.SaveDocument2 bstrPath, bstrName, bstrDescription,  
lDocumentType, lDocumentFileType, bstrSecurityClass, varabyContents,  
vbIsPrivate, lContentType, vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	The document repository folder that will contain the document. Input argument. String subtype.
<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>bstrDescription</i>	The description of the document. Input argument. String subtype.
<i>lDocumentType</i>	The document type of the document. Document types are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . If the document is a data form, this method also validates the document's definition. Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the document. File types are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 452 . Input argument. Long subtype.
<i>bstrSecurityClass</i>	The name of the security class for the document. Input argument. String subtype.
<i>varabyContents</i>	The document's contents as an array of bytes. Anything other than an array of bytes will throw an error. Input argument.
<i>vbIsPrivate</i>	Specifies whether the folder is public or private. Pass TRUE for private, FALSE for public. Input argument. Boolean subtype.
<i>lContentType</i>	The type of documents that the folder can store, if you are saving a folder. If are saving a document type other than a folder, you can pass any Long, as <i>SaveDocumentEx</i> will ignore this argument's value in that case. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 451 . Input argument. Long subtype.
<i>vbOverwriteExisting</i>	Indicates whether an existing document of the same name should be overwritten. Pass TRUE to overwrite, FALSE otherwise. Input argument. Boolean subtype.

SaveDocumentEx

Saves a document in the document repository and specifies the document's content type and privacy flag, given a string containing the content of the document to be created.

Tip: To save a document without specifying its content type and privacy flag, use [SaveDocument](#). To save documents using an array of bytes, or to save custom binary documents, use [SaveDocument2](#).

Syntax

```
<HFMwManageDocuments>.SaveDocumentEx bstrPath, bstrName, bstrDescription,  
lDocumentType, lDocumentFileType, bstrSecurityClass, bstrDocument,  
vbIsPrivate, lContentType, vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	The document repository folder that will contain the document. Input argument. String subtype.
<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>bstrDescription</i>	The description of the document. Input argument. String subtype.
<i>lDocumentType</i>	The document type of the document. Document types are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Do not pass WEBOM_DOCTYPE_CUSTOM to this method. To save custom documents, use SaveDocument2 . Input argument. Long subtype.
<i>lDocumentFileType</i>	The file type of the document. File types are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.
<i>bstrSecurityClass</i>	The name of the security class for the document. Input argument. String subtype.
<i>bstrDocument</i>	The content to be saved in the document. Tip: GetDocument returns a string that contains the content of an existing document. The HFMwQueryDef method GetReportDefinition returns a string that can be used to create a journal report. Input argument. String subtype.
<i>vbIsPrivate</i>	Specifies whether the folder is public or private. Pass TRUE for private, FALSE for public. Input argument. Boolean subtype.
<i>lContentType</i>	The type of documents that the folder can store, if you are saving a folder. If are saving a document type other than a folder, you can pass any Long, as SaveDocumentEx will ignore this argument's value in that case. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Input argument. Long subtype.

Argument	Description
<i>vbOverwriteExisting</i>	Indicates whether an existing document of the same name should be overwritten. Pass TRUE to overwrite, FALSE otherwise. Input argument. Boolean subtype.

SetWebSession

Associates an HFMwManageDocuments object with the HFMwSession object for an application.

Syntax

```
<HFMwManageDocuments>.SetWebSession varpIUnkHFMwSession
```

Argument	Description
<i>varpIUnkHFMwSession</i>	The HFMwSession object for the application. Input argument.

Example

SetWebSession is used in the example for [DeleteDocuments](#).

HFMwWorkspace Component

The HFMwWorkspace component exposes Financial Management's task list feature. For example, the following functionality is exposed:

- Add documents to task lists
- Delete documents from task lists.
- Get existing task list documents.
- Test whether a task list contains a given document.

Task lists themselves are documents stored in the document repository, meaning that the HFMwManageDocuments component applies to task lists. For example, to get an existing task list you use [GetDocument](#), and to save a task list you use [SaveDocumentEx](#).

HFMwWorkspace members are described in [“HFMwWorkspace Methods” on page 282](#) and [“HFMwWorkspace Properties” on page 288](#).

Obtaining an HFMwWorkspace Object Reference

Create an HFMwWorkspace object with `Server.CreateObject`. After creating the object, call [SetWebSession](#) to associate the object with the HFMwSession object for the application with which you are working:

```
chFMDocuments.SetWebSession chFMSession
Set chFMWs = Server.CreateObject("Hyperion.HFMwWorkspace")
```

```
' cHFMSession is a previously set HFMwSession object
cHFMWs.SetWebSession cHFMSession
```

Initializing an HFMwWorkspace Instance

After obtaining an HFMwWorkspace object reference, you must initialize the object by passing the XML string that represents the task list to either the [ParseWorkspaceDocument](#) method or the `xml` property. You get the task list's XML string from one of the following methods:

- If you are creating a new task list, use the string returned by [GetBlankWorkspaceDocument](#).
- If you are working with an existing task list, use the string returned by the HFMwManageDocuments method [GetDocument](#).

HFMwWorkspace Methods

The HFMwWorkspace component contains the methods described in the following topics.

AddDocument

Adds a document to the task list represented by the HFMwWorkspace instance.

Syntax

```
<HFMwWorkspace>.AddDocument bstrDocName, bstrDocPath, lDocType,
lDocFileType
```

Argument	Description
----------	-------------

<i>bstrDocName</i>	The name of the document. Input argument. String subtype.
--------------------	--

<i>bstrDocPath</i>	The path of the document. Input argument. String subtype.
--------------------	--

<i>lDocType</i>	The document type of the document. Valid values are represented by the HFMwConstants type library constants listed in “Document Type Constants” on page 451 , Input argument. Long subtype.
-----------------	--

<i>lDocFileType</i>	The file type of the document. Valid values are represented by the HFMwConstants type library constants listed in “Document File Type Constants” on page 452 , Input argument. Long subtype.
---------------------	---

AddTask

Adds a task to the task list represented by the HFMwWorkspace instance. You specify the task's name and the URL of the file that contains the task's functionality.

Syntax

```
<HFMwWorkspace>.AddTask (bstrName, bstrLink)
```

Argument Description

bstrName The name of the task.

Input argument. String subtype.

bstrLink The URL of the file for the task. For example, this could be the URL of an .ASP file or a Web page.

Input argument. String subtype.

Return Value

Indicates whether the task was successfully added to the task list. TRUE indicates success, FALSE failure. Boolean subtype.

EnumDocuments

Enumerates the names of the documents in the task list, as well as the documents' file types and document types.

This information is returned in arrays that have a one-to-one correspondence with each other.

Syntax

```
<HFMwWorkspace>.EnumDocuments (pvarastrPaths, pvarbstrTypes,  
pvarbstrFileTypes)
```

Argument Description

pvarastrPaths Returns an array of the documents' paths.

Input/output argument.

pvarbstrTypes Returns an array of the documents' document types. Valid values are represented by the HFMwConstants type library constants listed in ["Document Type Constants" on page 451](#).

Input/output argument.

pvarbstrFileTypes Returns an array of the documents' file types. Valid values are represented by the HFMwConstants type library constants listed in ["Document File Type Constants" on page 452](#).

Input/output argument.

Return Value

Returns an array of the documents' names.

GetBlankWorkspaceDocument

Sets the HFMwWorkspace instance to a blank task list with a given name, path, security class, and description, and returns an XML string that represents the task list. For further processing, you can pass this string to [ParseWorkspaceDocument](#).

Syntax

```
<HFMWorkspace>.GetBlankWorkspaceDocument (bstrPath, bstrName,  
bstrSecClass, bstrDesc)
```

Argument	Description
----------	-------------

<i>bstrPath</i>	The path in which to create the task list. Input argument. String subtype.
<i>bstrName</i>	The name of the task list. Input argument. String subtype.
<i>bstrSecClass</i>	The label of the task list's security class. Input argument. String subtype.
<i>bstrDesc</i>	The task list's description. Input argument. String subtype.

Return Value

Returns an XML string that represents the task list. String subtype.

GetDocumentAt

Returns a string containing the content of the document at the specified index, as well as the document's path, document type, and file type.

Syntax

```
<HFMWorkspace>.GetDocumentAt (lIndex, pvarbstrDocPath, plDocType,  
plDocFileType)
```

Argument	Description
----------	-------------

<i>lIndex</i>	The index of the document. Each document in a task list is indexed. Input argument. Long subtype.
<i>pvarbstrDocPath</i>	Returns the document's path. Input/output argument.
<i>plDocType</i>	Returns the document's document type. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Input/output argument. Long subtype.
<i>plDocFileType</i>	Returns the document's file type. Valid values are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input/output argument. Long subtype.

Return Value

Returns a string containing the document's content.

GetTaskNameAndIdByUrl

For internal use.

GetTaskNameAndUrlById

Returns the name of a task, given the task's ID.

Syntax

```
<HFMwWorkspace>.GetTaskNameAndUrlById idTask, pvarbstrName, pvarbstrUrl
```

Argument	Description
----------	-------------

<i>idTask</i>	The ID of the task. Valid values are represented by the HFMConstants enumeration described in “Task ID Constants” on page 462 .
---------------	---

Input argument. Long subtype.

<i>pvarbstrName</i>	Returns the name of the task.
---------------------	-------------------------------

Input/output argument.

<i>pvarbstrUrl</i>	<i>For internal use.</i>
--------------------	--------------------------

Input/output argument.

GetWorkspaceDocument

Returns an XML string that represents the task list. You can pass the return value to methods that require the task list's XML string. For example, you can pass this to `HFMwDocuments.SaveDocumentEx`.

Syntax

```
<HFMwWorkspace>.GetWorkspaceDocument ( )
```

Return Value

Returns an XML string that represents the task list.

Note: You do not need to directly manipulate the XML string to change the task list's properties, because these properties are exposed. For information on the task list properties, see [“HFMwWorkspace Properties” on page 288](#).

IsDocumentExists

Indicates whether the task list represented by the HFMwWorkspace instance contains the specified document.

Syntax

```
<HFMwWorkspace>.IsDocumentExists (bstrName, bstrPath, lDocType, lFileType)
```

Argument Description

<i>bstrName</i>	The name of the document. Input argument. String subtype.
<i>bstrPath</i>	The path of the document. Input argument. String subtype.
<i>IDocType</i>	The document's document type. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 . Input argument. Long subtype.
<i>IFileType</i>	The document's file type. Valid values are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 . Input argument. Long subtype.

Return Value

Indicates whether the task list contains the document. Returns TRUE if the task list contains the document, FALSE otherwise. Boolean subtype.

ParseWorkspaceDocument

Initializes the HFMwWorkspace instance to the task list represented by an XML string.

Note: Calling this method serves the same purpose as setting the [xml](#) property.

Syntax

```
<HFMwWorkspace>.ParseWorkspaceDocument bstrXml
```

Argument Description

<i>bstrXml</i>	The XML string that represents the task list. Get this string from methods that return XML representations of task lists. For example, call GetBlankWorkspaceDocument to get a string that represents a blank task list, or HFMwManageDocuments.GetDocument to get a string that represents an existing task list. Input argument. String subtype.
----------------	--

RemoveDocument

Removes the specified document from the task list, given the document's name, path, document type, and file type.

Tip: To remove a document by its index, use [RemoveDocumentAt](#).

Syntax

```
<HFMWorkspace>.RemoveDocument pbstrName, pbstrPath, plType, plFileType
```

Argument	Description
----------	-------------

<i>pbstrName</i>	The name of the document.
------------------	---------------------------

Input argument. String subtype.

<i>pbstrPath</i>	The path of the document.
------------------	---------------------------

Input argument. String subtype.

<i>plType</i>	The document's document type. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 451 .
---------------	---

Input argument. Long subtype.

<i>plFileType</i>	The document's file type. Valid values are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 452 .
-------------------	--

Input argument. Long subtype.

RemoveDocumentAt

Removes the document at the specified index.

Tip: To remove a document by its name, path, document type, and file type, use [RemoveDocument](#).

Syntax

```
<HFMWorkspace>.RemoveDocumentAt lIndex
```

Argument	Description
----------	-------------

<i>lIndex</i>	The index of the document to be removed. Each document in a task list is indexed.
---------------	---

Tip: You can derive the index from the [Count](#) property.

Input argument. Long subtype.

SetWebSession

Associates an HFMWorkspace object with the HFMwSession object for an application.

Syntax

```
<HFMWorkspace>.SetWebSession varpIUnkHFMwSession
```

Argument	Description
----------	-------------

<i>varpIUnkHFMwSession</i>	The HFMwSession object for the application.
----------------------------	---

Argument	Description
	Input argument.

SwapDocPosition

Moves a document from one position to another within the list of the task list's documents.

Syntax

```
<HFMWorkspace>.SwapDocPosition (lStart, lEnd)
```

Argument Description

<i>lStart</i>	The index of the document to be moved. Input argument. Long subtype.
<i>lEnd</i>	The index of the position to which the document is being moved. Input argument. Long subtype.

Return Value

Indicates whether the document was successfully moved. TRUE indicates success, FALSE indicates failure. Boolean subtype.

HFMWorkspace Properties

The HFMWorkspace component contains the properties described in the following topics.

Count

Returns a count of the task list's documents.

Read-only.

Description

Gets or sets the task list's description.

Read-write.

FileType

Returns the file type of the task list object. A task list's file type is represented by the `WEBOM_DOCFILETYPE_XML` constant, which is a member of the HFMwConstants type library enumeration described in [“Document File Type Constants” on page 452](#).

Read-only.

Height

For internal use.

Read-write.

Name

Gets or sets the task list's name.

Read-write.

Path

Gets or sets the task list's path.

Read-write.

Private

Gets or sets the task list's privacy flag. TRUE indicates that the task list is private, FALSE otherwise.

Read-write.

SecClass

Gets or sets the task list's security class, using the security class name.

Read-write.

ShowNames

Gets or sets whether the names of the task list's documents are displayed in the user interface. TRUE indicates that the document names are displayed, FALSE otherwise.

Read-write.

SubWorkspace

Gets or sets whether the task list can be added to other task lists. TRUE indicates that the task list can be added, FALSE that it cannot.

Read-write.

TaskFlow

For internal use.

Read-write.

Type

Returns the document type of the task list object. A task list's document type is represented by the `WEBOM_DOCTYPE_WORKSPACE` constant, which is a member of the `HFmwConstants` type library enumeration described in [“Document Type Constants” on page 451](#).

Read-only.

Width

For internal use.

Read-write.

xml

Gets or sets the XML string to which the `HFmwWorkspace` instance is initialized.

Note: Setting this property performs the same purpose as calling [ParseWorkspaceDocument](#).

Read-write.

The HF MwSecurity type library contains the HF MwSecurity component. This component provides methods that enumerate an application's security classes, indicate whether a user has rights to perform a given task, and return other types of security information.

Obtaining an HF MwSecurity Object Reference

Create an HF MwSecurity object with `Server.CreateObject`. After creating the object, call [SetWebSession](#) to associate the object with the HF MwSession object for the application for which you want to obtain security information:

```
Set cHF MwSecurity = Server.CreateObject("Hyperion.HF MwSecurity")
' cHF MwSession is a previously set HF MwSession object
cHF MwSecurity.SetWebSession cHF MwSession
```

HF MwSecurity Component Methods

The HF MwSecurity component contains the methods described in the following topics.

AddSecurityClasses

Adds the specified security classes.

Syntax

```
<HF MwSecurity>.AddSecurityClasses (bstrClassesXml)
```

Argument	Description
<i>bstrClassesXml</i>	An XML string that lists the classes to add. Following is the structure of the string, which must contain one <Class> element per security class: <pre><Classes><Class><Name>security class name</Name></Class></Classes></pre> Input argument. String subtype.

Return Value

Returns an XML string containing the IDs assigned to the security classes. Following is the structure of the string. There is one <Class> element per security class:

```
<Classes><Class><Name>security class name</Name><Id>security class ID</Id></Class></Classes>
```

AllowRulesLoadForEPMAApp

For internal use.

DeleteSecurityClasses

Deletes the specified security classes.

Syntax

```
<HFMwSecurity>.DeleteSecurityClasses bstrClassesXml
```

Argument	Description
----------	-------------

<i>bstrClassesXml</i>	An XML string that lists the classes to delete. Following is the structure of the string, which must contain one <Class> element per security class:
-----------------------	--

```
<Classes><Class><Name>security class name</Name></Class></Classes>
```

Input argument. String subtype.

DoesConnectedUserHaveApplicationAdminRights

Indicates whether the connected user has application administrator rights.

Note: To have application administrator rights, a user must be either the application owner or assigned to the Application Administrator role.

Syntax

```
<HFMwSecurity>.DoesConnectedUserHaveApplicationAdminRights()
```

Return Value

Returns TRUE if the user has application administrator rights, FALSE otherwise.

EnumRoles

Returns the names of all Financial Management security roles.

Syntax

```
<HFMwSecurity>.EnumRoles pvarabstrRoleNames
```

Argument	Description
----------	-------------

<i>pvarabstrRoleNames</i>	Returns an array of the role names.
---------------------------	-------------------------------------

Input/output argument.

Example

The following example prints the role names to the browser.

```
Sub printSecurityRoles()  
Dim cHFMSecurity, vaRoleNames  
Set cHFMSecurity = Server.CreateObject("Hyperion.HFMwSecurity")  
' cHFMSession is a previously set HFMwSession object  
cHFMSecurity.SetWebSession cHFMSession  
cHFMSecurity.EnumRoles vaRoleNames  
For i = 0 to uBound(vaRoleNames)  
    Response.Write "<p>" & vaRoleNames(i) & "</p>"  
Next  
End Sub
```

EnumSecurityClasses

Returns the labels and internal IDs of all the security classes in the current Financial Management application. The arrays have a one-to-one correspondence.

Syntax

```
<HFMwSecurity>.EnumSecurityClasses pvaralIDs, pvarabstrSecurityClasses
```

Argument	Description
----------	-------------

<i>pvaralIDs</i>	Returns an array of security class IDs.
------------------	---

Output argument.

<i>pvarabstrSecurityClasses</i>	Returns an array of security class labels.
---------------------------------	--

Tip: The array items are returned in alphabetical order, with [Default] always the first item.

Output argument.

Example

The following example prints the labels of an application's security classes to the browser.

```
Sub printSecurityNames()  
Dim cHFMSecurity, vaIDs, vaClasses  
Set cHFMSecurity = Server.CreateObject("Hyperion.HFMwSecurity")  
'g_cHFMSession is a previously set HFMwSession object  
cHFMSecurity.SetWebSession g_cHFMSession  
cHFMSecurity.EnumSecurityClasses vaIDs, vaClasses  
For i = 0 to uBound(vaClasses)  
    Response.Write "<p>" & vaClasses(i) & "</p>"  
Next
```

End Sub

EnumSecurityClassesForConnectedUser

Returns the labels and internal IDs of all the security classes to which the connected user has a given level of access rights. The arrays have a one-to-one correspondence.

Syntax

```
<HFMwSecurity>.EnumSecurityClassesForConnectedUser lRequiredAccessRights,  
pvaralIDs, pvarabstrSecurityClasses
```

Argument	Description
<i>lRequiredAccessRights</i>	The level of access rights. Valid values are represented by the HFMConstants type library constants listed in “Access Rights Constants” on page 438 . Input argument. Long subtype.
<i>pvaralIDs</i>	Returns an array containing the IDs of the security classes to which the user has the specified level of access rights. Input/output argument.
<i>pvarabstrSecurityClasses</i>	Returns an array containing the labels of the security classes to which the user has the specified level of access rights. Input/output argument.

EnumSecurityClassRights

Returns an XML string containing the access and e-mail alerting rights for the specified users to the specified security classes.

Syntax

```
<HFMwSecurity>.EnumSecurityClassRights (bstrUsersXml, bstrClassesXml)
```

Argument	Description
<i>bstrUsersXml</i>	An XML string that specifies the users' usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user: <pre><Users><User><Name>username</Name><Id>security identifier</Id></User></Users></pre> Input argument. String subtype.
<i>bstrClassesXml</i>	An XML string that specifies the names and IDs of the security classes. Following is the structure of the string, which must contain one <Class> element per security class: <pre><Classes><Class><Name>security class name</Name><Id>security class Id</Id></Class></Classes></pre> Input argument. String subtype.

Return Value

Returns an XML string that specifies the users' access and e-mail alerting rights to the security classes. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Class> element per security class. Access rights IDs are represented by the HFMConstants enumeration described in [“Access Rights Constants” on page 438](#):

```
<UserRights><User name="username" id="security identifier" ><Class
name="security class name" id="security class ID" alert="Boolean"
rights="access rights ID" /></User></UserRights>
```

EnumSecurityClassRightsForBiPub

Generates a report in specified format (PDF, RTF, HTML, or XLS) and returns an XML string containing the access and e-mail alerting rights for the specified users to the specified security classes.

Syntax

```
<HFMwSecurity>.EnumSecurityClassRightsForBiPub(bstrUsersXml,
bstrClassesXml, lReportFormat, bstrReportTemplateFileName, ReportFileOut)
```

Argument	Description
<i>bstrUsersXml</i>	<p>An XML string that specifies the users' usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user:</p> <pre><Users><User><Name>username</Name><Id>security identifier</Id></User></Users></pre> <p>Input argument. String subtype.</p>
<i>bstrClassesXml</i>	<p>An XML string that specifies the names and IDs of the security classes. Following is the structure of the string, which must contain one <Class> element per security class:</p> <pre><Classes><Class><Name>security class name</ Name><Id>security class Id</Id></Class></Classes></pre> <p>Input argument. String subtype.</p>
<i>lReportFormat</i>	<p>Value that specifies the report output format:</p> <ul style="list-style-type: none">● 1 = PDF format● 2 = RTF format● 3 = HTML format● 4 = XLS format● -1 = Financial Management format <p>Input argument. Long subtype.</p>
<i>bstrReportTemplateFileName</i>	<p>The BI Publisher report template name that is used to generate the report.</p> <p>Input argument. String subtype.</p>
<i>ReportFileOut</i>	<p>Specifies the file name of the report generated.</p>

Argument	Description
	Input argument. String subtype.

Return Value

Returns an XML string that specifies the users' access and e-mail alerting rights to the security classes. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Class> element per security class. Access rights IDs are represented by the HFMConstants enumeration described in [“Access Rights Constants” on page 438](#):

```
<UserRights><User name="username" id="security identifier" ><Classes
NODISPLAY='Yes/No'><Class name="security class name" id="security class ID"
alert="Boolean" rights="access rights ID" /></Classes><Roles
NODISPLAY='Yes/No'></Roles></User></UserRights>
```

String subtype.

EnumSecurityClassRightsAndRoles

Returns a localized XML string that lists the specified users' access rights to the specified security classes and the roles to which the users are assigned.

Syntax

```
<HFMwSecurity>.EnumSecurityClassRightsAndRoles (bstrUsersXml,
bstrClassesXml)
```

Argument	Description
<i>bstrLanguage</i>	String (ByVal). The two-letter language code representing the language in which to return the string. To obtain language codes, use GetLanguageCountryCodeFromLanguageId . Input argument. String subtype.
<i>bstrUsersXml</i>	An XML string that specifies the users' usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user: <pre><Users><User><Name>username</Name><Id>security identifier</Id></User></Users></pre> Input argument. String subtype.
<i>bstrClassesXml</i>	An XML string that specifies the names and IDs of the security classes. Following is the structure of the string, which must contain one <Class> element per security class: <pre><Classes><Class><Name>security class name</Name><Id>security class Id</Id></Class></Classes></pre> Input argument. String subtype.

Return Value

Returns an XML string that specifies the users' assigned roles and security class access and e-mail alerting rights. Following is the structure of the string. There is one <User> element per

user; each <User> element contains one <Role> element per assigned role, and one <Class> element per security class. The access rights flags are represented by the HFMConstants enumeration described in [“Access Rights Constants” on page 438](#).

```
<UserRightsRoles><User name="username" ><Roles><Role>role name</Role></Roles><Classes><Class name="security class name" id="security class ID" alert="Boolean" rights="access rights flag" /></Classes></User></UserRightsRoles>
```

EnumSecurityClassRightsAndRolesForBiPub

Generates report in specified format (PDF, RTF, HTML or XLS) and returns a localized XML string that lists the specified users’ access rights to the specified security classes and the roles to which the users are assigned.

Syntax

```
<HFMwSecurity>.EnumSecurityClassRightsAndRolesForBiPub(bstrLanguage, bstrUsersXml, bstrClassesXml, lReportFormat, bstrReportTemplateFileName, ReportFileOut)
```

Argument	Description
bstrLanguage	String (ByVal). The two-letter language code representing the language in which to return the string. To obtain language codes, use GetLanguageCountryCodeFromLanguageId . Input argument. String subtype.
bstrUsersXml	An XML string that specifies the users’ usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user: <Users><User><Name>username</Name><Id>security identifier</Id></User></Users> Input argument. String subtype.
bstrClassesXml	An XML string that specifies the names and IDs of the security classes. Following is the structure of the string, which must contain one <Class> element per security class: <Classes><Class><Name>security class name</Name><Id>security class Id</Id></Class></Classes> Input argument. String subtype.
lReportFormat	Value that specifies the report output format: <ul style="list-style-type: none">● 1 = PDF format● 2 = RTF format● 3 = HTML format● 4 = XLS format● -1 = Financial Management format Input argument. Long subtype.
bstrReportTemplateFileName	The BI Publisher report template name that is used to generate the report.

Argument	Description
	Input argument. String subtype.
<i>ReportFileOut</i>	Specifies the file name of the report generated.
	Input argument. String subtype.

Return Value

Returns an XML string that specifies the users' assigned roles and security class access and e-mail alerting rights. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Role> element per assigned role, and one <Class> element per security class. The access rights flags are represented by the HFMConstants enumeration described in [“Access Rights Constants” on page 438](#).

```
<UserRightsRoles><User name="username" ><Roles NODISPLAY='Yes/No'><Role>role name</Role><Classes NODISPLAY='Yes/No'><Class name="security class name" id="security class ID" alert="Boolean" rights="access rights flag" /></Classes></User></UserRightsRoles>
```

String subtype.

EnumUsers

Deprecated - use [EnumUsers2](#).

EnumUsers2

Returns the usernames and security identifiers of all the application's users. The usernames and security identifiers are returned in arrays that have a one-to-one correspondence.

Syntax

```
<HFMwSecurity>.EnumUsers2 pvarabstrSIDs, pvarabstrUserNames
```

Argument	Description
<i>pvarabstrSIDs</i>	Returns an array containing the security identifiers. Input/output argument.
<i>pvarabstrUserNames</i>	Returns an array containing the usernames. Input/output argument.

EnumUsersInGroup

Returns the security identifiers, usernames, and identity types of the users in a user group.

Syntax

```
<HFMwSecurity>.EnumUsersInGroup (bstrGroupsXml, lBitmaskFlags)
```

Argument	Description
----------	-------------

<i>bstrGroupsXml</i>	An XML string that specifies the user groups' names and security identifiers. Following is the structure of the string, which must contain one <User> element per group:
----------------------	--

```
<Users><User><Name>group name</Name><Id>security identifier</Id></User></Users>
```

Input argument. String subtype.

<i>lBitmaskFlags</i>	A bitmask that specifies the identity types of the users to return. Valid values for the bits are represented by the HFMConstants enumeration described in “User Groups - User Type Flag Constants” on page 445 .
----------------------	---

For example, to exclude bits for application administrators from the array, pass HFM_USER_GROUP_ENUM_EXCLUDEADMINS.

Input argument. Long subtype.

Return Value

Returns an XML string that specifies the users' usernames, security identifiers, and identity type bitmasks. Following is the structure of the string. There is one <Group> element per user group; each <Group> element contains one <User> element per user. Valid values for the bits in the identity type bitmasks are represented by the HFMConstants enumeration described in [“Identity Type Constants” on page 445](#).

```
<Groups><Group name="group name" id="security identifier" ><User name="username" id="security identifier" type="identity type bitmask" /></Group></Groups>
```

EnumUsersRoles

Returns a localized XML string that lists the roles to which the specified users are assigned.

Syntax

```
<HFMwSecurity>.EnumUsersRoles (bstrUsersXml)
```

Argument	Description
----------	-------------

<i>bstrLanguage</i>	String (ByVal). The two-letter language code representing the language in which to return the string. To obtain language codes, use GetLanguageCountryCodeFromLanguageId .
---------------------	--

<i>bstrUsersXml</i>	An XML string that specifies the users' usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user:
---------------------	---

```
<Users><User><Name>username</Name><Id>security identifier</Id></User></Users>
```

Input argument. String subtype.

Return Value

Returns an XML string that list the users' assigned roles. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Role> element per assigned role.

```
<UserRights><User name="username" id="security identifier" ><Role>role  
label</Role></User></UserRights>
```

EnumUsersRolesForBiPub

Generates report in specified format (PDF, RTF, HTML or XLS) and returns a localized XML string that lists the roles to which the specified users are assigned.

Syntax

```
<HFMwSecurity>.EnumUsersRolesForBiPub(bstrLanguage, bstrUsersXml,  
lReportFormat, bstrReportTemplateFileName, ReportFileOut)
```

Argument	Description
<i>bstrLanguage</i>	String (ByVal). The two-letter language code representing the language in which to return the string. To obtain language codes, use GetLanguageCountryCodeFromLanguageId .
<i>bstrUsersXml</i>	<p>An XML string that specifies the users' usernames and security identifiers. Following is the structure of the string, which must contain one <User> element per user:</p> <pre><Users><User><Name>username</Name><Id>security identifier</Id></User></Users></pre> <p>Input argument. String subtype.</p>
<i>lReportFormat</i>	<p>Value that specifies the report output format:</p> <ul style="list-style-type: none">● 1 = PDF format● 2 = RTF format● 3 = HTML format● 4 = XLS format● -1 = Financial Management format <p>Input argument. Long subtype.</p>
<i>bstrReportTemplateFileName</i>	<p>The BI Publisher report template name that is used to generate the report.</p> <p>Input argument. String subtype.</p>
<i>ReportFileOut</i>	<p>Specifies the file name of the report generated.</p> <p>Input argument. String subtype.</p>

Return Value

Returns an XML string that list the users' assigned roles. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Role> element per assigned role.

```
<UserRights><User name="username" id="security identifier" ><Roles  
NODISPLAY='Yes/No'><Role> role label</Role></Roles><Classes NODISPLAY='Yes/  
No'><Classes></User></UserRights>
```

String subtype.

EnumUsersInGroupForBiPub

Generates report in specified format (PDF, RTF, HTML or XLS) and returns the security identifiers, usernames, and identity types of the users in a user group.

Syntax

```
<HFMwSecurity>.EnumUsersInGroupForBiPub(bstrGroupsXml, lBitmaskFlags,  
lReportFormat, bstrReportTemplateFileName, ReportFileOut)
```

Argument	Description
<i>bstrGroupsXml</i>	<p>An XML string that specifies the user groups' names and security identifiers. Following is the structure of the string, which must contain one <User> element per group:</p> <pre><Users><User><Name>group name</Name><Id>security identifier</Id></User></Users></pre> <p>Input argument. String subtype.</p>
<i>lBitmaskFlags</i>	<p>A bitmask that specifies the identity types of the users to return. Valid values for the bits are represented by the HFMConstants enumeration described in “User Groups - User Type Flag Constants” on page 445.</p> <p>For example, to exclude bits for application administrators from the array, pass HFM_USER_GROUP_ENUM_EXCLUDEADMINS.</p> <p>Input argument. Long subtype.</p>
<i>lReportFormat</i>	<p>Value that specifies the report output format:</p> <ul style="list-style-type: none">● 1 = PDF format● 2 = RTF format● 3 = HTML format● 4 = XLS format● -1 = Financial Management format <p>Input argument. Long subtype.</p>
<i>bstrReportTemplateFileName</i>	<p>The BI Publisher report template name that is used to generate the report.</p> <p>Input argument. String subtype.</p>
<i>ReportFileOut</i>	<p>Specifies the file name of the report generated.</p> <p>Input argument. String subtype.</p>

Return Value

Returns an XML string that specifies the users' usernames, security identifiers, and identity type bitmasks. Following is the structure of the string. There is one <Group> element per user group;

each <Group> element contains one <User> element per user. Valid values for the bits in the identity type bitmasks are represented by the HFMConstants enumeration described in [“Identity Type Constants” on page 445](#).

```
<Groups NODISPLAY='Yes/No'><Group name="group name" id="security
identifier" ><GUser name" username" id="security identifier" type="identity
type bitmask" /></Group></Groups>
```

String subtype.

EnumUsersWithFilter

Performs a filtered search that returns the security identifiers and usernames of matching users and groups. Filtering options include wildcard searching on usernames and filtering by user category.

Syntax

```
<HFMwSecurity>.EnumUsersWithFilter bstrFilter,
lBitFieldPrincipalsToReturn, pvarabstrSIDs, pvarabstrUserNames
```

Argument	Description
<i>bstrFilter</i>	The string by which to search user and group names. For wildcard searches, use asterisks (*). Input argument. String subtype.
<i>lBitFieldPrincipalsToReturn</i>	A bitmask that represents the user categories by which to filter. Valid values for the bits are listed in “Search Filters” on page 445 . Input argument. Long subtype.
<i>pvarabstrSIDs</i>	Returns the security identifiers for the users and groups that match the search criteria. Input/output argument.
<i>pvarabstrUserNames</i>	Returns the names of the users and groups that match the search criteria. Input/output argument.

Example

The following function returns an array of the user and group names that begin with the specified string.

```
Function GetUsersByPrefix(sPrefix)
Dim cSecurity, vaIds, vaNames
Set cSecurity = Server.CreateObject("Hyperion.HFMwSecurity")
'g_chFMSession represents an HFMwSession instance
cSecurity.SetWebSession g_chFMSession
cSecurity.EnumUsersWithFilter sPrefix & "*", HFM_USER_GROUP_ENUM_GROUPS _
    & HFM_USER_GROUP_ENUM_USERS, vaIds, vaNames
GetUsersByPrefix = vaNames
End Function
```

GetOwner

This method is not supported as of Release 4.1.

GetRulesMode

For internal use.

GetSecurityClassAccessForAllUsers

Deprecated - use [GetSecurityClassAccessForAllUsers2](#).

GetSecurityClassAccessForAllUsers2

Returns the access rights to a given security class for the current application's users. `GetSecurityClassAccessForAllUsers2` returns arrays of security identifiers, usernames, and access rights; the arrays have a one-to-one correspondence.

Syntax

```
<HFMwSecurity>.GetSecurityClassAccessForAllUsers2 lSecurityClassID,  
pvarabstrUserSIDs, pvarabstrUserNames, pvaralRights
```

Argument	Description
<i>lSecurityClassID</i>	The ID of the security class. Tip: To get the ID for a security class name, use GetSecurityClassID . Input argument. Long subtype.
<i>pvarabstrUserSIDs</i>	Returns an array containing the users' security identifiers. Input/output argument.
<i>pvarabstrUserNames</i>	Returns an array containing the users' usernames. Input/output argument.
<i>pvaralRights</i>	Returns an array containing the users' access rights to the security class. Valid values are represented by the HFMConstants type library constants listed in " Access Rights Constants " on page 438. Input/output argument.

Example

The following example prints an application's usernames and their access rights to a given security class to the browser. The subroutine takes a security class name; the ID for this security class is obtained with `GetSecurityClassID`, and then passed to `GetSecurityClassAccessForAllUsers2`.

```

Sub printClassRights(sSecName)
Dim cHFMSecurity, lSecID, vaUserIDs, vaUserNames, vaRights
Dim sRight
Set cHFMSecurity = Server.CreateObject("Hyperion.HFMwSecurity")
' cHFMSession is a previously set HFMwSession object
cHFMSecurity.SetWebSession cHFMSession
lSecID = cHFMSecurity.GetSecurityClassID(sSecName)
cHFMSecurity.GetSecurityClassAccessForAllUsers2 lSecID, vaUserIDs, _
    vaUserNames, vaRights
For i = 0 to uBound(vaUserNames)
Select Case vaRights(i)
    Case HFM_ACCESS_RIGHTS_NONE
        sRight = "None"
    Case HFM_ACCESS_RIGHTS_READANDPROMOTE
        sRight = "Promote"
    Case HFM_ACCESS_RIGHTS_READONLY
        sRight = "Read"
    Case HFM_ACCESS_RIGHTS_ALL
        sRight = "All"
End Select
    Response.Write "<p>" & vaUserNames(i) & ": " & sRight & "</p>"
Next
End Sub

```

GetSecurityClassID

Returns the ID for a given security class name.

Syntax

```
<HFMwSecurity>.GetSecurityClassID (bstrSecurityClass)
```

Argument	Description
<i>bstrSecurityClass</i>	The security class name.

Input argument. String subtype.

Return Value

Returns the ID of the specified security class.

Example

GetSecurityClassID is used in the example for [GetSecurityClassAccessForAllUsers2](#).

GetSecurityClassRightsForConnectedUser

Returns the connected user's access rights for a given security class.

Syntax

```
<HFMwSecurity>.GetSecurityClassRightsForConnectedUser (bstrSecurityClass)
```

Argument	Description
----------	-------------

<i>bstrSecurityClass</i>	The name of the security class. Input argument. String subtype.
--------------------------	--

Return Value

The user's access rights to the specified security class. Valid values are represented by the HFMConstants type library constants listed in [“Access Rights Constants” on page 438](#).

GetUserAccessForAllSecurityClasses

Deprecated - use [GetUserAccessForAllSecurityClasses2](#).

GetUserAccessForAllSecurityClasses2

Returns a given user's access rights to the current application's security classes. `GetUserAccessForAllSecurityClasses2` returns arrays of security class IDs, security class names, and the user's access rights; the arrays have a one-to-one correspondence.

Syntax

```
<HFMwSecurity>.GetUserAccessForAllSecurityClasses2 bstrUserSID,  
pvaralSecurityClassIDs, pvarabstrSecurityClasses, pvaralRights
```

Argument	Description
----------	-------------

<i>bstrUserSID</i>	The user's security identifier. Input argument. String subtype.
<i>pvaralSecurityClassIDs</i>	Returns an array containing the IDs that represent the application's security classes. Input/output argument.
<i>pvarabstrSecurityClasses</i>	Returns an array containing the names of the application's security classes. Input/output argument.
<i>pvaralRights</i>	Returns an array of the access rights for the security classes. Valid values are represented by the HFMConstants type library constants listed in “Access Rights Constants” on page 438 . Input/output argument.

GetUserID

Deprecated - use [GetUserSID](#).

GetUserInfoFromUniqueID

Deprecated. Use [GetUserInfoFromUniqueID2](#).

GetUserInfoFromUniqueID2

Returns user information such as the username, first name, and last name, given the user's external authentication token and security identifier.

Syntax

```
<HFMwSecurity>.GetUserInfoFromUniqueID2 bstrSSOToken, bstrUniqueUserID,  
pvarbstrUserName, pbastrFullUserName, pvarbstrFirstName, pvarbstrLastName,  
pvarbstrDesc, pvarbstrEMail
```

Argument	Description
<i>bstrSSOToken</i>	The user's external authentication token. Tip: You can get an obtain a token from GetSSOTokenUsingWebSecurityAgentCredentials . Input argument. String subtype.
<i>bstrUniqueUserID</i>	The user's security identifier. Input argument. String subtype.
<i>pvarbstrUserName</i>	Returns the user's username. Output argument.
<i>pbastrFullUserName</i>	Returns the user's fully qualified username. Output argument.
<i>pvarbstrFirstName</i>	Returns the user's first name. Output argument.
<i>pvarbstrLastName</i>	Returns the user's last name. Output argument.
<i>pvarbstrDesc</i>	Returns the user's description. Output argument.
<i>pvarbstrEMail</i>	Returns the user's e-mail address. Output argument.

GetUserSID

Returns the security identifier of a given user.

Syntax

```
<HFMwSecurity>.GetUserSID (bstrName)
```

Argument	Description
<i>bstrName</i>	The username of the user. Input argument. String subtype.

Return Value

Returns the security identifier.

IsClassicHFMAApplication

Indicates whether an application is a Classic application.

Syntax

```
<HFMwSecurity>.IsClassicHFMAApplication()
```

Return Value

Boolean subtype. Returns TRUE for a Classic application, FALSE for a Performance Management Architect application.

IsConnectedUserAllowedToPerformTask

Indicates whether the connected user has rights to perform the specified task.

Syntax

```
<HFMwSecurity>.IsConnectedUserAllowedToPerformTask (lTaskID)
```

Argument	Description
<i>lTaskID</i>	The ID of the task. Valid values are represented by the HFMConstants type library constants listed in “Task Constants” on page 440 . Input argument. Long subtype.

Return Value

Returns TRUE if the user has rights, FALSE otherwise.

IsConnectedUserInRole

Indicates whether the connected user has access rights to the specified role.

Syntax

```
<HFMwSecurity>.IsConnectedUserInRole (lRoleID)
```

Argument	Description
<i>lRoleID</i>	The ID of the role. Valid values are represented by the HFMConstants enumeration described in “Role Constants” on page 438 . Input argument. Long subtype.

Return Value

Returns TRUE if the user has rights, FALSE otherwise. Boolean subtype.

LogInfo

Writes an HRESULT and an associated description to the application log.

Syntax

```
<HFMwSecurity>.LogInfo hresLog, bstrLog
```

Argument	Description
<i>hresLog</i>	The HRESULT to log. Input argument. Long subtype.
<i>bstrLog</i>	The description to log. Input argument. String subtype.

SaveSecurityClassRights

Specifies one or more user's access and e-mail alerting rights to the specified security classes.

Syntax

```
<HFMwSecurity>.SaveSecurityClassRights bstrRightsXml
```

Argument	Description
<i>bstrRightsXml</i>	An XML string that specifies the users' access and e-mail alerting rights to the security classes. Following is the structure of the string. There is one <User> element per user; each <User> element contains one <Class> element per security class. Access rights IDs are represented by the HFMConstants enumeration described in “Access Rights Constants” on page 438 :

Argument	Description
	<pre><UserRights><User name="username" id="security identifier" ><Class name="security class name" id="security class ID" alert="Boolean" rights="access rights ID" /></User></UserRights></pre>
	Input argument. String subtype.

SetWebSession

Associates an HFMwSecurity object with the HFMwSession object for an application

Syntax

```
<HFMwSecurity>.SetWebSession varpIDispHFMwSession
```

Argument	Description
<i>varpIDispHFMwSession</i>	The HFMwSession object for the application.
	Input argument.

Example

SetWebSession is used in the example for [EnumSecurityClasses](#).

In This Chapter

HF MwSystemInfo Component: Methods	311
--	-----

The HF MwSystemInfo type library contains the HF MwSystemInfo component, which exposes various system-related features. For example, this component includes methods that work with audit information, enable and disable new connections, and log off users.

Obtaining an HF MwSystemInfo Object Reference

Create an HF MwSystemInfo object with `Server.CreateObject`. After creating the object, call `SetWebSession` to associate the object with the HF MwSession object for the application with which you want to work:

```
Set cHFMSysInfo = Server.CreateObject("Hyperion.HF MwSystemInfo")
'cHFMSession is an HF MwSession object reference.
cHFMSysInfo.SetWebSession cHFMSession
```

HF MwSystemInfo Component: Methods

The HF MwSystemInfo component contains the methods described in the following topics.

ClearAuditTasks

Deletes the task audit history for all tasks that meet the specified criteria. Criteria include date range, application server, and user.

Note: To delete audit information for a given task, use `ClearAuditTasks2`.

Syntax

```
<HF MwSystemInfo>.ClearAuditTasks dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, lActivityUserID
```

Argument	Description
----------	-------------

<code>dStartTime</code>	The starting date and time of the date range. Pass a Double that can be cast to a Date format.
-------------------------	--

Argument	Description
	Input argument. Double subtype.
<i>dEndTime</i>	The ending date and time of the date range. Pass a Double that can be cast to a Date format. Input argument. Double subtype.
<i>vbAllServers</i>	A flag that determines whether to delete the task audit history for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to delete the task audit history. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllUsers</i>	A flag that determines whether to delete the task audit history for all users. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument. Input argument. Boolean subtype.
<i>lActivityUserID</i>	The activity user ID of the user for whom to delete the task audit history. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. Tip: You can get the activity user ID of all users on the system with the HFMwManageApplications component's EnumUsersOnSystem method. Input argument. Long subtype.

Example

The following subroutine deletes the audit history for a given application server.

```
Sub clearTaskAuditServer(sServer)
Dim cHFMSysInfo
Set cHFMSysInfo = Server.CreateObject("Hyperion.HFMwSystemInfo")
'cHFMSession is an HFMwSession object reference.
cHFMSysInfo.SetWebSession cHFMSession
cHFMSysInfo.ClearAuditTasks 0, CDbl(Now()), FALSE, sServer, _
    TRUE, 0
End Sub
```

ClearAuditTasks2

Deletes the audit history for a given task that meets the specified criteria. Criteria include date range, application server, and user.

Note: To delete audit information for all tasks, either set the *vbAllTasks* argument to FALSE or use [ClearAuditTasks](#).

```
<HFMwSystemInfo>.ClearAuditTasks2 dStartTime, dEndTime, vbAllServers,
bstrServer, vbAllUsers, bstrActivityUserName, vbAllTasks, lActivityTaskID
```


Argument	Description
<i>dStartTime</i>	The starting date and time of the date range. Pass a Double that can be cast to a Date format. Input argument. Double subtype.
<i>dEndTime</i>	The ending date and time of the date range. Pass a Double that can be cast to a Date format. Input argument. Double subtype.
<i>vbAllServers</i>	A flag that determines whether to delete the task audit history for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to delete the task audit history. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllUsers</i>	A flag that determines whether to delete the task audit history for all users. Pass TRUE for all users, FALSE to specify a user with the <i>bstrActivityUserName</i> argument. Input argument. Boolean subtype.
<i>bstrActivityUserName</i>	The user name of the user for whom to delete the task audit history. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllTasks</i>	Specifies whether audit information for all tasks or a given task is deleted. Pass TRUE for all tasks, FALSE to specify a task with the <i>lActivityTaskID</i> argument. Input argument. Boolean subtype.
<i>lActivityTaskID</i>	The ID of the task for which to delete audit information. This argument is used only if the <i>vbAllTasks</i> argument is set to FALSE. Task IDs are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453 . Input argument. Long subtype.

Example

The following subroutine deletes the audit history for a given task. The subroutine takes the ID of the task to be deleted.

```
Sub ClearTaskAuditAllUsers(lTaskId)
Dim cHFMSysInfo
Set cHFMSysInfo = Server.CreateObject("Hyperion.HFMwSystemInfo")
'cHFMSession is an HFMwSession object reference.
cHFMSysInfo.SetWebSession cHFMSession
cHFMSysInfo.ClearAuditTasks2 0, CDbl(Now()), True, "", _
    True, "", False, lTaskId
End Sub
```

EnableNewConnections

Enables or disables new Financial Management connections for the specified user and application server criteria.

Tip: To enable new connections for a given cluster or application, use `HFMwManageApplications.EnableNewConnections`.

Syntax

```
<HFMwSystemInfo>.EnableNewConnections vbEnable, vbAllServers, bstrServer, vbAllUsers, lActivityUserID
```

Argument	Description
<i>vbEnable</i>	A flag that determines whether new connections are enabled or disabled. Pass TRUE to enable new connections, FALSE to disable. Input argument. Boolean subtype.
<i>vbAllServers</i>	A flag that determines whether to enable or disable connections for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument. Input argument. Boolean subtype.
<i>bstrServer</i>	The name of the application server for which to enable or disable new connections. This is used only if the <i>vbAllServers</i> argument is set to FALSE. Input argument. String subtype.
<i>vbAllUsers</i>	A flag that determines whether to enable connections for all users. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument. Input argument. Boolean subtype.
<i>lActivityUserID</i>	The activity user ID of the user for whom to enable connections. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. Input argument. Long subtype.

EnumActivityServers

Returns the names of the application servers for which there are task audit and data audit records.

Syntax

```
<HFMwSystemInfo>.EnumActivityServers()
```

Return Value

Returns an array containing the names of the application servers.

Note: This is a 1-based array.

EnumActivityUsers

Returns the usernames of all users who have performed at least one activity in the application.

Syntax

```
<HFMwSystemInfo>.EnumActivityUsers
```

Return Value

Returns an array containing the usernames.

Note: This is a 1-based array.

EnumAuditTasks

Returns task audit information from a given range of audit records that meet the selection criteria. Criteria include date range, application server, and user. Audit information is returned in arrays that have a one-to-one correspondence.

Note: To return audit information for a given task, use [EnumAuditTasks2](#).

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *pvarlTotalRecords* argument returns the total number of records that match the selection criteria. To iterate through all the matching records, in the first call to `EnumAuditTasks` pass 0 to *lStartRecord*, then use the count returned by *pvarlTotalRecords* to loop through the remaining records.

Note: The number of matching records can change after you call `EnumAuditTasks`. For example, a user might delete some or all of the audit records.

Syntax

```
<HFMwSystemInfo>.EnumAuditTasks dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, bstrActivityUserName, lStartRecord, lEndRecord,  
pvaravarActivityUserID, pvaravarActivityUserName,  
pvaravarActivitySessionID, pvaravarServerName, pvaravarActivityCode,  
pvaravarStartTime, pvaravarEndTime, pvaravarDescription,  
pvaravarModuleNames, pvarlTotalRecords
```

Argument	Description
<i>dStartTime</i>	The starting date and time of the date range. Pass a Double that can be cast to a Date format. Input argument. Double subtype.
<i>dEndTime</i>	The ending date and time of the date range. Pass a Double that can be cast to a Date format. Input argument. Double subtype.

Argument	Description
<i>vbAllServers</i>	<p>A flag that determines whether audit information for all application servers is returned. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServer</i>	<p>The name of the application server for which to return audit information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllUsers</i>	<p>A flag that determines whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>bstrActivityUserName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrActivityUserName</i>	<p>The username of the user for whom to return audit information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>lStartRecord</i>	<p>The index of the first record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>lEndRecord</i>	<p>The index of the last record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>pvaravarActivityUserID</i>	<p>Returns an array containing the activity user IDs for the users who performed the tasks.</p> <p>The corresponding item in the array returned by the <i>pvaravarActivityUserName</i> argument contains the username associated with an activity user ID.</p> <p>Input/output argument.</p>
<i>pvaravarActivityUserName</i>	<p>Returns an array containing the usernames of the users who performed the tasks.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaravarActivitySessionID</i>	<p>Returns an array containing the internal IDs of the sessions in which the tasks occurred.</p> <p>Input/output argument.</p>
<i>pvaravarServerName</i>	<p>Returns an array containing the names of the application servers on which the tasks occurred.</p> <p>Note: This is a 1-based array.</p> <p>Input/output argument.</p>
<i>pvaravarActivityCode</i>	<p>Returns an array containing the IDs of the activities performed by the users. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453.</p> <p>Input/output argument.</p>
<i>pvaravarStartTime</i>	<p>Returns an array containing the starting times of the activities. Array items are formatted as Doubles that can be cast to the Date format.</p> <p>Input/output argument.</p>

Argument	Description
<i>pvaravarEndTime</i>	Returns an array containing the ending times of the activities. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarDescription</i>	Returns an array containing the tasks' descriptions. Note: This is a 1-based array. Input/output argument.
<i>pvaravarModuleNames</i>	Returns an array containing the names of the tasks' modules. Note: This is a 1-based array. Input/output argument.
<i>pvarlTotalRecords</i>	Returns the total number of audit records in the database that meet the filtering criteria. Input/output argument.

EnumAuditTasks2

Returns task audit information from a given range of audit records that meet the selection criteria. Criteria include task, date range, application server, and user. Audit information is returned in arrays that have a one-to-one correspondence.

Note: To return audit information for all tasks, either set the *vbAllTasks* argument to FALSE or use [EnumAuditTasks](#).

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *pvarlTotalRecords* argument returns the total number of records that match the selection criteria. To iterate through all the matching records, in the first call to `EnumAuditTasks2` pass 0 to *lStartRecord*, then use the count returned by *pvarlTotalRecords* to loop through the remaining records.

Note: The number of matching records can change after you call `EnumAuditTasks2`. For example, a user might delete some or all of the audit records.

Syntax

```
<HFmwSystemInfo>.EnumAuditTasks2 dStartTime, dEndTime, vbAllServers,
bstrServer, vbAllUsers, bstrActivityUserName, vbAllTasks, lActivityTaskID,
lStartRecord, lEndRecord, pvaravarActivityUserID,
pvaravarActivityUserName, pvaravarActivitySessionID, pvaravarServerName,
pvaravarActivityCode, pvaravarStartTime, pvaravarEndTime,
pvaravarDescription, pvaravarModuleNames, pvarlTotalRecords
```

Argument	Description
<i>dStartTime</i>	<p>The starting date and time of the date range. Pass a Double that can be cast to a Date format.</p> <p>Tip: To search for all dates, pass 0.</p> <p>Input argument. Double subtype.</p>
<i>dEndTime</i>	<p>The ending date and time of the date range. Pass a Double that can be cast to a Date format.</p> <p>Tip: To search for all dates, pass 0.</p> <p>Input argument. Double subtype.</p>
<i>vbAllServers</i>	<p>A flag that determines whether audit information for all application servers is returned. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServer</i>	<p>The name of the application server for which to return audit information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllUsers</i>	<p>A flag that determines whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>bstrActivityUserName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrActivityUserName</i>	<p>The username of the user for whom to return audit information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllTasks</i>	<p>Specifies whether audit information for all tasks or a given task is returned. Pass TRUE for all tasks, FALSE to specify a task with the <i>lActivityTaskID</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lActivityTaskID</i>	<p>The ID of the task for which to return audit information. This argument is used only if the <i>vbAllTasks</i> argument is set to FALSE.</p> <p>Task IDs are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453.</p> <p>Input argument. Long subtype.</p>
<i>lStartRecord</i>	<p>The index of the first record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>lEndRecord</i>	<p>The index of the last record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>
<i>pvaravarActivityUserID</i>	<p>Returns an array containing the activity user IDs for the users who performed the tasks.</p> <p>The corresponding item in the array returned by the <i>pvaravarActivityUserName</i> argument contains the username associated with an activity user ID.</p> <p>Input/output argument.</p>
<i>pvaravarActivityUserName</i>	<p>Returns an array containing the usernames of the users who performed the tasks.</p> <p>Note: This is a 1-based array.</p>

Argument	Description
	Input/output argument.
<i>pvaravarActivitySessionID</i>	Returns an array containing the internal IDs of the sessions in which the tasks occurred. A user can have multiple sessions. Input/output argument.
<i>pvaravarServerName</i>	Returns an array containing the names of the application servers on which the tasks occurred. Note: This is a 1-based array. Input/output argument.
<i>pvaravarActivityCode</i>	Returns an array containing the IDs of the tasks performed by the users. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453 . Input/output argument.
<i>pvaravarStartTime</i>	Returns an array containing the starting times of the tasks. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarEndTime</i>	Returns an array containing the ending times of the tasks. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarDescription</i>	Returns an array containing the tasks’ descriptions. Note: This is a 1-based array. Input/output argument.
<i>pvaravarModuleNames</i>	Returns an array containing the names of the tasks’ modules. Note: This is a 1-based array. Input/output argument.
<i>pvarITotalRecords</i>	Returns the total number of audit records in the database that meet the filtering criteria. Input/output argument.

EnumProhibitConnections

Returns information on the applications, application servers, and users for which connections have been disabled. The information is returned in arrays that have a one-to-one correspondence.

Tip: To return this information for only a given server cluster, use `HFMwManageApplications.EnumProhibitConnections`.

Syntax

```
<HFMwSystemInfo>.EnumProhibitConnections pvaravarvbAllApps,  
pvaravarbstrAppNames, pvaravarvbAllServers, pvaravarbstrServerNames,  
pvaravarvbAllUsers, pvaravarlActivityUserIDs, pvaravarbstrActivityUserNames
```

Argument	Description
<i>pvaravarvbAllApps</i>	<p>Returns an array that indicates whether connections are disabled for all applications. Array items can contain 0 or -1:</p> <ul style="list-style-type: none">● -1 indicates connections to all applications are disabled.● 0 indicates that only connections to the application returned by the corresponding <i>pvaravarbstrAppNames</i> argument's array item are disabled. <p>Input/output argument.</p>
<i>pvaravarbstrAppNames</i>	<p>Returns an array containing the names of the applications for which connections are disabled. Application names are returned only when the corresponding <i>pvaravarvbAllApps</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravarvbAllApps</i> argument's array item contains -1, this array item contains the string "AllApps".</p> <p>Input/output argument.</p>
<i>pvaravarvbAllServers</i>	<p>Returns an array that indicates whether connections are disabled for all application servers. Array items can contain 0 or -1:</p> <ul style="list-style-type: none">● -1 indicates connections to all application servers are disabled.● 0 indicates that only connections to the application server returned by the corresponding <i>pvaravarbstrServerNames</i> argument's array item are disabled. <p>Input/output argument.</p>
<i>pvaravarbstrServerNames</i>	<p>Returns an array containing the names of the application servers for which connections are disabled. Application server names are returned only when the corresponding <i>pvaravarvbAllServers</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravarvbAllServers</i> argument's array item contains -1, this array item contains the string "AllServers".</p> <p>Input/output argument.</p>
<i>pvaravarvbAllUsers</i>	<p>Returns an array that indicates whether connections are disabled for all users. Array items can contain 0 or -1:</p> <ul style="list-style-type: none">● -1 indicates connections for all users are disabled.● 0 indicates that only connections for the user represented by the corresponding <i>pvaravarlActivityUserIDs</i> and <i>pvaravarbstrActivityUserNames</i> arguments' array items are disabled. <p>Input/output argument.</p>
<i>pvaravarlActivityUserIDs</i>	<p>Returns an array containing the activity user IDs of the users for whom connections are disabled. Valid IDs are returned only when the corresponding <i>pvaravarvbAllUsers</i> argument's array item contains 0.</p>

Argument	Description
	<p>The corresponding item in the array returned by the <i>pvaravbstrActivityUserNames</i> argument contains the username associated with an activity user ID.</p> <p>Note: If the corresponding <i>pvaravrbAllUsers</i> argument's array item contains -1, this array item contains -1.</p> <p>Input/output argument.</p>
<i>pvaravbstrActivityUserNames</i>	<p>Returns an array containing the usernames of the users for whom connections are disabled. Usernames are returned only when the corresponding <i>pvaravrbAllUsers</i> argument's array item contains 0.</p> <p>Note: This is a 1-based array.</p> <p>Note: If the corresponding <i>pvaravrbAllUsers</i> argument's array item contains -1, this array item contains an empty string.</p> <p>Input/output argument.</p>

EnumRunningTasks

Returns information about the running tasks that meet the selection criteria. Criteria include task type, user, application server, user session, and task status. Task information is returned in arrays that have a one-to-one correspondence.

Note: To also return flags that indicate whether the tasks currently are running or stopped, use [EnumRunningTasksEx](#).

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *plTotalRecords* argument returns the total number of records that match the selection criteria. To iterate through all the matching records, in the first call to *EnumRunningTasks* pass 0 to *lStartRecord*, then use the count returned by *plTotalRecords* to loop through the remaining records.

Note: The number of matching records can change after you call *EnumRunningTasks*. For example, a new task could be added.

Syntax

```
<HFmwSystemInfo>.EnumRunningTasks (vbAllTaskTypes, lTaskType, vbAllUsers,
bstrUserName, vbAllServers, bstrServerName, vbAllSessions, vbAllStatus,
lStatus, lStartRecord, lEndRecord, pvaravarTaskIDs, pvaravarTaskTypes,
pvaravarTaskProgress, pvaravarTaskStatus, pvaravarUserNames,
pvaravarServerNames, pvaravarStartTimes, pvaravarStartRunningTimes,
pvaravarLastUpdateTimes, pvaravarDescriptions, pvaravarLogFiles)
```

Argument	Description
<i>vbAllTaskTypes</i>	<p>A flag that specifies whether to filter for a specific type of task. Pass TRUE to return information for all tasks, FALSE to filter by the type of task specified by the <i>lTaskType</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lTaskType</i>	<p>Long (ByVal). The ID of the task type for which to return information. This argument is used only if the <i>vbAllTaskTypes</i> argument is set to FALSE.</p> <p>Following are the constants that represent valid values. These are from the HFMConstants enumeration tagUSERACTIVITYCODE:</p> <ul style="list-style-type: none"> ● USERACTIVITYCODE_CONSOLIDATION: consolidations ● USERACTIVITYCODE_DATA_LOAD: data loads ● USERACTIVITYCODE_EA_EXPORT: Extended Analytics exports ● USERACTIVITYCODE_IC_POSTALL: post all intercompany transactions ● USERACTIVITYCODE_IC_UNPOSTALL: unpost all intercompany transactions ● USERACTIVITYCODE_IC_DELETEALL: delete all intercompany transactions ● USERACTIVITYCODE_IC_UNMATCHALL: unmatched all intercompany transactions ● USERACTIVITYCODE_IC_AUTOMATCHBYID: automatch intercompany transactions by ID ● USERACTIVITYCODE_IC_AUTOMATCHBYACCT: automatch intercompany transactions by account ● USERACTIVITYCODE_IC_MATCHINGRPTBYID: matching intercompany transactions by ID report ● USERACTIVITYCODE_IC_MATCHINGRPTBYACCT: matching intercompany transactions by account report ● USERACTIVITYCODE_IC_TRANSACTIONRPT: intercompany transactions report <p>Input argument. Long subtype.</p>
<i>vbAllUsers</i>	<p>A flag that specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <i>bstrUserName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrUserName</i>	<p>The username for which to return task information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllServers</i>	<p>A flag that specifies whether to filter by application server. Pass TRUE for all application servers, FALSE to filter by the application server specified with the <i>bstrServerName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServerName</i>	<p>The name of the application server for which to return task information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllSessions</i>	<p>Specifies whether to return information for all user sessions or only the session for the connected user. Pass TRUE for all sessions, FALSE otherwise.</p> <p>Input argument. Boolean subtype.</p>

Argument	Description
<i>vbAllStatus</i>	Specifies whether to filter by task status. Pass TRUE for all statuses, FALSE to filter by the status specified with the <i>lStatus</i> argument. Input argument. Boolean subtype.
<i>lStatus</i>	The task status by which to filter. This argument is used only if the <i>vbAllStatus</i> argument is set to FALSE. Valid values are represented by the HFMConstants type library constants listed in “Task Status Constants” on page 452 . Input argument. Long subtype.
<i>lStartRecord</i>	The index of the first record in the range of records to retrieve. This is a zero-based index. Input argument. Long subtype.
<i>lEndRecord</i>	The index of the last record in the range of records to retrieve. This is a zero-based index. Input argument. Long subtype.
<i>pvaravarTaskIDs</i>	Returns an array containing the task IDs that identify the running tasks. Input/output argument.
<i>pvaravarTaskTypes</i>	Returns an array containing the tasks’ types. Valid values are represented by the constants listed as valid values for the <i>lTaskType</i> argument. Input/output argument.
<i>pvaravarTaskProgress</i>	Returns an array containing the progress complete percentages of the running tasks. Input/output argument.
<i>pvaravarTaskStatus</i>	Returns an array containing the tasks’ statuses. Valid values are represented by the HFMConstants type library constants listed in “Task Status Constants” on page 452 . Input/output argument.
<i>pvaravarUserNames</i>	Returns an array containing the usernames of the running tasks’ users. Input/output argument.
<i>pvaravarServerNames</i>	Returns an array containing the names of the application servers for the running tasks. Input/output argument.
<i>pvaravarStartTimes</i>	Returns an array containing the timestamps of the tasks’ scheduled start times. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarStartRunningTimes</i>	Returns an array containing the timestamps of the tasks’ actual start times. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarLastUpdateTimes</i>	Returns an array containing the timestamps of the last time the tasks were updated. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarDescriptions</i>	Returns an array containing the descriptions for the tasks. The array is returned as a String subtype.

Argument	Description
	Input/output argument.
<i>pvaravarLogFiles</i>	Returns an array containing the file names and paths of the tasks' log files.
	Input/output argument.

Return Value

Returns a count of the number of tasks returned by `EnumRunningTasks`.

EnumRunningTasksEx

Returns information about the running tasks that meet the selection criteria; the information returned includes an array of flags that indicate whether the tasks currently are running or stopped. Criteria include task type, user, application server, user session, and task status. Task information is returned in arrays that have a one-to-one correspondence.

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *plTotalRecords* argument returns the total number of records that match the selection criteria. To iterate through all the matching records, in the first call to `EnumRunningTasksEx` pass 0 to *lStartRecord*, then use the count returned by *plTotalRecords* to loop through the remaining records.

Note: The number of matching records can change after you call `EnumRunningTasksEx`. For example, a new task could be added.

Syntax

```
<HFMwSystemInfo>.EnumRunningTasksEx (vbAllTaskTypes, lTaskType,
vbAllUsers, bstrUserName, vbAllServers, bstrServerName, vbAllSessions,
vbAllStatus, lStatus, lStartRecord, lEndRecord, pvaravarTaskIDs,
pvaravarTaskTypes, pvaravarTaskProgress, pvaravarTaskStatus,
pvaravarUserNames, pvaravarServerNames, pvaravarStartTimes,
pvaravarStartRunningTimes, pvaravarLastUpdateTimes, pvaravarDescriptions,
pvarbStopTaskFlags, pvaravarLogFiles)
```

Argument	Description
<i>vbAllTaskTypes</i>	<p>A flag that specifies whether to filter for a specific type of task. Pass TRUE to return information for all tasks, FALSE to filter by the type of task specified by the <i>lTaskType</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lTaskType</i>	<p>Long (ByVal). The ID of the task type for which to return information. This argument is used only if the <i>vbAllTaskTypes</i> argument is set to FALSE.</p> <p>Following are the constants that represent valid values. These are from the <code>HFMConstants</code> enumeration <code>tagUSERACTIVITYCODE</code>:</p> <ul style="list-style-type: none"> ● <code>USERACTIVITYCODE_CONSOLIDATION</code>: consolidations ● <code>USERACTIVITYCODE_DATA_LOAD</code>: data loads

Argument	Description
	<ul style="list-style-type: none"> ● USERACTIVITYCODE_EA_EXPORT: Extended Analytics exports ● USERACTIVITYCODE_IC_POSTALL: post all intercompany transactions ● USERACTIVITYCODE_IC_UNPOSTALL: unpost all intercompany transactions ● USERACTIVITYCODE_IC_DELETEALL: delete all intercompany transactions ● USERACTIVITYCODE_IC_UNMATCHALL: unmatched all intercompany transactions ● USERACTIVITYCODE_IC_AUTOMATCHBYID: automatch intercompany transactions by ID ● USERACTIVITYCODE_IC_AUTOMATCHBYACCT: automatch intercompany transactions by account ● USERACTIVITYCODE_IC_MATCHINGRPTBYID: matching intercompany transactions by ID report ● USERACTIVITYCODE_IC_MATCHINGRPTBYACCT: matching intercompany transactions by account report ● USERACTIVITYCODE_IC_TRANSACTIONRPT: intercompany transactions report <p>Input argument. Long subtype.</p>
<i>vbAllUsers</i>	<p>A flag that specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <i>bstrUserName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrUserName</i>	<p>The username for which to return task information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllServers</i>	<p>A flag that specifies whether to filter by application server. Pass TRUE for all application servers, FALSE to filter by the application server specified with the <i>bstrServerName</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServerName</i>	<p>The name of the application server for which to return task information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllSessions</i>	<p>Specifies whether to return information for all user sessions or only the session for the connected user. Pass TRUE for all sessions, FALSE otherwise.</p> <p>Input argument. Boolean subtype.</p>
<i>vbAllStatus</i>	<p>Specifies whether to filter by task status. Pass TRUE for all statuses, FALSE to filter by the status specified with the <i>lStatus</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lStatus</i>	<p>The task status by which to filter. This argument is used only if the <i>vbAllStatus</i> argument is set to FALSE. Valid values are represented by the HFMConstants type library constants listed in “Task Status Constants” on page 452.</p> <p>Input argument. Long subtype.</p>
<i>lStartRecord</i>	<p>The index of the first record in the range of records to retrieve. This is a zero-based index.</p> <p>Input argument. Long subtype.</p>

Argument	Description
<i>lEndRecord</i>	The index of the last record in the range of records to retrieve. This is a zero-based index. Input argument. Long subtype.
<i>pvaravarTaskIDs</i>	Returns an array containing the task IDs that identify the running tasks. Input/output argument.
<i>pvaravarTaskTypes</i>	Returns an array containing the tasks' types. Valid values are represented by the constants listed as valid values for the <i>lTaskType</i> argument. Input/output argument.
<i>pvaravarTaskProgress</i>	Returns an array containing the progress complete percentages of the running tasks. Input/output argument.
<i>pvaravarTaskStatus</i>	Returns an array containing the tasks' statuses. Valid values are represented by the HFMConstants type library constants listed in "Task Status Constants" on page 452 . Input/output argument.
<i>pvaravarUserNames</i>	Returns an array containing the usernames of the running tasks' users. Input/output argument.
<i>pvaravarServerNames</i>	Returns an array containing the names of the application servers for the running tasks. Input/output argument.
<i>pvaravarStartTimes</i>	Returns an array containing the timestamps of the tasks' scheduled start times. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarStartRunningTimes</i>	Returns an array containing the timestamps of the tasks' actual start times. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarLastUpdateTimes</i>	Returns an array containing the timestamps of the last time the tasks were updated. Array items are formatted as Doubles that can be cast to the Date format. Input/output argument.
<i>pvaravarDescriptions</i>	Returns an array containing the descriptions for the tasks. The array is returned as a String subtype. Input/output argument.
<i>pvarbStopTaskFlags</i>	Returns an array of Booleans that indicate whether the tasks are running or stopped. An array item returns TRUE if the task is stopped, FALSE if it is running. Input/output argument.
<i>pvaravarLogFiles</i>	Returns an array containing the file names and paths of the tasks' log files. Input/output argument.

Return Value

Returns a count of the number of tasks returned by `EnumRunningTasksEx`.

GetActivityCodeDesc

Returns the description of a given type of activity.

Syntax

```
<HFMwSystemInfo>.GetActivityCodeDesc (lActivityCode)
```

Argument	Description
----------	-------------

<i>lActivityCode</i>	The ID of the activity. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 453 .
----------------------	---

Input argument. Long subtype.

Return Value

Returns the activity’s description. String subtype.

GetKillUserStatus

Indicates whether an administrator has logged off the current user.

Syntax

```
<HFMwSystemInfo>.GetKillUserStatus pvbKill, pvbWarn, pvardTimestamp
```

Argument	Description
----------	-------------

<i>pvbKill</i>	Indicates whether the current user has been logged off. Returns TRUE if the user has been logged off, FALSE otherwise.
----------------	--

Input/output argument.

<i>pvbWarn</i>	<i>For internal use.</i>
----------------	--------------------------

<i>pvardTimestamp</i>	Returns the timestamp of when the user was logged off or warned. The return value is a Double that can be cast to the Date format.
-----------------------	--

Input/output argument.

GetRunningTaskLogFromServerFile

Copies a given task’s log file from the application server to the client, using the specified file name and path. After the system copies the log file, it deletes the file from the server.

Syntax

```
<HFMwSystemInfo>.GetRunningTaskLogFromServerFile lTaskID, bstrFileOnClient
```

Argument	Description
----------	-------------

<i>lTaskID</i>	The ID of the task.
----------------	---------------------

Argument	Description
	Tip: You can obtain the IDs of all running tasks with EnumRunningTasks . Input argument. Long subtype.
<i>bstrFileOnClient</i>	The name and path of the log file to create on the client. Input argument. String subtype.

GetRunningTaskProgress

Returns a given running task's percentage complete, status, last update time, and description.

Syntax

```
<HFMwSystemInfo>.GetRunningTaskProgress (lTaskID, pvlStatus,
pvdLastUpdateTime, pvbstrDesc)
```

Argument	Description
<i>lTaskID</i>	The ID of the task. Tip: You can obtain the IDs of all running tasks with EnumRunningTasks . Input argument. Long subtype.
<i>pvlStatus</i>	Returns the status of the task. Valid values are represented by the HFMConstants type library constants listed in "Task Status Constants" on page 452 . Input/output argument.
<i>pvdLastUpdateTime</i>	Returns the timestamp of the last time the task was updated. The timestamp is formatted as a Double that can be cast to the Date format. Input/output argument.
<i>pvbstrDesc</i>	Returns the task's description. Input/output argument.

Return Value

Returns the progress complete percentage of the task.

GetRunningTaskStatus

Returns the status of a given running task.

Syntax

```
<HFMwSystemInfo>.GetRunningTaskStatus (lTaskID)
```

Argument	Description
<i>lTaskID</i>	The ID of the task.

Argument Description

Tip: You can obtain the IDs of all running tasks with [EnumRunningTasks](#).
Input argument. Long subtype.

Return Value

Returns the status of the task. Valid values are represented by the HFMConstants type library constants listed in [“Task Status Constants” on page 452](#).

KillUsers

Logs off users. You can log off all users on all application servers, or log off only given users, application servers, and sessions.

Tip: To log off users from a given server cluster or application, use `HFMwManageApplications.KillUsers`.

Syntax

```
<HFMwSystemInfo>.KillUsers vbAllServers, bstrServer, vbAllUsers, lActivityUserID, vbAllSessions, lActivitySessionID
```

Argument	Description
<i>vbAllServers</i>	<p>A flag that determines whether to log off users from all application servers. Pass TRUE to log users off from all application servers, FALSE to log users off the server specified in the <i>bstrServer</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>bstrServer</i>	<p>The name of the application server for which to log users off. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.</p> <p>Input argument. String subtype.</p>
<i>vbAllUsers</i>	<p>A flag that determines whether to log off all users. Pass TRUE to log off all users, FALSE to log off the user specified in the <i>lActivityUserID</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lActivityUserID</i>	<p>The activity user ID of the user to log off. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>Tip: You can get the activity user ID of all users on the system with the HFMwManageApplications component's EnumUsersOnSystem method.</p> <p>Input argument. Long subtype.</p>
<i>vbAllSessions</i>	<p>A flag that determines whether to log off all user sessions. Pass TRUE to log off all sessions, FALSE to log off the session specified in the <i>lActivitySessionID</i> argument.</p> <p>Input argument. Boolean subtype.</p>
<i>lActivitySessionID</i>	<p>The ID of the session to log off. This argument is used only if the <i>vbAllSessions</i> argument is set to FALSE.</p>

Argument	Description
	Input argument. Long subtype.

SetCurrentModule

Sets the current module using a module name.

Tip: To set the current module with a module ID, use [SetCurrentModuleEx](#).

Syntax

```
<HFMwSystemInfo>.SetCurrentModule bstrModuleName
```

Argument	Description
<i>bstrModuleName</i>	The string to set as the current module. Input argument. String subtype.

SetCurrentModuleEx

Sets the current module using a module ID.

Tip: To set the current module with a module name, use [SetCurrentModule](#).

Syntax

```
<HsvSystemInfo>.SetCurrentModuleEx lModuleNameResource
```

Argument	Description
<i>lModuleNameResource</i>	Long (ByVal). The module ID. Valid values are represented by the HFMConstants enumeration tagMODULEIDS, which is described in “Module ID Constants” on page 468 .

SetWebSession

Associates an HFMwSystemInfo object with the HFMwSession object for an application.

Syntax

```
<HFMwSystemInfo>.SetWebSession varpIDispHFMwSession
```

Argument	Description
<i>varpIDispHFMwSession</i>	The HFMwSession object for the application. Input argument.

Example

SetWebSession is used in the example for [EnumAuditTasks](#).

StopRunningTasks

For internal use.

WarnUsersForShutDown

For internal use.

In This Chapter

HFMwJournal Component	333
HFMwTemplate Component.....	341
HFMwManageJournals Component.....	348
HFMwQueryDef Component.....	370

The HFMwJournals type library exposes Financial Management’s journal features, and contains the following components:

- The HFMwJournal component, which represents journals.
- The HFMwTemplate component, which represents journal templates.
- The HFMwManageJournals component, which processes journals and templates. This component also enables you to open and close periods, to create journals from templates, and to execute queries on journals and templates.
- The HFMwQueryDef component, which is used to define filtering and sorting criteria for journals and templates. This component also enables you to create journal reports based on the specified criteria.

Tip: To edit the data of journals and templates, use the HFMwJournal and HFMwTemplate components. To save or process the journals and templates represented by these components, use the HFMwManageJournals component.

HFMwJournal Component

The HFMwJournal component exposes the following journal functionality:

- Insert data into a journal using the component’s properties.
- Get and set the number of line items in a journal.
- Test whether a journal is a single-entity journal.

For information on this component’s members, see “[HFMwJournal Methods](#)” on page 334 and “[HFMwJournal Properties](#)” on page 336.

Tip: To save a journal to the database, use the HFMwManageJournals method [SaveJournal](#).

Obtaining an HFMwJournal Object Reference

Various HFMwManageJournals methods create HFMwJournal objects, including the following:

- HFMwManageJournals.[CreateJournal](#)
- HFMwManageJournals.[CreateJournalFromTemplate](#)
- HFMwManageJournals.[GetJournal](#)
- HFMwManageJournals.[GetJournalFromXML](#)

HFMwJournal Methods

The HFMwJournal component contains the methods described in the following topics.

GetAsXML

Gets the XML representation of the HFMwJournal object.

Note: You can pass the return value to the HFMwManageJournals method [GetJournalFromXML](#).

Syntax

```
<HFMwJournal>.GetAsXML()
```

Return Value

Returns the XML representation of the HFMwJournal object.

GetNumberOfLineItemEntries

Deprecated - use [GetNumberOfLineItems](#).

GetNumberOfLineItems

Returns the number of line item entries in the journal.

Syntax

```
<HFMwJournal>.GetNumberOfLineItems()
```

Return Value

Returns the count of line items.

IsNewJournal

Indicates whether the HFMwJournal object represents a new journal. In other words, this method indicates whether the journal exists in the database.

Syntax

```
<HFMwJournal>.IsNewJournal()
```

Return Value

Returns TRUE if the object represents a new journal, FALSE otherwise.

IsSingleEntity

Indicates whether the HFMwJournal object represents a single entity journal.

Syntax

```
<HFMwJournal>.IsSingleEntity()
```

Return Value

Returns TRUE if the object represents a single entity journal, FALSE otherwise.

SetNumberOfLineItemEntries

Deprecated - use [SetNumberOfLineItems](#).

SetNumberOfLineItems

Sets the number of line items for the journal. If the journal already contains line items, then the following rules apply:

- If the new number of line items is less than the existing number of line items, then the applicable number of line items are deleted, with the last items deleted first. For example, if the journal has four line items and you pass 2, `SetNumberOfLineItems` deletes the third and fourth line items.
- If the new number of line items is greater than the existing number of line items, the new line items are initialized to the data contained by the last of the existing line items. For example, if the journal has six line items and you pass 8, then `SetNumberOfLineItems` sets the seventh and eighth line items to the data contained by the sixth.

Syntax

```
<HFMwJournal>.SetNumberOfLineItems lNumItems
```

Argument	Description
----------	-------------

<i>lNumItems</i>	The new number of line items.
------------------	-------------------------------

Argument	Description
----------	-------------

	Input argument. Long subtype.
--	-------------------------------

HF MwJournal Properties

The HF MwJournal component contains the properties described in the following topics.

balanceType

Gets or sets the journal's balance type. Valid values are represented by the HF MConstants type library constants listed in [“Balance Type Constants” on page 428](#).

Read-write.

description

Gets or sets the journal's description.

Read-write.

group

Gets or sets the journal's group.

Read-write.

label

Gets or sets the journal's label.

Read-write.

Example

label is used in the example for [CreateJournalFromTemplate](#).

liAccount

Gets or sets the Account dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HF MwJournal>.liAccount(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index.
----------	---

Argument	Description
----------	-------------

	Input argument.
--	-----------------

liAmount

Gets or sets the amount for a given line item. The amount is a string formatted according to the applicable attributes of the line item's Account and Entity dimension members.

Read-write.

Syntax

```
<HFMwJournal>.liAmount(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index.
	Input argument.

liCustom1

Gets or sets the Custom 1 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liCustom1(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index.
	Input argument.

liCustom2

Gets or sets the Custom 2 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liCustom2(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index.
	Input argument.

liCustom3

Gets or sets the Custom 3 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liCustom3(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liCustom4

Gets or sets the Custom 4 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liCustom4(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liDebitCreditUnit

Gets or sets whether a given line item is a debit or credit. Valid values are represented by the HFMConstants type library constants listed in [“Debit/Credit Constants” on page 429](#).

Read-write.

Syntax

```
<HFMwJournal>.liDebitCreditUnit(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liDescription

Gets or sets the description of a given line item.

Read-write.

Syntax

```
<HFMwJournal>.liDescription(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liEntity

Gets or sets the child Entity dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liEntity(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liICP

Gets or sets the Intercompany Partner dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liICP(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liParent

Gets or sets the parent Entity dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwJournal>.liParent(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

oldLabel

Gets the old label of the journal. In other words, if you create the HFMwJournal object from an existing journal and assign a new label with the `label` property, `oldLabel` stores the existing journal's label.

Note: If the HFMwJournal object represents a new journal, `oldLabel` returns an empty string.

Read-only.

period

Gets the label of the journal's Period dimension member.

Read-only.

scenario

Gets the label of the journal's Scenario dimension member.

Read-only.

security

Gets or sets the security class of the journal, using the security class label.

Read-write.

singleEntity

Gets or sets the child Entity dimension member for a single entity journal, using the member's label. If the HFMwJournal object does not represent a single entity journal, this property returns an empty string.

Read-write.

singleParent

Gets or sets the parent Entity dimension member for a single entity journal, using the member's label. If the HFMwJournal object does not represent a single entity journal, this property returns an empty string.

Read-write.

status

Gets the status of the journal. Valid values are represented by the HFMConstants type library constants listed in [“Journal Status Constants” on page 430](#).

Read-only.

type

Gets the journal’s type. Valid values are represented by the HFMConstants type library constants listed in [“Journal Type Constants” on page 430](#).

Read-only.

value

Gets the label of the journal’s Value dimension member.

Read-only.

year

Gets the label of the journal’s Year dimension member.

Read-only.

HFMwTemplate Component

The HFMwTemplate component exposes the following journal template functionality:

- Insert data into a template using the component’s properties.
- Get and set the number of line items in a template.
- Test whether a template is a single-entity template.

For information on this component’s members, see [“HFMwTemplate Methods” on page 342](#) and [“HFMwTemplate Methods” on page 342](#).

Tip: To save a template to the database, use the HFMwManageJournals method [SaveTemplate](#).

Obtaining an HFMwTemplate Object Reference

The following HFMwManageJournals methods create HFMwTemplate objects.

- [CreateTemplate](#)
- [GetTemplate](#)

- [GetTemplateFromXML](#)

HF MwTemplate Methods

The HF MwTemplate component contains the methods described in the following topics.

CopyDataToJournal

For internal use.

CreateJournal

For internal use.

GetAsXML

Returns an XML representation of the HF MwTemplate object.

Note: You can pass the return value to the HF MwManageJournals method [GetTemplateFromXML](#).

Syntax

```
<HF MwTemplate>.GetAsXML( )
```

Return Value

Returns an XML string that represents the template.

GetNumberOfLineItems

Returns the number of line items in the template.

Syntax

```
<HF MwTemplate>.GetNumberOfLineItems( )
```

Return Value

Returns the count of line items.

IsSingleEntity

Indicates whether the HF MwTemplate object represents a single entity journal.

Syntax

```
<HF MwTemplate>.IsSingleEntity( )
```

Return Value

Returns TRUE if the object represents a single entity journal, FALSE otherwise.

SetNumberOfLineItems

Sets the number of line items in the template.

Note: If the new number of line items is greater than the existing number of line items, the new line items are initialized to the values in the last of the existing line items.

Syntax

```
<HFMwTemplate>.SetNumberOfLineItems(i)
```

Argument	Description
----------	-------------

<i>i</i>	The new number of line items. Input argument. Long subtype.
----------	--

HFMwTemplate Properties

The HFMwTemplate component contains the properties described in the following topics.

balanceType

Gets or sets the template's balance type. Valid values are represented by the HFMConstants type library constants listed in [“Balance Type Constants” on page 428](#).

Read-write.

description

Gets or sets the template's description.

Read-write.

group

Gets or sets the template's group.

Read-write.

label

Gets or sets the template's label.

Read-write.

liAccount

Gets or sets the Account dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liAccount(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liAmount

Gets or sets the amount for a given line item.

Read-write.

Syntax

```
<HFMwTemplate>.liAmount(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liCustom1

Gets or sets the Custom 1 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liCustom1(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liCustom2

Gets or sets the Custom 2 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liCustom2(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liCustom3

Gets or sets the Custom 3 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liCustom3(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liCustom4

Gets or sets the Custom 4 dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liCustom4(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liDebitCreditUnit

Gets or sets whether a given line item is a debit or credit. Valid values are represented by the HFMConstants type library constants listed in [“Debit/Credit Constants” on page 429](#).

Read-write.

Syntax

```
<HFMwTemplate>.liDebitCreditUnit(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liDescription

Gets or sets the description of a given line item.

Read-write.

Syntax

```
<HFMwTemplate>.liDescription(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liEntity

Gets or sets the child Entity dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liEntity(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

liICP

Gets or sets the Intercompany Partner dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liICP(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index.
----------	---

Argument	Description
----------	-------------

	Input argument.
--	-----------------

liParent

Gets or sets the parent Entity dimension member for a given line item, using the member's label.

Read-write.

Syntax

```
<HFMwTemplate>.liParent(i)
```

Argument	Description
----------	-------------

<i>i</i>	The number of the line item. This is a 0-based index. Input argument.
----------	--

oldLabel

Gets the old label of the template. In other words, if you create the HFMwTemplate object from an existing template and assign a new label with the [label](#) property, `oldLabel` stores the existing template's label.

Note: If the HFMwTemplate object represents a new template, `oldLabel` returns an empty string.

Read-only.

security

Gets or sets the security class of the template, using the security class label.

Read-write.

singleEntity

Gets or sets the child Entity dimension member for a single entity template, using the member's label. If the HFMwTemplate object does not represent a single entity template, this property returns an empty string.

Read-write.

singleParent

Gets or sets the parent Entity dimension member for a single entity template, using the member's label. If the HFMwTemplate object does not represent a single entity template, this property returns an empty string.

Read-write.

templateType

Gets the template's type. Valid values are represented by the HFMConstants type library constants listed in [“Template Type Constants” on page 432](#).

Read-only.

type

For internal use.

valueDimension

Gets or sets a recurring template's Value dimension member, using the member's label.

Caution! This property applies only to recurring templates.

Read-write.

HFMwManageJournals Component

The HFMwManageJournals component exposes various journal features, including those listed below:

- Process journals by submitting them, posting them, and so on.
- Create journals from templates.
- Create HFMwJournal and HFMwTemplate object references.
- Open and close periods.
- Delete journals and templates.
- Generate journals from recurring templates.
- Validate journal line items.
- Execute queries on and return data for journals and templates that meet the specified criteria.
- Add, get, and remove journal groups.

Obtaining an HFMwManageJournals Object Reference

Create an HFMwManageJournals object with `Server.CreateObject`. After creating the object, call `SetWebSession` to associate the object with the HFMwSession object that represents the application:

```
Set cHFMMwManageJournals = Server.CreateObject _  
    ("Hyperion.HFMwManageJournals")  
'cHFMSession is an HFMwSession object reference  
cHFMMwManageJournals.SetWebSession(cHFMSession)
```

HFMwManageJournals Methods

The HFMwManageJournals component contains the methods described in the following topics.

AddJournalGroup

Adds a journal group to the application.

Caution! The user must be assigned to the Journals Administrator role to successfully execute this operation.

Syntax

```
<HFMwManageJournals>.AddJournalGroup bstrGroup, bstrDescription
```

Argument	Description
<i>bstrGroup</i>	The name of the group to be added. Input argument. String subtype.
<i>bstrDescription</i>	A description of the group. Input argument. String subtype.

ClosePeriodMultiple

Closes a given set of periods for the specified year and scenario.

Syntax

```
<HFMwManageJournals>.ClosePeriodMultiple (bstrScenario, bstrYear,  
vaPeriods)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.

Argument	Description
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>vaPeriods</i>	An array containing the labels of the Period dimension members to close. Input argument.

Return Value

Returns an array of HRESULTS that indicate whether the periods were successfully closed. The array has a one-to-one correspondence with the array passed to the *vaPeriods* argument. A value of 0 means that the corresponding period was closed, a non-zero error number indicates the period was not closed.

CreateJournal

Creates an HFMwJournal object, applying the specified label, journal type, and Point of View to the journal.

Tip: To save the journal, pass the HFMwJournal object returned by `CreateJournal` to [SaveJournal](#).

Syntax

```
<HFMwManageJournals>.CreateJournal (bstrLabel, lType, bstrScenario,
bstrYear, bstrPeriod, bstrValue)
```

Argument	Description
<i>bstrLabel</i>	The label for the journal. Input argument. String subtype.
<i>lType</i>	The journal type. Valid values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 430 . Input argument. Long subtype.
<i>bstrScenario</i>	The label of the journal’s Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal’s Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal’s Period dimension member. Input argument. String subtype.
<i>bstrValue</i>	The label of the journal’s Value dimension member. Input argument. String subtype.

Return Value

Returns an HFMwJournal object reference.

CreateJournalFromTemplate

Creates an HFMwJournal object from a given template, initializing the journal with the template's data.

Tip: To save the journal, pass the HFMwJournal object returned by `CreateJournalFromTemplate` to [SaveJournal](#).

Syntax

```
<HFMwManageJournals>.CreateJournalFromTemplate (bstrScenario, bstrYear,  
bstrPeriod, bstrValue, varbCreateAsAutoReversing, bstrLabel)
```

Argument	Description
<i>bstrScenario</i>	The label of the journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal's Period dimension member. Input argument. String subtype.
<i>bstrValue</i>	The label of the journal's Value dimension member. Input argument. String subtype.
<i>varbCreateAsAutoReversing</i>	A flag that determines whether the journal is auto-reversing. Pass TRUE to create an auto-reversing journal, FALSE otherwise. Input argument. Boolean subtype.
<i>bstrLabel</i>	The label of the template. If the label is invalid, an error is returned. Input argument. String subtype.

Return Value

Returns an HFMwJournal object reference.

Example

The following subroutine creates and saves a new journal. `CreateJournalFromTemplate` creates an HFMwJournal object, using the template, Point of View, and auto-reversing flag passed to the subroutine. `HFMwJournal.Label` sets the journal label, using the label passed to the function. `SaveJournal` then saves the journal.

```
Sub AddNewJournal(sScen, sYear, sPer, sVal, bAutoRev, _
```

```

        sTemplate, sLabel)
Dim cHFMMManageJournals, cHFMJJournal
Set cHFMMManageJournals = Server.CreateObject _
    ("Hyperion.HFMwManageJournals")
'g_chFMSession is an HFMwSession object reference
cHFMMManageJournals.SetWebSession(g_chFMSession)
Set cHFMJJournal = cHFMMManageJournals.CreateJournalFromTemplate _
    (sScen, sYear, sPer, sVal, bAutoRev, sTemplate)
cHFMJJournal.label = sLabel
cHFMMManageJournals.SaveJournal cHFMJJournal
End Sub

```

CreateTemplate

Creates an HFMwTemplate object, initializing it with the specified label and template type.

Syntax

```
<HFMwManageJournals>.CreateTemplate (lType, lTemplateType, varbstrLabel)
```

Argument	Description
<i>lType</i>	<p>The journal type for the template. In this release, the method only supports templates for regular journals, and so the only valid value is the JTF_REGULAR constant.</p> <p>Note: The JTF_REGULAR constant is one of the HFMConstants type library constants listed in “Journal Type Constants” on page 430.</p> <p>Input argument. Long subtype.</p>
<i>lTemplateType</i>	<p>The template type. Valid values are represented by the HFMConstants type library constants listed in “Template Type Constants” on page 432.</p> <p>Input argument. Long subtype.</p>
<i>varbstrLabel</i>	<p>Label of the template.</p> <p>Input argument. String subtype.</p>

Return Value

Returns an HFMwTemplate object reference.

DeleteJournal

Deletes the specified journal.

Syntax

```
<HFMwManageJournals>.DeleteJournal bstrLabel, bstrScenario, bstrYear,
bstrPeriod
```

Argument	Description
<i>bstrLabel</i>	The label of the journal to be deleted.

Argument	Description
	Input argument. String subtype.
<i>bstrScenario</i>	The label of the journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal's Period dimension member. Input argument. String subtype.

DeleteJournals

Deletes the specified journals for a given scenario, year, and period.

Syntax

```
<HFMwManageJournals>.DeleteJournals (varabstrLabels, bstrScenario,
bstrYear, bstrPeriod)
```

Argument	Description
<i>varabstrLabels</i>	An array containing the labels of the journals to be deleted. Input argument.
<i>bstrScenario</i>	The label of the journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal's Period dimension member. Input argument. String subtype.

Return Value

Returns an array of HRESULTS that indicate whether the journals were successfully deleted. The array has a one-to-one correspondence with the array passed to the *varabstrLabels* argument. A value of 0 means that the corresponding journal was deleted, a non-zero error number indicates the journal was not deleted.

DeleteTemplate

Deletes the specified template.

Syntax

```
<HFMwManageJournals>.DeleteTemplate bstrLabel
```

Argument	Description
----------	-------------

<i>bstrLabel</i>	The label of the template to delete. Input argument. String subtype.
------------------	---

DeleteTemplates

Deletes the specified templates.

Caution!	The user must be assigned to the Manage Templates role in order for this method to successfully execute.
-----------------	--

Syntax

```
<HFMwManageJournals>.DeleteTemplates (varabstrLabels)
```

Argument	Description
----------	-------------

<i>varabstrLabels</i>	An array containing the labels of the templates to be deleted. Input argument.
-----------------------	---

Return Value

Returns an array of HRESULTS that indicate whether the templates were successfully deleted. The array has a one-to-one correspondence with the array passed to the *varabstrLabels* argument. A value of 0 means that the corresponding template was deleted, a non-zero error number indicates the template was not deleted.

EnumJournalGroups

Enumerates all the journal groups available for the application.

Caution!	The user must be assigned to the Read Journals role in order for this method to successfully execute.
-----------------	---

Syntax

```
<HFMwManageJournals>.EnumJournalGroups pvarabstrGroups,  
pvarabstrDescriptions
```

Argument	Description
----------	-------------

<i>pvarabstrGroups</i>	Returns an array containing the labels of the journal groups. Input/output argument.
------------------------	---

<i>pvarabstrDescriptions</i>	Returns an array containing the descriptions of the journal groups.
------------------------------	---

Argument	Description
	Input/output argument.

ExecuteQuery

Returns the labels of the journals or templates that match the specified columns, sort options, and filtering criteria.

To display the journals or template that meet the criteria, pass the labels returned by this method to [GetJournalsDisplayData](#) or [GetTemplatesDisplayData](#).

Syntax

```
<HFMwManageJournals>.ExecuteQuery (lType, bstrScenario, bstrYear,
bstrPeriod, bstrValue, varalColumns, varalSortOptions, varavFilters)
```

Argument	Description
<i>lType</i>	<p>A flag that determines whether to return labels of journals or templates. Pass 0 for journals, 1 for templates.</p> <p>Input argument. Long subtype.</p>
<i>bstrScenario</i>	<p>The label of the journals' Scenario dimension member.</p> <p>Note: This argument is ignored for templates.</p> <p>Input argument. String subtype.</p>
<i>bstrYear</i>	<p>The label of the journals' Year dimension member.</p> <p>Note: This argument is ignored for templates.</p> <p>Input argument. String subtype.</p>
<i>bstrPeriod</i>	<p>The label of the journals' Period dimension member.</p> <p>Note: This argument is ignored for templates.</p> <p>Input argument. String subtype.</p>
<i>bstrValue</i>	<p>The label of the Value dimension member for the journals or templates.</p> <p>Input argument. String subtype.</p>
<i>varalColumns</i>	<p>The journal or template columns for which to return data. Get these columns from <code>HFMwQueryDef.GetDefAsVariants</code>.</p> <p>Input argument.</p>
<i>varalSortOptions</i>	<p>The sort options for the journals or templates. Get these sort options from <code>HFMwQueryDef.GetDefAsVariants</code>.</p> <p>Input argument.</p>
<i>varavFilters</i>	<p>The filters for the journals or templates. Get these filters from <code>HFMwQueryDef.GetDefAsVariants</code>.</p> <p>Input argument.</p>

Return Value

Returns the labels of the journals or templates that match the specified criteria.

Example

ExecuteQuery is used in the example for [GetJournalsDisplayData](#).

ExecuteQueryLabelsAndIDs

Returns the labels and IDs of the journals or templates that match the specified columns, sort options, and filtering criteria. To display the journals or template that meet the criteria, pass the labels returned by this method to [GetJournalsDisplayData](#) or [GetTemplatesDisplayData](#). To display the journals or template that meet the criteria, pass the IDs returned by this method to [GetJournalsDisplayDataFromIDs](#) or [GetTemplatesDisplayDataFromIDs](#).

Syntax

```
<HFMwManageJournals>.ExecuteQueryLabelsAndIDs lType, bstrScenario,  
bstrYear, bstrPeriod, bstrValue, varalColumns, varalSortOptions,  
varavFilters, pvaralIDs, pvarasLabels
```

Argument	Description
<i>lType</i>	A flag that determines whether to return labels of journals or templates. Pass 0 for journals, 1 for templates. Long subtype.
<i>bstrScenario</i>	The label of the journals' Scenario dimension member. Note: This argument is ignored for templates. Input argument. String subtype.
<i>bstrYear</i>	The label of the journals' Year dimension member. Note: This argument is ignored for templates. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journals' Period dimension member. Note: This argument is ignored for templates. Input argument. String subtype.
<i>bstrValue</i>	The label of the Value dimension member for the journals or templates. Input argument. String subtype.
<i>varalColumns</i>	The journal or template columns for which to return data. Get these columns from HFMwQueryDef.GetDefAsVariants . Input argument.
<i>varalSortOptions</i>	The sort options for the journals or templates. Get these sort options from HFMwQueryDef.GetDefAsVariants .

Argument	Description
	Input argument.
<i>varavFilters</i>	The filters for the journals or templates. Get these filters from <code>HFMwQueryDef.GetDefAsVariants</code> . Input argument.
<i>pvarallDs</i>	Array Long subtype of journal IDs returned. Input/output argument.
<i>pvarasLabels</i>	Array of String Subtype of journals labels returned. Input/output argument.

GenerateJournalFromRecurringTemplate

Generates a journal from a given recurring template.

Syntax

```
<HFMwManageJournals>.GenerateJournalFromRecurringTemplate bstrScenario,
bstrYear, bstrPeriod, bstrLabel
```

Argument	Description
<i>bstrScenario</i>	The label of the journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal's Period dimension member. Input argument. String subtype.
<i>bstrLabel</i>	The label of the recurring template. Input argument. String subtype.

GetJournal

Creates an HFMwJournal object that represents a given journal.

Caution! If the journal specified in the arguments does not exist, an error is thrown.

Syntax

```
<HFMwManageJournals>.GetJournal (bstrLabel, bstrScenario, bstrYear,
bstrPeriod)
```

Argument	Description
<i>bstrLabel</i>	The label of the journal. Input argument. String subtype.
<i>bstrScenario</i>	The journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The journal's Period dimension member. Input argument. String subtype.

Return Value

Returns an HFMwJournal object reference.

Example

GetJournal is used in the example for [ValidateLineItems](#).

GetJournalFromXML

Creates an HFMwJournal object from an XML string.

Tip: You can use this method to create a new HFMwJournal object from an existing HFMwJournal object.

Syntax

```
<HFMwManageJournals>.GetJournalFromXML (bstrXML)
```

Argument Description

<i>bstrXML</i>	The XML string. Get this string from the HFMwJournal method GetAsXML . Input argument. String subtype.
----------------	---

Return Value

Returns an HFMwJournal object reference.

GetJournalsDisplayData

Gets the data for the specified journals. GetJournalsDisplayData returns data only for the columns specified in the *varalColumns* argument.

Syntax

```
<HFMMwManageJournals>.GetJournalsDisplayData (varasLabels, bstrScenario,  
bstrYear, bstrPeriod, varalColumns)
```

Argument	Description
----------	-------------

<i>varasLabels</i>	An array containing the labels of the journals for which to return data.
--------------------	--

Tip: You can get the labels of all journals that match given criteria with [ExecuteQuery](#).
Input argument.

<i>bstrScenario</i>	The label of the journals' Scenario dimension member. Input argument. String subtype.
---------------------	--

<i>bstrYear</i>	The label of the journals' Year dimension member. Input argument. String subtype.
-----------------	--

<i>bstrPeriod</i>	The label of the journals' Period dimension member. Input argument. String subtype.
-------------------	--

<i>varalColumns</i>	An array of that identifies the columns for which to return data. To set this argument, use either of the following techniques:
---------------------	---

- Pass the columns returned by the HFMwQueryDef method [GetDefAsVariants](#).
- Construct the array by setting the array items to the desired columns. Columns are represented by the HFMConstants type library constants listed in ["Journal Column Display Constants" on page 428](#).

Input argument. Array items are of a Long subtype.

Return Value

Returns an array of arrays containing data for the specified journals and columns. There is one array element for each journal specified by the *varasLabels* argument; each element then contains an array of journal data, with the size of that array corresponding to the columns specified by the *varalColumns* argument.

Note: Some journal data, such as journal type and status, are represented by HFMConstants type library constants. For more information, see ["Journal-Related Constants" on page 427](#).

Example

The following function returns the labels, statuses, and types of the journals for the specified Point of View, with the journals sorted by label. The columns, filtering, and sorting options are set with various HFMwQueryDef properties; note that the For . . . Next loops set the filters so that journals of all statuses, types, and balance types are returned.

```
Function GetJournalsForPOV(sScen, sYear, sPer, sVal)  
Dim cHFMMwManageJournals, cHFMMwQueryDef  
Dim vaCols, vaSortOpts, vaFilters, vaLabels, vaRet  
Set cHFMMwManageJournals = Server.CreateObject _
```

```

        ("Hyperion.HFMwManageJournals")
'g_chFMSession is an HFMwSession object reference
cHFMMManageJournals.SetWebSession(g_chFMSession)
Set cHFMQueryDef = Server.CreateObject("Hyperion.HFMwQueryDef")
cHFMQueryDef.ResetAll
cHFMQueryDef.columnContents(0) = COLUMN_JOURNALLABEL
cHFMQueryDef.columnSortOption(0) = JOURNALREPORT_SORT_ASCENDING
cHFMQueryDef.columnContents(1) = COLUMN_JOURNALSTATUS
cHFMQueryDef.columnContents(2) = COLUMN_JOURNALTYPE
For i = 0 to WEBOM_JOURNAL_NUM_STATUS_FILTER - 1
    cHFMQueryDef.statusFilter(i) = TRUE
Next
For i = 0 to WEBOM_JOURNAL_NUM_TYPE_FILTER - 1
    cHFMQueryDef.typeFilter(i) = TRUE
Next
For i = 0 to WEBOM_JOURNAL_NUM_BALANCE_TYPE_FILTER - 1
    cHFMQueryDef.balanceTypeFilter(i) = TRUE
Next
cHFMQueryDef.GetDefAsVariants 0, vaCols, vaSortOpts, vaFilters
vaLabels= cHFMMManageJournals.ExecuteQuery (0, _
    sScen, sYear, sPer, sVal, vaCols, vaSortOpts, vaFilters)
vaRet = cHFMMManageJournals.GetJournalsDisplayData(vaLabels, _
    sScen, sYear, sPer, vaCols)
GetJournalsForPOV= vaRet
End Function

```

GetJournalsDisplayDataFromIDs

Retrieves the data for the specified journals. Returns data only for the columns specified in the `varalColumns` argument.

Syntax

```

<HFMwManageJournals>.GetJournalsDisplayDataFromIDs(varalIDs, lScenario,
lYear, lPeriod, varalColumns)

```

Argument	Description
<i>varalIDs</i>	<p>An array containing the IDs of the journals for which to return data.</p> <p>Tip: You can get the IDs of all journals that match given criteria with <code>ExecuteQueryLabelsAndIDs</code>.</p> <p>Input argument.</p>
<i>lScenario</i>	<p>The member ID of the cell's Scenario dimension member.</p> <p>Input argument. Long subtype.</p>
<i>lYear</i>	<p>The member ID of the cell's Year dimension member.</p> <p>Input argument. Long subtype.</p>
<i>lPeriod</i>	<p>The member ID of the cell's Period dimension member.</p> <p>Input argument. Long subtype.</p>

Argument	Description
----------	-------------

<i>varalColumns</i>	An array of that identifies the columns for which to return data. To set this argument, use either of the following techniques:
---------------------	---

- Pass the columns returned by the HFMwQueryDef method [GetDefAsVariants](#).
- Construct the array by setting the array items to the desired columns. Columns are represented by the HFMConstants type library constants listed in [“Journal Column Display Constants” on page 428](#).

Input argument. Array items are of a Long subtype.

Return Value

GetJournalUsingIDs

Creates an HFMwJournal object that represents a given journal.

Caution! If the journal specified in the arguments does not exist, an error is thrown.

Syntax

```
<HFMwManageJournals>.GetJournalUsingIDs(lJourID, bstrScenario, bstrYear, bstrPeriod)
```

Argument	Description
----------	-------------

<i>lJourID</i>	The ID of the journal.
----------------	------------------------

Input argument. Long subtype.

<i>bstrScenario</i>	The journal's Scenario dimension member.
---------------------	--

Input argument. String subtype.

<i>bstrYear</i>	The journal's Year dimension member.
-----------------	--------------------------------------

Input argument. String subtype.

<i>bstrPeriod</i>	The journal's Period dimension member.
-------------------	--

Input argument. String subtype.

Return Value

GetPeriodStatusMultiple

Return an array containing the statuses of a given set of periods for the specified scenario and year.

Syntax

```
<HFMwManageJournals>.GetPeriodStatusMultiple (bstrScenario, bstrYear, vaPeriods)
```

Argument	Description
----------	-------------

<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>vaPeriods</i>	An array containing the labels of the Period dimension members. Input argument.

Return Value

Returns an array of the periods' statuses. This array has a one-to-one correspondence with the array passed to the *vaPeriods* argument. Valid values are represented by the HFMConstants type library constants listed in [“Period Status Constants” on page 429](#).

Example

The following function returns descriptions of the specified periods' statuses.

```
Function getPerStatusStrings(sScen, sYear, vaPers)
Dim vaRet
vaRet = g_cHFMwManageJournals.GetPeriodStatusMultiple(sScen, _
    sYear, vaPers)
For i = lBound(vaRet) to uBound(vaRet)
    Select Case vaRet(i)
        Case JPS_CLOSED
            vaRet(i) = "Closed"
        Case JPS_OPENED
            vaRet(i) = "Opened"
        Case JPS_UNOPENED
            vaRet(i) = "Unopened"
    End Select
Next
getPerStatusStrings = vaRet
End Function
```

GetTemplate

Creates an HFMwTemplate object that represents a given template.

Caution!	The user must be assigned to the Manage Templates role in order for this method to successfully execute.
-----------------	--

Syntax

```
<HFMwManageJournals>.GetTemplate (bstrLabel)
```

Argument	Description
<i>bstrLabel</i>	The label of the journal template. Input argument. String subtype.

Return Value

Returns an HFMwTemplate object reference.

GetTemplateFromXML

Creates an HFMwTemplate object from an XML string.

Tip: You can use this method to create a new HFMwTemplate object from an existing HFMwTemplate object.

Syntax

```
<HFMwManageJournals>.GetTemplateFromXML (bstrXML)
```

Argument	Description
<i>bstrXML</i>	The XML string. Get this string from the HFMwTemplate method GetAsXML . Input argument. String subtype.

Return Value

Returns an HFMwTemplate object reference.

GetTemplatesDisplayData

Gets the data for the specified templates. `GetTemplatesDisplayData` returns data only for the columns specified in the *varalColumns* argument.

Syntax

```
<HFMwManageJournals>.GetTemplatesDisplayData (varabstrLabels, varalColumns)
```

Argument	Description
<i>varabstrLabels</i>	An array containing the labels of the templates for which to return data. Tip: You can get the labels of all templates that match given criteria with ExecuteQuery . Input argument.

Argument	Description
<i>varalColumns</i>	<p>An array of that identifies the columns for which to return data. To set this argument, use either of the following techniques:</p> <ul style="list-style-type: none"> ● Pass the columns returned by the HFMwQueryDef method GetDefAsVariants. ● Construct the array by setting the array items to the desired columns. Columns are represented by the HFMConstants type library constants listed in “Template Column Display Constants” on page 431. <p>Input argument.</p>

Return Value

Returns an array of arrays containing data for the specified templates and columns. There is one array element for each template specified with the *varabstrLabels* argument; each element then contains an array of template data, with the size of that array corresponding to the columns specified with the *varalColumns* argument.

Note: Some template data, such as template type and balance type, are represented by HFMConstants type library constants. For more information, see [“Journal-Related Constants” on page 427](#).

Example

The following function returns the descriptions, groups, and template types of the specified templates. The template type longs are translated to strings using the HFMConstants type library constants listed in [“Template Type Constants” on page 432](#).

```
Function getTemplatesDescGroupType(vaLabels)
'g_chFMManageJournals is a global HFMwManageJournals object
Dim vaCols(2), vaRet
vaCols(0) = COLUMN_TEMPLATEDESCRIPTION
vaCols(1) = COLUMN_TEMPLATEGROUP
vaCols(2) = COLUMN_TEMPLATETYPE
vaRet = g_chFMManageJournals.GetTemplatesDisplayData(vaLabels, _
    vaCols)
'Translate template type longs to strings
For i = lBound(vaRet) to uBound(vaRet)
    Select Case vaRet(i,2)
        Case TTF_RECURRING
            vaRet(i,2) = "Recurring"
        Case TTF_STANDARD
            vaRet(i,2) = "Standard"
    End Select
Next
getTemplatesDescGroupType = vaRet
End Function
```

GetTemplatesDisplayDataFromIDs

Retrieves the data for the specified templates. Returns data only for the columns specified in the `varalColumns` argument.

Syntax

```
<HFMwManageJournals>.GetTemplatesDisplayDataFromIDs(varalIDs, varalColumns)
```

Argument	Description
----------	-------------

<i>varalIDs</i>	An array containing the IDs of the templates for which to return data.
-----------------	--

Tip: You can get the IDs of all journals that match given criteria with `ExecuteQueryLabelsAndIDs`.

Input argument.

<i>varalColumns</i>	An array of that identifies the columns for which to return data. To set this argument, use either of the following techniques:
---------------------	---

- Pass the columns returned by the HFMwQueryDef method [GetDefAsVariants](#).
- Construct the array by setting the array items to the desired columns. Columns are represented by the HFMConstants type library constants listed in [“Template Column Display Constants” on page 431](#).

Input argument. Array items are of a Long subtype

Return Value

GetVariance

Calculates the variance, credit, and debit totals for the specified journal. The totals are returned as formatted strings; the formatting is determined by the applicable attributes of the journal's Entity and Account dimensions.

Syntax

```
<HFMwManageJournals>.GetVariance piUnkHFMwJournal, pvarbstrTotalDebits,  
pvarabstrTotalCredits, pvarbstrTotalAmount
```

Argument	Description
----------	-------------

<i>piUnkHFMwJournal</i>	The HFMwJournal object reference that represents the journal.
-------------------------	---

Input argument.

<i>pvarbstrTotalDebits</i>	Returns a string that contains the journal's total debits.
----------------------------	--

Input/output argument.

<i>pvarabstrTotalCredits</i>	Returns a string that contains the journal's total credits.
------------------------------	---

Input/output argument.

<i>pvarbstrTotalAmount</i>	Returns a string that contains the journal's variance.
----------------------------	--

Argument	Description
	Note: The variance is the absolute difference between the total debits and credits. Input/output argument.

GetVarianceForTemplate

Calculates the variance, credit, and debit totals for the specified template. The totals are returned as formatted strings; the formatting is determined by the applicable attributes of the template's Entity and Account dimensions.

Syntax

```
<HFMwManageJournals>.GetVarianceForTemplate pIUnkHFMwTemplate,  
pvarbstrTotalDebits, pvarbstrTotalCredits, pvarbstrTotalVariance
```

Argument	Description
<i>pIUnkHFMwTemplate</i>	The HFMwTemplate object reference that represents the journal. Input argument.
<i>pvarbstrTotalDebits</i>	Returns a string that contains the template's total debits. Input/output argument.
<i>pvarbstrTotalCredits</i>	Returns a string that contains the template's total credits. Input/output argument.
<i>pvarbstrTotalVariance</i>	Returns a string that contains the template's variance. Note: The variance is the absolute difference between the total debits and credits. Input/output argument.

OpenPeriodMultiple

Opens one or more periods for the specified scenario and year.

Syntax

```
<HFMwManageJournals>.OpenPeriodMultiple (bstrScenario, bstrYear, vaPeriods)
```

Argument	Description
<i>bstrScenario</i>	The label of the Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the Year dimension member. Input argument. String subtype.
<i>vaPeriods</i>	An array containing the labels of the Period dimension members.

Argument	Description
	Input argument.

Return Value

Returns an array of HRESULTS that indicate whether the periods were successfully opened. The array has a one-to-one correspondence with the array passed to the *vaPeriods* argument. A value of 0 means that the corresponding period was opened, a non-zero error number indicates the period was not opened.

PerformBatchAction

Performs the specified action on a given set of journals. For example, you can approve or post journals with this method.

Syntax

```
<HFMwManageJournals>.PerformBatchAction (lAction, varabstrJournalLabels,
bstrScenario, bstrYear, bstrPeriod)
```

Argument	Description
<i>lAction</i>	A flag that identifies the action to be performed. Valid values are represented by the HFMConstants type library constants listed in “Journal Action Constants” on page 427 , Input argument. Long subtype.
<i>varabstrJournalLabels</i>	An array containing labels of the journals on which to perform the action. Input argument.
<i>bstrScenario</i>	The label of the journal's Scenario dimension member. Input argument. String subtype.
<i>bstrYear</i>	The label of the journal's Year dimension member. Input argument. String subtype.
<i>bstrPeriod</i>	The label of the journal's Period dimension member. Input argument. String subtype.

Return Value

Returns an array of HRESULTS that indicate whether the action was successfully performed for the journals. The array has a one-to-one correspondence with the array passed to the *varabstrJournalLabels* argument. A value of 0 means that the action succeeded for the corresponding journal, a non-zero error number indicates the action failed.

RemoveAllJournalGroups

Removes all journal groups from the application. The connected user must be assigned to the Journals Administrator role for this method to successfully execute.

Syntax

```
<HFMwManageJournals>.RemoveAllJournalGroups
```

RemoveJournalGroup

Removes the specified journal group from the application. The connected user must be assigned to the Journals Administrator role for this method to successfully execute.

Caution! If the group does not exist, or if the group is used in a journal, an error is thrown.

Syntax

```
<HFMwManageJournals>.RemoveJournalGroup bstrGroup
```

Argument	Description
----------	-------------

<i>bstrGroup</i>	The name of the journal group to delete. Input argument. String subtype.
------------------	---

SaveJournal

Saves the journal represented by a given HFMwJournal object.

Syntax

```
<HFMwManageJournals>.SaveJournal varpIUnkHFMwJournal
```

Argument	Description
----------	-------------

<i>varpIUnkHFMwJournal</i>	The HFMwJournal object reference for the journal to be saved. Input argument.
----------------------------	--

SaveJournal2

Saves the journal represented by a given HFMwJournal object. Returns the ID of the saved Journal.

Syntax

```
<HFMwManageJournals>.SaveJournal2 ( varpIUnkHFMwJournal )
```


Argument	Description
<i>varpIUnkHFMwJournal</i>	The HFMwJournal object reference for the journal to be saved. Input argument.

Return Value

Long subtype.

SaveTemplate

Saves the template represented by a given HFMwTemplate object.

Syntax

```
<HFMwManageJournals>.SaveTemplate varpIUnkHFMwTemplate
```

Argument	Description
<i>varpIUnkHFMwTemplate</i>	The HFMwTemplate object reference for the template to be saved. Input argument.

SetWebSession

Associates an HFMwManageJournals object with the HFMwSession object for an application.

Syntax

```
<HFMwManageJournals>.SetWebSession varpIDispHFMwSession
```

Argument	Description
<i>varpIDispHFMwSession</i>	The HFMwSession object for the application. Input argument.

ValidateLineItems

Validates a given journal's line items.

Syntax

```
<HFMwManageJournals>.ValidateLineItems (vardispHFMwJournal)
```

Argument	Description
<i>vardispHFMwJournal</i>	The HFMwJournal object reference for the journal to be validated Input argument.

Return Value

Returns an array containing the Financial Management error numbers for the line items, with one array item for each line item. If a line item is valid, the corresponding item in this array has a value of 0, otherwise the array item has a non-zero error number.

Note: If the journal header contains invalid data, `ValidateLineItems` throws an error.

Example

The following function validates the specified journal's line items. `GetJournal` creates an `HFMwJournal` object reference for the journal. This object reference is passed to `ValidateLineItems`, and the return value is assigned as the example function's return value.

```
Function validateJournal(sJournal, sScen, sYear, sPer)
Dim cHFMJournal, vaRet
'g_cHFMManageJournals is a global HFMwManageJournals object
Set cHFMJournal = g_cHFMManageJournals.GetJournal (sJournal, _
    sScen, sYear, sPer)
vaRet = g_cHFMManageJournals.ValidateLineItems(cHFMJournal)
validateJournal = vaRet
End Function
```

HFMwQueryDef Component

The `HFMwQueryDef` component provides properties and methods that set journal and template column, filtering, and sorting criteria. You set these criteria with the component's properties and methods, then return the criteria with `GetDefAsVariants`. You then find the journals or templates that meet the criteria by passing the `GetDefAsVariants` return values to the `HFMwManageJournals` method `ExecuteQuery`.

In addition, you can create a journal report based upon an `HFMwQueryDef` instance's properties. To create a journal report, take the following steps.

- To create a journal report:
 - 1 Set the `HFMwQueryDef` properties for a report's label, Point of View, and security class.
 - 2 Call `GetReportDefinition` to get a string that represents the `HFMwQueryDef` instance's properties.
 - 3 Pass the string to one of the `HFMwManageDocuments` methods that save documents.

For information on the component's members, see "[HFMwQueryDef Methods](#)" on page 371 and "[HFMwQueryDef Properties](#)" on page 372.

Obtaining an HFMwQueryDef Object Reference

Create an `HFMwQueryDef` object with `Server.CreateObject`.

HFMwQueryDef Methods

The HFMwQueryDef component contains the methods described in the following topics.

GetDefAsVariants

Returns the columns, sort options, and filters that have been set for the HFMwQueryDef object reference. You can pass these return values to HFMwManageJournals.[ExecuteQuery](#).

Tip: To set the columns, sort options, and filters, use the properties listed in “[HFMwQueryDef Properties](#)” on page 372.

Syntax

```
<HFMwQueryDef>.GetDefAsVariants lType, pvaralColumns, pvaralSortOptions,  
pvaravFilters
```

Argument	Description
<i>lType</i>	A flag that determines whether the operation is for journals or templates. Pass 0 for journals, 1 for templates. Input argument. Long subtype.
<i>pvaralColumns</i>	The columns that have been set for the HFMwQueryDef object. Input/output argument.
<i>pvaralSortOptions</i>	The sort options that have been set for the HFMwQueryDef object. Input/output argument.
<i>pvaravFilters</i>	The filters that have been set for the HFMwQueryDef object. Input/output argument.

GetReportDefinition

Returns a string that represents the properties of the HFMwQueryDef instance, with the string formatted in report syntax. You can create a journal report by passing the string to the HFMwManageDocuments methods [SaveDocument](#) or [SaveDocumentEx](#).

Syntax

```
<HFMwQueryDef>.GetReportDefinition()
```

Return Value

Returns the string that represents the HFMwQueryDef instance’s properties in report syntax.

InitializeFromReportDefinition

For internal use.

ResetAll

Resets the HFMwQueryDef object's columns and filters to their default values.

Syntax

```
<HFMwQueryDef>.ResetAll
```

ResetColumnContents

Resets the HFMwQueryDef object's columns to their default values.

Syntax

```
<HFMwQueryDef>.ResetColumnContents
```

ResetFilters

Resets the HFMwQueryDef object's filters to their default values.

Syntax

```
<HFMwQueryDef>.ResetFilters
```

Example

ResetFilters is used in the example for [statusFilter](#).

SetWebSession

For internal use.

HFMwQueryDef Properties

The HFMwQueryDef component contains the properties described in the following topics.

balanceTypeFilter

Gets or sets a given balance type filter for the HFMwQueryDef object reference. The property accesses a collection of filters for the various balance types; each filter can be set to TRUE or FALSE.

Read-write.

Syntax

```
<HFMwQueryDef>.balanceTypeFilter(nIndex)
```

Argument	Description
----------	-------------

<i>nIndex</i>	The index of the balance type filter. Valid index values are represented by the HFMConstants type library constants listed in “Balance Type Filter Constants” on page 432 . Input argument. Integer subtype.
---------------	---

Example

The following snippet turns on the balance filter for balanced journals.

```
'cHFMQueryDef is an HFMwQueryDef object reference.  
cHFMQueryDef.balanceTypeFilter _  
    (WEBOM_JOURNAL_BALANCE_TYPE_BALANCED) = true
```

columnContents

Gets or sets the columns returned in query results. The property accesses a collection of columns. Each item in the collection is set to a column to be displayed. Valid values for this property are represented by the HFMConstants type library constants listed in [“Journal Column Display Constants” on page 428](#).

Read-write.

Syntax

```
<HFMwQueryDef>.columnContents(lColumnIndex)
```

Argument	Description
----------	-------------

<i>lColumnIndex</i>	The index of the column. Input argument. Long subtype.
---------------------	---

Example

The following snippet sets the columns filter to show journal labels, statuses, types, descriptions, groups, and entities. [columnSortOption](#) specifies that the journals will be sorted by labels in ascending order.

```
'cHFMQueryDef is an HFMwQueryDef object reference.  
cHFMQueryDef.columnContents(0) = COLUMN_JOURNALLABEL  
cHFMQueryDef.columnSortOption(0) = JOURNALREPORT_SORT_ASCENDING  
cHFMQueryDef.columnContents(1) = COLUMN_JOURNALSTATUS  
cHFMQueryDef.columnContents(2) = COLUMN_JOURNALTYPE  
cHFMQueryDef.columnContents(3) = COLUMN_JOURNALDESCRIPTION  
cHFMQueryDef.columnContents(4) = COLUMN_JOURNALGROUP  
cHFMQueryDef.columnContents(5) = COLUMN_JOURNALLINEITEMENTITY
```

columnContentsOption

For internal use.

columnRepeatOption

For internal use.

columnSortOption

Gets or sets whether to sort by a given column, as well as whether to sort in ascending or descending order. The property accesses a collection of column sorting options. Each item in the collection applies to the corresponding item in the [columnContents](#) property's collection of columns.

Valid values for this property are represented by the HFMConstants type library constants listed in [“Journal Report Sort Option Constants” on page 429](#).

Tip: To exclude a column from the sort order, use the NUM_JOURNALREPORT_SORT_FLAGS constant.

Read-write.

Syntax

```
<HFMwQueryDef>.columnSortOption(lColumnIndex)
```

Argument	Description
----------	-------------

<i>lColumnIndex</i>	The index of the column in the collection of the columnContents property's columns. Input argument. Long subtype.
---------------------	--

Example

[columnSortOption](#) is used in the example for [columnContents](#).

columnTotalOption

For internal use.

description

For internal use.

descriptionFilter

Gets or sets the journal or template description by which to filter query results.

Tip: You can use the percent sign (%) as a wildcard.

Read-write.

entityFilter

Gets or sets the Entity dimension member by which to filter query results. The member is identified by its label.

Read-write.

filter

For internal use.

Read-write.

groupFilter

Gets or sets the journal group label by which to filter query results.

Tip: You can use the percent sign (%) as a wildcard.

Read-write.

label

Gets or sets the label for a journal report.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance's properties.

Read-write.

labelFilter

Gets or sets the journal or template label by which to filter query results.

Tip: You can use the percent sign (%) as a wildcard.

Read-write.

period

Gets or sets the Period dimension member for a journal report's Point of View, using the member's label.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance's properties.

You can also get or set the Period dimension member with [povString](#).

Read-write.

povString

Gets or sets the Period, Year, Scenario, and Value dimension members for a journal report's Point of View. The Point of View is represented with the following syntax, with labels of dimension members replacing *Scenario*, *Year*, *Period*, and *Value*:

```
S#Scenario.Y#Year.P#Period.V#Value
```

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance's properties.

You can also get or set these dimension members with the [period](#), [year](#), [scenario](#), and [value](#) properties.

Read-write.

Example

The following example sets the Scenario dimension to Actual, the Year dimension to 2003, the Period dimension to October, and the Value dimension to <Entity Curr Adjs>:

```
'cHFMQueryDef is an HFMwQueryDef object reference.  
cHFMQueryDef.povstring = "S#Actual.Y#2003.P#October." & _  
    "V#<Entity Curr Adjs>"
```

scenario

Gets or sets the Scenario dimension member for a journal report's Point of View, using the member's label.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance's properties.

You can also get or set the Scenario dimension member with [povString](#).

Read-write.

security

Gets or sets the security class for a journal report, using the security class's label.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance's properties.

Read-write.

statusFilter

Gets or sets a given status filter for the HFMwQueryDef object. The property accesses a collection of filters for the various journal statuses; each filter can be set to TRUE or FALSE.

Read-write.

Syntax

```
<HFMwQueryDef>.statusFilter(nIndex)
```

Argument	Description
<i>nIndex</i>	The index of the filter in the collection of status filters. Valid index values are represented by the HFMConstants type library constants listed in "Journal Status Filter Constants" on page 432 . Input argument. Integer subtype.

Example

The following snippet sets the HFMwQueryDef object to filter journals of Working status. [ResetFilters](#) sets the filters to their default state, then `statusFilter` turns on the filter for the Working status.

```
'cHFMQueryDef is an HFMwQueryDef object reference.  
cHFMQueryDef.ResetFilters  
cHFMQueryDef.statusFilter(WEBOM_JOURNAL_STATUS_WORKING) = TRUE
```

stylesheet

For internal use.

Read-write.

totalColumnCount

Gets or sets the number of columns in the HFMwQueryDef object.

Read-write.

type

For internal use.

Read-write.

typeFilter

Gets or sets a given journal type filter for the HFMwQueryDef object. The property accesses a collection of filters for the various journal types; each filter can be set to TRUE or FALSE.

Read-write.

Syntax

```
<HFMwQueryDef>.typeFilter(nIndex)
```

Argument Description

<i>nIndex</i>	The index of the filter in the collection of journal type filters. Valid index values are represented by the HFMConstants type library constants listed in “Journal Type Filter Constants” on page 433 . Input argument. Integer subtype.
---------------	--

Example

typeFilter is used in the example for the HFMwManageJournals method [GetJournalsDisplayData](#).

value

Gets or sets the Value dimension member for a journal report’s Point of View, using the member’s label.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance’s properties.

You can also get or set the Value dimension member with [povString](#).

Read-write.

xml

Gets or sets the state of the HFMwQueryDef using an XML string.

Read-write.

year

Gets or sets the Year dimension member for a journal report’s Point of View, using the member’s label.

Tip: You can use [GetReportDefinition](#) to create a journal report based upon the HFMwQueryDef instance’s properties.

You can also get or set the Year dimension member with `povString`.
Read-write.

In This Chapter

HFMwStringUtility Component	381
HFMwFileReader Component	383
HFMwEncodedFileWriter Component.....	386

The HFMwUtilities type library contains the following components, which enable you to work with strings and files:

- The HFMwStringUtility component provides methods that work with strings.
- The HFMwFileReader component provides methods that read files.
- The HFMwEncodedFileWriter component provides methods that write to files.

HFMwStringUtility Component

The HFMwStringUtility component provides methods that help you work with strings. The component contains the methods described in the following topics.

CalcCRC32

Returns the CRC-32 checksum for a given string.

Syntax

```
<HFMwStringUtility>.CalcCRC32 (bstrPlain)
```

Argument	Description
----------	-------------

<i>bstrPlain</i>	The string for which to return the CRC-32 checksum. Input argument. String subtype.
------------------	--

Return Value

Returns the CRC-32 checksum for the string.

ConvertFromUTF8

Converts a string encoded in the UTF-8 format to a regular string.

Syntax

```
<HFMwStringUtility>.ConvertFromUTF8 (bstrUTF8String)
```

Argument	Description
----------	-------------

<i>bstrUTF8String</i>	The UTF-8 encoded string to be converted.
-----------------------	---

Caution! Do not pass an empty string. Passing an empty string causes an error.
Input argument. String subtype.

Return Value

Returns the converted string.

IsValidValue

Indicates whether a string is a valid value for the specified type of Financial Management string. For example, IsValidValue can indicate whether a string would be a valid application name or a valid dimension member label.

Syntax

```
<HFMwStringUtility>.IsValidValue (bstrValue, lValueType)
```

Argument	Description
----------	-------------

<i>bstrValue</i>	The string to be evaluated. Input argument. String subtype.
------------------	--

<i>lValueType</i>	The type of Financial Management string. Pass one of the HFMConstants type library constants listed in "Validation Type Constants" on page 467 . Input argument. Long subtype.
-------------------	---

Return Value

Returns TRUE if the string represents a valid value, FALSE otherwise.

Example

The following function indicates whether the passed string would be a valid cell description.

```
Function IsValidLineItemDesc(sDesc)
Dim cStringUtility
Set cStringUtility = Server.CreateObject("Hyperion.HFMwStringUtility")
IsValidLineItemDesc = cStringUtility.IsValidValue(sDesc, _
HFM_VALIDATIONTYPE_CELL_DESCRIPTION)
End Function
```

ReverseUnicodeByteOrder

Reverses the byte order for a Unicode string. If the string contains a Unicode signature mask, the mask is stripped.

Syntax

```
<HFMwStringUtility>.ReverseUnicodeByteOrder (bstrSring)
```

Argument Description

<i>bstrSring</i>	The string to be reversed.
	Input argument. String subtype.

Return Value

Returns the reversed string. If no reversing is required, a copy of the original string is returned.

HFMwFileReader Component

The HFMwFileReader component provides methods for reading files. You can read files as either text or binary files:

- To read a file as a text file, open it with [OpenAsText](#). You then can loop through the file by calling [ReadLine](#) until the end of the file is reached; test for the end of the file with [IsEOF](#). After you finish reading the file, call [Close](#).
- To read a file as a binary file, open it with [OpenAsBinary](#). You then can loop through the file by calling [Read](#) until the end of the file is reached; test for the end of the file with [IsEOF](#). After you finish reading the file, call [Close](#).

The HFMwFileReader component contains the methods described in the following topics.

Close

Closes a file that had been opened for reading.

Syntax

```
<HFMwFileReader>.Close
```

Example

Close is used in the example for [OpenAsText](#).

IsEOF

Indicates whether the HFMwFileReader object is at the end of the file being read.

Tip: To read a text file, use `IsEOF` with [ReadLine](#). To read a binary file, use `IsEOF` with [Read](#).

Syntax

```
<HFMwFileReader>.IsEOF ( )
```

Return Value

Returns TRUE if the end of the file has been reached, FALSE otherwise. Boolean subtype.

Example

`IsEOF` is used in the example for [OpenAsText](#).

OpenAsBinary

Opens a file for reading as a binary file.

Tip: To read a binary file, call [Read](#) in a loop that runs until [IsEOF](#) indicates that the end of the file has been reached.

Syntax

```
<HFMwFileReader>.OpenAsBinary bstrFileName
```

Argument	Description
----------	-------------

<i>bstrFileName</i>	The name and path of the file to open. Input argument. String subtype.
---------------------	---

OpenAsText

Opens a file for reading as a text file.

Syntax

```
<HFMwFileReader>.OpenAsText bstrFileName
```

Argument	Description
----------	-------------

<i>bstrFileName</i>	The name and path of the file to open. Input argument. String subtype.
---------------------	---

Example

The following function opens the specified file as a text file, then loops through it using [ReadLine](#) and [IsEOF](#). When the end of the file is reached, [Close](#) is called.

```
Function loopThruTextFile(sFile)
```



```

Dim sTxt
Set cFileReader = Server.CreateObject("Hyperion.HFMwFileReader")
cFileReader.OpenAsText sFile
Do Until cFileReader.IsEOF() = TRUE
    sTxt = cFileReader.ReadLine()
    ' ... insert code to execute for each text line
Loop
cFileReader.Close
End Function

```

Read

Reads data from a binary file.

Tip: To read a file, call `Read` in a loop that runs until `IsEOF` indicates that the end of the file has been reached.

Syntax

```
<HFMwFileReader>.Read()
```

Return Value

Returns the file as an array of bytes.

ReadLine

Reads a line from a text file.

Tip: To read all the lines in a text file, call `ReadLine` in a loop that runs until `IsEOF` indicates that the end of the file has been reached.

Syntax

```
<HFMwFileReader>.ReadLine()
```

Return Value

Returns the line being read.

Example

`ReadLine` is used in the example for [OpenAsText](#).

Size

Returns the number of characters in the file being read.

Syntax

```
<HFMwFileReader>.Size()
```

Return Value

Returns the number of characters. Long subtype.

HFMwEncodedFileWriter Component

The HFMwEncodedFileWriter component provides methods that enable you to write to files. The component contains the methods described in the following topics.

Close

Closes a file that has been opened for writing.

Syntax

```
<HFMwEncodedFileWriter>.Close
```

Example

Close is used in the example for [Open](#).

Open

Opens a file for writing and sets the file's encoding. If the file does not exist, Open creates it; if the file exists, Open overwrites it.

Syntax

```
<HFMwEncodedFileWriter>.Open bstrFileName, lEncoding
```

Argument	Description
----------	-------------

<i>bstrFileName</i>	The name and path of the file to open. Input argument. String subtype.
---------------------	---

<i>lEncoding</i>	The type of encoding to apply to the file. Valid values are represented by the HFMConstants type library constants listed in “Extracted File Encoding Constants” on page 452 . Input argument. Long subtype.
------------------	---

Example

The following function writes the specified text stream to the specified file, with ANSI encoding applied.

```
Function createFile(sFile, sTxt)
```

```

Set cFileWriter = _
Server.CreateObject("Hyperion.HFMwEncodedFileWriter")
cFileWriter.Open sFile, EXTRACT_FILE_ENCODING_ANSI
cFileWriter.Write sTxt
cFileWriter.Close
End Function

```

Write

Writes the specified text stream to a file that has been opened for writing.

Syntax

```
<HFMwEncodedFileWriter>.Write bstrStream
```

Argument Description

bstrStream The text stream to be written to the file.
Input argument. String subtype.

Example

Write is used in the example for [Open](#).

WriteBinary

Dumps the binary content of an IStream pointer to a file.

Note: Call [Open](#) to open the target file. After you have finished writing to the file, call [Close](#).

Syntax

```
<HFMwEncodedFileWriter>.WriteBinary varIUnkStream
```

Argument Description

varIUnkStream The IStream pointer.
Input argument.

In This Chapter

HFMwSharesCalc Component Methods.....	389
---------------------------------------	-----

The HFMwSharesCalc type library contains the HFMwSharesCalc component. This component enables you to calculate ownership for a Financial Management application.

Obtaining an HFMwSharesCalc Object Reference

Create an HFMwSharesCalc object with `Server.CreateObject`. After creating the object, you must associate it with the HFMwSession object for the application in which you want to calculate ownership:

```
set cHFMSharesCalc = _
Server.CreateObject("Hyperion.HFMwSharesCalc")
' cHFMSession is a previously set HFMwSession object
cHFMSharesCalc.SetWebSession cHFMSession
```

HFMwSharesCalc Component Methods

The HFMwSharesCalc component contains the methods described in the following topics.

SetWebSession

Associates an HFMwSharesCalc object with the HFMwSession object for the application in which you want to calculate ownership.

Tip: HFMwSession object references are obtained with the HFMwManageApplications component's [OpenApplication](#) method.

Syntax

```
<HFMwSharesCalc>.SetWebSession varpIUnkHFMwSession
```

Argument

Description

varpIUnkHFMwSession The HFMwSession object to be associated with the HFMwSharesCalc object.

Argument	Description
	Input argument.

SharesCalculation

Calculates ownership for the specified scenario, year, entity, and range of periods. Flags specify the calculation type and calculation mode.

Syntax

```
<HFMwSharesCalc>.SharesCalculation bstrScenario, bstrYear,
varavarbstrPeriods, bstrParent, lCalcType, lCalcMode
```

Argument	Description
<i>bstrScenario</i>	The name of the scenario. Input argument. String subtype.
<i>bstrYear</i>	The name of the year. Input argument. String subtype.
<i>varavarbstrPeriods</i>	An array containing the names of the periods for which the calculation is to be performed. Input argument.
<i>bstrParent</i>	The name of the entity that is to be used as the parent for the calculation. Input argument. String subtype.
<i>lCalcType</i>	Specifies the type of calculation to be performed. Pass one of the HFMConstants type library constants listed in “Share Calculation Types Constants” on page 466 . Input argument. Long subtype.
<i>lCalcMode</i>	Specifies the calculation mode. Pass one of the HFMConstants type library constants listed in “Share Calculation Ownership Constants” on page 466 . Input argument. Long subtype.

In This Chapter

Error Handling with Financial Management.....	391
HsvResourceManager Object Methods.....	392
Error Message Lookup Utility.....	400

This chapter explains how to use the HsvResourceManager type library to handle Financial Management errors and to work with localized resource strings. The chapter provides an overview of Financial Management error handling, describes the HsvResourceManager object, and explains how to use a utility with which you can look up error number descriptions.

Error Handling with Financial Management

Financial Management provides a mechanism for error handling that offers the following benefits:

- Error message localization – Error messages will be translated into whatever languages has been Financial Management.
- Centralized location – Previous Financial Management releases stored error messages in the event logs of application servers. To overcome the limitations of that approach, error messages are now stored in the database.
- Dynamic error messages – For certain errors, the specific item involved in the error is dynamically specified. For example, instead of a vague error message that says “invalid account”, a dynamic error message would specify the name of the invalid account. For example, if the invalid account name was Sales1, the message would read something like this:
`Sales1 is not a valid account.`
- Additional technical information – To aid in debugging, technical information such as the source file and the line number are returned in addition to the traditional user-readable error messages.

To accomplish these objectives, Financial Management error handling works as follows:

- In the IErrorInfo interface, Financial Management returns an XML string as the error description, and the HRESULT as the error number. The XML string contains a wealth of information that is not normally included in an error description.

Note: In VBScript, the `Err` object is used to access error information in the `IErrorInfo` interface.

- You pass the XML string and the `HRESULT` to the `HsvResourceManager` object, which returns the applicable error message strings.

HsvResourceManager Object Methods

The DLL for the `HsvResourceManager` type library is `HsvResourceManager.dll`. The `HsvResourceManager` type library contains one object—the `HsvResourceManager` object.

Note: The `HsvResourceManager` type library provides an interface to the *Resource Manager*. The Resource Manager exposes error message strings for Financial Management. If a Financial Management release has been localized into a given language, the error message strings will also be localized.

Obtaining an HsvResourceManager Object Reference

There are two ways in which you can obtain an `HsvResourceManager` object reference:

- Use `CreateObject` as shown in the following example:

```
Set cResourceManager = Server.CreateObject_  
("Hyperion.HsvResourceManager")
```

Note: If you use `CreateObject`, you must specify that error message resources for the Web tier are needed by passing the `HFM_WEB_TIER` constant to `Initialize`.

- Use the `HFMwSession` component's `resourceManager` property.

If no application is open, you must use `CreateObject`. If an application is open, you can use either technique.

The following topics describe how to handle error with the `HsvResourceManager` object, the format of technical error strings returned by some methods, and the object's methods.

Handling Errors with the HsvResourceManager Object

To handle errors with the `HsvResourceManager` object, take the following steps.

➤ To handle errors:

- 1 Set an `HsvResourceManager` object reference.
- 2 If you created the `HsvResourceManager` object reference with `CreateObject`, specify that Web tier resources are desired by passing the `HFM_WEB_TIER` constant to `Initialize`.
- 3 Do one of the following:
 - Pass the XML string and the `HRESULT` to `GetFormattedError` or `GetFormattedErrorWithLineFeed`.

- Pass the HRESULT to [GetResourceStringFromHR](#).

All of these methods return error messages that you can display to users. However, `GetFormattedError` and `GetFormattedErrorWithLineFeed` also return more technical information that may help debug an issue.

Note: These methods take language IDs. If Financial Management has been localized into a language, then error messages for that language will be available, otherwise error messages will be returned in Financial Management's default language. You can enumerate the languages for which localized error message strings are available with [GetAvailableLanguages](#).

System Message Detail Strings

Some methods return strings of technical information for system messages such as errors. These strings contain a uniquely identifying Error Reference Number, followed by various fields of information. These fields are delimited by semi-colons, and each has a label that is followed by a colon, as in the following example:

```
Error Reference Number: {219EB33B-BF50-11D6-A43E-0000863DCCF1}
Num: 0x800415c6; Type: 1; DTime: 9/3/02 12:20:10 PM; Svr: GSZAB01; File:
CHsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

The following table describes the fields:

Table 8 Fields in System Message Detail Strings

Field	Description
Num	The error number in hexadecimal form.
Type	<i>This is for internal use only.</i>
DTime	The TimeStamp of the error.
Svr	The machine name of the computer on which the error occurred.
File	The name of the source code file to which the error applies.
Line	The line number in the source code file to which the error applies.
Ver	The version number of the DLL to which the error applies.

GetAvailableLanguages

Returns arrays that identify the languages for which resources are available. Resources are available only for those languages into which Financial Management has been localized.

`GetAvailableLanguages` returns an array of language IDs and an array of language names. These arrays have a one-to-one correspondence with each other.

Syntax

```
<HsvResourceManager>.GetAvailableLanguages pvarastrLanguagesIDs,  
pvarastrLanguages
```

Argument	Description
<i>pvarastrLanguagesIDs</i>	Variant. Returns an array of language IDs that identify the languages for which resources are available. Tip: The HFMConstants enumeration tagHFM_LANGUAGES represents language IDs that are valid for all releases; for more information, see “Supported Language Constants” on page 413 .
<i>pvarastrLanguages</i>	Variant. Returns an array of strings that describe the languages for which resources are available.

Example

GetAvailableLanguages is used in the example for [GetFormattedError](#).

GetCurrentVersion

Returns the version number of Financial Management.

Syntax

```
<HsvResourceManager>.GetCurrentVersion()
```

Return Value

Variant. Returns the version number as a String subtype.

GetFormattedDateTime

Converts a given timestamp to a formatted string. You can choose from several date and time formats, including formats that return only the date or the time.

Syntax

```
<HsvResourceManager>.GetFormattedDateTime(lFormatType, dDateTime)
```

Argument	Description
<i>lFormatType</i>	Long (ByVal). The format of the string to return. Valid values are represented by the HFMConstants type library constants listed in “Date and Time Format Constants” on page 465 .
<i>dDateTime</i>	Double (ByVal). The timestamp.

Return Value

Variant. The formatted string.

Example

The following function uses the 24-hour time format to return the time segment of a given timestamp.

```
Function getTime24String(dTimeStamp As Double) As String
Dim cResourceManager As HsvResourceManager
Set cResourceManager = New HsvResourceManager
cResourceManager.Initialize HFM_TIER1
getTime24String = cResourceManager.GetFormattedDateTime _
    (HFM_DATE_TIME_FORMAT_HHHH_MI_SS, CDb1(Now))
End Function
```

GetFormattedError

Returns two strings in a given language that provide information on an error. One string contains a user-readable description of the error; the other string contains technical details for debugging purposes. The error is identified by an error number and description.

Tip: To apply a given line feed character to the technical details string, use [GetFormattedErrorWithLineFeed](#).

Syntax

```
<HsvResourceManager>.GetFormattedError lLanguageId, hr, bstrXMLError,
bstrDefaultError, pvarbstrFormattedError, pvarbstrTechnicalError
```

Argument	Description
<i>lLanguageId</i>	Long (ByVal). The language ID of the language. You can obtain the IDs of a Financial Management release's valid languages with GetAvailableLanguages . Tip: The HFMConstants enumeration tagHFM_LANGUAGES represents language IDs that are valid for all releases; for more information, see "Supported Language Constants" on page 413 .
<i>hr</i>	Long (ByVal). The HRESULT that identifies the error. Tip: In VBScript, you can get the HRESULT with the <code>Number</code> property of the <code>Err</code> object.
<i>bstrXMLError</i>	String (ByVal). The XML string that describes the error. Tip: In VBScript, you can get the XML string with the <code>Description</code> property of the <code>Err</code> object.
<i>bstrDefaultError</i>	String (ByVal). A default error message string. If the error is not generated by Financial Management, the specified default string is returned by the <i>pvarbstrFormattedError</i> argument. The way this works is that <code>GetFormattedError</code> first looks for a resource string that corresponds to the XML string passed in the <i>bstrXMLError</i> argument. If no matching resource string is found, <code>GetFormattedError</code> looks for a resource string that corresponds to the passed HRESULT. If no matching resource string is found, then the default string is returned.
<i>pvarbstrFormattedError</i>	Variant. Returns a simple description of the error. This is text that can be displayed to the user.

Argument	Description
<i>pvarbstrTechnicalError</i>	Variant. Returns detailed technical information regarding the error. For more information, see “System Message Detail Strings” on page 393 .

Example

The following function takes a language ID and returns the corresponding error message. The passed language ID is compared to the languages returned by [GetAvailableLanguages](#); if no match is found, the language ID for the default language is passed to `GetFormattedError`.

```
Function getHFMErrors(lId)
Dim cResourceManager, vaIDs, vaNames
Dim lLanguageID, vUserError, vTechError
Set cResourceManager = Server.CreateObject("Hyperion.HsvResourceManager")
cResourceManager.Initialize HFM_WEB_TIER
cResourceManager.GetAvailableLanguages vaIDs, vaNames
' Cast the passed language ID as a String
lId = CStr(lId)
' Set a default language ID.
lLanguageID = HFM_LANGUAGE_DEFAULT
For i = LBound(vaIDs) To UBound(vaIDs)
    If lId = vaIDs(i) Then
        ' If the passed language is available, override the
        ' previously set default language ID.
        lLanguageID = lId
        Exit For
    End If
Next
cResourceManager.GetFormattedError lLanguageID, Err.Number, _
    Err.Description, "Unknown Error", vUserError, vTechError
getHFMErrors = vUserError
End Function
```

GetFormattedErrorWithLineFeed

Returns two strings in a given language that provide information on an error. One string contains a user-readable description of the error; the other string contains technical details for debugging purposes, with a specified line feed character applied to the technical details string. The error is identified by an error number and description.

Tip: If you do not need to apply a specific line feed character, use [GetFormattedError](#).

Syntax

```
<HsvResourceManager>.GetFormattedErrorWithLineFeed lLanguageId, hr,
bstrXMLError, bstrDefaultError, bstrLineFeed, pvarbstrFormattedError,
pvarbstrTechnicalError
```

Argument	Description
<i>lLanguageId</i>	Long (ByVal). Identifies the language for which the strings will be returned. Pass one of the HFMConstants type library constants listed in “Supported Language Constants” on page 413 .
<i>hr</i>	Long (ByVal). The HRESULT that identifies the error. In Visual Basic and VBScript, you can get the HRESULT with the <code>Number</code> property of the <code>Err</code> object.
<i>bstrXMLError</i>	String (ByVal). The XML string that describes the error. In Visual Basic and VBScript, you can get the XML string with the <code>Description</code> property of the <code>Err</code> object.
<i>bstrDefaultError</i>	String (ByVal). A default error message string. If the error is not generated by Financial Management, the specified default string is returned by the <i>pvarbstrFormattedError</i> argument. The way this works is that <code>GetFormattedError</code> first looks for a resource string that corresponds to the XML string passed in the <i>bstrXMLError</i> argument. If no matching resource string is found, <code>GetFormattedError</code> looks for a resource string that corresponds to the passed HRESULT. If no matching resource string is found, then the default string is returned.
<i>bstrLineFeed</i>	String (ByVal). The line feed character to apply to the <i>pvarbstrTechnicalError</i> argument's string.
<i>pvarbstrFormattedError</i>	Variant. Returns a simple description of the error. This is text that can be displayed to the user.
<i>pvarbstrTechnicalError</i>	Variant. Returns detailed technical information regarding the error. For more information, see “System Message Detail Strings” on page 393 .

GetFormattedResourceString

For internal use.

GetHelpDirectoryForLanguageID

Returns the name of the subdirectory that contains the Financial Management help system for a given language.

Syntax

```
<HsvResourceManager>.GetHelpDirectoryForLanguageID(lLanguageId)
```

Argument Description

lLanguageId Long (ByVal). Identifies the language for which the subdirectory will be returned. Pass one of the HFMConstants type library constants listed in [“Supported Language Constants” on page 413](#).

Return Value

Variant. Returns the subdirectory name.

GetHFMLanguageIdFromUserLanguages

Returns the Financial Management ID for the language identified by a given language code. The method returns IDs only of languages for which resources have been translated.

Syntax

```
<HsvResourceManager>.GetHFMLanguageIdFromUserLanguages (bstrLanguages)
```

Argument	Description
----------	-------------

<i>bstrLanguages</i>	String (ByVal). The two-letter language code. To obtain language codes, use GetLanguageCountryCodeFromLanguageId .
----------------------	--

Return Value

Variant. Returns the language ID.

Note: If a language code represents a language for which Financial Management does not provide translated resources, the ID for the default language is returned. To return the languages for which translated resources exist, use [GetAvailableLanguages](#).

GetLanguageCountryCodeFromLanguageId

Returns the two-letter language code for a given language ID.

Syntax

```
<HsvResourceManager>.GetLanguageCountryCodeFromLanguageId (lLanguageId)
```

Argument	Description
----------	-------------

<i>lLanguageId</i>	Long (ByVal). The language ID. You can obtain the IDs of a Financial Management release's valid languages with GetAvailableLanguages . In addition, you can pass the constants in the HFMConstants enumeration tagHFM_LANGUAGES. This enumeration represents language IDs that are valid for all releases; for more information, see "Supported Language Constants" on page 413 .
--------------------	---

Return Value

String. Returns the language code.

GetResourceString

Not supported - do not use.

GetResourceStringFromHR

Returns the resource string for an HRESULT in the specified language.

Syntax

```
<HsvResourceManager>.GetStringFromHR(lLanguageId, hr)
```

lLanguageId Long (ByVal). Identifies the language for which the error message string will be returned. Pass one of the HFMConstants type library constants listed in [“Supported Language Constants” on page 413](#).

hr Long (ByVal). The HRESULT that identifies the error.

Tip: In VBScript, you can get the HRESULT with the `Number` property of the `Err` object.

Return Value

String. Returns the resource string.

GetUserDisplayDateTimeFormats

Returns an array of strings that describe the date and time formats into which Financial Management can convert timestamps. These formats are represented by the constants listed in [“Date and Time Format Constants” on page 465](#).

Tip: To format a timestamp, use [GetFormattedDateTime](#).

Syntax

```
<HsvResourceManager>.GetUserDisplayDateTimeFormats pvaravarDateTimeFormats
```

Argument	Description
<i>pvaravarDateTimeFormats</i>	Variant. Returns an array of strings that describe the formats.

GetUserLanguageFromHFMLanguageId

Returns the two-letter language code for the language represented by a given Financial Management language ID.

Syntax

```
<HsvResourceManager>.GetUserLanguageFromHFMLanguageId (lLanguageId)
```

Argument	Description
<i>lLanguageId</i>	Long (ByVal). The language ID. You can obtain the IDs of a Financial Management release’s valid languages with GetAvailableLanguages .
Tip: The HFMConstants enumeration tagHFM_LANGUAGES represents language IDs that are valid for all releases; for more information, see “Supported Language Constants” on page 413 .	

Return Value

Variant. Returns a string containing the language code.

GetWindowsDateFormat

Returns the Windows date separator character and short date format for the computer.

Syntax

```
<HsvResourceManager>.GetWindowsDateFormat pvarbstrDateSeparator,  
pvarbstrDateOrder
```

Argument	Description
----------	-------------

<i>pvarbstrDateSeparator</i>	Variant. Returns a string containing the date separator character.
------------------------------	--

<i>pvarbstrDateOrder</i>	Variant. Returns a string containing the short date format.
--------------------------	---

Initialize

Specifies the tier for which resources are needed. Each Financial Management tier has its own DLL for resources, and if you created the object with `CreateObject`, you must call `Initialize` before calling any of the other `HsvResourceManager` methods.

Caution! If you create the `HsvResourceManager` object with the `resourceManager` property, do not call `Initialize`. If `Initialize` is called in this case, an error will occur.

Syntax

```
<HsvResourceManager>.Initialize sTier
```

sTier Integer (ByVal). Identifies the tier for which you want to use resources. When programming with the Web components, pass the `HFM_WEB_TIER` constant.

Tip: `HFM_WEB_TIER` is one of the `HFMConstants` type library constants listed in “Tier Constants” on page 437.

Example

`Initialize` is used in the example for `GetFormattedError`.

Error Message Lookup Utility

You can use a utility to look up error messages associated with Financial Management’s HRESULTS. The utility is installed in Financial Management’s Consultant Utilities directory and has a file name of `HFMErrorLookup.exe`.

After launching the utility, take the following steps to look up an error number.

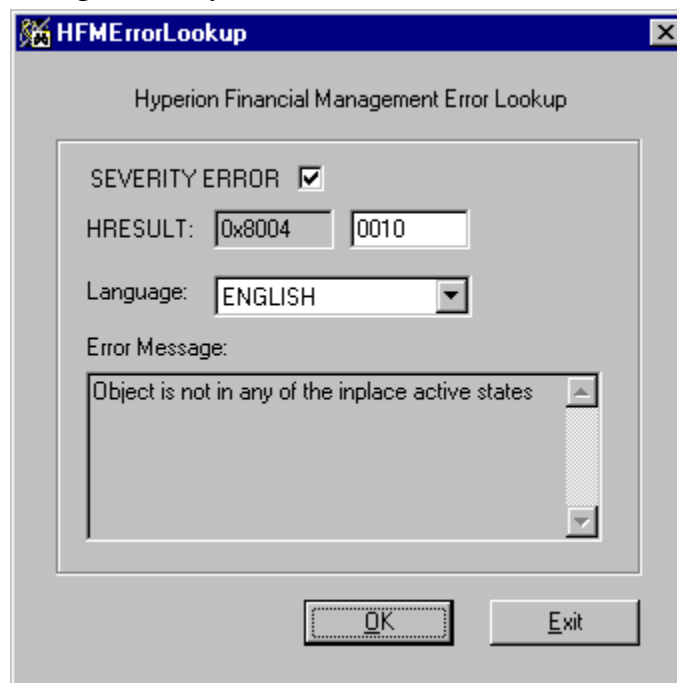
► To look up an error number:

1 Perform one of the following steps:

- a. To look up a severity HRESULT, select the **SEVERITY ERROR** check box. Severity HRESULTs are prefixed with 0x8004.
- b. To look up a success HRESULT, clear the **SEVERITY ERROR** check box. Success HRESULTs are prefixed with 0x0004.

Note: This check box determines the HRESULT prefix that displays in the read-only HRESULT text box to the left of the editable text box.

- 2** In the editable **HRESULT** text box, enter the portion of the HRESULT that is not displayed in the read-only text box.
- 3** From the **Language** drop-down list, select the language in which you want to view the error message.
- 4** Click **OK** to view the message for the specified HRESULT. In the following example, the utility displays the message for severity error 0x80040010:





Constants: The HFMConstants Type Library

In This Appendix

Metadata-Related Constants	403
Calculation, Transaction, and Metadata Status Constants	421
Journal-Related Constants	427
Process Management Constants	433
Report Column Constants	436
Consolidation Type Constants	437
Tier Constants	437
Security Constants	438
Web Constants	446
Extracted File Encoding Constants	452
Task Status Constants	452
User Activity Constants	453
Entity Transaction Detail Report Constants	456
Member Selection Constants	459
Miscellaneous Constants	464

This appendix lists constants for Financial Management. The DLL for the HFMConstants type library is `HFMConstants.dll`.

Note: The HFMConstants type library also contains enumerations that are not documented in this appendix. These are undocumented because they are used internally, and not by the methods and properties of the exposed object model.

Metadata-Related Constants

The HFMConstants type library includes numerous metadata-related constants. The following categories of metadata constants are provided:

- “Account Dimension Constants” on page 404
- “Custom Dimension Constants” on page 407
- “Consolidation Method Constants” on page 409
- “Currency Attribute Constant” on page 411

- [“Entity Dimension Constants” on page 411](#)
- [“View Dimension Constants” on page 412](#)
- [“Supported Language Constants” on page 413](#)
- [“Dimension-Related Constants” on page 413](#)
- [“Intercompany Partner Constants” on page 415](#)
- [“Period Dimension Constants” on page 416](#)
- [“Scenario Attribute Constants” on page 417](#)
- [“Member List Constants” on page 418](#)
- [“Value Dimension Constants” on page 418](#)
- [“Application Setting Attribute ID Constants” on page 420](#)

Account Dimension Constants

The HFMConstants type library provides the following types of constants for the Account dimension:

- [“Account Attribute Constants” on page 404](#)
- [“System Account Constants” on page 405](#)
- [“Account Member List Constants” on page 406](#)
- [“Account Type Constants” on page 406](#)

Account Attribute Constants

The following constants represent attributes of accounts.

Table 9 tagACCOUNT_ATTRIBS Enumeration

Constant	Description
ATTRIB_ACCOUNT_CALC_ATTRIBUTE	CalcAttribute attribute.
ATTRIB_ACCOUNT_CUSTOM1_TOP_MEMBER	Custom1TopMember attribute.
ATTRIB_ACCOUNT_CUSTOM2_TOP_MEMBER	Custom2TopMember attribute.
ATTRIB_ACCOUNT_CUSTOM3_TOP_MEMBER	Custom3TopMember attribute.
ATTRIB_ACCOUNT_CUSTOM4_TOP_MEMBER	Custom4TopMember attribute.
ATTRIB_ACCOUNT_ENABLE_CUSTOM1_AGGR	EnableCustom1Aggr attribute.
ATTRIB_ACCOUNT_ENABLE_CUSTOM2_AGGR	EnableCustom2Aggr attribute.
ATTRIB_ACCOUNT_ENABLE_CUSTOM3_AGGR	EnableCustom3Aggr attribute.
ATTRIB_ACCOUNT_ENABLE_CUSTOM4_AGGR	EnableCustom4Aggr attribute.

Constant	Description
ATTRIB_ACCOUNT_ENABLE_DATA_AUDIT	EnableDataAudit attribute.
ATTRIB_ACCOUNT_ICP_TOP_MEMBER	ICPTopMember attribute.
ATTRIB_ACCOUNT_IS_CALCULATED	IsCalculated attribute.
ATTRIB_ACCOUNT_IS_CONSOLIDATED	IsConsolidated attribute.
ATTRIB_ACCOUNT_IS_ICP	IsICP attribute.
ATTRIB_ACCOUNT_MAX	Represents the last account attribute; use this to loop through the account attributes.
ATTRIB_ACCOUNT_MIN	Represents the first account attribute; use this to loop through the account attributes.
ATTRIB_ACCOUNT_NUM_ATTRIBS	Represents the total number of account attributes.
ATTRIB_ACCOUNT_NUM_DECIMAL_PLACES	NumDecimalPlaces attribute.
ATTRIB_ACCOUNT_PLUG_ACCOUNT	PlugAcct attribute.
ATTRIB_ACCOUNT_SECURITY_CLASS	SecurityClass attribute.
ATTRIB_ACCOUNT_TYPE	AccountType attribute.
ATTRIB_ACCOUNT_USERDEF1	UserDefined1 attribute.
ATTRIB_ACCOUNT_USERDEF2	UserDefined2 attribute.
ATTRIB_ACCOUNT_USERDEF3	UserDefined3 attribute.
ATTRIB_ACCOUNT_USES_LINE_ITEMS	UsesLineItems attribute.
ATTRIB_ACCOUNT_XBRL_TAGS	XBRLTags attribute.

System Account Constants

The following constants represent system accounts.

Table 10 tagACCOUNTCONSTANTS Enumeration

Constant	Description
MEMBERACTIVESTATUS	Active account.
MEMBERCONSOL1	Consol1 account.
MEMBERCONSOL2	Consol2 account.
MEMBERCONSOL3	Consol3 account.
MEMBERCONSOLIDATIONMETHOD	Method account.
MEMBERDIRECTPERCENTOWNERSHIP	DOWN account.

Constant	Description
MEMBERFIRSTSYSTEMACCOUNT	Represents the first system account; use this to loop through the system accounts.
MEMBERLASTSYSTEMACCOUNT	Represents the last system account; use this to loop through the system accounts.
MEMBERPERCENTCONSOLIDATION	PCON account.
MEMBERPERCENTCONTROL	PCTRL account.
MEMBERPERCENTOWNERSHIP	POWN account.
MEMBERSHARESOUTSTANDING	SharesOutstanding account.
MEMBERSHARESOWNED	SharesOwned account.
MEMBERSHARESPERCENTOWNED	Shares%Owned account.
MEMBERVOTINGOUTSTANDING	VotingOutstanding account.
MEMBERVOTINGOWNED	VotingOwned account.
MEMBERVOTINGPERCENTOWNED	Voting%Owned account.

Account Member List Constants

The following constants represent system-defined member lists for accounts.

Table 11 tagACCOUNTMEMBERLISTS Enumeration

Constant	Description
ACCOUNT_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
ACCOUNT_MEMBER_LIST_ANCESTORS	Ancestors member list.
ACCOUNT_MEMBER_LIST_BASE	Base member list.
ACCOUNT_MEMBER_LIST_CHILDREN	Children member list.
ACCOUNT_MEMBER_LIST_DESCENDANTS	Descendants member list.
ACCOUNT_MEMBER_LIST_PARENTS	Parents member list.
ACCOUNT_MEMBER_LIST_SYSTEM	System member list.
NUM_PREDEFINED_ACCOUNT_MEMBER_LISTS	Represents the number of system-defined account member lists.

Account Type Constants

The following constants represent the valid account types.

Table 12 tagACCOUNTTYPES Enumeration

Constant	Description
ACCOUNTTYPE_ASSET	Asset account.
ACCOUNTTYPE_BALANCE	Balance account.
ACCOUNTTYPE_BALANCERECURRING	BalanceRecurring account.
ACCOUNTTYPE_CURRENCYRATE	CurrencyRate account.
ACCOUNTTYPE_DYNAMIC	Dynamic account.
ACCOUNTTYPE_EXPENSE	Expense account.
ACCOUNTTYPE_FLOW	Flow account.
ACCOUNTTYPE_GROUPLABEL	Grouplabel account.
ACCOUNTTYPE_INCOME	This constant has been replaced by the ACCOUNTTYPE_REVENUE constant.
ACCOUNTTYPE_LIABILITY	Liability account.
ACCOUNTTYPE_MAX	Represents the last account type; use this to loop through the account types.
ACCOUNTTYPE_MIN	Represents the first account type; use this to loop through the account types.
ACCOUNTTYPE_NONFINANCIAL	Nonfinancial account. Note: This account type is no longer used, but the constant has been left in for backwards compatibility purposes.
ACCOUNTTYPE_REVENUE	Revenue account.
ACCOUNTTYPE_SYSTEM	System account.
ACCOUNTTYPE_TEXT	Text account. Note: This account type is no longer used, but the constant has been left in for backwards compatibility purposes.

Custom Dimension Constants

The HFMConstants type library provides the following types of constants for the custom 1, 2, 3, and 4 dimensions:

- [“Custom 1 and 2 Dimension Member Lists” on page 407](#)
- [“Custom 3 and 4 Dimension Member Lists” on page 408](#)
- [“Custom Dimension Attributes” on page 408](#)

Custom 1 and 2 Dimension Member Lists

The following constants represent system-defined member lists for the Custom 1 and Custom 2 dimensions.

Table 13 tagC1ANDC2MEMBERLISTS Enumeration

Constant	Description
CUSTOM1_MEMBER_LIST_CONSOL_METHODS	ConsolMethods member list. Note: This applies only to the Custom 1 dimension.
CUSTOM1AND2_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
CUSTOM1AND2_MEMBER_LIST_ANCESTORS	Ancestors member list.
CUSTOM1AND2_MEMBER_LIST_BASE	Base member list.
CUSTOM1AND2_MEMBER_LIST_CHILDREN	Children member list.
CUSTOM1AND2_MEMBER_LIST_CURRENCIES	Currencies member list.
CUSTOM1AND2_MEMBER_LIST_DESCENDANTS	Descendants member list.
CUSTOM1AND2_MEMBER_LIST_PARENTS	Parents member list.
NUM_PREDEFINED_CUSTOM1_MEMBER_LISTS	Represents the number of system-defined member lists for the Custom 1 dimension.
NUM_PREDEFINED_CUSTOM2_MEMBER_LISTS	Represents the number of system-defined member lists for the Custom 2 dimension.

Custom 3 and 4 Dimension Member Lists

The following constants represent system-defined member lists for the Custom 3 and Custom 4 dimensions.

Table 14 tagC3ANDC4MEMBERLISTS Enumeration

Constant	Description
CUSTOM3AND4_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
CUSTOM3AND4_MEMBER_LIST_ANCESTORS	Ancestors member list.
CUSTOM3AND4_MEMBER_LIST_BASE	Base member list.
CUSTOM3AND4_MEMBER_LIST_CHILDREN	Children member list.
CUSTOM3AND4_MEMBER_LIST_DESCENDANTS	Descendants member list.
CUSTOM3AND4_MEMBER_LIST_PARENTS	Parents member list.
NUM_PREDEFINED_CUSTOM3AND4_MEMBER_LISTS	Represents the number of system-defined member lists for the Custom 3 and 4 dimensions.

Custom Dimension Attributes

The following constants represent attributes of the Custom 1, 2, 3, and 4 dimensions.

Table 15 tagCUSTOM_ATTRIBS Enumeration

Constant	Description
ATTRIB_CUSTOM_IS_CALCULATED	IsCalculated attribute.
ATTRIB_CUSTOM_MAX	Represents the last Custom dimension attribute; use this to loop through the attributes.
ATTRIB_CUSTOM_MIN	Represents the first Custom dimension attribute; use this to loop through the attributes.
ATTRIB_CUSTOM_NUM_ATTRIBS	Represents the total number of Custom dimension attributes.
ATTRIB_CUSTOM_SECURITY_CLASS	SecurityClass attribute.
ATTRIB_CUSTOM_SWITCH_SIGN	SwitchSignForFlow attribute.
ATTRIB_CUSTOM_SWITCH_TYPE	SwitchTypeForFlow attribute.
ATTRIB_CUSTOM_USERDEF1	UserDefined1 attribute.
ATTRIB_CUSTOM_USERDEF2	UserDefined2 attribute.
ATTRIB_CUSTOM_USERDEF3	UserDefined3 attribute.

Consolidation Method Constants

The HFMConstants type library provides the following types of constants for consolidation methods:

- [“Consolidation Method Attribute Constants” on page 409](#)
- [“Consolidation Methods: Control Attribute Constants” on page 410](#)
- [“Consolidation Methods: ToPercentControlComp Attribute Constants” on page 410](#)
- [“Consolidation Methods: PercentConsol Attribute Constants” on page 410](#)

Consolidation Method Attribute Constants

The following constants represent the attributes of consolidation methods.

Table 16 tagCONSOLMETHOD_ATTRIBS Enumeration

Constant	Description
ATTRIB_CONSOLMETH_CONTROL	Control attribute.
ATTRIB_CONSOLMETH_ISHOLDINGMETHOD	IsHoldingMethod attribute.
ATTRIB_CONSOLMETH_MAX	Represents the last attribute in this enumeration; use this to loop through the attributes.
ATTRIB_CONSOLMETH_MIN	Represents the first attribute in this enumeration; use this to loop through the attributes.
ATTRIB_CONSOLMETH_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.
ATTRIB_CONSOLMETH_PCTCONSOL	PercentConsol attribute.

Constant	Description
ATTRIB_CONSOLMETH_TOPCTCTRL	ToPercentControl attribute.
ATTRIB_CONSOLMETH_TOPCTCTRLCOMP	ToPercentControlComp attribute.
ATTRIB_CONSOLMETH_USEDBYCALCROUTINE	UsedByCalcRoutine attribute.

Consolidation Methods: Control Attribute Constants

The following constants represent the valid values for the Control attribute of consolidation methods.

Table 17 tagCONSOLMETHOD_CONTROL Enumeration

Constant	Description
CONSOLMETHOD_CONTROL_EMPTY	The attribute does not have a value.
CONSOLMETHOD_CONTROL_FULL	Full.
CONSOLMETHOD_CONTROL_LIMITED	Limited.
CONSOLMETHOD_CONTROL_NO	No.

Consolidation Methods: ToPercentControlComp Attribute Constants

The following constants represent the valid values for the ToPercentControlComp attribute of consolidation methods.

Table 18 tagCONSOLMETHOD_TOPCTCTRLLOPS Enumeration

Constant	Description
CONSOLMETHOD_TOPCTCTRLLOP_LESSTHAN	Less-than (<) operator.
CONSOLMETHOD_TOPCTCTRLLOP_LESSTHANOREQUAL	Less-than-or-equal-to (<=) operator.
CONSOLMETHOD_TOPCTCTRLLOP_NONE	The attribute does not have a value.

Consolidation Methods: PercentConsol Attribute Constants

The following constants represent system-defined values for the PercentConsol attribute.

Table 19 tagCONSOLPCTCONSTANTS Enumeration

Constant	Description
CONSOLPCT_POWN	POWN system account.
CONSOLPCT_POWNMIN	POWNMIN system account.

Constant	Description
CONSOLPCT_UNSPECIFIED	No value is specified for this attribute.

Currency Attribute Constant

The following constants represent currency attributes.

Table 20 tagCURRENCY_ATTRIBS Enumeration

Constant	Description
ATTRIB_CURRENCY_ICT_DISPLAY	DisplayInICT attribute.
ATTRIB_CURRENCY_MAX	Represents the last attribute in this enumeration; use this to loop through the attributes.
ATTRIB_CURRENCY_MIN	Represents the first attribute in this enumeration; use this to loop through the attributes.
ATTRIB_CURRENCY_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.
ATTRIB_CURRENCY_SCALE	Scale attribute.
ATTRIB_CURRENCY_TRANSOP	TranslationOperator attribute.

Entity Dimension Constants

The HFMCconstants type library provides the following types of constants for entities:

- [“Entity Attribute Constants” on page 411](#)
- [“Entity Member List Constants” on page 412](#)

Entity Attribute Constants

The following constants represent attributes of Entity dimension members.

Table 21 tagENTITY_ATTRIBUTES Enumeration

Constant	Description
ATTRIB_EAP_SECURITY_CLASS	SecurityAsPartner attribute.
ATTRIB_ENTITY_ALLOW_ADJS_FROM_CHILDREN	AllowAdjFromChildren attribute.
ATTRIB_ENTITY_ALLOW_ADJUSTMENTS	AllowAdjs attribute.
ATTRIB_ENTITY_DEFAULT_VALUE_ID	DefCurrency attribute.
ATTRIB_ENTITY_HOLDING_COMPANY	HoldingCompany attribute.
ATTRIB_ENTITY_IS_ICP	IsICP attribute.
ATTRIB_ENTITY_MAX	Represents the last attribute in this enumeration; use this to loop through the attributes.

Constant	Description
ATTRIB_ENTITY_MIN	Represents the first attribute in this enumeration; use this to loop through the attributes.
ATTRIB_ENTITY_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.
ATTRIB_ENTITY_SECURITY_CLASS	SecurityClass attribute.
ATTRIB_ENTITY_USERDEF1	UserDefined1 attribute.
ATTRIB_ENTITY_USERDEF2	UserDefined2 attribute.
ATTRIB_ENTITY_USERDEF3	UserDefined3 attribute.

Entity Member List Constants

The following constants represent member lists that have been system-defined for the Entity dimension.

Table 22 tagENTITYMEMBERLISTS Enumeration

Constant	Description
ENTITY_MEMBER_LIST_ADJUSTMENT_ENTITIES	Adjustment Entities member list.
ENTITY_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
ENTITY_MEMBER_LIST_ANCESTORS	Ancestors member list.
ENTITY_MEMBER_LIST_BASE	Base member list.
ENTITY_MEMBER_LIST_CHILDREN	Children member list.
ENTITY_MEMBER_LIST_DESCENDANTS	Descendants member list.
ENTITY_MEMBER_LIST_PARENT_ADJUSTMENT_ENTITIES	Parent Adjustment Entities member list.
ENTITY_MEMBER_LIST_PARENTS	Parents member list.
NUM_PREDEFINED_ENTITY_MEMBER_LISTS	Represents the number of member lists that have been system-defined for the Entity dimension.

View Dimension Constants

The HFMConstants type library provides the following types of constants for the View dimension:

- [“Frequencies Constants” on page 412](#)
- [“View Dimension Member Constants” on page 413](#)

Frequencies Constants

The following constants represent attributes of frequencies.

Table 23 tagFREQUENCYCONSTANTS Enumeration

Constant	Description
MEMBERLOWESTFREQAVAILABLE	Represents the lowest internal ID number that is available to represent user-defined frequencies. ID numbers lower than the number represented by this constant identify system-defined frequencies.
MEMBERYEARLY	Yearly frequency.

View Dimension Member Constants

The following constants represent members of the View dimension.

Table 24 tagVIEWCONSTANTS Enumeration

Constant	Description
MEMBERLOWESTVIEWAVAILABLE	Represents the lowest internal ID number that is available to represent user-defined View dimension members. ID numbers lower than the number represented by this constant identify system-defined members.
MEMBERPERIODIC	Periodic member.
MEMBERSENARIOVIEW	<Scenario View> member.
MEMBERYTD	YTD member.

Supported Language Constants

The following constants represent languages supported by Financial Management.

Table 25 tagHFM_LANGUAGES Enumeration

Constant	Description
HFM_LANGUAGE_DEFAULT	Represents the default language.
HFM_LANGUAGE_ENGLISH	English.
HFM_LANGUAGE_INSTALLED	Represents the installed language.
HFM_LANGUAGE_USER_DEFAULT	Represents the user's default language.
HFM_NO_LANGUAGE	Used to get non-translated resource strings.

Dimension-Related Constants

The HFMConstants type library provides the following types of dimension-related constants:

- [“Dimension ID Constants” on page 414](#)
- [“Constant Representing All Dimensions” on page 414](#)

- [“Dimension Member Constants” on page 414](#)

Dimension ID Constants

The following constants represent internal IDs of Financial Management’s dimensions.

Table 26 tagHFMDIMENSIONS Enumeration

Constant	Description
DIMENSIONACCOUNT	Account dimension.
DIMENSIONCUSTOM1	Custom 1 dimension.
DIMENSIONCUSTOM2	Custom 2 dimension.
DIMENSIONCUSTOM3	Custom 3 dimension.
DIMENSIONCUSTOM4	Custom 4 dimension.
DIMENSIONENTITY	Entity dimension.
DIMENSIONICP	Intercompany Partner dimension.
DIMENSIONPARENT	Parent entity, Entity dimension.
DIMENSIONPERIOD	Period dimension.
DIMENSIONSENARIO	Scenario dimension.
DIMENSIONVALUE	Value dimension.
DIMENSIONVIEW	View dimension.
DIMENSIONYEAR	Year dimension.
NUMDIMENSIONS	Represents the number of dimensions.
NUMDIMENSIONS_INCLUDINGPARENT	Represents the number of dimensions, counting the parent entity as a dimension.
DIMENSION_LBOUND	Represents the first dimension in the set of dimensions; use this to loop through the dimensions.
DIMENSION_UBOUND	Represents the last dimension in the set of dimensions; use this to loop through the dimensions.

Constant Representing All Dimensions

The tagHFMDIMENSIONSALL enumeration contains the DIMENSION_ALL constant, which represents all dimensions (as opposed to the constants listed in [“Dimension ID Constants” on page 414](#), which represent specific dimensions).

Dimension Member Constants

The following constants can be useful when working with dimension members.

Table 27 tagPOVDEFAULTS Enumeration

Constant	Description
MEMBERALL	Specifies that you want to use all members of a dimension.
MEMBERALLEXCEPTNONE	Used for the Intercompany Partner dimension to ignore the [ICP None] member.
MEMBERANYONE	Specifies that you want to use any one member of a dimension.
MEMBERDEFAULT	Represents the default member of a dimension.
MEMBERNONE	Some dimensions, such as the Account dimension, have a member named [None]. This constant represents such members.
MEMBERNOTUSED	Specifies that no specific member applies.

Intercompany Partner Constants

The HFMCconstants type library provides the following types of constants for the Intercompany Partner dimension:

- [“Intercompany Partner Member Constants” on page 415](#)
- [“Intercompany Partner Member List Constants” on page 416](#)
- [“Intercompany Partner Member Attribute Constants” on page 416](#)

Intercompany Partner Member Constants

The following constants represent members of the Intercompany Partner dimension.

Table 28 tagICPCONSTANTS Enumeration

Constant	Description
MEMBERFIRSTSYSTEMICP	Represents the first member of the Intercompany Partner dimension; use this to loop through the dimension's members.
MEMBERICPALL	<i>For internal use.</i>
MEMBERICPDEFAULT	<i>For internal use.</i>
MEMBERICPNONE	Represents the [ICP None] member.
MEMBERICPTOP	Represents the [ICP Top] member.
MEMBERINTERCOMPANYENTITIES	Represents the [ICP Entities] member.
MEMBERLASTSYSTEMICP	Represents the last member of the Intercompany Partner dimension; use this to loop through the dimension's members
MEMBERLOWESTICPAVAILABLE	Represents the lowest internal ID number that is available to represent user-defined Intercompany Partner dimension members. ID numbers lower than the number represented by this constant identify system-defined members.

Intercompany Partner Member List Constants

The following constants represent system-defined member lists for the Intercompany Partner dimension.

Table 29 tagICPMEMBERLISTS Enumeration

Constant	Description
ICP_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
ICP_MEMBER_LIST_ANCESTORS	Ancestors member list.
ICP_MEMBER_LIST_BASE	Base member list.
ICP_MEMBER_LIST_CHILDREN	Children member list.
ICP_MEMBER_LIST_DESCENDANTS	Descendants member list.
ICP_MEMBER_LIST_PARENTS	Parents member list.
ICP_MEMBER_LIST_SYSTEM	System member list.
NUM_PREDEFINED_ICP_MEMBER_LISTS	The number of system-defined member lists.

Intercompany Partner Member Attribute Constants

The following table lists constants for Intercompany Partner dimension attributes.

Table 30 tagICP_ATTRIBS Enumeration

Constant	Description
ATTRIB_ICP_MAX	Represents the last attribute in this enumeration; use this to loop through the attributes.
ATTRIB_ICP_MIN	Represents the first attribute in this enumeration; use this to loop through the attributes.
ATTRIB_ICP_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.
ATTRIB_ICP_SECURITY_CLASS	SecurityClass attribute.

Period Dimension Constants

The HFMConstants type library provides the following types of constants for the Period dimension:

- [“Period Member Constant” on page 416](#)
- [“Period Member List Constants” on page 417](#)

Period Member Constant

The tagPERIODCONSTANTS class contains the MEMBERYEAR constant, which represents the Year member.

Tip: Year is a system-defined member of the Period dimension.

Period Member List Constants

The following constants represent system-defined member lists for the Period dimension.

Table 31 tagPERIODMEMBERLISTS Enumeration

Constant	Description
NUM_PREDEFINED_PERIOD_MEMBER_LISTS	The number of system-defined member lists for the Period dimension.
PERIOD_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
PERIOD_MEMBER_LIST_ANCESTORS	Ancestors member list.
PERIOD_MEMBER_LIST_CHILDREN	Children member list.
PERIOD_MEMBER_LIST_DESCENDANTS	Descendants member list.
PERIOD_MEMBER_LIST_FIFTH_GENERATION	Fifth Generation member list.
PERIOD_MEMBER_LIST_FIRST_GENERATION	First Generation member list.
PERIOD_MEMBER_LIST_FOURTH_GENERATION	Fourth Generation member list.
PERIOD_MEMBER_LIST_PARENTS	Parents member list.
PERIOD_MEMBER_LIST_SECOND_GENERATION	Second Generation member list.
PERIOD_MEMBER_LIST_SIXTH_GENERATION	Sixth Generation member list.
PERIOD_MEMBER_LIST_THIRD_GENERATION	Third Generation member list.

Scenario Attribute Constants

The following constants represent attributes of the Scenario dimension.

Table 32 tagSCENARIO_ATTRIBUTES Enumeration

Constant	Description
ATTRIB_SCENARIO_CONSOLIDATE_YTD	ConsolidateYTD attribute.
ATTRIB_SCENARIO_DEFAULT_FREQ_FOR_IC_TRANSACTIONS	DefFreqForICTrans attribute.
ATTRIB_SCENARIO_DEFAULT_FREQUENCY	DefaultFreq attribute.
ATTRIB_SCENARIO_DEFAULT_VIEW	DefaultView attribute.
ATTRIB_SCENARIO_ENABLE_DATA_AUDIT	EnableDataAudit attribute.
ATTRIB_SCENARIO_MAX	Represents the last attribute in this enumeration; use this to loop through the attributes.

Constant	Description
ATTRIB_SCENARIO_MAX_REVIEW_LEVEL	MaximumReviewLevel attribute.
ATTRIB_SCENARIO_MIN	Represents the first attribute in this enumeration; use this to loop through the attributes.
ATTRIB_SCENARIO_MISSING_DATA_ADJ	ZeroViewForAdj attribute.
ATTRIB_SCENARIO_MISSING_DATA_NON_ADJ	ZeroViewForNonadj attribute.
ATTRIB_SCENARIO_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.
ATTRIB_SCENARIO_SECURITY_CLASS	SecurityClass attribute.
ATTRIB_SCENARIO_SUPPORTS_PROCESS_FLOW	SupportsProcessManagement attribute.
ATTRIB_SCENARIO_USERDEF1	UserDefined1 attribute.
ATTRIB_SCENARIO_USERDEF2	UserDefined2 attribute.
ATTRIB_SCENARIO_USERDEF3	UserDefined3 attribute.
ATTRIB_SCENARIO_USES_LINE_ITEMS	UsesLineItems attribute.

Member List Constants

The following constants represent system-defined member lists. Unlike some of the other constants, these apply to member lists of all dimensions.

Table 33 tagSYSTEMLISTS Enumeration

Constant	Description
MEMBER_LIST_ALL_HIERARCHY	[Hierarchy] member list.
MEMBER_LIST_DESCENDANTS	[Descendants] member list.
MEMBER_LIST_DML_START	Dynamic member list.

Value Dimension Constants

The HFMConstants type library provides the following types of constants for the Value dimension:

- [“Value Dimension Member Constants” on page 418](#)
- [“Value Dimension Member List Constants” on page 419](#)

Value Dimension Member Constants

The following constants represent Value dimension members.

Table 34 tagVALUECONSTANTS Enumeration

Constant	Description
MEMBERALLENTITYVALUES	<i>For internal use.</i>
MEMBERALLNODEVALUES	<i>For internal use.</i>
MEMBERCONTRIBUTION	[Contribution]
MEMBERCONTRIBUTIONADJS	[Contribution Adjs]
MEMBERCONTRIBUTIONTOTAL	[Contribution Total]
MEMBERDERIVEPROPORTION	<i>For internal use.</i>
MEMBERELIMINATION	[Elimination]
MEMBERENTITYCURRENCY	[Entity Currency]
MEMBERENTITYCURRENCYADJS	[Entity Curr Adjs]
MEMBERENTITYCURRENCYTOTAL	[Entity Curr Total]
MEMBERFIRSTSPECIALVALUE	<i>For internal use.</i>
MEMBERLASTSPECIALVALUE	<i>For internal use.</i>
MEMBERLOWESTVALUEAVAILABLE	Represents the lowest internal ID number that is available to represent user-defined Value dimension members. ID numbers lower than the number represented by this constant identify system-defined members.
MEMBERPARENT	[Parent]
MEMBERPARENTADJS	[Parent Adjs]
MEMBERPARENTCURRENCY	[Parent Currency]
MEMBERPARENTCURRENCYADJS	[Parent Curr Adjs]
MEMBERPARENTCURRENCYTOTAL	[Parent Curr Total]
MEMBERPARENTTOTAL	[Parent Total]
MEMBERPROPORTION	[Proportion]
MEMBERSPECIALCURRENCY	<i>For internal use.</i>
MEMBERSPECIALCURRENCYADJ	<i>For internal use.</i>
MEMBERSPECIALCURRENCYTOTAL	<i>For internal use.</i>

Value Dimension Member List Constants

The following constants represent system-defined member lists for the Value dimension.

Table 35 tagVALUEMEMBERLISTS Enumeration

Constant	Description
NUM_PREDEFINED_VALUE_MEMBER_LISTS	The number of system-defined member lists for the Value dimension.
VALUE_MEMBER_LIST_ADJS	Adjustments member list.
VALUE_MEMBER_LIST_ALL_HIERARCHY	Hierarchy member list.
VALUE_MEMBER_LIST_DEFAULT_CURRENCIES	Default Currencies member list.
VALUE_MEMBER_LIST_DESCENDANTS	Descendants member list.
VALUE_MEMBER_LIST_INPUTS	Inputs member list.
VALUE_MEMBER_LIST_TOTALS	Totals member list.

Application Setting Attribute ID Constants

The following constants represent IDs of application setting attributes.

Table 36 tagAPPSETTING_ATTRIBS Enumeration

Constant	Description
ATTRIB_APPSETTING_CONSOLIDATIONRULES	ConsolidationRules attribute.
ATTRIB_APPSETTING_DEFAULTCURRENCY	DefaultCurrency attribute.
ATTRIB_APPSETTING_DEFAULTRATEFORBALANCEACCOUNTS	DefaultRateForBalanceAccounts attribute.
ATTRIB_APPSETTING_DEFAULTRATEFORFLOWACCOUNTS	DefaultRateForFlowAccounts attribute.
ATTRIB_APPSETTING_DEFAULTVALUEFORACTIVE	DefaultValueForActive attribute.
ATTRIB_APPSETTING_ENABLEMETADATASECURITYFILTERING	EnableMetadataSecurityFiltering attribute.
ATTRIB_APPSETTING_ICPENTITIESAGGREGATIONWEIGHT	ICPEntitiesAggregationWeight attribute.
ATTRIB_APPSETTING_MAX_CELLTEXT_SIZE	MaxCellTextSize attribute.
ATTRIB_APPSETTING_MAX_DOC_ATTACHMENT_SIZE	MaxDocAttachmentSize attribute.
ATTRIB_APPSETTING_MAX_NUM_DOC_ATTACHMENTS	MaxNumDocAttachments attribute.
ATTRIB_APPSETTING_NODESECURITY	NodeSecurity attribute.
ATTRIB_APPSETTING_ORGBYPERIODAPPLICATION	OrgByPeriodApplication attribute.
ATTRIB_APPSETTING_USEPVAFORBALANCEACCOUNTS	UsePVAForBalanceAccounts attribute.
ATTRIB_APPSETTING_USEPVAFORFLOWACCOUNTS	UsePVAForFlowAccounts attribute.
ATTRIB_APPSETTING_USESECURITYFORACCOUNTS	UseSecurityForAccounts attribute.

Constant	Description
ATTRIB_APPSETTING_USESECURITYFORCUSTOM1	UseSecurityForCustom1 attribute.
ATTRIB_APPSETTING_USESECURITYFORCUSTOM2	UseSecurityForCustom2 attribute.
ATTRIB_APPSETTING_USESECURITYFORCUSTOM3	UseSecurityForCustom3 attribute.
ATTRIB_APPSETTING_USESECURITYFORCUSTOM4	UseSecurityForCustom4 attribute.
ATTRIB_APPSETTING_USESECURITYFORENTITIES	UseSecurityForEntities attribute.
ATTRIB_APPSETTING_USESECURITYFORICP	UseSecurityForICP attribute.
ATTRIB_APPSETTING_USESECURITYFORSCENARIOS	UseSecurityForScenarios attribute.
ATTRIB_APPSETTING_VALIDATIONACCOUNT	ValidationAccount attribute.
ATTRIB_APPSETTING_MIN	Represents the first application setting attribute; use this to loop through the attributes.
ATTRIB_APPSETTING_MAX	Represents the last application setting attribute; use this to loop through the attributes.
ATTRIB_APPSETTING_NUM_ATTRIBS	Represents the total number of attributes in this enumeration.

Calculation, Transaction, and Metadata Status Constants

The HFMConstants type library provides the following types of status-related constants:

- [“Cell Calculation Status Constants” on page 421](#)
- [“Cell Status Constants” on page 422](#)
- [“Cell Metadata Status Constants” on page 422](#)
- [“Subcube Period Calculation Status Constants” on page 423](#)
- [“Transaction Dimension Constants” on page 424](#)
- [“Cell Transaction Status Constants” on page 425](#)
- [“Calculation Status Statistic Constants” on page 426](#)

Cell Calculation Status Constants

The following constants represent calculation statuses of cells.

Table 37 tagCALCSTATUSHIGHBITS Enumeration

Constant	Description
CELLSTATUS_ERROR	An error applies to the cell.
CELLSTATUS_INUSE	No data has been entered or processed for the cell’s subcube.

Constant	Description
CELLSTATUS_LOCKED	The cell is locked, meaning users are not allowed to modify the cell's data.
CELLSTATUS_NEEDSCHARTLOGIC	A calculation should be run for the cell.
CELLSTATUS_NEEDSCONSOLIDATION	A consolidation should be run for the cell.
CELLSTATUS_NEEDSTRANSlation	A translation should be run for the cell.
CELLSTATUS_NODATAINTABLE	No data exists for the cell's scenario and year.
CELLSTATUS_OK_BUT_SYSTEM_CHANGED	The system has changed because someone has loaded rules or metadata. A recalculation via Force Calculate is recommended but is not required.
CELLSTATUS_PROCESS_FLOW_BIT1	<i>For internal use.</i>
CELLSTATUS_PROCESS_FLOW_BIT2	<i>For internal use.</i>
CELLSTATUS_PROCESS_FLOW_BIT3	<i>For internal use.</i>
CELLSTATUS_PROCESS_FLOW_BIT4	<i>For internal use.</i>
CELLSTATUS_VALUEMEMBER_NEEDS_CALC	A calculation should be run for the cell's Value dimension member.
CELLSTATUS_VALUEMEMBER_NODATAINTABLE	No data exists for the cell.

Cell Status Constants

The following constants represent statuses of cells.

Table 38 tagCALCSTATUSLOWBITS Enumeration

Constant	Description
CELLSTATUS_DERIVED	The cell's data is derived.
CELLSTATUS_HASICDETAILTRANS	The cell contains intercompany transactions.
CELLSTATUS_HASTRANSCTIONS	The cell contains line items.
CELLSTATUS_NODATA	The cell has no data.
CELLSTATUS_PARENT_LEVEL_INPUT	The cell contains data that has been manually entered at the parent entity level.

Cell Metadata Status Constants

The following constants represent metadata statuses for cells.

Table 39 tagCALCSTATUSMIDDLEBITS Enumeration

Constant	Description
CELLSTATUS_CANREAD	The cell cannot be read.

Constant	Description
CELLSTATUS_CANWRITE	The cell cannot be written to.
CELLSTATUS_HASTEXT	The cell contains a cell text description.
CELLSTATUS_INVALID	Invalid metadata combination.
CELLSTATUS_ISADJUSTMENTMEMBER	The cell's Value dimension member is an ADJS member.
CELLSTATUS_ISINPUTFREQUENCY	The cell is an input-level time period.
CELLSTATUS_NOREADACCESS	The user does not have read access to the cell.
CELLSTATUS_NOWRITEACCESS	The user does not have write access to the cell.
CELLSTATUS_SUPPORTSCHARTLOGIC	The cell supports calculations.
CELLSTATUS_SUPPORTSCONSOLIDATION	The cell supports consolidations.
CELLSTATUS_SUPPORTSLINEITEMS	The cell supports line items.
CELLSTATUS_SUPPORTSPERIODALLOCATIONS	The cell is a parent-level time period whose children are input cells.
CELLSTATUS_SUPPORTSTRANSlation	The cell supports translations.

Subcube Period Calculation Status Constants

The following constants represent calculation statuses for periods in subcubes.

Table 40 tagCALCULATIONSTATUS Enumeration

Constant	Description
CALCSTATUS_ADJ_IS_NODATA	The cells that intersect the period and the "Adjs" member of the subcube's Value dimension triplet contain no data.
CALCSTATUS_ADJ_NEEDS_CALC	The cells that intersect the period and the "Adjs" member of the subcube's Value dimension triplet require a calculation.
CALCSTATUS_CONSOLIDATION_TRANSACTIONS_ARE_INVALID	The subcube cells have invalid consolidation transactions.
CALCSTATUS_CONTRIBUTIONADJ_IS_NODATA	The subcube cells that intersect the period and the [Contribution Adjs] Value dimension member contain no data.
CALCSTATUS_CONTRIBUTIONADJ_NEEDS_CALC	The subcube cells that intersect the period and the [Contribution Adjs] Value dimension member require a calculation.
CALCSTATUS_ELIMINATION_IS_NODATA	The subcube cells that intersect the period and the [Elimination] Value dimension member contain no data.
CALCSTATUS_ELIMINATION_NEEDS_CALC	The subcube cells that intersect the period and the [Elimination] Value dimension member require a calculation.

Constant	Description
CALCSTATUS_INPUT_IS_NODATA	The cells that intersect the period and the input member of the subcube's Value dimension triplet contain no data.
CALCSTATUS_INPUT_NEEDS_CALC	The cells that intersect the period and the input member of the subcube's Value dimension triplet require a calculation.
CALCSTATUS_INUSE	The period of the subcube is in use; data was entered or a calculation was performed.
CALCSTATUS_LOCKED	No one is allowed to modify data for the subcube cells that intersect the period.
CALCSTATUS_NEEDSCHARTLOGIC	A calculation should be run for the subcube cells that intersect the period.
CALCSTATUS_NEEDSCONSOLIDATION	A consolidation should be run for the subcube cells that intersect the period.
CALCSTATUS_NEEDSTRANSlation	A translation should be run for the subcube cells that intersect the period.
CALCSTATUS_NODATA	No data exists for the subcube cells that intersect the period.
CALCSTATUS_OK_BUT_SYSTEM_CHANGED	The system has changed because someone has loaded rules or metadata. A recalculation via Force Calculate is recommended but is not required.
CALCSTATUS_PARENTADJ_IS_NODATA	The subcube cells that intersect the period and the [Parent Adjs] Value dimension member contain no data.
CALCSTATUS_PARENTADJ_NEEDS_CALC	The subcube cells that intersect the period and the [Parent Adjs] Value dimension member require a calculation.
CALCSTATUS_PROCESS_FLOW_BIT1	<i>For internal use.</i>
CALCSTATUS_PROCESS_FLOW_BIT2	<i>For internal use.</i>
CALCSTATUS_PROCESS_FLOW_BIT3	<i>For internal use.</i>
CALCSTATUS_PROCESS_FLOW_BIT4	<i>For internal use.</i>
CALCSTATUS_PROPORTION_IS_NODATA	The subcube cells that intersect the period and the [Proportion] Value dimension member contain no data.
CALCSTATUS_PROPORTION_NEEDS_CALC	The subcube cells that intersect the period and the [Proportion] Value dimension member require a calculation.

Transaction Dimension Constants

The following constants represent the transaction dimensions for statutory consolidations.

Table 41 tagTRANSACTION_DIMENSIONS Enumeration

Constant	Description
TRANSACTION_DIMENSIONS_CUR_ENTITY	The current entity.
TRANSACTION_DIMENSIONS_CUR_PARENT	The parent of the current entity.
TRANSACTION_DIMENSIONS_CUR_PERIOD	The current period.

Constant	Description
TRANSACTION_DIMENSIONS_DES_ACCOUNT	The destination account.
TRANSACTION_DIMENSIONS_DES_CUSTOM1	The destination member of the Custom 1 dimension.
TRANSACTION_DIMENSIONS_DES_CUSTOM2	The destination member of the Custom 2 dimension.
TRANSACTION_DIMENSIONS_DES_CUSTOM3	The destination member of the Custom 3 dimension.
TRANSACTION_DIMENSIONS_DES_CUSTOM4	The destination member of the Custom 4 dimension.
TRANSACTION_DIMENSIONS_DES_ENTITY	The destination entity.
TRANSACTION_DIMENSIONS_DES_ICP	The destination member of the Intercompany Partner dimension.
TRANSACTION_DIMENSIONS_DES_VALUE	The destination member of the Value dimension.
TRANSACTION_DIMENSIONS_SRC_ACCOUNT	The source account.
TRANSACTION_DIMENSIONS_SRC_CUSTOM1	The source member of the Custom 1 dimension.
TRANSACTION_DIMENSIONS_SRC_CUSTOM2	The source member of the Custom 2 dimension.
TRANSACTION_DIMENSIONS_SRC_CUSTOM3	The source member of the Custom 3 dimension.
TRANSACTION_DIMENSIONS_SRC_CUSTOM4	The source member of the Custom 4 dimension.
TRANSACTION_DIMENSIONS_SRC_ENTITY	The source entity.
TRANSACTION_DIMENSIONS_SRC_ICP	The source member of the Intercompany Partner dimension.
TRANSACTION_DIMENSIONS_SRC_PARENT	The parent of the source entity.
TRANSACTION_DIMENSIONS_SRC_PERIOD	The source period.
TRANSACTION_DIMENSIONS_SRC_SCENARIO	The source scenario.
TRANSACTION_DIMENSIONS_SRC_VALUE	The source member of the Value dimension.
TRANSACTION_DIMENSIONS_SRC_VIEW	The source member of the View dimension.
TRANSACTION_DIMENSIONS_SRC_YEAR	The source year.
TRANSACTION_DIMENSIONS_TOTAL	Represents the total number of transaction dimensions.

Cell Transaction Status Constants

The following constants represent transaction statuses for cells.

Table 42 tagTRANSACTIONTYPECONSTANTS Enumeration

Constant	Description
TRANSACTIONTYPE_AGGREGATED	The cell's data was calculated by aggregation.

Constant	Description
TRANSACTIONTYPE_CALCULATED	The cell's data was calculated by a rule.
TRANSACTIONTYPE_DERIVED	The cell's data was calculated by time intelligence.
TRANSACTIONTYPE_HASICDETAILTRANS	The cell has intercompany transactions.
TRANSACTIONTYPE_HASTRANSACTIONS	The cell contains line items.
TRANSACTIONTYPE_INPUT	The cell's data was input.
TRANSACTIONTYPE_NODATA	The cell contains no data.
TRANSACTIONTYPE_PARENT_LEVEL_INPUT	The cell's data has been manually entered at the parent entity level.

Calculation Status Statistic Constants

The following constants represent calculation status statistics.

Table 43 tagCALCSTATUSSTATISTICS Enumeration

Constant	Description
CALCSTATUS_STATSCOL_NODATA	Represents the NoData status.
CALCSTATUS_STATSCOL_OK	Represents the OK status.
CALCSTATUS_STATSCOL_OKSC	Represents the OKSC status.
CALCSTATUS_STATSCOL_CH	Represents the CH status.
CALCSTATUS_STATSCOL_CN	Represents the CN status.
CALCSTATUS_STATSCOL_TR	Represents the TR status.
CALCSTATUS_STATSCOL_OKND	Represents the OKND status.
CALCSTATUS_STATSCOL_CHND	Represents the CHND status.
CALCSTATUS_STATSCOL_CNND	Represents the CNND status.
CALCSTATUS_STATSCOL_TRND	Represents the TRND status.
CALCSTATUS_STATSCOL_LOCKED	Represents the Locked status.
CALCSTATUS_STATSCOL_ALL	Represents all calculation statuses.
CALCSTATUS_STATSCOL_MIN	Represents the first constant in this enumeration.
CALCSTATUS_STATSCOL_MAX	Represents the last constant in this enumeration.
CALCSTATUS_STATSCOL_UNKNOWN	Unknown status.

Journal-Related Constants

The HFMConstants type library includes constants that apply to journals and to journal templates. The following categories of journal-related constants are provided:

- “Journal Action Constants” on page 427
- “Balance Type Constants” on page 428
- “Journal Column Display Constants” on page 428
- “Debit/Credit Constants” on page 429
- “Period Status Constants” on page 429
- “Journal Report Display Option Constants” on page 429
- “Journal Report Sort Option Constants” on page 429
- “Journal Report Total Flag Constants” on page 430
- “Journal Status Constants” on page 430
- “Journal Type Constants” on page 430
- “Journal Dimension Members Display Option Constants” on page 431
- “Journal Web Session Parameter Constants” on page 431
- “Template Column Display Constants” on page 431
- “Template Type Constants” on page 432
- “Balance Type Filter Constants” on page 432
- “Journal Status Filter Constants” on page 432
- “Journal Type Filter Constants” on page 433

Journal Action Constants

The following constants represent journal actions.

Table 44 tagJOURNALACTION Enumeration

Constant	Description
JBA_APPROVE	Approve journal.
JBA_DELETE	Delete journal.
JBA_POST	Post journal.
JBA_REJECT	Reject journal.
JBA_SUBMIT	Submit journal.
JBA_UNPOST	Unpost journal.
JBA_UNSUBMIT	Unsubmit journal.

Balance Type Constants

The following constants represent balance types of journals.

Table 45 tagJOURNALBALANCETYPEFLAGS Enumeration

Constant	Description
JBTF_BALANCED	Balanced journal.
JBTF_BALANCED_BY_ENTITY	Balanced-by-entity journal.
JBTF_UNBALANCED	Unbalanced journal.

Journal Column Display Constants

The following constants represent journals' display columns.

Table 46 tagJOURNALCOLUMNSFORVIEWANDFILTER Enumeration

Constant	Description
COLUMN_JOURNALAPPROVEDBY	Approved By column.
COLUMN_JOURNALAPPROVEDON	Approved On column.
COLUMN_JOURNALBALANCEATTRIBUTE	Balance Type column.
COLUMN_JOURNALCREATEDBY	Created By column.
COLUMN_JOURNALCREATEDON	Date Created column.
COLUMN_JOURNALDESCRIPTION	Description column.
COLUMN_JOURNALGROUP	Group column.
COLUMN_JOURNALLABEL	Label column.
COLUMN_JOURNALLASTACTEDBY	Posted By column.
COLUMN_JOURNALLASTACTIONON	Date Posted column.
COLUMN_JOURNALLINEITEMENTITY	Entity column.
COLUMN_JOURNALLINEITEMPARENT	Parent column.
COLUMN_JOURNALSECURITYCLASS	Security Class column.
COLUMN_JOURNALSTATUS	Status column.
COLUMN_JOURNALTRUNCATEDDESCRIPTION	Short Description column.
COLUMN_JOURNALTYPE	Type column.
NUMJOURNALCOLUMNS	Represents the number of journal columns.

Debit/Credit Constants

The following constants represent debits and credits.

Table 47 tagDEBITCREDITUNIT Enumeration

Constant	Description
JE_TYPE_CREDIT	Credit.
JE_TYPE_DEBIT	Debit.
JE_TYPE_UNIT	<i>For internal use.</i>

Period Status Constants

The following constants represent period statuses of journals.

Table 48 tagJOURNALPERIODSTATUS Enumeration

Constant	Description
JPS_CLOSED	Closed period.
JPS_OPENED	Open period.
JPS_UNOPENED	Unopened period.

Journal Report Display Option Constants

The following constants represent display options for journal reports.

Table 49 tagJOURNALREPORTDISPLAYFLAGS Enumeration

Constant	Description
JOURNALREPORT_DISPLAY_NOREPEAT	NoRepeat.
JOURNALREPORT_DISPLAY_REPEAT	Repeat.
NUM_JOURNALREPORT_DISPLAY_FLAGS	Represents the number of available display options.

Journal Report Sort Option Constants

The following constants represent sorting options for journal reports.

Table 50 tagJOURNALREPORTSORTFLAGS Enumeration

Constant	Description
JOURNALREPORT_SORT_ASCENDING	Ascending sort order.

Constant	Description
JOURNALREPORT_SORT_DESCENDING	Descending sort order.
NUM_JOURNALREPORT_SORT_FLAGS	Represents the number of available sorting options.

Journal Report Total Flag Constants

The following constants represent options for journal report totalling.

Table 51 tagJOURNALREPORTTOTALFLAGS Enumeration

Constant	Description
JOURNALREPORT_TOTAL_NO	No totals.
JOURNALREPORT_TOTAL_YES	Yes – display totals.
NUM_JOURNALREPORT_TOTAL_FLAGS	Represents the number of available totalling options.

Journal Status Constants

The following constants represent journal statuses.

Table 52 tagJOURNALSTATUSFLAGS Enumeration

Constant	Description
JSF_APPROVED	Approved status.
JSF_POSTED	Posted status.
JSF_REJECTED	Rejected status.
JSF_SUBMITTED	Submitted status.
JSF_WORKING	Working status.

Journal Type Constants

The following constants represent journal types.

Table 53 tagJOURNALTYPEFLAGS Enumeration

Constant	Description
JTF_AUTOREVERSAL	System-generated autoreversing journal. When an autoreversing journal is posted, Financial Management automatically generates a journal in the <i>next</i> period. In this journal the debits and credits are transposed.
JTF_AUTOREVERSING	Autoreversing journal.

Constant	Description
JTF_REGULAR	Regular journal.
JTF_UNIT	<i>For internal use.</i>

Journal Dimension Members Display Option Constants

The following constants represent display options for dimension member labels and descriptions.

Table 54 tagJOURNALREPORTSHOWDESCRIPTIONFLAGS Enumeration

Constant	Description
JOURNALREPORT_SHOW_BOTH	Show both labels and descriptions.
JOURNALREPORT_SHOW_DESCRIPTION	Show only descriptions.
JOURNALREPORT_SHOW_LABEL	Show only labels.
NUM_JOURNALREPORT_SHOWDESCRIPTION_FLAGS	The total number of constants in this enumeration.

Journal Web Session Parameter Constants

The following constants represent default settings for Web sessions.

Table 55 tagJOURNALWEBSESSIONPARAMETERS Enumeration

Constant	Description
JWSP_BUFFER_SIZE	<i>For internal use.</i>
JWSP_NUM_ENTRIES_DISPLAYED	Represents the default number of journals that are displayed on a page.

Template Column Display Constants

The following constants represent display columns of journal templates.

Table 56 tagTEMPLATECOLUMNSFORVIEWANDFILTER Enumeration

Constant	Description
COLUMN_TEMPLATEBALANCEATTRIBUTE	Balance Type column.
COLUMN_TEMPLATEDESCRIPTION	Description column.
COLUMN_TEMPLATEGROUP	Group column.
COLUMN_TEMPLATELABEL	Label column.

Constant	Description
COLUMN_TEMPLATELINEITEMENTITY	Entity column.
COLUMN_TEMPLATELINEITEMPARENT	Parent column.
COLUMN_TEMPLATESECURITYCLASS	Security Class column.
COLUMN_TEMPLATETRUNCATEDDESCRIPTION	Short Description column.
COLUMN_TEMPLATETYPE	Type column.
COLUMN_TEMPLATEVALUEDIMENSION	Value column.
NUMTEMPLATECOLUMNS	Represents the number of available display columns.

Template Type Constants

The following constants represent types of journal templates.

Table 57 tagTEMPLATETYPEFLAGS Enumeration

Constant	Description
TTF_RECURRING	Recurring template.
TTF_STANDARD	Standard template.

Balance Type Filter Constants

The following constants represent balance type filters. For example, these are used with the `HFMwQueryDef.balanceTypeFilter` property.

Table 58 tagWEBOM_JOURNAL_BALANCE_TYPE_FILTER Enumeration

Constant	Description
WEBOM_JOURNAL_BALANCE_TYPE_BALANCED	Balanced journal filter.
WEBOM_JOURNAL_BALANCE_TYPE_BALANCED_BY_ENTITY	Balanced by entity journal filter.
WEBOM_JOURNAL_BALANCE_TYPE_UNBALANCED	Unbalanced journal filter.
WEBOM_JOURNAL_NUM_BALANCE_TYPE_FILTER	The total number of filters in this enumeration.

Journal Status Filter Constants

The following constants represent journal status filters. For example, these are used with the `HFMwQueryDef.statusFilter` property.

Table 59 tagWEBOM_JOURNAL_STATUS_FILTER Enumeration

Constant	Description
WEBOM_JOURNAL_STATUS_APPROVED	Approved status filter.
WEBOM_JOURNAL_STATUS_POSTED	Posted status filter.
WEBOM_JOURNAL_STATUS_REJECTED	Rejected status filter.
WEBOM_JOURNAL_STATUS_SUBMITTED	Submitted status filter.
WEBOM_JOURNAL_STATUS_WORKING	Working status filter.
WEBOM_JOURNAL_NUM_STATUS_FILTER	The total number of filters in this enumeration.

Journal Type Filter Constants

The following constants represent journal type filters. For example, these are used with the `HFMwQueryDef.typeFilter` property.

Table 60 tagWEBOM_JOURNAL_TYPE_FILTER Enumeration

Constant	Description
WEBOM_JOURNAL_TYPE_AUTOREVERSAL	System-generated autoreversing journal filter.
WEBOM_JOURNAL_TYPE_AUTOREVERSING	Autoreversing journal filter.
WEBOM_JOURNAL_TYPE_REGULAR	Regular journal filter.
WEBOM_JOURNAL_NUM_TYPE_FILTER	The total number of filters in this enumeration.

Process Management Constants

The `HFMConstants` type library provides the following types of process management-related constants:

- “Process Management Action Constants” on page 433
- “Process Management Review Level Constants” on page 434
- “Process Management Filters” on page 435
- “Process Management Validation Constants” on page 435
- “Process Management Sorting Constants” on page 436

Process Management Action Constants

The following constants represent process management actions.

Table 61 CEnumProcessFlowActions Enumeration

Constant	Description
PROCESS_FLOW_ACTION_APPROVE	Approve action.
PROCESS_FLOW_ACTION_PROMOTE	Promote action.
PROCESS_FLOW_ACTION_PUBLISH	Publish action.
PROCESS_FLOW_ACTION_REJECT	Reject action.
PROCESS_FLOW_ACTION_SIGN_OFF	Sign off action.
PROCESS_FLOW_ACTION_START	Start action.
PROCESS_FLOW_ACTION_SUBMIT	Submit action.
NUM_PROCESS_FLOW_ACTIONS	Represents the total number of process management actions.

Process Management Review Level Constants

The following constants represent process management review levels.

Table 62 CEnumProcessFlowStates Enumeration

Constant	Description
PROCESS_FLOW_STATE_APPROVED	Approved.
PROCESS_FLOW_STATE_FIRST_PASS	First pass.
PROCESS_FLOW_STATE_NOT_STARTED	Not started.
PROCESS_FLOW_STATE_NOT_SUPPORTED	Process management is not supported. This constant represents either of the following conditions: <ul style="list-style-type: none">● Process management is not enabled for the process unit's Scenario dimension member.● The connected user does not have the required security access for the process management operation.
PROCESS_FLOW_STATE_PUBLISHED	Published.
PROCESS_FLOW_STATE_REVIEW1	Review Level 1.
PROCESS_FLOW_STATE_REVIEW10	Review Level 10.
PROCESS_FLOW_STATE_REVIEW2	Review Level 2.
PROCESS_FLOW_STATE_REVIEW3	Review Level 3.
PROCESS_FLOW_STATE_REVIEW4	Review Level 4.
PROCESS_FLOW_STATE_REVIEW5	Review Level 5.
PROCESS_FLOW_STATE_REVIEW6	Review Level 6.

Constant	Description
PROCESS_FLOW_STATE_REVIEW7	Review Level 7.
PROCESS_FLOW_STATE_REVIEW8	Review Level 8.
PROCESS_FLOW_STATE_REVIEW9	Review Level 9.
PROCESS_FLOW_STATE_SUBMITTED	Submitted.
NUM_PROCESS_FLOW_STATES	Represents the total number of process management review levels.

Process Management Filters

The following constants represent process management filters.

Table 63 CEnumProcessFlowFilters Enumeration

Constant	Description
PROCESS_FLOW_FILTER_ABOVESTATE	Show only those process units with review levels above a given review level.
PROCESS_FLOW_FILTER_ALLSTATES	Show process units of all review levels.
PROCESS_FLOW_FILTER_BELOWSTATE	Show only those process units with review levels below a given review level.
PROCESS_FLOW_FILTER_FAILONLY	Show only process units that failed validation for promotion.
PROCESS_FLOW_FILTER_NONE	Do not filter by review level.
PROCESS_FLOW_FILTER_ONLYSTATE	Show only those process units with a given review level.
PROCESS_FLOW_FILTER_PASSFAIL	Do not filter process units by whether they passed or failed validation for promotion.
PROCESS_FLOW_FILTER_PASSONLY	Show only process units that passed validation for promotion.

Process Management Validation Constants

The following constants represent statuses related to validation for promotion.

Table 64 CEnumProcessFlowPassFailStatus Enumeration

Constant	Description
PROCESS_FLOW_STATUS_FAILED	Failed validation.
PROCESS_FLOW_STATUS_PASSED	Passed validation.
PROCESS_FLOW_STATUS_NOTSUPPORTED	<i>For internal use.</i>
NUM_PROCESS_FLOW_STATUS	The total number of constants in this enumeration.

Process Management Sorting Constants

The following constants represent sorting options for process units.

Table 65 CEnumProcessFlowSortOrders Enumeration

Constant	Description
PROCESS_FLOW_SORT_NONE	No sorting is applied.
PROCESS_FLOW_SORT_PASSFAILASC	Sort by validation status in ascending order.
PROCESS_FLOW_SORT_PASSFAILDESC	Sort by validation status in descending order.
PROCESS_FLOW_SORT_STATEASC	Sort by review level in ascending order.
PROCESS_FLOW_SORT_STATEDESC	Sort by review level in descending order.

Report Column Constants

The following constants represent columns in reports.

Table 66 tagREPORTECOLUMNSFORVIEWANDFILTER Enumeration

Constant	Description
COLUMN_REPORTAPPROVEDBY	Approved By column.
COLUMN_REPORTAPPROVEDON	Date Approved column.
COLUMN_REPORTBALANCEATTRIBUTE	Balance Type column.
COLUMN_REPORTCREATEDBY	Created By column.
COLUMN_REPORTCREATEDON	Date Created column.
COLUMN_REPORTDESCRIPTION	Description column.
COLUMN_REPORTGROUP	Group column.
COLUMN_REPORTLABEL	Label column.
COLUMN_REPORTLASTACTEDBY	Posted By column.
COLUMN_REPORTLASTACTIONON	Date Posted column.
COLUMN_REPORTLINEITEMACCOUNT	Account column.
COLUMN_REPORTLINEITEMCUSTOM1	Custom 1 column.
COLUMN_REPORTLINEITEMCUSTOM2	Custom 2 column.
COLUMN_REPORTLINEITEMCUSTOM3	Custom 3 column.
COLUMN_REPORTLINEITEMCUSTOM4	Custom 4 column.

Constant	Description
COLUMN_REPORTLINEITEMDESCRIPTION	Line item description constant.
COLUMN_REPORTLINEITEMENTITY	Entity column.
COLUMN_REPORTLINEITEMICP	Intercompany Partner column.
COLUMN_REPORTLINEITEMPARENT	Parent column.
COLUMN_REPORTSECURITYCLASS	Security Class column.
COLUMN_REPORTSTATUS	Status column.
COLUMN_REPORTTRUNCATEDDESCRIPTION	Short Description column.
COLUMN_REPORTTYPE	Type column.
NUMREPORTCOLUMNS	Represents the number of available report columns.

Consolidation Type Constants

The following constants represent types of consolidations.

Table 67 tagCONSOLIDATIONTYPE Enumeration

Constant	Description
CONSOLIDATE_ALL	Consolidate All.
CONSOLIDATE_ALLWITHDATA	Consolidate All With Data.
CONSOLIDATE_ENTITYONLY	Consolidate Entity Only.
CONSOLIDATE_FORCEENTITYONLY	Consolidate Force Entity Only.
CONSOLIDATE_IMPACTED	Consolidate Impacted.

Tier Constants

The following constants represent tiers of Financial Management's multi-tier architecture.

Table 68 tagHFM_TIERS Enumeration

Constant	Description
HFM_TIER1	Client tier.
HFM_TIER2	Application server tier.
HFM_WEB_TIER	Web server tier.

Security Constants

The following categories of constants represent security access rights, roles, and tasks:

- [“Access Rights Constants” on page 438](#)
- [“Role Constants” on page 438](#)
- [“Task Constants” on page 440](#)
- [“User Groups - User Type Flag Constants” on page 445](#)
- [“Identity Type Constants” on page 445](#)

Access Rights Constants

The following constants represent access rights.

Table 69 tagHFM_NUM_ACCESS_TYPES Enumeration

Constant	Description
HFM_ACCESS_RIGHTS_ALL	All
HFM_ACCESS_RIGHTS_NONE	None
HFM_ACCESS_RIGHTS_READANDPROMOTE	Promote
HFM_ACCESS_RIGHTS_READONLY	Read
HFM_ACCESS_RIGHTS_UNSPECIFIED	<i>For internal use.</i>
HFM_ACCESS_RIGHTS_VIEW	Metadata.

Role Constants

The following constants represent roles.

Table 70 tagHFM_ROLE_ENUM Enumeration

Constant	Description
HFM_NUM_ROLES	Represents the number of roles for an application.
HFM_ROLE_ADVANCED_USER	Advanced User
HFM_ROLE_APPLICATION_ADMINISTRATOR	Application Administrator
HFM_ROLE_APPROVE_JOURNALS	Approve Journals
HFM_ROLE_CONSOLIDATE_ALL	Consolidate All
HFM_ROLE_CONSOLIDATE_ALL_DATA	Consolidate
HFM_ROLE_CREATE_INTEGRATIONS	Create Integrations

Constant	Description
HFM_ROLE_CREATE_JOURNALS	Create Journals
HFM_ROLE_CREATE_UNBALANCED_JOURNALS	Create Unbalanced Journals
HFM_ROLE_DATA_FORM_WRITEBACK_EXCEL	Enable write back in Web Grid
HFM_ROLE_DB_MANAGEMENT	Database Management.
HFM_ROLE_DEFAULT	Default Role
HFM_ROLE_EXTENDED_ANALYTICS	Extended Analytics
HFM_ROLE_GENERATE_RECURRING	Generate Recurring
HFM_ROLE_IC_AUTO_MATCH_BY_ACCOUNT	Intercompany Transaction Auto Match by Account
HFM_ROLE_IC_AUTO_MATCH_BY_ID	Intercompany Transaction Auto Match by ID
HFM_ROLE_IC_MANUAL_MATCH	Intercompany Transaction Manual Match
HFM_ROLE_IC_MANUAL_MATCH_TOL	Intercompany Transaction Manual Match with Tolerance
HFM_ROLE_IC_MATCH_TEMPLATE	Intercompany Transaction Match Template
HFM_ROLE_IC_POST_UNPOST	Intercompany Transaction Post/Unpost
HFM_ROLE_IC_TRANSACTION_ADMIN	Intercompany Transaction Admin
HFM_ROLE_IC_TRANSACTION_USER	Intercompany Transaction User
HFM_ROLE_IC_UNMATCH	Intercompany Transaction Unmatch
HFM_ROLE_INTERCOMPANY_EMAIL_ALERTING	Receive Email Alerts for Intercompany Transactions
HFM_ROLE_JOURNALS_ADMINISTRATOR	Journals Administrator
HFM_ROLE_LOAD_EXCEL_DATA	Load Excel Data
HFM_ROLE_LOAD_SYSTEM	Load System
HFM_ROLE_LOCK_DATA	Lock Data
HFM_ROLE_MANAGE_CUSTOM_DOCUMENTS	Manage Custom Documents
HFM_ROLE_MANAGE_DATA_ENTRY_FORMS	Manage Data Forms
HFM_ROLE_MANAGE_MODELS	Manage Models
HFM_ROLE_MANAGE_OWNERSHIP	Manage Ownership
HFM_ROLE_MANAGE_TEMPLATES	Manage Templates
HFM_ROLE_POST_JOURNALS	Post Journals
HFM_ROLE_PROCESS_FLOW_REVIEWER1	Reviewer 1

Constant	Description
HFM_ROLE_PROCESS_FLOW_REVIEWER10	Reviewer 10
HFM_ROLE_PROCESS_FLOW_REVIEWER2	Reviewer 2
HFM_ROLE_PROCESS_FLOW_REVIEWER3	Reviewer 3
HFM_ROLE_PROCESS_FLOW_REVIEWER4	Reviewer 4
HFM_ROLE_PROCESS_FLOW_REVIEWER5	Reviewer 5
HFM_ROLE_PROCESS_FLOW_REVIEWER6	Reviewer 6
HFM_ROLE_PROCESS_FLOW_REVIEWER7	Reviewer 7
HFM_ROLE_PROCESS_FLOW_REVIEWER8	Reviewer 8
HFM_ROLE_PROCESS_FLOW_REVIEWER9	Reviewer 9
HFM_ROLE_PROCESS_FLOW_SUBMITTER	Submitter
HFM_ROLE_PROCESS_FLOW_SUPERVISOR	Review Supervisor
HFM_ROLE_PROCESS_MGMT_EMAIL_ALERTING	Receive Email Alerts for Process Management
HFM_ROLE_READ_JOURNALS	Read Journals
HFM_ROLE_RUN_ALLOCATION	Run Allocation
HFM_ROLE_RUN_CONSOLIDATION	Consolidate
HFM_ROLE_RUN_INTEGRATIONS	Run Integrations
HFM_ROLE_SAVE_SYSTEM_REPORT_ON_SERVER	Save System Report On Server
HFM_ROLE_PROVISIONING_MANAGER	Provisioning Manager .
HFM_ROLE_SECURITY_ADMINISTRATOR	Provisioning Manager (<i>deprecated</i>). This constant is deprecated; instead, use HFM_ROLE_PROVISIONING_MANAGER.
HFM_ROLE_TASK_AUTOMATION	Task Automation
HFM_ROLE_UNLOCK_DATA	Unlock Data
HFM_ROLE_WEBGRID_WRITE_BACK	Enable Write Back in Web Grid

Task Constants

The following constants represent tasks.

Table 71 tagHFM_TASK_ENUM Enumeration

Constant	Description
HFM_NUM_TASKS	Represents the number of tasks for an application.

Constant	Description
HFM_TASK_ADVANCED_USER_UI	Advanced User for UI
HFM_TASK_APPLICATION_CREATE_APPLICATION	Create Application
HFM_TASK_APPLICATION_DEFINE_APPLICATION_PROFILE	Define Application Profile
HFM_TASK_APPLICATION_DELETE_APPLICATION	Delete Application
HFM_TASK_CLOSE_APPLICATION	Close Application
HFM_TASK_CREATE_INTERCOMPANY_MATCHING_REPORT	Create Intercompany Matching Report
HFM_TASK_DATA_AUDIT	Data Audit
HFM_TASK_DATA_EXPLORER_ALLOCATE	Allocate
HFM_TASK_DATA_EXPLORER_CALCULATE	Calculate
HFM_TASK_DATA_EXPLORER_CALCULATE_CONTRIBUTION	Calculate Contribution
HFM_TASK_DATA_EXPLORER_CALCULATE_OWNERSHIP	Calculate Ownership
HFM_TASK_DATA_EXPLORER_CELL_ADJUSTMENTS	Cell Adjustments
HFM_TASK_DATA_EXPLORER_CELL_INFORMATION	Cell Information
HFM_TASK_DATA_EXPLORER_CELL_LINE_ITEM_DETAIL	Cell Line Item Detail
HFM_TASK_DATA_EXPLORER_CELL_TEXT	Cell Text
HFM_TASK_DATA_EXPLORER_COMMIT_DATA	Commit Data
HFM_TASK_DATA_EXPLORER_CONSOLIDATE	Consolidate
HFM_TASK_DATA_EXPLORER_CONSOLIDATE_ALL	Consolidate All
HFM_TASK_DATA_EXPLORER_CONSOLIDATE_ALL_WITH_DATA	Consolidate All with Data
HFM_TASK_DATA_EXPLORER_CONSOLIDATE_TRANSACTION	Consolidation Transaction
HFM_TASK_DATA_EXPLORER_FORCE_CALCULATE	Force Calculate
HFM_TASK_DATA_EXPLORER_FORCE_CALCULATE_CONTRIBUTION	Force Calculate Contribution
HFM_TASK_DATA_EXPLORER_FORCE_TRANSLATE	Force Translate
HFM_TASK_DATA_EXPLORER_GRID_SYSTEM_REPORT	Data Explorer Grid - System Report
HFM_TASK_DATA_EXPLORER_LOCK	Lock Data
HFM_TASK_DATA_EXPLORER_MANAGE_PROCESS	Manage Process
HFM_TASK_DATA_EXPLORER_OPEN_GRID	Open Grid
HFM_TASK_DATA_EXPLORER_REFRESH_DATA	Refresh Data

Constant	Description
HFM_TASK_DATA_EXPLORER_SAVE_SETTINGS	Save Data Explorer Settings
HFM_TASK_DATA_EXPLORER_SETTINGS	Data Explorer Settings
HFM_TASK_DATA_EXPLORER_TRANSLATE	Translate
HFM_TASK_DATA_EXPLORER_UNLOCK	Unlock Data
HFM_TASK_DATABASE_MANAGEMENT	Database Management
HFM_TASK_DB_MANAGEMENT	Database Management
HFM_TASK_ENTER_DATA	Enter Data
HFM_TASK_ENTER_SHARES_DATA	Enter Shares Data
HFM_TASK_ESSBASE	Oracle Essbase
HFM_TASK_EXPLORE_DATA	Explore Data
HFM_TASK_EXTENDED_ANALYTICS	Extended Analytics
HFM_TASK_EXTRACT_DATA	Extract Data
HFM_TASK_EXTRACT_JOURNALS	Extract Journals
HFM_TASK_EXTRACT_MEMBER_LISTS	Extract Member Lists
HFM_TASK_EXTRACT_METADATA	Extract Metadata
HFM_TASK_EXTRACT_RULES	Extract Rules
HFM_TASK_EXTRACT_SECURITY	Extract Security
HFM_TASK_ICT_AUTO_MATCH_BY_ACCOUNT	Intercompany Transaction Auto Match by Account
HFM_TASK_ICT_AUTO_MATCH_BY_ID	Intercompany Transaction Auto Match by ID
HFM_TASK_ICT_CREATE_TRANSACTIONS	Create Intercompany Transactions
HFM_TASK_ICT_DELETE_TRANSACTIONS	Delete Intercompany Transactions
HFM_TASK_ICT_DRILL_THROUGH	Drill Through Intercompany Transactions
HFM_TASK_ICT_EDIT_TRANSACTIONS	Edit Intercompany Transactions
HFM_TASK_ICT_EXTRACT_TRANSACTIONS	Extract Intercompany Transactions
HFM_TASK_ICT_LOAD_TRANSACTIONS	Load Intercompany Transactions
HFM_TASK_ICT_LOCK_UNLOCK_ENTITY	Intercompany Transactions - Lock/Unlock Entities
HFM_TASK_ICT_MANAGE_MATCHING_TEMPLATE	Intercompany Transaction - Manage Match Template
HFM_TASK_ICT_MANAGE_PERIODS	Intercompany Transaction - Manage Periods

Constant	Description
HFM_TASK_ICT_MANAGE_REASON_CODE	Intercompany Transaction - Manage Reason Codes
HFM_TASK_ICT_MANUAL_MATCH	Intercompany Transaction Manual Match
HFM_TASK_ICT_MANUAL_MATCH_TOL	Intercompany Transaction Manual Match with Tolerance
HFM_TASK_ICT_MATCH_REPORT_BY_ACCOUNT	Intercompany Transaction Match Report by Account
HFM_TASK_ICT_MATCH_REPORT_BY_ID	Intercompany Transaction Match Report by ID
HFM_TASK_ICT_POST_TRANSACTIONS	Post Intercompany Transactions
HFM_TASK_ICT_PROCESS_IC_TRANSACTIONS	Process Intercompany Transactions
HFM_TASK_ICT_TRANSACTION_REPORT	Intercompany Transaction Report
HFM_TASK_ICT_UNMATCH_TRANSACTIONS	Intercompany Transaction Unmatch
HFM_TASK_ICT_UNPOST_TRANSACTIONS	Intercompany Transaction Unpost
HFM_TASK_JOURNALS_ENUM_REC_TEMPLATES	Enumerate recurring journal templates.
HFM_TASK_JOURNALS_ENUM_STD_TEMPLATES	Enumerate standard journal templates.
HFM_TASK_JOURNALS_OPEN_REC_TEMPLATES	Open recurring journal templates.
HFM_TASK_JOURNALS_OPEN_STD_TEMPLATES	Open standard journal templates.
HFM_TASK_JOURNALS_PROCESS_JOURNALS	Process Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_ALLOW_UNBALANCED	Allow Unbalanced Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_APPROVE	Approve Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_DELETE	Delete Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_NEW	New Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_OPEN	Open Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_POST	Post Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_REJECT	Reject Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_SUBMIT	Submit Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_SYSTEM_REPORT	Journals System Report
HFM_TASK_JOURNALS_PROCESS_JOURNALS_UNPOST	Unpost Journals
HFM_TASK_JOURNALS_PROCESS_JOURNALS_UNSUBMIT	Unsubmit Journals
HFM_TASK_JOURNALS_SETUP_JOURNALS	Setup Journals
HFM_TASK_JOURNALS_SETUP_JOURNALS_GENERATE_RECURRING	Generate Recurring Journals

Constant	Description
HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_GROUPS	Manage Journal Groups
HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_PERIODS	Manage Periods
HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_TEMPLATES	Manage Templates
HFM_TASK_JOURNALS_SETUP_JOURNALS_TEMPLATES	Journal Templates
HFM_TASK_LOAD_DATA	Load Data
HFM_TASK_LOAD_DATA_REPLACE_BY_SECURITY	Load Data - Replace by Security mode
HFM_TASK_LOAD_JOURNALS	Load Journals
HFM_TASK_LOAD_MEMBER_LISTS	Load Member Lists
HFM_TASK_LOAD_METADATA	Load Metadata
HFM_TASK_LOAD_RULES	Load Rules
HFM_TASK_LOAD_SECURITY	Load Security
HFM_TASK_LOGOFF	Log Off
HFM_TASK_MANAGE_CUSTOM_DOCUMENTS	Manage Custom Documents
HFM_TASK_MANAGE_DATA_ENTRY_FORMS	Manage Data Forms
HFM_TASK_MANAGE_METADATA	Manage Metadata
HFM_TASK_MANAGE_OWNERSHIP	Manage Ownership
HFM_TASK_MANAGE_RULES	Manage Rules
HFM_TASK_NONE	<i>For internal use.</i>
HFM_TASK_OPEN_APPLICATION	Open Application
HFM_TASK_PIVOT_DATA	Pivot Data
HFM_TASK_RETRIEVE_DATA	Retrieve Data
HFM_TASK_RUN_DYNAMIC_INTERCOMPANY_MATCHING_REPORT	Run Dynamic Intercompany Matching Report
HFM_TASK_SECURITY_CONFIGURE_ACCESS	Configure Access
HFM_TASK_SECURITY_DEFINE_SECURITY_CLASSES	Define Security Classes
HFM_TASK_SECURITY_DEFINE_USERS_AND_GROUPS	Define Users and Groups
HFM_TASK_SYSTEM_REPORTS	System Reports
HFM_TASK_SYSTEM_REPORTS_SAVE_REMOTELY	Save File Remotely
HFM_TASK_TASK_AUDIT	Task Audit

Constant	Description
HFM_TASK_TASK_AUTOMATION	Task Automation
HFM_TASK_USER_PREFERENCES	User Preferences
HFM_TASK_USERS_ON_SYSTEM	Users on System
HFM_TASK_WEBGRID_WRITE_BACK	Enable write back in Web Grid

User Groups - User Type Flag Constants

Some HFMwSecurity methods use bitmasks that represent flags for types of users in user groups. The following table lists constants that represent the bits.

Table 72 tagHFM_USER_GROUP_ENUM_FLAGS Enumeration

User Type Flag	Constant
Exclude administrators	HFM_USER_GROUP_ENUM_EXCLUDEADMINS
User groups	HFM_USER_GROUP_ENUM_GROUPS
Include nested user groups	HFM_USER_GROUP_ENUM_RECURSIVE
Users	HFM_USER_GROUP_ENUM_USERS

Identity Type Constants

Some HFMwSecurity methods use bitmasks that represent *identity types*, which are types of users. The following table lists constants that represent the bits.

Table 73 tagHFM_IDENTITY_TYPE_FLAGS Enumeration

User Type	Constant
Application Administrator	HFM_IDENTITY_TYPE_APP_ADMIN
User Group	HFM_IDENTITY_TYPE_GROUP
User	HFM_IDENTITY_TYPE_USER
Invalid security identifier	HFM_IDENTITY_TYPE_INVALID_SID

Search Filters

The following table lists constants that represent bits for bitmasks in filtered searches, such as those performed by [EnumUsersWithFilter](#).

Table 74 HFM_USER_GROUP_ENUM_FLAGS Enumeration

Filter	Constant
Exclude application administrators	HFM_USER_GROUP_ENUM_EXCLUDEADMINS
Include groups	HFM_USER_GROUP_ENUM_GROUPS
Return groups recursively—groups within groups—if the HFM_USER_GROUP_ENUM_GROUPS flag is set.	HFM_USER_GROUP_ENUM_RECURSIVE
Include users	HFM_USER_GROUP_ENUM_FLAGS

Web Constants

The HFMConstants type library includes the following categories of constants for the Web components.

- [“Data Explorer Task Constants” on page 446](#)
- [“Data Explorer Process Management Constants” on page 447](#)
- [“Data Grid Definition Constants” on page 448](#)
- [“Data Grid Member Expansion Mode Constants” on page 448](#)
- [“Data Grid Dimension Expansion Mode Constants” on page 449](#)
- [“Data Grid Transaction Information Constants” on page 449](#)
- [“Data Information Display Constants” on page 449](#)
- [“Data Display Page Constant” on page 450](#)
- [“Member Display Constants” on page 450](#)
- [“Metadata Information Constants” on page 450](#)
- [“Document Type Constants” on page 451](#)
- [“Document File Type Constants” on page 452](#)
- [“Task ID Constants” on page 462](#)
- [“Process Control Grid Constants” on page 463](#)

Data Explorer Task Constants

The following constants represent Data Explorer tasks.

Table 75 tagWEBOM_DATAGRID_TASKMASK_ENUM Enumeration

Constant	Description
WEBOM_DATAGRID_TASK_ALLOCATE	Allocate.
WEBOM_DATAGRID_TASK_CALCULATE	Calculate.

Constant	Description
WEBOM_DATAGRID_TASK_CALCULATECONTRIBUTION	Calculate contribution.
WEBOM_DATAGRID_TASK_CELLADJUSTMENTS	Cell adjustments.
WEBOM_DATAGRID_TASK_CELLINFORMATION	Cell information.
WEBOM_DATAGRID_TASK_CELLLINEITEMDETAIL	Line item detail.
WEBOM_DATAGRID_TASK_CELLTEXT	Cell text.
WEBOM_DATAGRID_TASK_CONSOLIDATE	Consolidate.
WEBOM_DATAGRID_TASK_CONSOLIDATEALL	Consolidate all.
WEBOM_DATAGRID_TASK_CONSOLIDATEALLWITHDATA	Consolidate all with data.
WEBOM_DATAGRID_TASK_DESTINATIONTRANSACTIONS	Destination transactions.
WEBOM_DATAGRID_TASK_ENTITYDETAILREPORT	Entity detail report.
WEBOM_DATAGRID_TASK_FORCECALCULATE	Force calculate.
WEBOM_DATAGRID_TASK_FORCECALCULATECONTRIBUTION	Force calculate contribution.
WEBOM_DATAGRID_TASK_FORCETRANSLATE	Force translate.
WEBOM_DATAGRID_TASK_ICTRANSACTIONREPORT	Intercompany transaction report.
WEBOM_DATAGRID_TASK_LOCK	Lock subcube.
WEBOM_DATAGRID_TASK_MANAGEPROCESS	Manage process.
WEBOM_DATAGRID_TASK_SOURCETRANSACTIONS	Source transactions.
WEBOM_DATAGRID_TASK_TRANSLATE	Translate.
WEBOM_DATAGRID_TASK_UNLOCK	Unlock subcube.

Data Explorer Process Management Constants

The following constants represent process management actions for data grids.

Table 76 tagWEBOM_DATAGRID_PROCESSFLOWACTION_ENUM Enumeration

Constant	Description
WEBOM_DATAGRID_PROCESSFLOWACTION_APPROVE	Approve.
WEBOM_DATAGRID_PROCESSFLOWACTION_PROMOTE	Promote.
WEBOM_DATAGRID_PROCESSFLOWACTION_PUBLISH	Publish.
WEBOM_DATAGRID_PROCESSFLOWACTION_REJECT	Reject.

Constant	Description
WEBOM_DATAGRID_PROCESSFLOWACTION_SIGNOFF	Sign off.
WEBOM_DATAGRID_PROCESSFLOWACTION_START	Start.
WEBOM_DATAGRID_PROCESSFLOWACTION_SUBMIT	Submit.

Data Grid Definition Constants

The following constants represent the types of grid definition information that can be returned.

Table 77 tagWEBOM_DATAGRID_DEFINITION_INFO_FLAGS Enumeration

Constant	Description
WEBOM_DATAGRID_DEFINITION_INFO_ALL	Return all available information for the grid's definition.
WEBOM_DATAGRID_DEFINITION_INFO_COLDIMS	Return information regarding the grid's column dimension members.
WEBOM_DATAGRID_DEFINITION_INFO_EXPANSIONS	Return information regarding the grid's expanded dimensions.
WEBOM_DATAGRID_DEFINITION_INFO_POV	Display information regarding the grid's Point of View.
WEBOM_DATAGRID_DEFINITION_INFO_PROCESS	Display Process Management information for the grid.
WEBOM_DATAGRID_DEFINITION_INFO_ROWDIRS	Display information regarding the grid's row dimension members.
WEBOM_DATAGRID_DEFINITION_INFO_UISTATE	Display information regarding the grid's user interface state. This includes information on the currently focused cell and the row and column headers.

Data Grid Member Expansion Mode Constants

The following constants represent dimension member expansion modes applicable to the HFMwDataGrid component.

Table 78 tagWEBOM_DATAGRID_MEMBER_EXPANSION_MODES Enumeration

Constant	Description
WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_NONE	The dimension is a flat list, and no members support expansion.
WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_NO_CHILDREN	The member has no children.
WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_IS_EXPANDED	The member has children and is expanded.
WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_IS_COLLAPSED	The member has children and is collapsed.

Data Grid Dimension Expansion Mode Constants

The following constants represent dimension expansion modes applicable to the HFMwDataGrid component.

Table 79 tagWEBOM_DATAGRID_DIMENSION_EXPANSION_MODES Enumeration

Constant	Description
WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_COLLAPSED	The dimension is collapsed.
WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_EXPANDED	The dimension is expanded.
WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_INHERITED	The dimension's state is inherited from the previous row or column.
WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_NONE	No expansion mode applies, as the dimension is the rightmost dimension in the row or column.

Data Grid Transaction Information Constants

The following constants represent transaction information applicable to the HFMwDataGrid component.

Table 80 tagWEBOM_DATAGRID_TRANSACTION_INFO_FLAGS Enumeration

Constant	Description
WEBOM_DATAGRID_TRANSACTION_INFO_ALL	Return all transaction information.
WEBOM_DATAGRID_TRANSACTION_INFO_DESTINATION	Return information for destination transactions.
WEBOM_DATAGRID_TRANSACTION_INFO_HEADER	Return header information for the transactions. For example, header information includes the dimension members of the cell to which the transaction information applies, the current username, the cell's data, and so on.
WEBOM_DATAGRID_TRANSACTION_INFO_SOURCE	Return information for source transactions.

Data Information Display Constants

The following constants represent the types of information that can be displayed in the data grid.

Table 81 tagWEBOM_DATAGRID_DATA_DISPLAY_FLAGS Enumeration

Constant	Description
WEBOM_DATAGRID_DATA_DISPLAY_CALCSTATUS	Display calculation statuses.
WEBOM_DATAGRID_DATA_DISPLAY_CONTROL_PANEL_STATUS	Display process control status.

Constant	Description
WEBOM_DATAGRID_DATA_DISPLAY_DATA	Display data.
WEBOM_DATAGRID_DATA_DISPLAY_LASTUSED	Display the last previously displayed type of information.
WEBOM_DATAGRID_DATA_DISPLAY_PROCESSFLOWSTATE	Display process flow states.

Data Display Page Constant

The tagWEBOM_DATAGRID_PAGE_FLAGS enum contains the WEBOM_DATAGRID_PAGE_LASTUSED constant, which represents the last page accessed in the data grid.

Member Display Constants

The following constants represent the ways in which dimension members are displayed in the data grid.

Table 82 tagWEBOM_DATAGRID_METADATA_DISPLAY_FLAGS Enumeration

Constant	Description
WEBOM_DATAGRID_METADATA_DISPLAY_LABELS	Display labels only.
WEBOM_DATAGRID_METADATA_DISPLAY_DESCRIPTIONS	Display descriptions only.
WEBOM_DATAGRID_METADATA_DISPLAY_BOTH	Display both labels and descriptions.

Metadata Information Constants

The following constants represent types of metadata information.

Table 83 tagWEBOM_METADATA_INFO_FLAGS Enumeration

Constant	Description
WEBOM_METADATA_INFO_ALL	Represents all of the metadata information represented by the other constants in this enum.
WEBOM_METADATA_INFO_DESCRIPTION	The description of the specified item.
WEBOM_METADATA_INFO_ID	The internal ID of the specified item.
WEBOM_METADATA_INFO_LABEL	The label of the specified item.
WEBOM_METADATA_INFO_NUMCHILDREN	The number of children of the specified dimension member.
WEBOM_METADATA_INFO_SORT_DESCENDING	Sort members in descending order.
WEBOM_METADATA_INFO_SORTBY_DESC	Sort members by description.

Constant	Description
WEBOM_METADATA_INFO_SORTBY_ID	Sort members by member ID.
WEBOM_METADATA_INFO_SORTBY_LABEL	Sort members by label.

Document Type Constants

The following constants represent document types.

Table 84 tagDOCUMENTTYPES

Constant	Description
WEBOM_DOCTYPE_ALL	All document types represented by this enum.
WEBOM_DOCTYPE_CUSTOM	<i>For internal use.</i>
WEBOM_DOCTYPE_FOLDER	Folder.
WEBOM_DOCTYPE_INVALID	Invalid document.
WEBOM_DOCTYPE_LB	The lower bound of the constants in this enum.
WEBOM_DOCTYPE_LINK	Link.
WEBOM_DOCTYPE_RELATEDCONTENT	Related content.
WEBOM_DOCTYPE_RPTDATAEXPLORER	Data explorer report.
WEBOM_DOCTYPE_RPTHYP	<i>For internal use.</i>
WEBOM_DOCTYPE_RPTICMATCHBYACCOUNT	Intercompany matching by account report.
WEBOM_DOCTYPE_RPTICMATCHBYTRANSID	Intercompany matching by ID report.
WEBOM_DOCTYPE_RPTICMATCHINGTEMPLATE	Intercompany matching template report.
WEBOM_DOCTYPE_RPTICMONITOR	<i>For internal use.</i>
WEBOM_DOCTYPE_RPTICTRANSACTION	Intercompany matching by account report.
WEBOM_DOCTYPE_RPTINTERCOMPANY	Intercompany report.
WEBOM_DOCTYPE_RPTJOURNAL	Journal report.
WEBOM_DOCTYPE_TASK	Task in a task list.
WEBOM_DOCTYPE_UB	The count of document types that are represented by constants in this enum.
WEBOM_DOCTYPE_WEBFORM	Web form.
WEBOM_DOCTYPE_WEBGRID	Web grid.
WEBOM_DOCTYPE_WORKSPACE	Task list.

Document File Type Constants

The following constants represent the file types of documents.

Table 85 tagDOCUMENTFILETYPES Enumeration

Constant	Description
WEBOM_DOCFILETYPE_RPTDEF	Report definition file, RPT format.
WEBOM_DOCFILETYPE_RPTXML	Report definition file, XML format.
WEBOM_DOCFILETYPE_RPTHTML	Report definition file, HTML format.
WEBOM_DOCFILETYPE_FORMDEF	Web form.
WEBOM_DOCFILETYPE_FOLDER	Folder.
WEBOM_DOCFILETYPE_XML	XML file.
WEBOM_DOCFILETYPE_CUSTOM	<i>For internal use.</i>
WEBOM_DOCFILETYPE_TASK	Task in a task list.
WEBOM_DOCFILETYPE_ALL	All file types represented by this enum.
WEBOM_DOCFILETYPE_LB	The lower bound of the constants in this enum.
WEBOM_DOCFILETYPE_UB	The count of file types that are represented by constants in this enum.

Extracted File Encoding Constants

The following constants represent encoding types of extracted files.

Table 86 tagEXTRACTFILEENCODING Enumeration

Constant	Description
EXTRACT_FILE_ENCODING_ANSI	ANSI encoding.
EXTRACT_FILE_ENCODING_STREAM	Binary data encoding.
EXTRACT_FILE_ENCODING_UTF8	UTF-8 encoding.
EXTRACT_FILE_ENCODING_UTF16	UTF-16 encoding.
NUM_EXTRACT_FILE_ENCODINGS	Represents the number of available encoding types.

Task Status Constants

The following constants represent task statuses.

Table 87 tagUSERACTIVITYSTATUS Enumeration

Constant	Description
USERACTIVITYSTATUS_ABORTED	Task is aborted.
USERACTIVITYSTATUS_COMPLETED	Task is completed.
USERACTIVITYSTATUS_NOT_RESPONDING	Task is not responding.
USERACTIVITYSTATUS_PAUSED	Task is paused.
USERACTIVITYSTATUS_RUNNING	Task is running.
USERACTIVITYSTATUS_SCHEDULED_START	Task is scheduled to start.
USERACTIVITYSTATUS_SCHEDULED_STOP	Task is scheduled to stop.
USERACTIVITYSTATUS_STARTING	Task is starting.
USERACTIVITYSTATUS_STOPPED	Task is stopped.
USERACTIVITYSTATUS_STOPPING	Task is stopping.
USERACTIVITYSTATUS_UNDEFINED	Task status is unknown.
USERACTIVITYSTATUS__LBOUND	Represents the lower bounds of this enumeration.
USERACTIVITYSTATUS__UBOUND	Represents the upper bounds of this enumeration.
NUM_USERACTIVITYSTATUS	Represents the number of statuses represented by this enumeration.

User Activity Constants

The following constants represent user activities.

Table 88 tagUSERACTIVITYCODE Enumeration

Constant	Activity
NUM_USERACTIVITYCODE	Represents the number of constants in this enumeration.
USERACTIVITYCODE__LBOUND	Represents the lower bounds of this enumeration.
USERACTIVITYCODE__UBOUND	Represents the upper bounds of this enumeration.
USERACTIVITYCODE_ALLOCATE	Allocate.
USERACTIVITYCODE_APPLICATION_DELETION	Delete Application.
USERACTIVITYCODE_ATTACH_DOCUMENT	Attach Document.
USERACTIVITYCODE_CHART_LOGIC	Chart Logic.
USERACTIVITYCODE_CONSOLIDATION	Consolidation.

Constant	Activity
USERACTIVITYCODE_CUSTOM_LOGIC	Custom Logic.
USERACTIVITYCODE_DATA_AUDIT_PURGED	Purge Data Audit Records.
USERACTIVITYCODE_DATA_CLEAR	Data Clear.
USERACTIVITYCODE_DATA_COPY	Data Copy.
USERACTIVITYCODE_DATA_DELETE_INVALID_RECORDS	Delete Invalid Records.
USERACTIVITYCODE_DATA_ENTRY	Data Entry.
USERACTIVITYCODE_DATA_EXTRACT	Data Extract.
USERACTIVITYCODE_DATA_EXTRACT_HAL	Data Extract via HAL.
USERACTIVITYCODE_DATA_LOAD	Data Load.
USERACTIVITYCODE_DATA_RETRIEVAL	Data Retrieval.
USERACTIVITYCODE_DATA_SCAN	Data Scan.
USERACTIVITYCODE_DETACH_DOCUMENT	Detach Document.
USERACTIVITYCODE_EA_DELETE	Extended Analytics - delete.
USERACTIVITYCODE_EA_EXPORT	Extended Analytics - extract.
USERACTIVITYCODE_EXTERNAL	Custom activity.
USERACTIVITYCODE_IC_AUTOMATCHBYACCT	Automatch Intercompany Transactions by Account.
USERACTIVITYCODE_IC_AUTOMATCHBYID	Automatch Intercompany Transactions by ID.
USERACTIVITYCODE_IC_CREATE_TRANSACTIONS	Create Intercompany Transaction.
USERACTIVITYCODE_IC_DELETE_TRANSACTIONS	Delete Intercompany Transaction.
USERACTIVITYCODE_IC_DELETEALL	Delete All Intercompany Transactions.
USERACTIVITYCODE_IC_EDIT_TRANSACTIONS	Edit Intercompany Transaction.
USERACTIVITYCODE_IC_LOCKUNLOCK_ENTITIES	Lock and Unlock Entities.
USERACTIVITYCODE_IC_MANAGE_PERIODS	Manage Periods.
USERACTIVITYCODE_IC_MANAGE_REASONCODES	Manage Reason Codes.
USERACTIVITYCODE_IC_MANUALMATCH_TRANSACTIONS	Manual Match Intercompany Transactions.
USERACTIVITYCODE_IC_MATCHINGRPTBYACCT	Intercompany Transaction Matching Report by Account.
USERACTIVITYCODE_IC_MATCHINGRPTBYID	Intercompany Transaction Matching Report by ID.
USERACTIVITYCODE_IC_POST_TRANSACTIONS	Post Intercompany Transaction.

Constant	Activity
USERACTIVITYCODE_IC_POSTALL	Post All Intercompany Transactions.
USERACTIVITYCODE_IC_TRANSACTIONRPT	Intercompany Transaction Report.
USERACTIVITYCODE_IC_TRANSACTIONS_EXTRACT	Extract Intercompany Transaction.
USERACTIVITYCODE_IC_TRANSACTIONS_LOAD	Load Intercompany Transaction.
USERACTIVITYCODE_IC_UNMATCH_TRANSACTIONS	Unmatch Intercompany Transactions.
USERACTIVITYCODE_IC_UNMATCHALL	Unmatch All Intercompany Transactions.
USERACTIVITYCODE_IC_UNPOST_TRANSACTIONS	Unpost Intercompany Transaction.
USERACTIVITYCODE_IC_UNPOSTALL	Unpost All Intercompany Transactions.
USERACTIVITYCODE_IDLE	Idle - no current activity.
USERACTIVITYCODE_JOURNAL_ENTRY	Journal Entry.
USERACTIVITYCODE_JOURNAL_POSTING	Journal Posting
USERACTIVITYCODE_JOURNAL_RETRIEVAL	Journal Retrieval.
USERACTIVITYCODE_JOURNAL_TEMPLATE_ENTRY	Journal Template Entry.
USERACTIVITYCODE_JOURNAL_UNPOSTING	Journal Unposting.
USERACTIVITYCODE_LOGOFF	Logoff.
USERACTIVITYCODE_LOGON	Logon.
USERACTIVITYCODE_LOGON_FAILURE	Logon Failure.
USERACTIVITYCODE_MEMBER_LIST_EXTRACT	Member List Extract.
USERACTIVITYCODE_MEMBER_LIST_LOAD	Member List Load.
USERACTIVITYCODE_MEMBER_LIST_SCAN	Member List Scan.
USERACTIVITYCODE_METADATA_EXTRACT	Metadata Extract.
USERACTIVITYCODE_METADATA_LOAD	Metadata Load.
USERACTIVITYCODE_METADATA_SCAN	Metadata Scan.
USERACTIVITYCODE_RULES_EXTRACT	Rules Extract.
USERACTIVITYCODE_RULES_LOAD	Rules Load.
USERACTIVITYCODE_RULES_SCAN	Rules Scan.
USERACTIVITYCODE_SECURITY_EXTRACT	Security Extract.
USERACTIVITYCODE_SECURITY_LOAD	Security Load.

Constant	Activity
USERACTIVITYCODE_SECURITY_SCAN	Security Scan.
USERACTIVITYCODE_TASK_AUDIT_PURGED	Purge Task Audit Records.
USERACTIVITYCODE_TRANSLATION	Translate.

Entity Transaction Detail Report Constants

The following categories of constants apply to Entity Transaction Detail reports:

- [“Entity Transaction Detail Header Options” on page 456](#)
- [“Entity Transaction Detail Display Options” on page 456](#)
- [“Entity Transaction Detail Information” on page 457](#)
- [“Entity Transaction Detail Row Types” on page 458](#)

Entity Transaction Detail Header Options

The following constants represent the types of header information exposed for Entity Transaction Detail reports.

Table 89 tagENTITYDETAILSHEADER Enumeration

Constant	Description
ENTITYDETAILS_HEADER_CONSOLIDATION_MODEYTD	Consolidation mode YTD.
ENTITYDETAILS_HEADER_ENTITY_CURRENCY	Entity currency.
ENTITYDETAILS_HEADER_MISSINGDATA_JOURNALS	Missing data journals view.
ENTITYDETAILS_HEADER_PARENT_CURRENCY	Parent currency.
ENTITYDETAILS_HEADER_SCENARIO_VIEW	Scenario view.
ENTITYDETAILS_HEADER_TIME	Application server time.
ENTITYDETAILS_HEADER_TOTAL	The total number of members in the tagENTITYDETAILSHEADER enumeration. Use this constant to loop.

Entity Transaction Detail Display Options

The following constants represent row and column display options for Entity Transaction Detail reports.

Table 90 tagENTITYDETAILSOPTIONS Enumeration

Constant	Description
ENTITYDETAILS_ACCOUNT	Account.
ENTITYDETAILS_ADJUSTMENTS	Journals.
ENTITYDETAILS_AMOUNT	Amount.
ENTITYDETAILS_BASE_RECORDS	Base details.
ENTITYDETAILS_CREDIT	Credit.
ENTITYDETAILS_CUSTOM1	Custom1.
ENTITYDETAILS_CUSTOM2	Custom2.
ENTITYDETAILS_CUSTOM3	Custom3.
ENTITYDETAILS_CUSTOM4	Custom4.
ENTITYDETAILS_DEBIT	Debit.
ENTITYDETAILS_DEST_TRANSACTIONS	Destination transactions.
ENTITYDETAILS_ENTITY	Entity.
ENTITYDETAILS_ICP	ICP.
ENTITYDETAILS_ID	ID.
ENTITYDETAILS_LID	Line items.
ENTITYDETAILS_REMARKS	Remarks.
ENTITYDETAILS_SRC_TRANSACTIONS	Source transactions.
ENTITYDETAILS_VALUE	Value.

Entity Transaction Detail Information

The following constants represent the types of information contained in rows of Entity Transaction Detail reports.

Table 91 tagENTITYDETAILSDIMENSIONS Enumeration

Constant	Description
ENTITYDETAILS_DIMENSION_ACCOUNT	The label of the row's Account dimension member.
ENTITYDETAILS_DIMENSION_AMOUNT	The amount for the row.
ENTITYDETAILS_DIMENSION_CREDIT	The row's credit amount.

Constant	Description
ENTITYDETAILS_DIMENSION_CUSTOM1	The label of the row's Custom 1 dimension member.
ENTITYDETAILS_DIMENSION_CUSTOM2	The label of the row's Custom 2 dimension member.
ENTITYDETAILS_DIMENSION_CUSTOM3	The label of the row's Custom 3 dimension member.
ENTITYDETAILS_DIMENSION_CUSTOM4	The label of the row's Custom 4 dimension member.
ENTITYDETAILS_DIMENSION_DEBIT	The row's debit amount.
ENTITYDETAILS_DIMENSION_ENTITY	The label of the row's Entity dimension member.
ENTITYDETAILS_DIMENSION_ICP	The label of the row's Intercompany Partner dimension member.
ENTITYDETAILS_DIMENSION_ID	The row's ID.
ENTITYDETAILS_DIMENSION_REMARKS	The row's remarks.
ENTITYDETAILS_DIMENSION_ROWTYPE	The row type. Valid values are represented by the HFMCconstants type library constants listed in “Entity Transaction Detail Row Types” on page 458 .
ENTITYDETAILS_DIMENSION_TOTAL	The total number of members in the tagENTITYDETAILSDIMENSIONS enumeration. Use this constant to loop.
ENTITYDETAILS_DIMENSION_VALUE	The label of the row's Value dimension member.

Entity Transaction Detail Row Types

The following constants represent the types of rows contained by Entity Transaction Detail reports.

Table 92 tagENTITYDETAILSROWTYPES Enumeration

Constant	Description
ENTITYDETAILS_ROWTYPE_BASE	The row contains a data record.
ENTITYDETAILS_ROWTYPE_DESTINATION	The row contains destination transactions.
ENTITYDETAILS_ROWTYPE_JOURNALS	The row contains journals.
ENTITYDETAILS_ROWTYPE_LID	The row contains line item details.
ENTITYDETAILS_ROWTYPE_PRIOR	The row contains the previous period's data record.
ENTITYDETAILS_ROWTYPE_REGULAR	The row contains a detail data record.
ENTITYDETAILS_ROWTYPE_SOURCE	The row contains source transactions.

Constant	Description
ENTITYDETAILS_ROWTYPE_SUBTOTALS	The row contains subtotal amounts - for example, the sum of the cell's line item details.
ENTITYDETAILS_ROWTYPE_TOTAL	The row contains total amounts.

Member Selection Constants

The constants listed in the following topics are used for member selection with the HFMwMbrSel and HFMwMbrSelDim components:

- “Dimension Value Type Constants” on page 459
- “Point of View Attribute Constants” on page 459
- “Processing Directive ID Constants” on page 461
- “Validation Error Flags” on page 462

Dimension Value Type Constants

The following constants represent types of dimension value types. For example, these are used with the HFMwMbrSel methods [buildValue](#) and [parseValue](#).

Table 93 tagWEBOM_MBRSEL_VALTYPES Enumeration

Constant	Description
WEBOM_MBRSEL_VALTYPE_FUNCTION	Function.
WEBOM_MBRSEL_VALTYPE_LIST	Member list.
WEBOM_MBRSEL_VALTYPE_MEMBER	Member label.
WEBOM_MBRSEL_VALTYPE_NONE	Unknown type.

Point of View Attribute Constants

The following constants represent Point of View attributes for the HFMwMbrSel and HFMwMbrSelDim components.

Table 94 tagWEBOM_MBRSEL_IDS Enumeration

Constant	Description
WEBOM_MBRSEL_ID_ALLOWLISTVALUE	<code>allowlist</code> attribute, which specifies whether a member list can be used for a dimension's value. TRUE if member lists can be used, FALSE otherwise.
WEBOM_MBRSEL_ID_ALLOWMEMBERVALUE	<code>allowmember</code> attribute, which specifies whether a member label can be used for a dimension's value. TRUE if member labels can be used, FALSE otherwise.

Constant	Description
WEBOM_MBRSEL_ID_BASEONLY	<code>baseonly</code> attribute, which specifies whether a member must be a base member of the dimension hierarchy in order to be valid. TRUE if the member must be a base member, FALSE otherwise.
WEBOM_MBRSEL_ID_DIMID	<code>dimid</code> attribute, which stores the internal ID of a dimension. Valid values are represented by the constants listed in “Dimension ID Constants” on page 414 that represent dimensions.
WEBOM_MBRSEL_ID_ENABLED	<code>enabled</code> attribute, which determines whether a dimension’s member can be changed in the user interface. TRUE indicates the dimension member can be changed, FALSE otherwise.
WEBOM_MBRSEL_ID_FILTERHIDDEN	<code>filterhidden</code> attribute, which specifies whether a member list drop-down list is displayed. You can use this attribute if you are working with a UI similar to the Financial Management UI for selecting members. TRUE to display the drop-down, FALSE to hide it.
WEBOM_MBRSEL_ID_FILTERLISTNAME	<code>filterlist</code> attribute, which contains the name of the member list if the dimension value is a member list.
WEBOM_MBRSEL_ID_FILTERLOCKED	<code>filterlocked</code> attribute, which specifies whether the member must be a child of the dimension value specified in the <code>filterroot</code> attribute. TRUE if the member must be a child of <code>filterroot</code> , FALSE otherwise.
WEBOM_MBRSEL_ID_FILTERPATH	<code>filterpath</code> attribute. <i>For internal use.</i>
WEBOM_MBRSEL_ID_FILTERROOT	<code>filterroot</code> attribute, used if the <code>filterlocked</code> attribute is TRUE. If <code>filterlocked</code> is TRUE, <code>filterroot</code> contains a dimension value specifying a member list and, optionally, either the top member or a function.
WEBOM_MBRSEL_ID_FILTERTOPNAME	<code>filtertop</code> attribute, which contains the top member or function specified for the <code>filterroot</code> attribute.
WEBOM_MBRSEL_ID_MODE	<i>For internal use.</i>
WEBOM_MBRSEL_ID_NAME	<code>name</code> attribute, which stores a dimension’s name.
WEBOM_MBRSEL_ID_SELECTED	<code>selected</code> attribute. <i>For internal use.</i>
WEBOM_MBRSEL_ID_SORTORDER	<code>order</code> attribute, which stores the dimension’s position in the sort order in cases where this attribute overrides the system’s native sort order. Dimensions are sorted from the lowest to the highest <code>sortOrder</code> value. Tip: <code>HFMwMbrSel.SortOnly</code> indicates whether an <code>HFMwMbrSel</code> instance uses the <code>sortOrder</code> attribute.
WEBOM_MBRSEL_ID_VALUE	<code>val</code> attribute, which contains the value that specifies the member label, member list, or function that specifies the dimension’s members. When specifying this attribute, you must follow the same syntax rules for specifying member lists, members, and functions that the user interface requires. To ensure that you are complying with these rules, use <code>HFMwMbrSel.buildValue</code> .
WEBOM_MBRSEL_ID_VISIBLE	<code>visible</code> attribute, which indicates whether the dimension is currently visible in the user interface. TRUE indicates that the dimension is visible, FALSE that it is not.
WEBOM_MBRSEL_ID__ALL	<i>For internal use.</i>

Constant	Description
WEBOM_MBRSEL_ID__CNT	Represents the total number of Point of View attributes.
WEBOM_MBRSEL_ID__LBOUND	Represents the first attribute in the set of attributes; use this to loop through the attributes.
WEBOM_MBRSEL_ID__UBOUND	Represents the last attribute in the set of attributes.

Processing Directive ID Constants

The following constants represent processing directives provided by the HFMwMbrSel component.

Table 95 tagWEBOM_MBRSEL_DIR_IDS Enumeration

Constant	Description
WEBOM_MBRSEL_DIR_ID_ALLOW_DYN_POV_LIST	<i>For internal use.</i>
WEBOM_MBRSEL_DIR_ID_FUNCS	Boolean flag that indicates whether Year and Period dimensions allow relative keywords.
WEBOM_MBRSEL_DIR_ID_SECURITY_AS_PARTNER	Boolean flag that specifies whether to use Security by Partner entity security.
WEBOM_MBRSEL_DIR_ID_SELECTED	The index of the selected dimension index, or -1 for if no dimension is selected.
WEBOM_MBRSEL_DIR_ID_SORTED	Boolean flag that indicates whether to distinguish between sorted and unsorted dimensions, and to display the former in order.
WEBOM_MBRSEL_DIR_ID_SUMMARY	<i>For internal use.</i>
WEBOM_MBRSEL_DIR_ID_SUMMARY_LABEL	<i>For internal use.</i>
WEBOM_MBRSEL_DIR_ID_VALIDATE_CUSTOMS	Flag that specifies whether and how to validate Custom dimension members based on the Account member's Custom dimension attributes. Valid values are represented by the constants listed in “Validation Error Flags” on page 462 .
WEBOM_MBRSEL_DIR_ID_VALIDATE_ICP	Flag that specifies whether and how to validate Intercompany Partner dimension members based on the Account member's ICPTopMember attribute. Valid values are represented by the constants listed in “Validation Error Flags” on page 462 .
WEBOM_MBRSEL_DIR_ID_VALIDATE_NO_EMPTY	Boolean flag that indicates whether all dimensions must contain a value.
WEBOM_MBRSEL_DIR_ID_VALIDATE_ORG_BY_PERIOD	Flag that specifies whether and how to validate Entity dimension members by Organization By Period settings. Valid values are represented by the constants listed in “Validation Error Flags” on page 462 .
WEBOM_MBRSEL_DIR_ID_VALIDATE_PERIOD	Flag that specifies whether and how to validate Scenario members by the Period member's frequency. Valid values are represented by the constants listed in “Validation Error Flags” on page 462 .
WEBOM_MBRSEL_DIR_ID_VER	Returns the release number of Financial Management. This is a read-only directive.

Constant	Description
WEBOM_MBRSEL_DIR_ID__ALL	<i>For internal use.</i>
WEBOM_MBRSEL_DIR_ID__CNT	Represents the total number of IDs in this enumeration.
WEBOM_MBRSEL_DIR_ID__LBOUND	Represents the first ID in the enumeration's set of IDs.
WEBOM_MBRSEL_DIR_ID__UBOUND	Represents the last ID in the enumeration's set of IDs.

Validation Error Flags

The following constants represent valid values for some of the constants listed in [“Processing Directive ID Constants” on page 461](#).

Table 96 tagWEBOM_MBRSEL_ERRORS Enumeration

Constant	Description
WEBOM_MBRSEL_ERROR_HARD	Validate.
WEBOM_MBRSEL_ERROR_OFF	Do not validate.
WEBOM_MBRSEL_ERROR_SOFT	Validate, but allow overriding.

Task ID Constants

The following constants represent IDs of tasks.

Table 97 tagWEBOM_DOCMGR_TASKS Enumeration

Constant	Description
WEBOM_DOCMGR_TASKS_CONSOLIDATION_TEMPLATE	Consolidation template.
WEBOM_DOCMGR_TASKS_DATA_AUDIT	Data audit.
WEBOM_DOCMGR_TASKS_DATABASE_MANAGEMENT	Database management.
WEBOM_DOCMGR_TASKS_EXTENDED_ANALYTICS	Extended Analytics.
WEBOM_DOCMGR_TASKS_EXTRACT_DATA	Extract data.
WEBOM_DOCMGR_TASKS_EXTRACT_JOURNALS	Extract journals.
WEBOM_DOCMGR_TASKS_EXTRACT_MEMBER_LISTS	Extract member lists.
WEBOM_DOCMGR_TASKS_EXTRACT_METADATA	Extract metadata.
WEBOM_DOCMGR_TASKS_EXTRACT_RULES	Extract rules.
WEBOM_DOCMGR_TASKS_EXTRACT_SECURITY	Extract security.

Constant	Description
WEBOM_DOCMGR_TASKS_EXTRACT_TRANSACTIONS	Extract intercompany transactions.
WEBOM_DOCMGR_TASKS_JOURNAL_TEMPLATES	Journal templates.
WEBOM_DOCMGR_TASKS_JOURNALS	Process journals.
WEBOM_DOCMGR_TASKS_LOAD_DATA	Load data.
WEBOM_DOCMGR_TASKS_LOAD_JOURNALS	Load journals.
WEBOM_DOCMGR_TASKS_LOAD_MEMBER_LISTS	Load member lists.
WEBOM_DOCMGR_TASKS_LOAD_METADATA	Load metadata.
WEBOM_DOCMGR_TASKS_LOAD_RULES	Load rules.
WEBOM_DOCMGR_TASKS_LOAD_SECURITY	Load security.
WEBOM_DOCMGR_TASKS_LOAD_TRANSACTIONS	Load intercompany transactions.
WEBOM_DOCMGR_TASKS_LOCKUNLOCKIC_TRANSACTIONS	Lock and unlock intercompany transactions.
WEBOM_DOCMGR_TASKS_MANAGE_GROUPS	Manage groups.
WEBOM_DOCMGR_TASKS_MANAGE_IC_REASONCODES	Manage intercompany reason codes.
WEBOM_DOCMGR_TASKS_MANAGE_PERIODS	Manage periods.
WEBOM_DOCMGR_TASKS_MANAGEICPERIODS_TRANSACTIONS	Manage intercompany transactions.
WEBOM_DOCMGR_TASKS_MONITORIC_TRANSACTIONS	Monitor intercompany transactions.
WEBOM_DOCMGR_TASKS_OWNERSHIP_MANAGEMENT	Ownership management.
WEBOM_DOCMGR_TASKS_PROCESS_MANAGERMENT	Process control.
WEBOM_DOCMGR_TASKS_PROCESSIC_TRANSACTIONS	Process intercompany transactions.
WEBOM_DOCMGR_TASKS_RUNNING_TASKS	Running tasks.
WEBOM_DOCMGR_TASKS_TASK_AUDIT	Task audit.
WEBOM_DOCMGR_TASKS_ID__LBOUND	Represents the lower bounds of this enumeration.
WEBOM_DOCMGR_TASKS_ID__UBOUND	Represents the upper bounds of this enumeration.
WEBOM_DOCMGR_TASKS_ID__CNT	Represents the number of constants in this enumeration.

Process Control Grid Constants

The following constants represent types of information that can be displayed on a data grid.

Table 98 Enum tagWEBOM_PROCESS_CONTROL_GRID_COLUMNS

Constant	Description
WEBOM_PROCESS_CONTROL_GRID_COLUMN_CALCSTATUS	Calculation status.
WEBOM_PROCESS_CONTROL_GRID_COLUMN_JRLSTATUS	Journal status (as related to the <Entity Curr Adjs> member of the Value dimension).
WEBOM_PROCESS_CONTROL_GRID_COLUMN_JRLSTATUS_CONTRIB	Journal status (as related to the <Parent Curr Adjs> member of the Value dimension).
WEBOM_PROCESS_CONTROL_GRID_COLUMN_PASSFAIL	Pass/fail status.
WEBOM_PROCESS_CONTROL_GRID_COLUMN_REVIEWLEVEL	Review level.
WEBOM_PROCESS_CONTROL_GRID_COLUMN_VALIDATION	Validation status.

Miscellaneous Constants

The following constants have been categorized as “miscellaneous”:

- [“Showing Public and Private Documents” on page 464](#)
- [“Date and Time Format Constants” on page 465](#)
- [“Member ID Range” on page 466](#)
- [“Number Defaults Constants” on page 466](#)
- [“Share Calculation Ownership Constants” on page 466](#)
- [“Share Calculation Types Constants” on page 466](#)
- [“Validation Type Constants” on page 467](#)
- [“Default Security Class Constant” on page 468](#)
- [“Module ID Constants” on page 468](#)

Showing Public and Private Documents

The following constants represent whether to show public or private documents. For example, these are used with `HFMwManageDocuments.EnumDocumentsEx`.

Table 99 tagENUMSHOWPRIVATEDOCS Enumeration

Constant	Description
ENUMSHOWPRIVATEDOCS_DONTSHOW	Show only public documents.
ENUMSHOWPRIVATEDOCS_SHOW	Show only private documents.

Constant	Description
ENUMSHOWPRIVATEDOCS_SHOWALL	Show both public & private documents.
ENUMSHOWPRIVATEDOCS_ID__LBOUND	Represents the lower bounds of this enumeration.

Date and Time Format Constants

The following constants represent date and time formats. For example, these are used with `HsvResourceManager.GetFormattedDateTime`.

The following conventions are used to describe the formats:

- *DD* - Date with a leading zero.
- *MM* - Month with a leading zero.
- *YYYY* - Year (four digits).
- *hh* - 12-hour time format.
- *HH* - 24-hour time format.
- *MI* - Minutes with a leading zero.
- *SS* - Seconds with a leading zero.
- *TT* - AM or PM.

Table 100 tagHFM_DATE_TIME_FORMAT Enumeration

Constant	Description
HFM_DATE_TIME_FORMAT_DD_MM_YY	Represents the format <i>DD/MM/YYYY</i> .
HFM_DATE_TIME_FORMAT_DD_MM_YY_HH_MI_SS	Represents the format <i>DD/MM/YYYY hh:MI:SS TT</i> .
HFM_DATE_TIME_FORMAT_DD_MM_YY_HHHH_MI_SS	Represents the format <i>DD/MM/YYYY HH:MI:SS</i> .
HFM_DATE_TIME_FORMAT_HH_MI_SS	Represents the format <i>hh:MI:SS TT</i> .
HFM_DATE_TIME_FORMAT_HHHH_MI_SS	Represents the format <i>HH:MI:SS</i> .
HFM_DATE_TIME_FORMAT_MM_DD_YY	Represents the format <i>MM/DD/YYYY</i> .
HFM_DATE_TIME_FORMAT_MM_DD_YY_HH_MI_SS	Represents the format <i>MM/DD/YYYY hh:MI:SS TT</i> .
HFM_DATE_TIME_FORMAT_MM_DD_YY_HHHH_MI_SS	Represents the format <i>MM/DD/YYYY HH:MI:SS</i> .
HFM_DATE_TIME_FORMAT_YY_MM_DD	Represents the format <i>YYYY/MM/DD</i> .
HFM_DATE_TIME_FORMAT_YY_MM_DD_HH_MI_SS	Represents the format <i>YYYY/MM/DD hh:MI:SS TT</i> .
HFM_DATE_TIME_FORMAT_YY_MM_DD_HHHH_MI_SS	Represents the format <i>YYYY/MM/DD HH:MI:SS</i> .
HFM_DATE_TIME_FORMAT_TOTAL	Returns a count of the constants in this enumeration.

Member ID Range

The following constants represent the upper and lower limits of the member IDs that Financial Management uses to identify dimension members.

Table 101 tagMEMBERIDLIMITS Enumeration

Constant	Description
MEMBERHIGHESTAVAILABLE	The upper limit.
MEMBERLOWESTAVAILABLE	The lower limit.

Number Defaults Constants

The following constants represent Financial Management's defaults for certain properties of numbers.

Table 102 tagPRECISIONCONSTANTS Enumeration

Constant	Description
DEFAULT_NUM_DECIMALS	The default number of digits to the right of decimal characters.
DEFAULT_SCALE	The default scale.

Share Calculation Ownership Constants

The following constants represent the different modes used to calculate ownership percentages based on shares.

Note: The parent entity passed must be a valid parent entity for the application. However, for SHARESCALC_ALL_IN_APP the MEMBERNOTUSED constant can be passed instead of a valid parent.

Table 103 tagSHARESCALCULATIONMODES Enumeration

Constant	Description
SHARESCALC_ALL_BELOW	Perform the calculation for the specified parent and each of all the parents beneath the specified parent.
SHARESCALC_ALL_IN_APP	Perform the calculation for all the parents in the application.
SHARESCALC_PARENT_ONLY	Perform the calculation for only the specified parent.

Share Calculation Types Constants

The following constants represent the different calculations that can be performed.

Note: Multiple calculations can be specified by using the `Or` operator with the applicable constants; for example, `SHARESCALC_CONTROL Or SHARESCALC_DIRECTOWN`.

Table 104 tagSHARESCALCULATIONTYPES Enumeration

Constant	Description
SHARESCALC_CONTROL	Calculate the Percent Control based on shares for all the dependents of the specified parent.
SHARESCALC_DIRECTOWN	Calculate the Direct Percent Ownership based on shares for all the dependents of the specified parent.
SHARESCALC_METHOD	Propose a Method of consolidation based on shares for all the dependents of the specified parent.
SHARESCALC_OWNERSHIP	Calculate the Percent Ownership based on shares for all the dependents of the specified parent.
SHARESCALC_PERCENTCONSOL	Calculate the Percent Consolidation based on shares for all the dependents of the specified parent.

Validation Type Constants

The following constants represent various types of information that can be validated.

Table 105 HFM_VALIDATIONTYPE_ENUM Enumeration

Constant	Description
HFM_NUMVALIDATIONTYPES	Represents the total number of available validation types.
HFM_VALIDATIONTYPE_ANNOTATION	Validate whether the string represents a valid annotation.
HFM_VALIDATIONTYPE_APPLICATION_DESCRIPTION	Validate whether the string represents a valid application description.
HFM_VALIDATIONTYPE_APPNAME	Validate whether the string represents a valid application name.
HFM_VALIDATIONTYPE_CELL_DESCRIPTION	Validate whether the string represents a valid cell description.
HFM_VALIDATIONTYPE_CLUSTERNAME	Validate whether the string represents a valid cluster name.
HFM_VALIDATIONTYPE_DOCUMENTNAME	Validate whether the string represents a valid document name.
HFM_VALIDATIONTYPE_JOURNAL_GROUP_LABEL	Validate whether the string represents a valid journal group label.
HFM_VALIDATIONTYPE_JOURNAL_LINEITEM_DESCRIPTION	Validate whether the string represents a valid description for a journal line item.
HFM_VALIDATIONTYPE_JOURNALDESCRIPTION	Validate whether the string represents a valid journal description.
HFM_VALIDATIONTYPE_JOURNALNAME	Validate whether the string represents a valid journal label.
HFM_VALIDATIONTYPE_LBOUND	Represents the lower bounds of the total number of available validation types.
HFM_VALIDATIONTYPE_LINEITEM_DESCRIPTION	Validate whether the string represents a valid description for a cell line item.
HFM_VALIDATIONTYPE_LOADEXTRACT_DELIMITER	Validate whether the string represents a valid delimiter for a load or extract file.

Constant	Description
HFM_VALIDATIONTYPE_MEMBERDESCRIPTION	Validate whether the string represents a valid description of a dimension member.
HFM_VALIDATIONTYPE_MEMBERLABEL	Validate whether the string represents a valid dimension member label.
HFM_VALIDATIONTYPE_PROFILE_FILE	<i>For internal use.</i>
HFM_VALIDATIONTYPE_REPORTDESCRIPTION	Validate whether the string represents a valid description of a report.
HFM_VALIDATIONTYPE_REPORTNAME	Validate whether the string represents a valid report label.
HFM_VALIDATIONTYPE_SECURITYCLASS	Validate whether the string represents a valid security class label.
HFM_VALIDATIONTYPE_USERNAME	Validate whether the string represents a valid username.

Default Security Class Constant

The tagSECURITYCONSTANTS class contains the SECURITYCLASSDEFAULT constant, which represents the ID of the [Default] security class.

Module ID Constants

The following constants represent module IDs.

Table 106 tagMODULEIDS Enumeration

Constant	Module
MODULEID_ADMINISTRATION	Administration
MODULEID_CELL_HISTORY	Cell History
MODULEID_CLOSE_APPLICATION	Close Application
MODULEID_CONSOLIDATION	Consolidation
MODULEID_CREATE_APPLICATION	Create Application
MODULEID_DATA_AUDIT	Data Audit
MODULEID_DATA_ENTRY_FORMS	Data Forms
MODULEID_DATABASE_MANAGEMENT	Database Management
MODULEID_DELETE_APPLICATION	Delete Application
MODULEID_DOCUMENT_EXTRACT	Document Extract
MODULEID_DOCUMENTS	Documents
MODULEID_EDIT_INTERCOMPANY_TRANSACTION	Edit Intercompany Transaction
MODULEID_EMPTY_WORKSPACE	Empty Task list

Constant	Module
MODULEID_ERROR	Error
MODULEID_EXPLORE_DATA	Explore Data
MODULEID_EXTENDED_ANALYTICS	Extended Analytics
MODULEID_EXTRACT_DATA	Extract Data
MODULEID_EXTRACT_JOURNALS	Extract Journals
MODULEID_EXTRACT_MEMBERLISTS	Extract Member Lists
MODULEID_EXTRACT_METADATA	Extract Metadata
MODULEID_EXTRACT_RULES	Extract Rules
MODULEID_EXTRACT_SECURITY	Extract Security
MODULEID_EXTRACT_TRANSACTIONS	Extract Intercompany Transactions
MODULEID_FAVORITES	Favorites
MODULEID_HAL	Oracle's Hyperion® Application Link
MODULEID_HFM	Financial Management
MODULEID_HOME	Home
MODULEID_IC_TRANSACTIONS	Intercompany Transactions
MODULEID_ICTRANS_MATCHING_REPORT_BY_ACCOUNT	Intercompany Transactions - Matching Report By Account
MODULEID_ICTRANS_MATCHING_REPORT_BY_TRANSACTION_ID	Intercompany Transactions - Matching Report By Transaction ID
MODULEID_JOURNALS	Journals
MODULEID_LINK	Link
MODULEID_LOAD_DATA	Load Data
MODULEID_LOAD_JOURNALS	Load Journals
MODULEID_LOAD_MEMBERLISTS	Load Member Lists
MODULEID_LOAD_METADATA	Load Metadata
MODULEID_LOAD_RULES	Load Rules
MODULEID_LOAD_SECURITY	Load Security
MODULEID_LOAD_TRANSACTIONS	Load Intercompany Transactions
MODULEID_LOCK_AND_UNLOCK_ENTITIES	Lock and Unlock Entities
MODULEID_LOGOFF	Logoff

Constant	Module
MODULEID_LOGON	Logon
MODULEID_MANAGE_GROUPS	Manage Groups
MODULEID_MANAGE_IC_REASON_CODES	Manage Intercompany Transaction Reason Codes
MODULEID_MANAGE_PERIODS	Manage Periods
MODULEID_MANAGE_SERVERS_AND_APPLICATIONS	Manage Servers and Applications
MODULEID_MEMBER_SELECTOR	Member Selector
MODULEID_MONITOR_INTERCOMPANY_TRANSACTIONS	Monitor Intercompany Transactions
MODULEID_NEW_INTERCOMPANY_TRANSACTION	New Intercompany Transaction
MODULEID_OFFICE_ADDIN	Office Add-in
MODULEID_OWNERSHIP_MANAGEMENT	Manage Ownership
MODULEID_PREFERENCES	Preferences
MODULEID_PROCESS_CONTROL_PANEL	Process Control Panel
MODULEID_PROCESS_INTERCOMPANY_TRANSACTIONS	Process Intercompany Transactions
MODULEID_PROCESS_MANAGEMENT	Process Management
MODULEID_REGISTER_APPLICATION	Register Application with Oracle's Hyperion® Shared Services
MODULEID_REGISTER_SMARTVIEW_PROVIDER	Register Smart View Provider
MODULEID_RELATED_CONTENT	Related Content
MODULEID_REPORTS	Reports
MODULEID_RUNNING_TASKS	Running Tasks
MODULEID_SCAN_ICTRANSACTIONS	Scan Intercompany Transactions
MODULEID_SELECT_APPLICATION	Select Application
MODULEID_SELECT_CLUSTER	Select Cluster
MODULEID_SYSTEM_MESSAGE_DETAILS	System Message Details
MODULEID_SYSTEM_MESSAGES	System Messages
MODULEID_TASK_AUDIT	Task Audit
MODULEID_TASK_AUTOMATION	Task Automation
MODULEID_UNKNOWN	Unknown Module
MODULEID_USERS_ON_SYSTEM	Users on System

Constant	Module
MODULEID_WEB_DATA_ENTRY_FORM_BUILDER	Data Form Builder
MODULEID_WORKSPACE	Task lists
MODULEID_WORKSPACE_PREFERENCES	Task list Preferences



XML String DTD Reference

In This Appendix

Member List Enumerations DTD	473
Dimension Member Enumerations DTD	474
Data Grid Definition DTD	474
Data Grid Data and Headers DTD	477
Process Management Information DTD	480
Line Item Detail DTD	481
Cell Text DTD	481
Point of View DTD	482
Transaction Information DTD	483
Cell Adjustment DTD	486
Cell Information DTD	486
Cell Attachments DTD	487

This appendix contains DTDs for the XML strings used by various methods.

Member List Enumerations DTD

```
<!DOCTYPE MBRLSTS [  
  <!ELEMENT LST EMPTY>  
  <!ATTLIST LST id CDATA #IMPLIED>  
  <!ATTLIST LST name CDATA #IMPLIED>  

```

Element	Description
---------	-------------

<i>MBRLSTS</i>	The root element.
----------------	-------------------

<i>LST</i>	Represents a member list, and contains the following attributes: <ul style="list-style-type: none">● <i>id</i> = The internal ID of the member list.● <i>name</i> = Member list name.
------------	--

Dimension Member Enumerations DTD

```
<!DOCTYPE MBRS [  
  <!ELEMENT MBR EMPTY>  
  <!ATTLIST MBR id CDATA #IMPLIED>  
  <!ATTLIST MBR name CDATA #IMPLIED>  
  <!ATTLIST MBR desc CDATA " ">  
  <!ATTLIST MBR nc CDATA "0">  

```

Element Description

MBRS Root element.

MBR Represents a dimension member, and contains the following attributes:

- `id` = Member ID.
- `name` = Member label.
- `desc` = Member description in the current language.
- `nc` = A count of the member's children.

Note: The DTD is defined with default values for `desc` and `nc` so that these attributes do not need to be specified in the XML document. This keeps the size of the XML document to a minimum. The DTD is returned as part of the XML document.

Data Grid Definition DTD

```
<!DOCTYPE GRIDDEF [  
  <!ELEMENT GRIDDEF ANY >  
  <!ATTLIST GRIDDEF maxColsPerPage CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF maxRowsPerPage CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF metaDisplayMode CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF dataDisplayMode CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF rowSuppressNoData CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF rowSuppressZero CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF scale CDATA #IMPLIED>  
  <!ATTLIST GRIDDEF numDecPlaces CDATA #IMPLIED>  
  <!ELEMENT RDIMS (DIM*)>  
  <!ELEMENT CDIMS (DIM*)>  
  <!ELEMENT DIM EMPTY>  
  <!ATTLIST DIM idx CDATA #IMPLIED>  
  <!ATTLIST DIM name CDATA #IMPLIED>  
  <!ELEMENT POV (POVDIM*)>  
  <!ATTLIST POV obpFilterEnabled CDATA \"0\">  
  <!ATTLIST POV obpScenario CDATA \"\">  
  <!ATTLIST POV obpYear CDATA \"\">  
  <!ATTLIST POV obpPeriod CDATA \"\">  
  <!ELEMENT POVDIM EMPTY>  
  <!ATTLIST POVDIM name CDATA #IMPLIED>  
  <!ATTLIST POVDIM memberList CDATA #IMPLIED>  
  <!ATTLIST POVDIM topMember CDATA #IMPLIED>  
  <!ATTLIST POVDIM value CDATA #IMPLIED>
```

```

<!--ATTLIST POVDIM fixedMember CDATA #IMPLIED>
<!--ATTLIST POVDIM useMemberList CDATA #IMPLIED>
<!--ELEMENT PROCOPTIONS (FILTER*)>
<!--ATTLIST PROCOPTIONS listStyle CDATA #IMPLIED>
<!--ATTLIST PROCOPTIONS periodDisplay CDATA #IMPLIED>
<!--ATTLIST PROCOPTIONS allPeriodsColumn CDATA #IMPLIED>
<!--ELEMENT FILTER EMPTY>
<!--ATTLIST FILTER reviewLevelFilter CDATA #IMPLIED>
<!--ATTLIST FILTER reviewLevelCond CDATA #IMPLIED>
<!--ATTLIST FILTER reviewLevelSort CDATA #IMPLIED>
<!--ATTLIST FILTER passFailFilter CDATA #IMPLIED>
<!--ATTLIST FILTER consolStatusFilter CDATA #IMPLIED>
<!--ELEMENT REXPS (EXPRC*)>
<!--ELEMENT CEXPS (EXPRC*)>
<!--ELEMENT EXPRC EXPDET*>
<!--ATTLIST EXPRC seq CDATA #IMPLIED>
<!--ELEMENT EXPDET EMPTY>
<!--ATTLIST EXPDET didx CDATA #IMPLIED>
<!--ATTLIST EXPDET mbr CDATA #IMPLIED>
<!--ATTLIST EXPDET dexp CDATA \"0\">
<!--ATTLIST EXPDET mexp CDATA \"0\">
<!--ELEMENT UISTATE HDRWIDTH*>
<!--ATTLIST UISTATE currRow CDATA #IMPLIED>
<!--ATTLIST UISTATE currCol CDATA #IMPLIED>
<!--ELEMENT HDRWIDTH EMPTY>
<!--ATTLIST HDRWIDTH type CDATA #IMPLIED>
<!--ATTLIST HDRWIDTH idx CDATA #IMPLIED>
<!--ATTLIST HDRWIDTH width CDATA #IMPLIED>
]>

```

Element	Description
<i>GRIDDEF</i>	<p>Root element. <GRIDDEF> contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>maxColsPerPage</code> = Maximum columns per page (corresponds to the maxColsPerPage property). ● <code>maxRowsPerPage</code> = Maximum rows per page (corresponds to the maxRowsPerPage property). ● <code>metaDisplayMode</code> = Metadata display mode (corresponds to the metadataDisplayMode property). ● <code>dataDisplayMode</code> = Data display mode (corresponds to <code>sDataDisplayMode</code> argument of GetGridPageWithExpansionInfoAsXML). ● <code>rowSuppressNoData</code> = <i>For internal use.</i> ● <code>rowSuppressZero</code> = <i>For internal use.</i> ● <code>scale</code> = Scale (corresponds to the scale specified in the Data Explorer Grid Settings dialog box). ● <code>numDecPlaces</code> = Number of digits to the right of the decimal point.
<i>RDIMS</i>	Defines the row dimensions. An <RDIMS> tag contains one <DIM> tag for each row dimension.
<i>CDIMS</i>	Defines the column dimensions. A <CDIMS> tag contains one <DIM> tag for each column dimension.
<i>DIM</i>	<p>Represents a row or column dimension. <DIM> tags are nested in <RDIMS> and <CDIMS> tags and contain the following attributes:</p> <ul style="list-style-type: none"> ● <code>idx</code> = Dimension index. The index is relative to the other row or column dimensions (corresponds to the <code>IDimIndex</code> argument used in “Expand” methods such as ExpandRowDimension).

Element	Description
	<ul style="list-style-type: none"> ● <code>name</code> = Dimension name.
<i>POV</i>	<p>Point of view information for the grid. A <POV> tag contains a <POVDIM> tag for each dimension and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>obpFilterEnabled</code> = Indicates whether active entity filtering is enabled for Organization by Period applications. 0 indicates that filtering is disabled; a non-zero value indicates that filtering is enabled. ● <code>obpScenario</code> = The Scenario dimension member to use if filtering is enabled. ● <code>obpYear</code> = The Year dimension member to use if filtering is enabled. ● <code>obpPeriod</code> = The Period dimension member to use if filtering is enabled.
<i>POVDIM</i>	<p>Represents a dimension in a Point of View. <POVDIM> tags are nested in <POV> tags and contain the following attributes:</p> <ul style="list-style-type: none"> ● <code>name</code> = Dimension name. ● <code>memberList</code> = The label or internal ID of the member list for the dimension. ● <code>created before Release 4.0 of Financial Management</code> ● <code>topMember</code> = The name of the dimension's top member. ● <code>value</code> = The names of the dimension's members for the Point of View. If there are multiple members, the members are delimited by semicolons. ● <code>fixedMember</code> = The name of the fixed Point of View member, which is used for page dimensions. ● <code>useMemberList</code> = Indicates whether the Point of View uses a member list to specify the dimension's members. A non-zero value indicates that the dimension members are specified by <code>memberList</code>, while 0 indicates that the dimension members are specified by <code>value</code>.
<i>PROCOPTIONS</i>	<p>Represents the display options specified by the following attributes:</p> <ul style="list-style-type: none"> ● <code>listStyle</code> = A flag indicating whether the grid represents members as a tree or a flat list. 1 represents tree, 0 represents list. ● <code>periodDisplay</code> = A flag indicating whether the grid displays one period or all periods. 0 represents one period, 1 represents all. ● <code>allPeriodsColumn</code> = A flag indicating whether the grid displays process levels or calculation statuses. 0 represents process levels, 3 represents calculation statuses.
<i>FILTER</i>	<p>Represents the filtering and sorting options specified by the following attributes:</p> <ul style="list-style-type: none"> ● <code>reviewLevelFilter</code> = A number that represents the review level by which to filter. Valid values are represented by the HFMConstants type library enumeration <code>CEnumProcessFlowStates</code>, which is described in “Process Management Review Level Constants” on page 434. ● <code>reviewLevelCond</code> = A number that represents the condition by which to filter the review level in the <code>reviewLevelFilter</code> attribute. Valid values are represented by the following members of the HFMConstants enumeration <code>CEnumProcessFlowFilters</code>, which is described in “Process Management Filters” on page 435: <ul style="list-style-type: none"> ○ <code>PROCESS_FLOW_FILTER_ABOVESTATE</code> ○ <code>PROCESS_FLOW_FILTER_BELOWSTATE</code> ○ <code>PROCESS_FLOW_FILTER_ONLYSTATE</code> ● <code>reviewLevelSort</code> = A number that represents the order in which to sort process units in the grid. Valid values are represented by the HFMConstants enumeration <code>CEnumProcessFlowSortOrders</code>, which is described in “Process Management Sorting Constants” on page 436.

Element	Description
	<ul style="list-style-type: none"> ● <code>passFailFilter</code> = A number that represents the validation for promotion state by which to filter. Valid values are represented by the following members of the <code>HFMConstants</code> enumeration <code>CEnumProcessFlowFilters</code>, which is described in “Process Management Filters” on page 435: <ul style="list-style-type: none"> ○ <code>PROCESS_FLOW_FILTER_PASSFAIL</code> ○ <code>PROCESS_FLOW_FILTER_PASSONLY</code> ○ <code>PROCESS_FLOW_FILTER_FAILONLY</code> ● <code>consolStatusFilter</code> = A number that represents the consolidation statuses by which to filter. Valid values are represented by the <code>HFMConstants</code> enumeration tag <code>CALCSTATUSSTATISTICS</code>, which is described in “Calculation Status Statistic Constants” on page 426.
<i>REXPS</i>	Represents the grid’s expanded rows, and contains one <code><EXPRC></code> tag for each expanded row.
<i>CEXPS</i>	Represents the grid’s expanded columns, and contains one <code><EXPRC></code> tag for each expanded column.
<i>EXPRC</i>	Represents an expanded row or column. This element contains one <code><EXPDET></code> tag for each dimension member in the row or column. The <code><EXPRC></code> element also contains the <code>seq</code> attribute, which represents the row or column number.
<i>EXPDET</i>	Represents a dimension member in an expanded row or column, and contains the following attributes: <ul style="list-style-type: none"> ● <code>didx</code> = The index of the dimension member within the set of members in the row or column. ● <code>mbr</code> = This can be the member label, member description, or both, depending upon the value to which the <code>metaDisplayMode</code> attribute of <code><GRIDDEF></code> is set. ● <code>dexp</code> = The dimension expansion mode. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Data Grid Dimension Expansion Mode Constants” on page 449. ● <code>mexp</code> = The member expansion mode. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Data Grid Member Expansion Mode Constants” on page 448.
<i>UISTATE</i>	Represents the state of the user interface. This element contains one <code><HDRWIDTH></code> tag for each row and column header in the grid, as well as the following attributes: <ul style="list-style-type: none"> ● <code>currRow</code> = The row coordinate of the cell that currently has the focus. ● <code>currCol</code> = The column coordinate of the cell that currently has the focus.
<i>HDRWIDTH</i>	Represents a row’s or column’s header, and contains the following attributes: <ul style="list-style-type: none"> ● <code>type</code> = Indicates whether the header is for a row or column. The letter “r” indicates a row, the letter “c” indicates a column. ● <code>idx</code> = The index that identifies the header within the set of rows or column headers. ● <code>width</code> = The width of the header, measured in pixels.

Data Grid Data and Headers DTD

```

<!DOCTYPE GRID [
  <!ELEMENT GRID ANY>
  <!ATTLIST GRID hPageCnt CDATA #IMPLIED>
  <!ATTLIST GRID vPageCnt CDATA #IMPLIED>
  <!ATTLIST GRID hPageNum CDATA #IMPLIED>
  <!ATTLIST GRID vPageNum CDATA #IMPLIED>
  <!ELEMENT RHDRS (HDR*)>
  <!ATTLIST RHDRS count CDATA #IMPLIED>

```

```

<!ELEMENT CHDRS (HDR*)>
<!--ATTLIST CHDRS count CDATA #IMPLIED-->
<!ELEMENT HDR EMPTY>
<!--ATTLIST HDR idx CDATA #IMPLIED-->
<!--ATTLIST HDR dim CDATA #IMPLIED-->
<!--ATTLIST HDR txt CDATA ""-->
<!--ATTLIST HDR exp CDATA "0"-->
<!--ATTLIST HDR gen CDATA "0"-->
<!ELEMENT CELLS (CELL*)>
<!--ELEMENT CELL (CELLTEXT?, LINEITEMDETAIL?, CELLATTACHMENTS?)-->
<!--ATTLIST CELL row CDATA #IMPLIED-->
<!--ATTLIST CELL col CDATA #IMPLIED-->
<!--ATTLIST CELL value CDATA ""-->
<!--ATTLIST CELL status CDATA #IMPLIED-->
<!--ATTLIST CELL color CDATA #IMPLIED-->
<!--ATTLIST CELL tasks CDATA "0"-->
<!ELEMENT CELLTEXT EMPTY>
<!--ATTLIST CELLTEXT text CDATA #IMPLIED-->
<!ELEMENT LINEITEMDETAIL (LINES)>
<!ELEMENT LINES (LINE*)>
<!ELEMENT LINE EMPTY>
<!--ATTLIST LINE idx CDATA #IMPLIED-->
<!--ATTLIST LINE desc CDATA #IMPLIED-->
<!--ATTLIST LINE data CDATA #IMPLIED-->
<!ELEMENT CELLATTACHMENTS (DOCUMENTS)>
<!ELEMENT DOCUMENTS (DOCUMENT*)>
<!ELEMENT DOCUMENT EMPTY>
<!--ATTLIST DOCUMENT file CDATA #IMPLIED-->
<!--ATTLIST DOCUMENT path CDATA #IMPLIED-->
]>

```

Element	Description
<i>GRID</i>	<p>Root element, and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>hPageCnt</code> = Total number of horizontal pages in the grid. ● <code>vPageCnt</code> = Total number of vertical pages in the grid. ● <code>hPageNum</code> = The horizontal page number of the current page (0-based). ● <code>vPageNum</code> = The vertical page number of the current page (0-based).
<i>RHDRS</i>	<p>Row header information. An <code><RHDRS></code> tag contains one <code><HDR></code> tag for each row header.</p> <p><code><RHDRS></code> contains the <code>count</code> attribute, which represents the number of rows on the grid page.</p> <p>Note: If you are using SaveDirtyCellsOnLastGridPageFromXML, do not include <code><RHDRS></code>.</p>
<i>CHDRS</i>	<p>Column header information. A <code><CHDRS></code> tag contains one <code><HDR></code> tag for each column header.</p> <p><code><CHDRS></code> contains the <code>count</code> attribute, which represents the number of columns on the grid page.</p> <p>Note: If you are using SaveDirtyCellsOnLastGridPageFromXML, do not include <code><CHDRS></code>.</p>
<i>HDR</i>	<p>Represents a row or column header, and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>idx</code> = Row or column number (0-based).

Element	Description
	<ul style="list-style-type: none"> ● <code>dim</code> = Dimension index, which is the 0-based index within the row or column for the dimension (corresponds to the <i>IDimIndex</i> parameter used in “Expand” methods such as ExpandRowDimension). ● <code>txt</code> = Row or column label. This may be either a member label, a member description, or a combination of the two, depending on the metadata display mode. This attribute is empty for unexpanded dimensions. ● <code>exp</code> = The current expansion state, which is represented by the following values: <ul style="list-style-type: none"> –1 = The dimension not expanded. 0 = Members are shown as a flat list, and no expansion is supported. 1 = The member has no children, and cannot be expanded. 2 = The member has children and is expanded. 3 = The member has children and is not expanded. ● <code>gen</code> = The generation level within the hierarchy (0=top level). This attribute has meaning only for hierarchical dimensions such as the Entity dimension. <p>Note: If you are using SaveDirtyCellsOnLastGridPageFromXML, do not include <code><HDR></code>.</p>
<i>CELLS</i>	Represents the collection of cells in the data grid. A <code><CELLS></code> tag contains one <code><CELL></code> tag for each cell.
<i>CELL</i>	<p>Represents a cell in the grid. A <code><CELL></code> element can contain 0 or 1 <code><CELLTEXT></code> elements, 0 or 1 <code><LINEITEMDETAIL></code> elements, and 0 or more <code><CELLATTACHMENTS></code> elements. <code><CELL></code> contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>row</code> = The row number (0-based). ● <code>col</code> = The column number (0-based). ● <code>value</code> = The cell value. The value will vary depending on the data display mode, which is specified by the <i>sDisplayDataMode</i> argument of GetGridPageWithExpansionInfoAsXML. <p>Note: For data, <code>value</code> represents the numeric data value; for calculation status, <code>value</code> represents the status; for process state, <code>value</code> represents the current Process Management state.</p> <ul style="list-style-type: none"> ● <code>status</code> = The cell's status. Cell statuses are described in “About Cell Statuses” on page 25. ● <code>color</code> = Represents the color that identifies the cell's status. For details on valid values, see the description of the <i>pvarColors</i> argument for GetGridPage. ● <code>tasks</code> = Indicates the Data Explorer tasks allowed for this cell. The numbers that are valid values for <code>tasks</code> are represented by the <i>HFMConstants</i> type library constants listed in “Data Explorer Task Constants” on page 446. <p>Note: If you are using SaveDirtyCellsOnLastGridPageFromXML, include only the <code>row</code>, <code>col</code>, and <code>value</code> attributes, and do not include <code>status</code>, <code>color</code> and <code>tasks</code>.</p>
<i>CELLTEXT</i>	Represents cell text for the cell. The <code>text</code> attribute contains the text.
<i>LINEITEMDETAIL</i>	The parent element of line item details for the cell.
<i>LINES</i>	An enumeration of the line item details. <code><LINES></code> contains a <code><LINE></code> tag for each line item detail.
<i>LINE</i>	Represents a line item, and contains the following attributes:

Element	Description
	<ul style="list-style-type: none"> ● <code>idx</code> = The index of the line item. ● <code>data</code> = The line item's numeric value. ● <code>desc</code> = The line item's description.
<i>CELLATTACHMENTS</i>	The parent element of the cell's attachments. <code><CELLATTACHMENTS></code> contains a <code><DOCUMENTS></code> tag, which in turn contains the cell attachment.
<i>DOCUMENTS</i>	Contains one <code><DOCUMENT></code> element per cell attachment.
<i>DOCUMENT</i>	Represents a cell attachment, and contains the following attributes: <ul style="list-style-type: none"> ● <code>file</code> = The name of the attachment. ● <code>path</code> = The path of the attachment.

Process Management Information DTD

```
<!DOCTYPE PROCFLOWINFO [
  <!ELEMENT PROCFLOWINFO ANY >
  <!ATTLIST PROCFLOWINFO scenario CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO year CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO period CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO entity CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO parent CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO value CDATA #IMPLIED>
  <!ATTLIST PROCFLOWINFO userName CDATA #IMPLIED>
  <!ELEMENT ROLES (ROLE*)>
  <!ELEMENT ROLE EMPTY>
  <!ATTLIST ROLE name CDATA #IMPLIED>
  <!ELEMENT HISTORY (ACTION*)>
  <!ELEMENT ACTION EMPTY>
  <!ATTLIST ACTION time CDATA #IMPLIED>
  <!ATTLIST ACTION user CDATA #IMPLIED>
  <!ATTLIST ACTION action CDATA #IMPLIED>
  <!ATTLIST ACTION state CDATA #IMPLIED>
  <!ATTLIST ACTION comment CDATA #IMPLIED>
]>
```

Element	Description
<i>PROCFLOWINFO</i>	The root element, which contains the following attributes: <ul style="list-style-type: none"> ● <code>scenario</code> = The Scenario dimension member. ● <code>year</code> = The Year dimension member. ● <code>period</code> = The Period dimension member. ● <code>entity</code> = The Entity dimension member. ● <code>parent</code> = The parent of entity. ● <code>value</code> = The Value dimension member. ● <code>username</code> = The username of the current user.
<i>ROLES</i>	An enumeration of the current user's roles. <code><ROLES></code> contains a <code><ROLE></code> tag for each role.

Element	Description
<i>ROLE</i>	Represents a Process Management role to which the user is assigned. <ROLE> contains the <code>name</code> attribute, which specifies the role name.
<i>HISTORY</i>	A history of all Process Management actions for the cell. A <HISTORY> tag contains an <ACTION> tag for each action.
<i>ACTION</i>	Represents a Process Management action for the cell, and contains the following attributes: <ul style="list-style-type: none"> ● <code>time</code> = The date and time of the action. ● <code>user</code> = The username of the user who performed the action. ● <code>action</code> = A number that represents the action. The valid values for <code>action</code> are represented by the HFMConstants type library constants listed in “Data Explorer Process Management Constants” on page 447. ● <code>state</code> = A number that represents the Process Management state after the action was taken. Valid values are listed in the description for the <code>pvarsProcessState</code> argument of ProcessFlowGetInfo (HFMwDataGrid component). ● <code>comment</code> = The comments entered for the action.

Line Item Detail DTD

```
<!DOCTYPE LINEITEMDETAIL [
  <!ELEMENT LINEITEMDETAIL ANY >
  <!ELEMENT LINES (LINE*)>
  <!ELEMENT LINE EMPTY>
  <!ATTLIST LINE idx CDATA #IMPLIED>
  <!ATTLIST LINE data CDATA #IMPLIED>
  <!ATTLIST LINE desc CDATA #IMPLIED>
  <!ATTLIST LINE isNew CDATA \"0\">
]>
```

Element	Description
<i>LINEITEMDETAIL</i>	The root element.
<i>LINES</i>	An enumeration of the line item details. <LINES> contains a <LINE> tag for each line item detail.
<i>LINE</i>	Represents a line item, and contains the following attributes: <ul style="list-style-type: none"> ● <code>idx</code> = The index of the line item. ● <code>data</code> = The line item's numeric value. ● <code>desc</code> = The line item's description. ● <code>isNew</code> = Indicates whether the line item is a new entry.

Cell Text DTD

```
<!DOCTYPE CELLTEXT [
  <!ELEMENT CELLTEXT ANY >
  <ATTLIST CELLTEXT text CDATA #IMPLIED>
```

]>

Element	Description
---------	-------------

CELLTEXT	The root element for cell text.
----------	---------------------------------

<CELLTEXT> contains the `text` attribute, which represents the cell text.

Point of View DTD

```
<!DOCTYPE POV [ \
  <!ATTLIST POV obpFilteringEnabled CDATA \"0\"> \
  <!ATTLIST POV obpScenario CDATA \"\"> \
  <!ATTLIST POV obpYear CDATA \"\"> \
  <!ATTLIST POV obpPeriod CDATA \"\"> \
  <!ELEMENT DIMENSIONS (DIMENSION*)> \
  <!ELEMENT DIMENSION EMPTY> \
  <!ATTLIST DIMENSION name CDATA #IMPLIED> \
  <!ATTLIST DIMENSION id CDATA #IMPLIED> \
  <!ATTLIST DIMENSION memberListID CDATA #IMPLIED> \
  <!ATTLIST DIMENSION topMember CDATA \"\"> \
  <!ATTLIST DIMENSION topMemberFieldVisible CDATA \"-1\"> \
  <!ATTLIST DIMENSION enabled CDATA \"-1\"> \
  <!ATTLIST DIMENSION canChangeTopMember CDATA \"-1\"> \
  <!ATTLIST DIMENSION canChangeMemberList CDATA \"-1\"> \
  <!ATTLIST DIMENSION selected CDATA \"0\"> \
  <!ATTLIST DIMENSION selectType CDATA \"0\"> \
  <!ATTLIST DIMENSION value CDATA #IMPLIED> \
  <!ATTLIST DIMENSION memberType CDATA \"-1\"> \
  <!ATTLIST DIMENSION fixedPOVMember CDATA \"\"> \
  <!ATTLIST DIMENSION visible CDATA \"-1\"> \
  <!ATTLIST DIMENSION sortOrder CDATA \"0\"> \
  <!ATTLIST DIMENSION allowListValue CDATA \"0\"> \
  <!ATTLIST DIMENSION allowNonListValue CDATA \"-1\"> \
  <!ATTLIST DIMENSION topFilter CDATA \"-1\"> \
]>
```

Element	Description
---------	-------------

POV	The root element when this DTD is used by the HFMwPOV and HFMwDataGrid components.
-----	--

Note: When this DTD is used by the HFMwDimension component, <POV> and <DIMENSIONS> are not applicable, and <DIMENSION> is the root element.

The <POV> element contains the following attributes, which are applicable when this DTD is used with the HFMwPOV component:

- `obpFilteringEnabled` = Indicates whether active entities are filtered for display on the grid. 0 indicates filtering is off, -1 that filtering is on.
- `obpScenario` = The label of the Scenario dimension member by which to filter, if filtering is on.
- `obpYear` = The label of the Year dimension member by which to filter, if filtering is on.
- `obpPeriod` = The label of the Period dimension member by which to filter, if filtering is on.

Tip: To turn filtering on or off, or to specify filtering criteria, use [SetOrgByPeriodFilteringInfo](#).

Element	Description
DIMENSIONS	<p>An enumeration of the dimensions. <DIMENSIONS> contains a <DIMENSION> tag for each dimension.</p> <p>Note: When this DTD is used by the HFMwDimension object, <DIMENSION> is the root element.</p>
DIMENSION	<p>Represents the Point of View information for a dimension, and contains the following attributes:</p> <ul style="list-style-type: none"> ● <code>name</code> = The dimension name. ● <code>id</code> = The internal ID of the dimension. Valid values for <code>id</code> are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 414 that represent dimensions. ● <code>memberListID</code> = The internal ID of the member list for the dimension. ● <code>topMember</code> = The name of the top member in the dimension hierarchy. If no top member is specified in the Point of View, this is an empty string. ● <code>topMemberFieldVisible</code>: <i>For internal use.</i> ● <code>enabled</code> = Indicates whether the dimension’s member can be changed in the user interface. TRUE indicates the dimension member can be changed, FALSE that it cannot be changed. ● <code>canChangeTopMember</code>: <i>For internal use.</i> ● <code>canChangeMemberList</code> = <i>For internal use.</i> ● <code>selected</code>: <i>For internal use.</i> ● <code>selectType</code>: <i>For internal use.</i> ● <code>value</code> = The labels of the dimension’s members in the Point of View. If there are multiple members, the list is delimited by semicolons. ● <code>memberType</code> = <i>For internal use.</i> ● <code>fixedPOVMember</code> = The default member of the dimension. This attribute is used for page dimensions. ● <code>visible</code> = Indicates whether the dimension is currently visible in the user interface. TRUE indicates that the dimension is visible, FALSE that it is not. ● <code>sortOrder</code> = Indicates the dimension’s position in the sort order, in cases where this attribute overrides the system’s native sort order. ● <code>allowListValue</code> = <i>For internal use.</i> ● <code>allowNonListValue</code> = <i>For internal use.</i> ● <code>topFilter</code> = <i>For internal use.</i>

Transaction Information DTD

```

<!DOCTYPE TRANSINFO [
  <!ELEMENT HDRS (HDR*)>
  <!ATTLIST HDRS count CDATA #IMPLIED>
  <!ELEMENT HDR EMPTY>
  <ATTLIST HDR seq CDATA #IMPLIED>
  <ATTLIST HDR label CDATA #IMPLIED>
  <ATTLIST HDR value CDATA #IMPLIED>
  <!ELEMENT SRCS (SRC*)>
  <ATTLIST SRCS count CDATA #IMPLIED>
  <ATTLIST SRCS totDestData CDATA #IMPLIED>
  <ATTLIST SRCS totSrcData CDATA #IMPLIED>
  <!ELEMENT SRC EMPTY>
  <ATTLIST SRC currEntity CDATA #IMPLIED>
  <ATTLIST SRC period CDATA #IMPLIED>

```

```

<ATTLIST SRC parent CDATA #IMPLIED>
<ATTLIST SRC entity CDATA #IMPLIED>
<ATTLIST SRC value CDATA #IMPLIED>
<ATTLIST SRC account CDATA #IMPLIED>
<ATTLIST SRC icp CDATA #IMPLIED>
<ATTLIST SRC custom1 CDATA #IMPLIED>
<ATTLIST SRC custom2 CDATA #IMPLIED>
<ATTLIST SRC custom3 CDATA #IMPLIED>
<ATTLIST SRC custom4 CDATA #IMPLIED>
<ATTLIST SRC destData CDATA #IMPLIED>
<ATTLIST SRC srcData CDATA #IMPLIED>
<ATTLIST SRC factor CDATA #IMPLIED>
<ATTLIST SRC nature CDATA #IMPLIED>
<!ELEMENT DESTS (DEST*)>
<ATTLIST DESTS count CDATA #IMPLIED>
<ATTLIST DESTS totDestData CDATA #IMPLIED>
<ATTLIST DESTS totSrcData CDATA #IMPLIED>
<!ELEMENT DEST EMPTY>
<ATTLIST DEST currEntity CDATA #IMPLIED>
<ATTLIST DEST scenario CDATA #IMPLIED>
<ATTLIST DEST year CDATA #IMPLIED>
<ATTLIST DEST period CDATA #IMPLIED>
<ATTLIST DEST parent CDATA #IMPLIED>
<ATTLIST DEST entity CDATA #IMPLIED>
<ATTLIST DEST value CDATA #IMPLIED>
<ATTLIST DEST account CDATA #IMPLIED>
<ATTLIST DEST icp CDATA #IMPLIED>
<ATTLIST DEST view CDATA #IMPLIED>
<ATTLIST DEST custom1 CDATA #IMPLIED>
<ATTLIST DEST custom2 CDATA #IMPLIED>
<ATTLIST DEST custom3 CDATA #IMPLIED>
<ATTLIST DEST custom4 CDATA #IMPLIED>
<ATTLIST DEST destData CDATA #IMPLIED>
<ATTLIST DEST srcData CDATA #IMPLIED>
<ATTLIST DEST factor CDATA #IMPLIED>
<ATTLIST DEST nature CDATA #IMPLIED>
]>

```

Element Description

HDRS An enumeration of the transaction's header information. <HDRS> contains a <HDR> tag for each element of header information.

<HDRS> also contains the `count` attribute, which represents the number of <HDR> tags that are children to <HDRS>.

Tip: Header information includes the dimension members of the cell to which the transaction information applies, the current username, the cell's data, and so on.

HDR Represents an element of header information, and contains the following attributes:

- `seq` = The ordinal position of the <HDR> tag within the set of <HDR> tags that are children to <HDRS>.
- `label` = The label of the header information.
- `value` = The value that corresponds to the label.

SRCS An enumeration of the source transactions. <SRCS> contains a <SRC> tag for each source transaction, as well as the following attributes:

- `count` = The number of <SRC> tags that are children to <SRCS>.

Element Description

- `totDestData` = The sum of the destination transaction amounts in the `<SRC>` tags.
- `totSrcData` = The sum of the source transaction amounts in the `<SRC>` tags.

SRC Represents a source transaction, and contains the following attributes:

- `currEntity` = The label of the transaction's current Entity dimension member.
- `period` = The label of the transaction's Period dimension member.
- `parent` = The label of the transaction's parent Entity dimension member.
- `entity` = The label of the transaction's child Entity dimension member.
- `value` = The label of the transaction's Value dimension member.
- `account` = The label of the transaction's Account dimension member.
- `icp` = The label of the transaction's Intercompany Partner dimension member.
- `custom1` = The label of the transaction's Custom 1 dimension member.
- `custom2` = The label of the transaction's Custom 2 dimension member.
- `custom3` = The label of the transaction's Custom 3 dimension member.
- `custom4` = The label of the transaction's Custom 4 dimension member.
- `destData` = The destination transaction amount.
- `srcData` = The source transaction amount.
- `factor` = The factor used in the transaction.
- `nature` = The audit string specified for the transaction.

DESTS An enumeration of the destination transactions. `<DESTS>` contains a `<DEST>` tag for each destination transaction, as well as the following attributes:

- `count` = The number of `<DEST>` tags that are children to `<DESTS>`.
- `totDestData` = The sum of the destination transaction amounts in the `<DEST>` tags.
- `totSrcData` = The sum of the source transaction amounts in the `<DEST>` tags.

DEST Represents a destination transaction, and contains the following attributes:

- `currEntity` = The label of the transaction's current Entity dimension member.
- `scenario` = The label of the transaction's Scenario dimension member.
- `year` = The label of the transaction's Year dimension member.
- `period` = The label of the transaction's Period dimension member.
- `parent` = The label of the transaction's parent Entity dimension member.
- `entity` = The label of the transaction's child Entity dimension member.
- `value` = The label of the transaction's Value dimension member.
- `account` = The label of the transaction's Account dimension member.
- `icp` = The label of the transaction's Intercompany Partner dimension member.
- `view` = The label of the transaction's View dimension member.
- `custom1` = The label of the transaction's Custom 1 dimension member.
- `custom2` = The label of the transaction's Custom 2 dimension member.
- `custom3` = The label of the transaction's Custom 3 dimension member.
- `custom4` = The label of the transaction's Custom 4 dimension member.
- `destData` = The destination transaction amount.
- `srcData` = The source transaction amount.

Element	Description
---------	-------------

- | | |
|--|---|
| | <ul style="list-style-type: none">● <code>factor</code> = The factor used in the transaction.● <code>nature</code> = The audit string specified for the transaction. |
|--|---|

Cell Adjustment DTD

```
<!DOCTYPE CELLADJS [  
  <!ELEMENT CELLADJS (ADJ*, TOTAL*)>  
  <!ELEMENT ADJ EMPTY>  
  <!ATTLIST ADJ seq CDATA #IMPLIED>  
  <!ATTLIST ADJ value CDATA #IMPLIED>  
  <!ELEMENT TOTAL EMPTY>  
  <!ATTLIST TOTAL value CDATA #IMPLIED>  

```

Element	Description
---------	-------------

<i>CELLADJS</i>	Represents the cell's adjustments. <CELLADJS> contains a <DEST> tag for each cell adjustment, and provides the total adjustment amount in the <TOTAL> tag.
-----------------	--

<i>ADJ</i>	Represents a cell adjustment, and contains the following attributes:
------------	--

- | | |
|--|---|
| | <ul style="list-style-type: none">● <code>seq</code> = The ordinal position of the <ADJ> tag within the set of <ADJ> tags that are children to <CELLADJS>.● <code>value</code> = A description of the cell adjustment. The description includes the adjustment's amount and journal label. |
|--|---|

<i>TOTAL</i>	Represents the total amount of the cell's adjustments. The amount is returned by the <code>value</code> attribute.
--------------	--

Cell Information DTD

```
<!DOCTYPE CELLINFO [  
  <!ATTLIST CELLINFO procUnit CDATA #IMPLIED>  
  <!ATTLIST CELLINFO povDetail CDATA #IMPLIED>  
  <!ATTLIST CELLINFO view CDATA #IMPLIED>  
  <!ATTLIST CELLINFO calcStatus CDATA #IMPLIED>  
  <!ATTLIST CELLINFO procLevel CDATA #IMPLIED>  
  <!ATTLIST CELLINFO cellStatus CDATA #IMPLIED>  
  <!ATTLIST CELLINFO cellSecurity CDATA #IMPLIED>  
  <!ATTLIST CELLINFO dispData CDATA #IMPLIED>  
  <!ATTLIST CELLINFO fullResData CDATA #IMPLIED>  
  <!ATTLIST CELLINFO storedData CDATA #IMPLIED>  

```

Element	Description
---------	-------------

<i>CELLINFO</i>	The root element, which contains the following attributes:
-----------------	--

- | | |
|--|--|
| | <ul style="list-style-type: none">● <code>procUnit</code> = The labels of the dimension members that the cell's process unit consists of. The labels are delimited by commas. (Process units are described in "About Process Units" on page 24.) |
|--|--|

Element	Description
---------	-------------

- | | |
|--|--|
| | <ul style="list-style-type: none">● <code>povDetail</code> = The labels of the cell's Account, Intercompany Partner, and Custom 1-4 dimension members. The labels are delimited by commas.● <code>view</code> = The label of the cell's View dimension member.● <code>calcStatus</code> = A string description of the cell's calculation status.● <code>procLevel</code> = A string description of the cell's process management review level.● <code>cellStatus</code> = A string description of the cell's status.● <code>cellSecurity</code> = A string description of the cell's security access rights.● <code>dispData</code> = The cell's displayed data. |
|--|--|

Note: Displayed data is rounded to the number of decimal places specified for the cell's account and includes the user's thousands separator preference.

- | | |
|--|---|
| | <ul style="list-style-type: none">● <code>fullResData</code> = The cell's full resolution data. |
|--|---|

Note: Full resolution data includes the user's thousands separator preference.

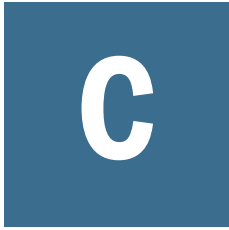
- | | |
|--|---|
| | <ul style="list-style-type: none">● <code>storedData</code> = The cell's stored data. |
|--|---|

Cell Attachments DTD

```
<!DOCTYPE CELLATTACHMENTS [  
  <!ELEMENT CELLATTACHMENTS ANY >  
  <!ELEMENT DOCUMENTS (DOCUMENT*) >  
  <!ELEMENT DOCUMENT EMPTY>  
  <!ATTLIST DOCUMENT file CDATA #IMPLIED>  
  <!ATTLIST DOCUMENT path CDATA #IMPLIED>  

```

Element	Description
<i>CELLATTACHMENTS</i>	The root element. <CELLATTACHMENTS> contains a <DOCUMENTS> tag, which in turn contains the cell attachment.
<i>DOCUMENTS</i>	Contains one <DOCUMENT> element per cell attachment.
<i>DOCUMENT</i>	Represents a cell attachment, and contains the following attributes: <ul style="list-style-type: none">● <code>file</code> = The name of the attachment.● <code>path</code> = The path of the attachment.



Change History

In This Appendix

Changes for Release 11.1.1.3.....	489
Changes for Release 11.1.1	490
Changes for Release 9.3.1.....	491
Changes for Release 9.3.0.1	492
Changes for Release 4.1	494
Changes for Release 4.0	496
Changes for Release 3.5.1.....	499
Changes for Release 3.5	500
Changes for Release 3.4	502
Changes for Release 3.0.4.....	506

This Appendix lists changes to the Financial Management Web object model that have occurred since the *Oracle Hyperion Financial Management, Fusion Edition Web Developer's Guide* was introduced in Release 3.0.

Changes for Release 11.1.1.3

The following topics list changes since release 11.1.1:

- [HFMwApplications](#)
- [HFMwData](#)
- [HFMwJournals](#)
- [HFMwSecurity](#)

HFMwApplications

The new method “[EnumUsersOnSystemEx2](#)” on page 42 has been added to the HFMwApplications component.

The new method “[GetDisableAspTechnicalErrorMessage](#)” on page 44 is for internal use.

HF MwData

The new method [“GetRegionsForCell”](#) on page 129 has been added to the HF MwData component:

HF MwJournals

The following methods have been added to the HF MwJournals component:

- [“ExecuteQueryLabelsAndIDs”](#) on page 356
- [“GetJournalsDisplayDataFromIDs”](#) on page 360
- [“GetJournalUsingIDs”](#) on page 361
- [“GetTemplatesDisplayDataFromIDs”](#) on page 365
- [“SaveJournal2”](#) on page 368

HF MwSecurity

The new method [“GetRulesMode”](#) on page 303 is for internal use.

Changes for Release 11.1.1

The following topics list changes since release 9.3.1:

[HF MwDataGrid Type Library](#)

[HF MwMetadata Type Library](#)

[HF MwSecurity Type Library](#)

[HF MwSession Type Library](#)

HF MwDataGrid Type Library

[“SetRowandColumnCount”](#) on page 250 has been added to the HF MwDataGrid component.

HF MwMetadata Type Library

The following methods have been added to the HF MwMetadata component:

- [“IsMemberAChildOf”](#) on page 66
- [“ApplicationSettingsTimeStamp”](#) on page 67
- [“ApplicationVersion”](#) on page 67
- [“ConsolidationMethodsTimeStamp”](#) on page 68
- [“CurrencyTimeStamp”](#) on page 68

- [“timestamp” on page 71](#)
- [“IsInputCell” on page 69](#)

HFMwSecurity Type Library

The following methods have been added to the HFMwSecurity component:

- [“EnumSecurityClassRightsAndRolesForBiPub” on page 297](#)
- [“EnumSecurityClassRightsForBiPub” on page 295](#)
- [“EnumUsersInGroupForBiPub” on page 301](#)
- [“EnumUsersRolesForBiPub” on page 300](#)
- [AllowRulesLoadForEPMAApp](#) is only for internal use.

HFMwSession Type Library

[“defaultProcessControlGridDef” on page 59](#) has been added to the HFMwSession component.

Changes for Release 9.3.1

The following topics list changes since release 9.3.0.1:

- [“HFMwSession Type Library” on page 491](#)
- [“HFMwData Type Library” on page 491](#)
- [“HFMwSecurity Type Library” on page 492](#)
- [“HFMwDataGrid Type Library” on page 492](#)

HFMwSession Type Library

[GetLicenseExpirationStatus](#) has been deprecated.

HFMwData Type Library

The following methods have been added to the HFMwManageProcess component:

- [GetPhaseSubmissionGroupAndPhaseForCell](#)
- [GetPhaseSubmissionReviewLevel](#)
- [PhasedSubmissionProcessManagementChangeStateForMultipleEntities](#)
- [PhaseSubmissionApprove](#)
- [PhaseSubmissionGetHistory](#)
- [PhaseSubmissionPromote](#)

- [PhaseSubmissionReject](#)
- [PhaseSubmissionSignOff](#)
- [PhaseSubmissionStart](#)
- [PhaseSubmissionSubmit](#)

HFMwSecurity Type Library

The following changes have been made:

- [GetUserInfoFromUniqueID2](#) has been added.
- [GetUserInfoFromUniqueID](#) has been deprecated in favor of [GetUserInfoFromUniqueID2](#).

HFMwDataGrid Type Library

The following methods and properties have been added:

- [GetUnassignedGroups](#)
- [colPhases](#)
- [processCtrlDispCols](#)

Changes for Release 9.3.0.1

The following topics list changes since release 4.1:

- [“Classic Applications” on page 492](#)
- [“HFMwApplications Type Library” on page 492](#)
- [“HFMwMetadata Type Library” on page 493](#)
- [“HFMwData Type Library” on page 493](#)
- [“HFMwDataGrid Type Library” on page 493](#)
- [“HFMwSecurity Type Library” on page 493](#)

Classic Applications

Some methods now are supported for only Classic applications. You can determine whether an application is a Classic application with the HFMwSecurity method [IsClassicHFMApplication](#).

HFMwApplications Type Library

The following changes have been made to the HFMwManageApplications component:

- The following methods have been added:
 - [IsValidApplication](#)
 - [EnumUserAppPreferences](#)
 - [GetDMEListenerConnectionInfo](#)
 - [UpdateUserAppPreferences](#)
- The following methods now apply only to Classic applications. For information on similar functionality for Oracle Hyperion EPM Architect, Fusion Edition see the *Hyperion Enterprise Performance Management Architect, Fusion Edition Web Services Developer's Guide* :
 - [CreateApplicationCAS](#)
 - [DeleteApplication](#)
 - [RegisterApplicationCAS](#)

HF Mw Metadata Type Library

The following properties have been added to the HF Mw Metadata component:

- [submissionPhaseFlag](#)
- [ValidationAccountByPhase](#)

HF Mw Data Type Library

The following methods have been added to the HF Mw ManageProcess component:

- [GetHistory2UsingPhaseId](#)
- [GetReviewLevelUsingPhaseID](#)

HF Mw DataGrid Type Library

The following members have been added to the HF Mw DataGrid component:

- [GetPhaseGroupInfoAsJavaScript](#)
- [GetProcessGridPageMultiWithExpansionInfoAsJavaScript](#)
- [phaseID](#)
- [SavePhaseGroupInfoUsingXML](#)
- [SetPhaseGroupInfoAsJavaScript](#)

HF Mw Security Type Library

The following changes have been made to the HF Mw Security component:

- The following methods have been added:

- [EnumUsersWithFilter](#)
- [IsClassicHFMApplication](#)
- The *bstrFilter* argument has been added to the following methods:
 - [EnumSecurityClassRightsAndRoles](#)
 - [EnumUsersRoles](#)

Changes for Release 4.1

The following topics list changes for Release 4.1:

- [“HFMwApplications Type Library” on page 494](#)
- [“HFMwSession Type Library” on page 494](#)
- [“HFMwMetadata Type Library” on page 494](#)
- [“HFMwSecurity Type Library” on page 495](#)
- [“HFMwDocuments Type Library” on page 495](#)
- [“HFMwSystemInfo Type Library” on page 496](#)
- [“HFMwDataGrid Type Library” on page 496](#)

HFMwApplications Type Library

The following changes have been made to the HFMwManageApplications component:

- The following methods have been added:
 - [CreateApplicationCAS](#)
 - [EnumProvisioningProjectNames](#)
 - [EnumUsersOnSystemEx](#)
 - [RegisterApplicationCAS](#)
- [CreateApplication](#) has been deprecated - use [CreateApplicationCAS](#).

HFMwSession Type Library

The following methods have been added to the HFMwSession component:

- [GetLicenseExpirationStatus](#)
- [HasSystemChanged](#)

HFMwMetadata Type Library

The following changes have been made to the HFMwMetadata type library:

- The [EnumMembers2](#) method has been added to the HFMwDimension component.

- The [EnumCurrencies](#) method has been added to the HFMwCurrencies component.
- The [ApplicationAttribute](#) property has been added to the HFMwMetadata component.

HFMwSecurity Type Library

The following changes have been made to the HFMwSecurity component:

- The following methods have been added:
 - [AddSecurityClasses](#)
 - [DeleteSecurityClasses](#)
 - [EnumSecurityClassRights](#)
 - [EnumSecurityClassRightsAndRoles](#)
 - [EnumUsers2](#)
 - [EnumUsersInGroup](#)
 - [EnumUsersRoles](#)
 - [GetSecurityClassAccessForAllUsers2](#)
 - [GetUserAccessForAllSecurityClasses2](#)
 - [GetUserInfoFromUniqueID](#)
 - [GetUserSID](#)
 - [IsConnectedUserInRole](#)
 - [LogInfo](#)
 - [SaveSecurityClassRights](#)
- [GetOwner](#) is no longer supported.
- The following methods have been deprecated:
 - [EnumUsers](#) - use [EnumUsers2](#).
 - [GetSecurityClassAccessForAllUsers](#) - use [GetSecurityClassAccessForAllUsers2](#).
 - [GetSecurityClassAccessForAllUsers](#) - use [GetSecurityClassAccessForAllUsers2](#).
 - [GetUserID](#) - use [GetUserSID](#).
- Users are now identified with security identifiers (SIDs) instead of numeric user IDs. All methods that had taken or returned user IDs are deprecated.

HFMwDocuments Type Library

The following changes have been made to the HFMwWorkspace component:

- The Boolean return value has been removed from [RemoveDocumentAt](#).
- The following members have been added:

- [SubWorkspace](#)
- [GetTaskNameAndUrlById](#)
- [GetTaskNameAndIdByUrl](#)

HF MwSystemInfo Type Library

The following methods have been added to the HF MwSystemInfo component:

- [EnumRunningTasksEx](#)
- [GetActivityCodeDesc](#)
- [SetCurrentModuleEx](#)

HF MwDataGrid Type Library

The following methods have been added to the HF MwDataGrid component:

- [ExecuteDataExplorerProcessManagementTask3](#)
- [GenerateAlerts](#)
- [GetCalcStatusSummaryAsXML](#)
- [GetOverlappingConsolidationInfoAsXML](#)
- [GetProcessGridPageWithExpansionInfoAsJavaScript](#)
- [SubmitProcessStateChangeFromXML](#)

The following properties have been added to the HF MwDataGrid component:

- [consolStatusFilter](#)
- [dataDisplayMode](#)
- [listStyle](#)

Changes for Release 4.0

The following topics list changes for Release 4.0:

- [“HF MwPOV Type Library - Removed” on page 497](#)
- [“HF MwMbrSel Type Library” on page 497](#)
- [“HF MwData Type Library” on page 497](#)
- [“HF MwMetadata Type Library” on page 497](#)
- [“HF MwDataGrid Type Library” on page 498](#)
- [“HF MwDocuments Type Library” on page 498](#)
- [“HF MwSecurity Type Library” on page 498](#)
- [“HF MwSession Type Library” on page 499](#)

- [“HFMwSystemInfo Type Library” on page 499](#)

HFMwPOV Type Library - Removed

The HFMwPOV type library has been removed. HFMwPOV has been replaced by the HFMwMbrSel type library, which provides similar functionality.

HFMwMbrSel Type Library

The HFMwMbrSel type library has been added. HFMwMbrSel enables you to get and set Points of View. For more information, see [Chapter 7, “HFMwMbrSel Type Library.”](#)

HFMwData Type Library

The following changes have been made to the HFMwData type library:

- The following changes have been made to the HFMwData component:
 - The first argument of [GetInputPeriods](#) now takes either a Scenario dimension member label or an array that specifies a Point of View. If a Point of View is specified, the return value flags read-only periods and omits periods in which line items cannot be entered.
 - The following HFMwData methods have been added:
 - [ClearData](#)
 - [ClearInvalidData](#)
 - [CopyData](#)
- The following methods have been added to the HFMwManageProcess component:
 - [Approve2](#)
 - [ChangeProcessManagementStateForMultipleEntities2](#)
 - [GetHistory2](#)
 - [Promote2](#)
 - [Publish2](#)
 - [Reject2](#)
 - [SignOff2](#)
 - [Start2](#)
 - [Submit2](#)

HFMwMetadata Type Library

The following changes have been made to the HFMwMetadata type library:

- The `HFMwCurrencies` component has been added. `HFMwCurrencies` exposes currency functionality and is described in “[HFMwCurrencies Component](#)” on page 101.
- The `GetDefaultParent` method has been added to the `HFMwDimension` component.
- The `GetDefaultCurrenciesForEntities` method has been added to the `HFMwEntities` component.
- The following items have been added to the `HFMwMetadata` component:
 - `ApplicationCurrency` property
 - `currencies` property
 - `EnableOrDisableUseSecurityAsPartner` method

HFMwDataGrid Type Library

- `SaveDirtyCellsOnLastGridPageFromXML` now takes an ASP Request object that contains an XML string. In previous releases, this method took an XML string.
- The following methods have been added to the `HFMwDataGrid` component:
 - `DefineGridUsingXMLFromRequest`
 - `ExecuteDataExplorerProcessManagementTask2`
 - `GetCellAttachmentsAsXML`
 - `GetConsolidationProgress`
 - `GetGridPageWithExpansionInfoAsJavaScript`
 - `ProcessFlowGetInfo2`
 - `SaveUIStateUsingXMLFromRequest`
 - `StartConsolidation`
 - `StopConsolidation`

HFMwDocuments Type Library

The following changes have been made to the `HFMwDocuments` type library:

- The `SaveDocument2` method has been added to the `HFMwManageDocuments` component.
- The `TaskFlow` property has been added to the `HFMwWorkspace` component.

HFMwSecurity Type Library

The `EnumSecurityClassesForConnectedUser` method has been added to the `HFMwSecurity` component.

HFMwSession Type Library

The following items have been added to the HFMwSession component. However, both items are only for internal use:

- [IsRunningTasks](#) method
- [IsBusy](#) property

HFMwSystemInfo Type Library

The following changes have been made to the HFMwSystemInfo component:

- The output argument for [EnumActivityUsers](#) has been removed; the usernames now are returned on the left-hand side.
- The *pvaravarModuleNames* argument has been added to [EnumAuditTasks](#).
- The *pvaravarModuleNames* argument has been added to [EnumAuditTasks2](#).
- The following methods have been added:
 - [EnumActivityServers](#)
 - [EnumRunningTasks](#)
 - [GetRunningTaskLogFromServerFile](#)
 - [GetRunningTaskProgress](#)
 - [GetRunningTaskStatus](#)
 - [StopRunningTasks](#)

Changes for Release 3.5.1

The following topics list changes for Release 3.5.1:

- “[HsvResourceManager Type Library](#)” on page 499
- “[Documentation Errors](#)” on page 500

HsvResourceManager Type Library

The following methods have been added to the HsvResourceManager component:

- [GetCurrentVersion](#)
- [GetFormattedDateTime](#)
- [GetFormattedErrorWithLineFeed](#)
- [GetFormattedResourceString](#)
- [GetHelpDirectoryForLanguageID](#)
- [GetHFMLanguageIdFromUserLanguages](#)

- [GetUserDisplayDataTimeFormats](#)
- [GetUserLanguageFromHFMLanguageId](#)
- [GetWindowsDateFormat](#)

Documentation Errors

The following documentation errors have been corrected:

- In “[Document File Type Constants](#)” on page 452, `WEBOM_DOCFILETYPE_UB` had previously been described as representing the upper bound of the constants in the enum. However, `WEBOM_DOCFILETYPE_UB` actually represents the count of file types that are represented by constants in the enum.
- In “[Document Type Constants](#)” on page 451, `WEBOM_DOCTYPE_UB` had previously been described as representing the upper bound of the constants in the enum. However, `WEBOM_DOCTYPE_UB` actually represents the count of document types that are represented by constants in the enum.

Changes for Release 3.5

The following topics list changes for Release 3.5:

- “[HFmwApplications Type Library](#)” on page 500
- “[HFmwData Type Library](#)” on page 500
- “[HFmwDocuments Type Library](#)” on page 501
- “[HFmwSecurity Type Library](#)” on page 501
- “[HFmwSystemInfo Type Library](#)” on page 501
- “[HFmwJournals Type Library](#)” on page 501
- “[HFmwDataGrid Type Library](#)” on page 502

HFmwApplications Type Library

The following methods have been added to the `HFmwManageApplications` component:

- [DoesUserHaveCreateApplicationRights](#)
- [DoesUserHaveSystemAdminRights](#)
- [GetSSOTokenUsingWebSecurityAgentCredentials](#)
- [IsWebSecurityAgentEnabled](#)

HFmwData Type Library

The following methods have been added to the `HFmwData` component:

- [GetCellHistory2](#)
- [EnumDataAuditItems2](#)

HF Mw Documents Type Library

The *lShowPrivateDocs* argument has been added to the HF Mw Manage Documents method [EnumDocumentsEx](#).

HF Mw Security Type Library

The [GetSecurityClassRightsForConnectedUser](#) method has been added to the HF Mw Security component.

HF Mw System Info Type Library

The following methods have been added to the HF Mw System Info component:

- [ClearAuditTasks2](#)
- [EnumAuditTasks2](#)

HF Mw Journals Type Library

The following methods have been added to the HF Mw Journal component:

- [GetNumberOfLineItems](#)
- [SetNumberOfLineItems](#)

The following methods have been deprecated in the HF Mw Journal component:

- [GetNumberOfLineItemEntries](#). Use [GetNumberOfLineItems](#) instead.
- [SetNumberOfLineItemEntries](#). Use [SetNumberOfLineItems](#) instead.

The following methods have been added to the HF Mw Query Def component:

- [GetReportDefinition](#)
- [InitializeFromReportDefinition](#)

The following properties have been added to the HF Mw Query Def component:

- [label](#)
- [period](#)
- [povString](#)
- [scenario](#)
- [security](#)
- [stylesheet](#)

- [value](#)
- [year](#)

HFMwDataGrid Type Library

The following properties have been added to the HFMwDataGrid component:

- [gridNumDecimals](#)
- [gridScale](#)

Changes for Release 3.4

The following topics list changes for Release 3.4:

- [“HFMwApplications Type Library” on page 502](#)
- [“HFMwSession Type Library” on page 502](#)
- [“HFMwData Type Library” on page 503](#)
- [“HFMwDocuments Type Library” on page 503](#)
- [“HFMwMetadata Type Library” on page 503](#)
- [“HFMwPOV Type Library” on page 504](#)
- [“HFMwSecurity Type Library” on page 505](#)
- [“HFMwJournals Type Library” on page 505](#)
- [“HFMwSystemInfo Type Library” on page 505](#)
- [“HFMwDataGrid Type Library” on page 505](#)
- [“HFMwUtilities Type Library” on page 506](#)

HFMwApplications Type Library

The following methods have been added to the HFMwManageApplications component:

- [EnableNewConnections](#)
- [EnumProhibitConnections](#)
- [EnumUsersOnSystem](#)
- [KillUsers](#)
- [WarnUsersForShutDown](#)

HFMwSession Type Library

The following methods have been added to the HFMwSession component:

- [GetUserSetting](#)

- [RemoveUserSetting](#)
- [SetUserSetting](#)

HF MwData Type Library

The following methods have been added to the HF MwData component:

- [ClearDataAuditItems](#)
- [EnumDataAuditItems](#)
- [GetCellHistory](#)
- [GetEntityDetails](#)
- [GetEntityDetailsHeader](#)

The following methods have been added to the HF MwManageProcess component:

- [ChangeProcessManagementStateForMultipleEntities](#)
- [GetProcessManagementRoleAccess](#)

HF MwDocuments Type Library

The HF MwWorkspace component has been added to expose Oracle Hyperion Financial Management, Fusion Edition task lists. For more information, see “[HF MwWorkspace Component](#)” on page 281.

The following methods have been added to the HF MwManageDocuments component:

- [CreateFolderEx](#)
- [DoesDocumentExist](#)
- [EnumDocumentsEx](#)
- [GetDocumentEx](#)
- [LoadDocumentsEx](#)
- [SaveDocumentEx](#)

HF MwMetadata Type Library

The [GetTopMemberOfCustomsForAccount](#) method has been added to the HF MwAccounts component.

The following methods have been added to the HF MwDimension component:

- [GetAttributeValue](#)
- [GetMemberDescription](#)

The [ValidationAccount](#) property has been added to the HF MwMetadata component.

The [defaultFrequency](#) property has been added to the HF MwScenarios component.

The following methods have been added to the HFMwPeriods component:

- [GetAncestorAtFrequency](#)
- [GetDescendantsAtFrequency](#)
- [GetFrequency](#)
- [GetPeriodsInFrequency](#)

HFMwPOV Type Library

The following methods and properties have been added to the HFMwPOV component:

- ClearFilters
- attributes
- count
- dim
- dimAttribute
- directive
- dump
- enabledOnly
- nextSortOrder
- scriptString
- sortedOnly
- valuedOnly
- visibleOnly
- xmlString

The following methods and properties have been deprecated in the HFMwPOV component:

- GetXML. Use xmlString instead.
- SetPOVUsingXML. Use xmlString instead.
- povDimension. Use dim instead.

The following properties have been added to the HFMwPOVDimension component:

- attribute
- dump

The following methods and properties have been deprecated in the HFMwPOVDimension component:

- GetXML. Use HFMwPOV.xmlString instead.
- canChangeMemberList. Use attribute instead.
- canChangeTopMember. Use attribute instead.

- enabled. Use attribute instead.
- fixedPOVMember. Use attribute instead.
- ID. Use attribute instead.
- memberListID. Use attribute instead.
- name. Use attribute instead.
- selected. Use attribute instead.
- selectType. Use attribute instead.
- topMember. Use attribute instead.
- topMemberFieldVisible. Use attribute instead.
- value. Use attribute instead.

HFMwSecurity Type Library

The following methods have been added to the HFMwSecurity component:

- [DoesConnectedUserHaveApplicationAdminRights](#)
- [EnumRoles](#)
- [EnumUsers](#)
- [GetOwner](#)
- [GetSecurityClassAccessForAllUsers](#)
- [GetSecurityClassID](#)
- [GetUserAccessForAllSecurityClasses](#)
- [GetUserID](#)

HFMwJournals Type Library

The HFMwJournals type library has been added. For more information, see [“HFMwJournals Type Library”](#) on page 333.

HFMwSystemInfo Type Library

The HFMwSystemInfo type library has been added. For information on its methods, see [Chapter 11, “HFMwSystemInfo Type Library.”](#)

HFMwDataGrid Type Library

The following methods have been added to the HFMwDataGrid component:

- [GetProcessManagementGridPage](#)
- [GetProcessManagementSummary](#)

- [SetProcessManagementFilterAndSortOptions](#)

The following properties have been added to the HFMwDataGrid component:

- [PassFailFilter](#)
- [ReviewLevelFilter](#)
- [ReviewLevelFilterCondition](#)
- [ReviewLevelSortOrder](#)
- [totalFilteredRowCount](#)
- [vFilteredPageCount](#)

HF Mw Utilities Type Library

The following changes have been made to the HFMwUtilities library's components"

- The [CalcCRC32](#) method has been added to the HFMwStringUtility component.
- The [WriteBinary](#) method has been added to the HFMwEncodedFileWriter component.

Changes for Release 3.0.4

The following topics list changes for Release 3.0.4:

- [“HF Mw Applications Type Library” on page 506](#)
- [“HF Mw Session Type Library” on page 506](#)
- [“HF Mw Metadata Type Library” on page 507](#)
- [“HF Mw Data Type Library” on page 507](#)
- [“HF Mw DataGrid Type Library” on page 507](#)
- [“HF Mw Utilities Type Library” on page 508](#)
- [“Miscellaneous Changes” on page 508](#)

HF Mw Applications Type Library

The following methods have been added to the HFMwApplications component:

- [SetLogonInfoSSO](#)
- [ValidateTokenOnClusterSSO](#)

SetLogonInfo has been deprecated. Use SetLogonInfoSSO instead.

HF Mw Session Type Library

The [RemoveFile](#) method has been added to the HFMwSession component.

The following properties have been added to the HFMwSession component:

- [extractEncoding](#)
- [showsDescriptions](#)
- [IsDebug](#)

HFMwMetadata Type Library

The [FindMemberOccurrencesInHierarchyEx](#) method has been added to the HFMwDimension component.

HFMwData Type Library

The following changes have been made to the HFMwData type library:

- The following methods have been added to the HFMwData component:
 - [GetFormattedNumber](#)
 - [GetDoubleFromText](#)
- The HFMwManageProcessFlow component has been added. For more information, see “[HFMwManageProcess Component](#)” on page 135.
- The HFMwCalculate component has been added. For more information, see “[HFMwCalculate Component](#)” on page 184.

HFMwDataGrid Type Library

The following methods have been added to the HFMwDataGrid component:

- [CollapseColumnDimension](#)
- [CollapseRowDimension](#)
- [DefinePOVUsingXML](#)
- [ExpandGridUsingXML](#)
- [GetCellAdjustments](#)
- [GetCellAdjustmentsAsXML](#)
- [GetCellInfo](#)
- [GetCellInfoAsXML](#)
- [GetDestinationTransactions](#)
- [GetGridDefinitionAsXMLEx](#)
- [GetSourceTransactions](#)
- [GetTransactionInfoAsXML](#)
- [SaveUIStateUsingXML](#)

- [SetOrgByPeriodFilteringInfo](#)

HF Mw Utilities Type Library

The HF Mw Utilities type library has been added. This library contains the following components:

- HF MwStringUtility, which is described in [“HF MwStringUtility Component” on page 381](#).
- HF MwFileReader, which is described in [“HF MwFileReader Component” on page 383](#).
- HF MwEncodedFileWriter, which is described in [“HF MwEncodedFileWriter Component” on page 386](#).

Miscellaneous Changes

- The error message lookup utility (`HF MwErrorLookup.exe`) is now located in the `Consultant Utilities` directory.
- Several elements have been added to the data grid definition DTD. This DTD is described in [“Data Grid Definition DTD” on page 474](#).

Index

Symbols

.per file, [31](#)

A

accounts (property), [67](#)
 actions, process management, constants for, [433](#)
 active entities, displaying, [249](#)
 activity user IDs, [25](#)
 AddDocument, [282](#)
 AddJournalGroup, [349](#)
 AddSecurityClasses, [291](#)
 AddTask, [282](#)
 adjustments, cell, returning, [219](#)
 as XML, [222](#)
 Administrator Group, [33](#)
 alerts, e-mail, [216](#)
 alias, [101](#)
 Allocate, [185](#)
 allowlist, Point of View attribute, [459](#)
 allowmember, Point of View attribute, [459](#)
 AllowRulesLoadForEPMAApp, [292](#)
 application creators, Creator Group, [33](#)
 application settings, getting, [67](#)
 ApplicationAttribute, [67](#)
 ApplicationCurrency, [67](#)
 applicationName, [58](#)
 applications
 Classic, testing whether, [307](#)
 creating, [31](#)
 deleting, [32](#)
 applications, Organization by Period
 filtering entities for, [249](#)
 indicating, [70](#)
 applications, statutory
 destination transactions, getting, [224](#)
 source transactions, getting, [238](#)
 transactions, as XML string, [240](#)

ApplicationSettingsTimeStamp, [67](#)
 ApplicationVersion, [67](#)
 ApplyUserSettings, [53](#)
 Approve, [136](#)
 Approve2, [137](#)
 arguments, input/output, [26](#)
 Attr, [199](#)
 AttrAll, [193](#)
 AttrDim, [193](#)
 audit, data
 ClearDataAuditItems, [106](#)
 EnumDataAuditItems, [110](#)
 EnumDataAuditItems2, [113](#)
 audit, task
 ClearAuditTasks, [311](#)
 EnumAuditTasks, [315](#)

B

balanceType
 HFMwJournal component, [336](#)
 HFMwTemplate component, [343](#)
 balanceTypeFilter, [372](#)
 baseonly, Point of View attribute, [460](#)
 buildValue, [190](#)

C

CalcCRC32, [381](#)
 calculate ownership, [390](#)
 calculate, ChartLogic method for, [185](#)
 calculation statuses
 cells, constants for, [421](#)
 subcube periods, constants for, [423](#)
 cell adjustments, returning, [219](#)
 as XML, [222](#)
 cell data, methods for, [105](#)
 cell statuses, [25](#)
 constants for, [421](#)

CELLSTATUS_NOREADACCESS, [423](#)
 ChangeProcessManagementStateForMultipleEntities
 , [138](#)
 ChangeProcessManagementStateForMultipleEntities
 2, [140](#)
 ChartLogic, [185](#)
 Classic application, testing whether, [307](#)
 ClearAuditTasks, [311](#)
 ClearAuditTasks2, [312](#)
 ClearData, [105](#)
 ClearDataAuditItems, [106](#)
 clearFilters, [191](#)
 ClearInvalidData, [107](#)
 Close
 HFMwEncodedFileWriter component, [386](#)
 HFMwFileReader component, [383](#)
 ClosePeriodMultiple, [349](#)
 clusterName, [58](#)
 CollapseColumnDimension, [203](#)
 CollapseColumnMemberInDimension, [203](#)
 CollapseRowDimension, [204](#)
 CollapseRowMemberInDimension, [205](#)
 colPhases, [254](#)
 columnContents, [373](#)
 columnContentsOption, [373](#)
 columnDimensions, [254](#)
 columnRepeatOption, [374](#)
 columnSortOption, [374](#)
 columnTotalOption, [374](#)
 Consolidate, [186](#)
 ConsolidationMethodsTimeStamp, [68](#)
 consolStatusFilter, [255](#)
 constants
 CELLSTATUS_NOREADACCESS, [423](#)
 MEMBER_LIST_ALL_HIERARCHY, [418](#)
 MEMBERNOTUSED, [415](#)
 ConvertFromUTF8, [382](#)
 CopyData, [107](#)
 CopyDataToJournal, [342](#)
 Core.asp, [26](#)
 Count, [194](#), [288](#)
 CRC-32 checksum, getting for a string, [381](#)
 CreateApplication, [31](#)
 CreateApplicationCAS, [31](#)
 CreateFolder, [266](#)
 CreateFolderEx, [267](#)
 CreateJournal

 HFMwManageJournals component, [350](#)
 HFMwTemplate component, [342](#)
 CreateJournalFromTemplate, [351](#)
 CreateTemplate, [352](#)
 Creator Group, [33](#)
 currencies, [68](#)
 currencies, HFMwCurrencies component, [101](#)
 currency ID, obtaining, [103](#)
 currency subcubes, [23](#)
 CurrencyTimeStamp, [68](#)
 current module, [330](#)
 Custom, [194](#), [200](#)
 custom settings
 getting, [55](#)
 setting, [57](#)
 custom1, [68](#)
 Custom1TopMember account attribute, returning,
 [72](#)
 custom2, [68](#)
 Custom2TopMember account attribute, returning,
 [72](#)
 custom3, [68](#)
 Custom3TopMember account attribute, returning,
 [72](#)
 custom4, [69](#)
 Custom4TopMember account attribute, returning,
 [72](#)
 cyclic redundancy check, [381](#)

D

data audit
 ClearDataAuditItems, [106](#)
 EnumDataAuditItems, [110](#)
 EnumDataAuditItems2, [113](#)
 data forms, working with, [21](#)
 data, methods for, [105](#)
 dataDisplayMode, [255](#)
 decimalSeparator, [58](#)
 Default security class, constant for, [468](#)
 defaultDataGridDef, [59](#)
 DefaultFreq attribute, returning, [74](#)
 DefaultFreq attribute, returning periods for, [128](#)
 defaultFrequency, [74](#)
 defaultMemberID, [101](#)
 defaultProcessControlGridDef, [59](#)
 DefCurrency attribute, returning, [98](#)
 DefineGridUsingXML, [205](#)

DefineGridUsingXMLFromRequest, 206
 DefinePOVUsingXML, 206
 DeleteApplication, 32
 DeleteDocuments, 268
 DeleteJournal, 352
 DeleteJournals, 353
 DeleteLineItemDetails, 109
 DeleteSecurityClasses, 292
 DeleteTemplate, 353
 DeleteTemplates, 354
 DeleteXMLErrorsFromDatabase, 33
 Description, 288
 description
 HFMwJournal component, 336
 HFMwQueryDef component, 374
 HFMwTemplate component, 343
 descriptionFilter, 374
 descriptionLanguage, 59
 destination transactions, getting, 224
 Dim, 195
 dimension
 HFMwAccounts component, 73
 HFMwCustom1 component, 81
 HFMwCustom2 component, 81
 HFMwCustom3 component, 82
 HFMwCustom4 component, 82
 HFMwEntities component, 79
 HFMwICPs component, 80
 HFMwMetadata component, 69
 HFMwPeriods component, 78
 HFMwScenarios component, 74
 HFMwValues component, 80
 HFMwViews component, 80
 HFMwYears component, 74
 dimensions, constants that represent, 414
 dimid, Point of View attribute, 460
 Directive, 195
 directive, named, HFMwMbrSel object, 194
 displayed data, returning, 221
 document repository, 265
 DoesConnectedUserHaveApplicationAdminRights, 292
 DoesDocumentExist, 269
 DoesUserHaveCreateApplicationRights, 33
 DoesUserHaveSystemAdminRights, 33
 domain, 52
 Dump, 195, 200

E
 e-mail alerting, 216
 enabled, Point of View attribute, 460
 EnabledOnly, 196
 EnableNewConnections
 HFMwManageApplications component, 34
 HFMwSystemInfo component, 314
 EnableOrDisableUseSecurityAsPartner, 64
 encoding, UTF-16, 59
 entities (property), 69
 entities, active, displaying, 249
 Entity Transaction Detail reports, 124
 entityFilter, 375
 EnumActivityServers, 314
 EnumActivityUsers, 315
 EnumApplicationNames, 35
 EnumAuditTasks, 315
 EnumAuditTasks2, 317
 EnumChildren, 83
 EnumChildrenAsXML, 85
 EnumClusterNames, 35
 EnumColumnHeaderInfo, 206
 EnumCurrencies, 102
 EnumDataAuditItems, 110
 EnumDataAuditItems2, 113
 EnumDescriptionLanguages, 65
 EnumDimensions, 65
 EnumDocuments
 HFMwManageDocuments component, 269
 HFMwWorkspace component, 283
 EnumDocumentsEx, 270
 EnumJournalGroups, 354
 EnumMemberLists, 86
 EnumMemberListsAsXML, 87
 EnumMembers, 88
 EnumMembers2, 90
 EnumMembersAsXML, 92
 EnumParents, 93
 EnumProhibitConnections
 HFMwManageApplications component, 35
 HFMwSystemInfo component, 319
 EnumProvisioningProjectNames, 37
 EnumRoles, 292
 EnumRowHeaderInfo, 207
 EnumRunningTasks, 321
 EnumRunningTasksEx, 324
 EnumSecurityClasses, 293

EnumSecurityClassesForConnectedUser, [294](#)
 EnumSecurityClassRights, [294](#)
 EnumSecurityClassRightsAndRoles, [296](#)
 EnumSecurityClassRightsAndRolesForBiPub, [297](#)
 EnumSecurityClassRightsForBiPub, [295](#)
 EnumUserAppPreferences, [37](#)
 EnumUsers, [298](#)
 EnumUsers2, [298](#)
 EnumUsersInGroup, [298](#)
 EnumUsersInGroupForBiPub, [301](#)
 EnumUsersOnSystem, [37](#)
 EnumUsersOnSystemEx, [40](#)
 EnumUsersOnSystemEx2, [42](#)
 EnumUsersRoles, [299](#)
 EnumUsersRolesForBiPub, [300](#)
 EnumUsersWithFilter, [302](#)
 Err object, Financial Management error handling and, [392](#)
 error handling, [391](#)
 error handling, reserved error numbers, [25](#)
 error numbers, utility for, [400](#)
 ExecuteAction, [207](#)
 ExecuteDataExplorerAction, [207](#)
 ExecuteDataExplorerProcessManagementTask, [208](#)
 ExecuteDataExplorerProcessManagementTask2, [210](#)
 ExecuteDataExplorerProcessManagementTask3, [211](#)
 ExecuteQuery, [355](#)
 ExecuteQueryLabelsAndIDs, [356](#)
 ExpandColumnDimension, [213](#)
 ExpandColumnMemberInDimension, [214](#)
 ExpandGridUsingXML, [214](#)
 ExpandRowDimension, [215](#)
 ExpandRowMemberInDimension, [215](#)
 external authentication
 logging on, [50](#)
 token, getting or validating, [51](#)
 extractEncoding, [59](#)

F

files
 reading, [383](#)
 writing, [386](#)
 files, extracting in Unicode, [59](#)
 FileType, [288](#)
 filter, [375](#)
 filterhidden, Point of View attribute, [460](#)
 filterlist, Point of View attribute, [460](#)

filterlocked, Point of View attribute, [460](#)
 filterpath, Point of View attribute, [460](#)
 filterroot, Point of View attribute, [460](#)
 filtertop, Point of View attribute, [460](#)
 FindMemberOccurrencesInHierarchy, [95](#)
 FindMemberOccurrencesInHierarchyEx, [96](#)
 frequency
 returning internal ID of, [77](#)
 returning periods in, [77](#)
 scenario's default, returning, [74](#)
 full resolution data, returning, [221](#)

G

GenerateAlerts, [216](#)
 GenerateJournalFromRecurringTemplate, [357](#)
 GetActivityCodeDesc, [327](#)
 GetAncestorAtFrequency, [75](#)
 GetAsXML, [334](#), [342](#)
 GetAttributeValue, [97](#)
 GetAvailableLanguages, [393](#)
 GetBlankWorkspaceDocument, [283](#)
 GetCalcStatusSummaryAsXML, [218](#)
 GetCellAdjustments, [219](#)
 GetCellAdjustmentsAsXML, [222](#)
 GetCellAttachmentsAsXML, [220](#)
 GetCellHistory, [118](#)
 GetCellHistory2, [120](#)
 GetCellInfo, [220](#)
 GetCellInfoAsXML, [222](#)
 GetCellLineItemDetailsAsXML, [223](#)
 GetCellText, [223](#)
 GetCellTextAsXML, [223](#)
 GetConsolidationProgress, [224](#)
 GetCurrencyID, [103](#)
 GetCurrencyTranslationOperator, [103](#)
 GetCurrentVersion, [394](#)
 GetDefAsVariants, [371](#)
 GetDefaultCurrenciesForEntities, [78](#)
 GetDefaultParent, [98](#)
 GetDescendantsAtFrequency, [76](#)
 GetDestinationTransactions, [224](#)
 GetDisableAspTechnicalErrorMessage, [44](#)
 GetDMELListenerConnectionInfo, [44](#)
 GetDocument, [273](#)
 GetDocumentAt, [284](#)
 GetDocumentEx, [274](#)
 GetDoubleFromText, [123](#)

[GetEntityDetails](#), 124
[GetEntityDetailsHeader](#), 126
[GetErrorListCount](#), 44
[GetErrorListInRange](#), 44
[GetFormattedDateTime](#), 394
[GetFormattedError](#), 395
[GetFormattedErrorWithLineFeed](#), 396
[GetFormattedNumber](#), 127
[GetFormattedResourceString](#), 397
[GetFrequency](#), 77
[GetGridDefinitionAsXML](#), 226
[GetGridDefinitionAsXMLeX](#), 227
[GetGridPage](#), 228
[GetGridPageWithExpansionInfo](#), 229
[GetGridPageWithExpansionInfoAsJavaScript](#), 231
[GetGridPageWithExpansionInfoAsXML](#), 232
[GetHelpDirectoryForLanguageID](#), 397
[GetHFMErrLogRecordSet](#), 44
[GetHFMLanguageIdFromUserLanguages](#), 398
[GetHistory](#), 141
[GetHistory2](#), 144
[GetHistory2UsingPhaseId](#), 145
[GetInputPeriods](#), 128
[GetJournal](#), 357
[GetJournalFromXML](#), 358
[GetJournalsDisplayData](#), 358
[GetJournalsDisplayDataFromIDs](#), 360
[GetJournalUsingIDs](#), 361
[GetKillUserStatus](#), 327
[GetLanguageCountryCodeFromLanguageId](#), 398
[GetLicenseExpirationStatus](#), 54
[GetMemberDescription](#), 99
[GetMemberID](#), 99
[GetMemberLabel](#), 100
[GetNumberOfLineItemEntries](#), 334
[GetNumberOfLineItems](#), 334, 342
[GetOverlappingConsolidationInfoAsXML](#), 232
[GetOwner](#), 303
[GetPeriodsInFrequency](#), 77
[GetPeriodStatusMultiple](#), 361
[GetPhaseGroupInfoAsJavaScript](#), 234
[GetPhaseSubmissionGroupAndPhaseForCell](#), 147
[GetPhaseSubmissionReviewLevel](#), 148
[GetPOVAsXML](#), 234
[GetPOVByTag](#), 54
[GetProcessGridPageMultiWithExpansionInfoAsJavaScript](#), 235
[GetProcessGridPageWithExpansionInfoAsJavaScript](#), 235
[GetProcessManagementGridPage](#), 236
[GetProcessManagementRoleAccess](#), 150
[GetProcessManagementSummary](#), 237
[GetRegionsForCell](#), 129
[GetReportDefinition](#), 371
[GetResourceString](#), 398
[GetResourceStringFromHR](#), 398
[GetReviewLevel](#), 150
[GetReviewLevelUsingPhaseID](#), 151
[GetRulesMode](#), 303
[GetRunningTaskLogFromServerFile](#), 327
[GetRunningTaskProgress](#), 328
[GetRunningTaskStatus](#), 328
[GetSecurityClassAccessForAllUsers](#), 303
[GetSecurityClassAccessForAllUsers2](#), 303
[GetSecurityClassID](#), 304
[GetSecurityClassRightsForConnectedUser](#), 304
[GetSourceTransactions](#), 238
[GetSSOTokenUsingWebSecurityAgentCredentials](#), 44
[GetTaskNameAndIdByUrl](#), 285
[GetTaskNameAndUrlById](#), 285
[GetTemplate](#), 362
[GetTemplateFromXML](#), 363
[GetTemplatesDisplayData](#), 363
[GetTemplatesDisplayDataFromIDs](#), 365
[GetTextCellLineItems](#), 130
[GetTopMemberOfCustomsForAccount](#), 72
[GetTransactionInfoAsXML](#), 240
[GetUnassignedGroups](#), 241
[GetUserAccessForAllSecurityClasses](#), 305
[GetUserAccessForAllSecurityClasses2](#), 305
[GetUserDisplayDataTimeFormats](#), 399
[GetUserID](#), 306
[GetUserInfoFromUniqueID](#), 306
[GetUserInfoFromUniqueID2](#), 306
[GetUserLanguageFromHFMLanguageId](#), 399
[GetUserSetting](#), 55
[GetUserSID](#), 307
[GetVariance](#), 365
[GetVarianceForTemplate](#), 366
[GetWindowsDateFormat](#), 400
[GetWorkspaceDocument](#), 285
[GetXMLErrorFromDatabase](#), 45
[GetXMLErrorsListFromDatabase](#), 45

gridNumDecimals, [255](#)

gridScale, [255](#)

group

 HFMwJournal component, [336](#)

 HFMwTemplate component, [343](#)

groupFilter, [375](#)

H

HasSystemChanged, [55](#)

Height, [289](#)

HFM_USER_GROUP_ENUM_FLAGS, [446](#)

HFMwAccounts

 dimension, [73](#)

 GetTopMemberOfCustomsForAccount, [72](#)

 object reference, obtaining, [72](#)

HFMwCalculate

 Allocate, [185](#)

 ChartLogic, [185](#)

 Consolidate, [186](#)

 object reference, obtaining, [184](#)

 SetWebSession, [187](#)

 Translate, [187](#)

HFMwCurrencies

 EnumCurrencies, [102](#)

 GetCurrencyID, [103](#)

 GetCurrencyTranslationOperator, [103](#)

 object reference, obtaining, [101](#)

HFMwCustom1

 dimension, [81](#)

 object reference, obtaining, [81](#)

HFMwCustom2

 dimension, [81](#)

 object reference, obtaining, [81](#)

HFMwCustom3

 dimension, [82](#)

 object reference, obtaining, [81](#)

HFMwCustom4

 dimension, [82](#)

 object reference, obtaining, [82](#)

HFMwData

 ClearData, [105](#)

 ClearDataAuditItems, [106](#)

 ClearInvalidData, [107](#)

 CopyData, [107](#)

 DeleteLineItemDetails, [109](#)

 EnumDataAuditItems, [110](#)

 EnumDataAuditItems2, [113](#)

GetCellHistory, [118](#)

GetCellHistory2, [120](#)

GetDoubleFromText, [123](#)

GetEntityDetails, [124](#)

GetEntityDetailsHeader, [126](#)

GetFormattedNumber, [127](#)

GetInputPeriods, [128](#)

GetRegionsForCell, [129](#)

GetTextCellLineItems, [130](#)

InsertLineItemDetails, [131](#)

object reference, obtaining, [105](#)

SetTextCellLineItems, [132](#)

SetTextCellsLineItems, [134](#)

SetWebSession, [135](#)

HFMwDataGrid

 CollapseColumnDimension, [203](#)

 CollapseColumnMemberInDimension, [203](#)

 CollapseRowDimension, [204](#)

 CollapseRowMemberInDimension, [205](#)

colPhases, [254](#)

columnDimensions, [254](#)

consolStatusFilter, [255](#)

dataDisplayMode, [255](#)

DefineGridUsingXML, [205](#)

DefineGridUsingXMLFromRequest, [206](#)

DefinePOVUsingXML, [206](#)

EnumColumnHeaderInfo, [206](#)

EnumRowHeaderInfo, [207](#)

ExecuteAction, [207](#)

ExecuteDataExplorerAction, [207](#)

ExecuteDataExplorerProcessManagementTask,
[208](#)

ExecuteDataExplorerProcessManagementTask2,
[210](#)

ExecuteDataExplorerProcessManagementTask3,
[211](#)

ExpandColumnDimension, [213](#)

ExpandColumnMemberInDimension, [214](#)

ExpandGridUsingXML, [214](#)

ExpandRowDimension, [215](#)

ExpandRowMemberInDimension, [215](#)

GenerateAlerts, [216](#)

GetCalcStatusSummaryAsXML, [218](#)

GetCellAdjustments, [219](#)

GetCellAdjustmentsAsXML, [222](#)

GetCellAttachmentsAsXML, [220](#)

GetCellInfo, [220](#)

- GetCellInfoAsXML, [222](#)
- GetCellLineItemDetailsAsXML, [223](#)
- GetCellText, [223](#)
- GetCellTextAsXML, [223](#)
- GetConsolidationProgress, [224](#)
- GetDestinationTransactions, [224](#)
- GetGridDefinitionAsXML, [226](#)
- GetGridDefinitionAsXMLEx, [227](#)
- GetGridPage, [228](#)
- GetGridPageWithExpansionInfo, [229](#)
- GetGridPageWithExpansionInfoAsJavaScript, [231](#)
- GetGridPageWithExpansionInfoAsXML, [232](#)
- GetOverlappingConsolidationInfoAsXML, [232](#)
- GetPhaseGroupInfoAsJavaScript, [234](#)
- GetPOVAsXML, [234](#)
- GetProcessGridPageMultiWithExpansionInfoAsJavaScript, [235](#)
- GetProcessGridPageWithExpansionInfoAsJavaScript, [235](#)
- GetProcessManagementGridPage, [236](#)
- GetProcessManagementSummary, [237](#)
- GetSourceTransactions, [238](#)
- GetTransactionInfoAsXML, [240](#)
- GetUnassignedGroups, [241](#)
- gridNumDecimals, [255](#)
- gridScale, [255](#)
- hPageCount, [255](#)
- listStyle, [256](#)
- maxColsPerPage, [256](#)
- maxRowsPerPage, [257](#)
- memberID, [257](#)
- memberListID, [258](#)
- metadataDisplayMode, [258](#)
- object reference, obtaining, [201](#)
- pages in grids, [202](#)
- parentID, [259](#)
- PassFailFilter, [259](#)
- phaseID, [259](#)
- processCtrlDispCols, [259](#)
- ProcessFlowGetInfo, [241](#)
- ProcessFlowGetInfo2, [243](#)
- ProcessFlowGetInfoAsXML, [246](#)
- ReviewLevelFilter, [259](#)
- ReviewLevelFilterCondition, [260](#)
- ReviewLevelSortOrder, [260](#)
- rowDimensions, [260](#)
- SaveDirtyCellsOnLastGridPage, [246](#)
- SaveDirtyCellsOnLastGridPageFromXML, [247](#)
- SaveLastGridPage, [247](#)
- SavePhaseGroupInfoUsingXML, [248](#)
- SaveUIStateUsingXML, [248](#)
- SaveUIStateUsingXMLFromRequest, [248](#)
- selectedMemberIDs, [261](#)
- selectedParentIDs, [261](#)
- SetOrgByPeriodFilteringInfo, [249](#)
- SetPhaseGroupInfoAsJavaScript, [249](#)
- SetProcessManagementFilterAndSortOptions, [249](#)
- SetRowandColumnCount, [250](#)
- SetWebSession, [251](#)
- StartConsolidation, [251](#)
- StopConsolidation, [252](#)
- SubmitProcessStateChangeFromXML, [252](#)
- suppressNodataColumns, [261](#)
- suppressNodataRows, [261](#)
- suppressZeroRows, [261](#)
- supressZeroColumns, [262](#)
- topMemberID, [262](#)
- totalColumnCount, [262](#)
- totalFilteredRowCount, [262](#)
- totalRowCount, [262](#)
- useList, [263](#)
- vFilteredPageCount, [263](#)
- vPageCount, [263](#)
- HFMwDimension
 - alias, [101](#)
 - defaultMemberID, [101](#)
 - EnumChildren, [83](#)
 - EnumChildrenAsXML, [85](#)
 - EnumMemberLists, [86](#)
 - EnumMemberListsAsXML, [87](#)
 - EnumMembers, [88](#)
 - EnumMembers2, [90](#)
 - EnumMembersAsXML, [92](#)
 - EnumParents, [93](#)
 - FindMemberOccurrencesInHierarchy, [95](#)
 - FindMemberOccurrencesInHierarchyEx, [96](#)
 - GetAttributeValue, [97](#)
 - GetDefaultParent, [98](#)
 - GetMemberDescription, [99](#)
 - GetMemberID, [99](#)
 - GetMemberLabel, [100](#)
 - IsMemberAChildOf, [66](#)
 - name, [101](#)
 - numMembers, [101](#)

- object reference, obtaining, [82](#)
- timestamp, [71](#)
- HFMwEncodedFileWriter
 - Close, [386](#)
 - Open, [386](#)
 - Write, [387](#)
 - WriteBinary, [387](#)
- HFMwEntities
 - dimension, [79](#)
 - GetDefaultCurrenciesForEntities, [78](#)
 - object reference, obtaining, [78](#)
- HFMwFileReader
 - Close, [383](#)
 - IsEOF, [383](#)
 - OpenAsBinary, [384](#)
 - OpenAsText, [384](#)
 - Read, [385](#)
 - ReadLine, [385](#)
 - Size, [385](#)
- HFMwICPs
 - dimension, [80](#)
 - object reference, obtaining, [80](#)
- HFMwJournal
 - balanceType, [336](#)
 - description, [336](#)
 - GetAsXML, [334](#)
 - GetNumberOfLineItemEntries, [334](#)
 - GetNumberOfLineItems, [334](#)
 - group, [336](#)
 - IsNewJournal, [335](#)
 - IsSingleEntity, [335](#)
 - label, [336](#)
 - liAccount, [336](#)
 - liAmount, [337](#)
 - liCustom1, [337](#)
 - liCustom2, [337](#)
 - liCustom3, [338](#)
 - liCustom4, [338](#)
 - liDebitCreditUnit, [338](#)
 - liDescription, [338](#)
 - liEntity, [339](#)
 - liICP, [339](#)
 - liParent, [339](#)
 - object reference, obtaining, [334](#)
 - oldLabel, [340](#)
 - period, [340](#)
 - scenario, [340](#)
 - security, [340](#)
 - SetNumberOfLineItemEntries, [335](#)
 - SetNumberOfLineItems, [335](#)
 - singleEntity, [340](#)
 - singleParent, [340](#)
 - status, [341](#)
 - type, [341](#)
 - value, [341](#)
 - year, [341](#)
- HFMwJournals type library, [333](#)
- HFMwManageApplications
 - CreateApplication, [31](#)
 - CreateApplicationCAS, [31](#)
 - DeleteApplication, [32](#)
 - DeleteXMLErrorsFromDatabase, [33](#)
 - DoesUserHaveCreateApplicationRights, [33](#)
 - DoesUserHaveSystemAdminRights, [33](#)
 - domain, [52](#)
 - EnableNewConnections, [34](#)
 - EnumApplicationNames, [35](#)
 - EnumClusterNames, [35](#)
 - EnumProhibitConnections, [35](#)
 - EnumProvisioningProjectNames, [37](#)
 - EnumUserAppPreferences, [37](#)
 - EnumUsersOnSystem, [37](#)
 - EnumUsersOnSystemEx, [40](#)
 - EnumUsersOnSystemEx2, [42](#)
 - GetDisableAspTechnicalErrorMessage, [44](#)
 - GetDMELListenerConnectionInfo, [44](#)
 - GetErrorListCount, [44](#)
 - GetErrorListInRange, [44](#)
 - GetHFMErrLogRecordSet, [44](#)
 - GetSSOTokenUsingWebSecurityAgentCredentials, [44](#)
 - GetXMLErrorFromDatabase, [45](#)
 - GetXMLErrorsListFromDatabase, [45](#)
 - IsValidApplication, [47](#)
 - IsWebSecurityAgentEnabled, [47](#)
 - KillUsers, [47](#)
 - OpenApplication, [48](#)
 - RegisterApplicationCAS, [49](#)
 - SetLogonInfo, [49](#)
 - SetLogonInfoSSO, [50](#)
 - UpdateUserAppPreferences, [51](#)
 - userName, [52](#)
 - ValidateTokenOnClusterSSO, [51](#)
 - WarnUsersForShutDown, [51](#)

HF MwManageDocuments

[CreateFolder](#), 266
[CreateFolderEx](#), 267
[DeleteDocuments](#), 268
[DoesDocumentExist](#), 269
[EnumDocuments](#), 269
[EnumDocumentsEx](#), 270
[GetDocument](#), 273
[GetDocumentEx](#), 274
[LoadDocuments](#), 275
[LoadDocumentsEx](#), 276
[object reference, obtaining](#), 265
[SaveDocument](#), 277
[SaveDocument2](#), 278
[SaveDocumentEx](#), 279
[SetWebSession](#), 281

HF MwManageJournals

[AddJournalGroup](#), 349
[ClosePeriodMultiple](#), 349
[CreateJournal](#), 350
[CreateJournalFromTemplate](#), 351
[CreateTemplate](#), 352
[DeleteJournal](#), 352
[DeleteJournals](#), 353
[DeleteTemplate](#), 353
[DeleteTemplates](#), 354
[EnumJournalGroups](#), 354
[ExecuteQuery](#), 355
[ExecuteQueryLabelsAndIDs](#), 356
[GenerateJournalFromRecurringTemplate](#), 357
[GetJournal](#), 357
[GetJournalFromXML](#), 358
[GetJournalsDisplayData](#), 358
[GetJournalsDisplayDataFromIDs](#), 360
[GetJournalUsingIDs](#), 361
[GetPeriodStatusMultiple](#), 361
[GetTemplate](#), 362
[GetTemplateFromXML](#), 363
[GetTemplatesDisplayData](#), 363
[GetTemplatesDisplayDataFromIDs](#), 365
[GetVariance](#), 365
[GetVarianceForTemplate](#), 366
[object reference, obtaining](#), 349
[OpenPeriodMultiple](#), 366
[PerformBatchAction](#), 367
[RemoveAllJournalGroups](#), 368
[RemoveJournalGroup](#), 368

[SaveJournal](#), 368
[SaveJournal2](#), 368
[SaveTemplate](#), 369
[SetWebSession](#), 369
[ValidateLineItems](#), 369

HF MwManageProcess

[Approve](#), 136
[Approve2](#), 137
[ChangeProcessManagementStateForMultipleEntities](#), 138
[ChangeProcessManagementStateForMultipleEntities2](#), 140
[GetHistory](#), 141
[GetHistory2](#), 144
[GetHistory2UsingPhaseId](#), 145
[GetPhaseSubmissionGroupAndPhaseForCell](#), 147
[GetPhaseSubmissionReviewLevel](#), 148
[GetProcessManagementRoleAccess](#), 150
[GetReviewLevel](#), 150
[GetReviewLevelUsingPhaseID](#), 151
[MapReviewLevelToString](#), 152
[object reference, obtaining](#), 135
[PhasedSubmissionProcessManagementChangeStateForMultipleEntities](#), 153
[PhaseSubmissionApprove](#), 155
[PhaseSubmissionGetHistory](#), 157
[PhaseSubmissionPromote](#), 159
[PhaseSubmissionReject](#), 161
[PhaseSubmissionSignOff](#), 163
[PhaseSubmissionStart](#), 165
[PhaseSubmissionSubmit](#), 167
[Promote](#), 169
[Promote2](#), 171
[Publish](#), 172
[Publish2](#), 173
[Reject](#), 175
[Reject2](#), 176
[SetWebSession](#), 177
[SignOff](#), 177
[SignOff2](#), 178
[Start](#), 179
[Start2](#), 180
[Submit](#), 182
[Submit2](#), 183

HF MwMbrSel

[AttrAll](#), 193
[AttrDim](#), 193

- buildValue, [190](#)
- clearFilters, [191](#)
- Count, [194](#)
- Custom, [194](#)
- Dim, [195](#)
- Directive, [195](#)
- Dump, [195](#)
- EnabledOnly, [196](#)
- NextSortOrder, [196](#)
- object reference, obtaining, [189](#)
- parseValue, [191](#)
- ScriptString, [197](#)
- setWebSession, [192](#)
- SortedOnly, [197](#)
- ValuedOnly, [197](#)
- VisibleOnly, [198](#)
- XmlDom, [198](#)
- XmlString, [198](#)
- HFMwMbrSelDim
 - Attr, [199](#)
 - Custom, [200](#)
 - Dump, [200](#)
 - object reference, obtaining, [199](#)
- HFMwMetadata
 - accounts, [67](#)
 - ApplicationAttribute, [67](#)
 - ApplicationCurrency, [67](#)
 - ApplicationSettingsTimeStamp, [67](#)
 - ApplicationVersion, [67](#)
 - ConsolidationMethodsTimeStamp, [68](#)
 - currencies, [68](#)
 - CurrencyTimeStamp, [68](#)
 - custom1, [68](#)
 - custom2, [68](#)
 - custom3, [68](#)
 - custom4, [69](#)
 - dimension, [69](#)
 - EnableOrDisableUseSecurityAsPartner, [64](#)
 - entities, [69](#)
 - EnumDescriptionLanguages, [65](#)
 - EnumDimensions, [65](#)
 - ICPs, [69](#)
 - IsInputCell, [69](#)
 - isOrgByPeriod, [70](#)
 - object reference, obtaining, [64](#)
 - periods, [70](#)
 - scenarios, [70](#)
 - SetWebSession, [66](#)
 - submissionPhaseFlag, [71](#)
 - ValidationAccount, [71](#)
 - ValidationAccountByPhase, [71](#)
 - values, [71](#)
 - views, [72](#)
 - years, [72](#)
- HFMwPeriods
 - dimension, [78](#)
 - GetAncestorAtFrequency, [75](#)
 - GetDescendantsAtFrequency, [76](#)
 - GetFrequency, [77](#)
 - GetPeriodsInFrequency, [77](#)
 - object reference, obtaining, [75](#)
- HFMwPOV, removed, [497](#)
- HFMwPOVDimension, removed, [497](#)
- HFMwQueryDef
 - balanceTypeFilter, [372](#)
 - columnContents, [373](#)
 - columnContentsOption, [373](#)
 - columnRepeatOption, [374](#)
 - columnSortOption, [374](#)
 - columnTotalOption, [374](#)
 - description, [374](#)
 - descriptionFilter, [374](#)
 - entityFilter, [375](#)
 - filter, [375](#)
 - GetDefAsVariants, [371](#)
 - GetReportDefinition, [371](#)
 - groupFilter, [375](#)
 - InitializeFromReportDefinition, [371](#)
 - label, [375](#)
 - labelFilter, [375](#)
 - object reference, obtaining, [370](#)
 - period, [375](#)
 - povString, [376](#)
 - ResetAll, [372](#)
 - ResetColumnContents, [372](#)
 - ResetFilters, [372](#)
 - scenario, [376](#)
 - security, [376](#)
 - SetWebSession, [372](#)
 - statusFilter, [377](#)
 - stylesheet, [377](#)
 - totalColumnCount, [377](#)
 - type, [377](#)
 - typeFilter, [378](#)

- value, [378](#)
- xml, [378](#)
- year, [378](#)
- HFMwScenarios
 - defaultFrequency, [74](#)
 - dimension, [74](#)
 - object reference, obtaining, [73](#)
- HFMwSecurity
 - AddSecurityClasses, [291](#)
 - AllowRulesLoadForEPMAApp, [292](#)
 - DeleteSecurityClasses, [292](#)
 - DoesConnectedUserHaveApplicationAdminRights, [292](#)
 - EnumRoles, [292](#)
 - EnumSecurityClasses, [293](#)
 - EnumSecurityClassesForConnectedUser, [294](#)
 - EnumSecurityClassRights, [294](#)
 - EnumSecurityClassRightsAndRoles, [296](#)
 - EnumSecurityClassRightsAndRolesForBiPub, [297](#)
 - EnumSecurityClassRightsForBiPub, [295](#)
 - EnumUsers, [298](#)
 - EnumUsers2, [298](#)
 - EnumUsersInGroup, [298](#)
 - EnumUsersInGroupForBiPub, [301](#)
 - EnumUsersRoles, [299](#)
 - EnumUsersRolesForBiPub, [300](#)
 - EnumUsersWithFilter, [302](#)
 - GetOwner, [303](#)
 - GetRulesMode, [303](#)
 - GetSecurityClassAccessForAllUsers, [303](#)
 - GetSecurityClassAccessForAllUsers2, [303](#)
 - GetSecurityClassID, [304](#)
 - GetSecurityClassRightsForConnectedUser, [304](#)
 - GetUserAccessForAllSecurityClasses, [305](#)
 - GetUserAccessForAllSecurityClasses2, [305](#)
 - GetUserID, [306](#)
 - GetUserInfoFromUniqueID, [306](#)
 - GetUserInfoFromUniqueID2, [306](#)
 - GetUserSID, [307](#)
 - IsClassicHFMAApplication, [307](#)
 - IsConnectedUserAllowedToPerformTask, [307](#)
 - IsConnectedUserInRole, [308](#)
 - LogInfo, [308](#)
 - object reference, obtaining, [291](#)
 - SaveSecurityClassRights, [308](#)
 - SetWebSession, [309](#)
- HFMwSession
 - applicationName, [58](#)
 - ApplyUserSettings, [53](#)
 - clusterName, [58](#)
 - decimalSeparator, [58](#)
 - defaultDataGridDef, [59](#)
 - defaultProcessControlGridDef, [59](#)
 - descriptionLanguage, [59](#)
 - extractEncoding, [59](#)
 - GetLicenseExpirationStatus, [54](#)
 - GetPOVByTag, [54](#)
 - GetUserSetting, [55](#)
 - HasSystemChanged, [55](#)
 - IsBusy, [60](#)
 - IsDebug, [60](#)
 - IsRunningTasks, [56](#)
 - language, [60](#)
 - metadata, [60](#)
 - object reference, obtaining, [53](#)
 - RemoveFile, [56](#)
 - RemovePOVByTag, [56](#)
 - RemoveUserSetting, [56](#)
 - resourceManager, [60](#)
 - session, [60](#)
 - SetPOVByTag, [57](#)
 - SetUserSetting, [57](#)
 - showsDescriptions, [61](#)
 - thousandsSeparator, [61](#)
 - userName, [61](#)
 - usesApplets, [61](#)
- HFMwSharesCalc
 - object reference, obtaining, [389](#)
 - SetWebSession, [389](#)
 - SharesCalculation, [390](#)
- HFMwStringUtility
 - CalcCRC32, [381](#)
 - ConvertFromUTF8, [382](#)
 - IsValidValue, [382](#)
 - ReverseUnicodeByteOrder, [383](#)
- HFMwSystemInfo
 - ClearAuditTasks, [311](#)
 - ClearAuditTasks2, [312](#)
 - EnableNewConnections, [314](#)
 - EnumActivityServers, [314](#)
 - EnumActivityUsers, [315](#)
 - EnumAuditTasks, [315](#)
 - EnumAuditTasks2, [317](#)
 - EnumProhibitConnections, [319](#)

- EnumRunningTasks, [321](#)
- EnumRunningTasksEx, [324](#)
- GetActivityCodeDesc, [327](#)
- GetKillUserStatus, [327](#)
- GetRunningTaskLogFromServerFile, [327](#)
- GetRunningTaskProgress, [328](#)
- GetRunningTaskStatus, [328](#)
- KillUsers, [329](#)
- object reference, obtaining, [311](#)
- SetCurrentModule, [330](#)
- SetCurrentModuleEx, [330](#)
- SetWebSession, [330](#)
- StopRunningTasks, [331](#)
- WarnUsersForShutDown, [331](#)
- HFMwTemplate
 - balanceType, [343](#)
 - CopyDataToJournal, [342](#)
 - CreateJournal, [342](#)
 - description, [343](#)
 - GetAsXML, [342](#)
 - GetNumberOfLineItems, [342](#)
 - group, [343](#)
 - IsSingleEntity, [342](#)
 - label, [343](#)
 - liAccount, [344](#)
 - liAmount, [344](#)
 - liCustom1, [344](#)
 - liCustom2, [344](#)
 - liCustom3, [345](#)
 - liCustom4, [345](#)
 - liDebitCreditUnit, [345](#)
 - liDescription, [346](#)
 - liEntity, [346](#)
 - liICP, [346](#)
 - liParent, [347](#)
 - object reference, obtaining, [341](#)
 - oldLabel, [347](#)
 - security, [347](#)
 - SetNumberOfLineItems, [343](#)
 - singleEntity, [347](#)
 - singleParent, [348](#)
 - templateType, [348](#)
 - type, [348](#)
 - valueDimension, [348](#)
- HFMwValues
 - dimension, [80](#)
 - object reference, obtaining, [79](#)
- HFMwViews
 - dimension, [80](#)
 - object reference, obtaining, [80](#)
- HFMwWorkspace
 - AddDocument, [282](#)
 - AddTask, [282](#)
 - Count, [288](#)
 - Description, [288](#)
 - EnumDocuments, [283](#)
 - FileType, [288](#)
 - GetBlankWorkspaceDocument, [283](#)
 - GetDocumentAt, [284](#)
 - GetTaskNameAndIdByUrl, [285](#)
 - GetTaskNameAndUrlById, [285](#)
 - GetWorkspaceDocument, [285](#)
 - Height, [289](#)
 - IsDocumentExists, [285](#)
 - Name, [289](#)
 - object reference, obtaining, [281](#)
 - ParseWorkspaceDocument, [286](#)
 - Path, [289](#)
 - Private, [289](#)
 - RemoveDocument, [286](#)
 - RemoveDocumentAt, [287](#)
 - SecClass, [289](#)
 - SetWebSession, [287](#)
 - ShowNames, [289](#)
 - SubWorkspace, [289](#)
 - SwapDocPosition, [288](#)
 - TaskFlow, [289](#)
 - Type, [290](#)
 - Width, [290](#)
 - xml, [290](#)
- HFMwYears
 - dimension, [74](#)
 - object reference, obtaining, [74](#)
- hPageCount, [255](#)
- HRESULTS, utility for, [400](#)
- HsvResourceManager
 - GetAvailableLanguages, [393](#)
 - GetFormattedError, [395](#)
 - GetString, [398](#)
 - GetStringFromHR, [398](#)
 - Initialize, [400](#)
 - object reference, obtaining, [392](#)
- HsvResourceManager object
 - GetCurrentVersion, [394](#)

GetFormattedDateTime, [394](#)
 GetFormattedErrorWithLineFeed, [396](#)
 GetFormattedResourceString, [397](#)
 GetHelpDirectoryForLanguageID, [397](#)
 GetHFMLanguageIdFromUserLanguages, [398](#)
 GetLanguageCountryCodeFromLanguageId, [398](#)
 GetUserDisplayDataTimeFormats, [399](#)
 GetUserLanguageFromHFMLanguageId, [399](#)
 GetWindowsDateFormat, [400](#)

I

ICPs, [69](#)
 identity types, [445](#)
 Initialize, [400](#)
 InitializeFromReportDefinition, [371](#)
 input periods for scenario, returning, [128](#)
 input/output arguments, [26](#)
 InsertLineItemDetails, [131](#)
 Intercompany Partner member attributes, constants
 for, [416](#)
 IsBusy, [60](#)
 IsClassicHFMAApplication, [307](#)
 IsConnectedUserAllowedToPerformTask, [307](#)
 IsConnectedUserInRole, [308](#)
 IsDebug, [60](#)
 IsDocumentExists, [285](#)
 IsEOF, [383](#)
 IsInputCell, [69](#)
 IsMemberAChildOf, [66](#)
 IsNewJournal, [335](#)
 isOrgByPeriod, [70](#)
 IsRunningTasks, [56](#)
 IsSingleEntity
 HFMAJournal component, [335](#)
 HFMATemplate component, [342](#)
 IsValidApplication, [47](#)
 IsValidValue, [382](#)
 IsWebSecurityAgentEnabled, [47](#)

J

journal adjustments to cells, returning, [219](#)
 as XML, [222](#)

K

KillUsers, [47](#), [329](#)

L

label, [375](#)
 HFMAJournal component, [336](#)
 HFMATemplate component, [343](#)
 labelFilter, [375](#)
 language, [60](#)
 liAccount
 HFMAJournal component, [336](#)
 HFMATemplate component, [344](#)
 liAmount
 HFMAJournal component, [337](#)
 HFMATemplate component, [344](#)
 liCustom1
 HFMAJournal component, [337](#)
 HFMATemplate component, [344](#)
 liCustom2
 HFMAJournal component, [337](#)
 HFMATemplate component, [344](#)
 liCustom3
 HFMAJournal component, [338](#)
 HFMATemplate component, [345](#)
 liCustom4
 HFMAJournal component, [338](#)
 HFMATemplate component, [345](#)
 liDebitCreditUnit
 HFMAJournal component, [338](#)
 HFMATemplate component, [345](#)
 liDescription
 HFMAJournal component, [338](#)
 HFMATemplate component, [346](#)
 liEntity
 HFMAJournal component, [339](#)
 HFMATemplate component, [346](#)
 liICP
 HFMAJournal component, [339](#)
 HFMATemplate component, [346](#)
 line item details, methods for, [105](#)
 liParent
 HFMAJournal component, [339](#)
 HFMATemplate component, [347](#)
 listStyle, [256](#)
 LoadDocuments, [275](#)
 LoadDocumentsEx, [276](#)
 log, writing to, [308](#)
 logging on, [50](#)
 LogInfo, [308](#)

M

MapReviewLevelToString, [152](#)
 maxColsPerPage, [256](#)
 maxRowsPerPage, [257](#)
 MEMBER_LIST_ALL_HIERARCHY, [418](#)
 memberID, [257](#)
 memberListID, [258](#)
 MEMBERNOTUSED, [415](#)
 MessageDisplayFunctions.asp, [26](#)
 metadata
 HFMwMetadata type library, [63](#)
 status, constants for, [422](#)
 metadata (property), [60](#)
 metadataDisplayMode, [258](#)
 module, current, [330](#)

N

Name, [289](#)
 name
 HFMwDimension component, [101](#)
 Point of View attribute, [460](#)
 named directive, HFMwMbrSel object, [194](#)
 NextSortOrder, [196](#)
 node subcubes, [23](#)
 numMembers, [101](#)

O

oldLabel
 HFMwJournal component, [340](#)
 HFMwTemplate component, [347](#)
 Open, [386](#)
 OpenApplication, [48](#)
 OpenAsBinary, [384](#)
 OpenAsText, [384](#)
 OpenPeriodMultiple, [366](#)
 order, Point of View attribute, [460](#)
 Organization by Period applications
 filtering entities for, [249](#)
 indicating, [70](#)
 ownership, calculating, [390](#)

P

pages, HFMwDataGrid component, [202](#)
 parentID, [259](#)
 parseValue, [191](#)
 ParseWorkspaceDocument, [286](#)

PassFailFilter, [259](#)
 Path, [289](#)
 PerformBatchAction, [367](#)
 period, [340](#), [375](#)
 periods, [70](#)
 phased submissions
 enabled, testing whether, [71](#)
 validation accounts, [71](#)
 PhasedSubmissionProcessManagementChangeState
 ForMultipleEntities, [153](#)
 phaseID, [259](#)
 PhaseSubmissionApprove, [155](#)
 PhaseSubmissionGetHistory, [157](#)
 PhaseSubmissionPromote, [159](#)
 PhaseSubmissionReject, [161](#)
 PhaseSubmissionSignOff, [163](#)
 PhaseSubmissionStart, [165](#)
 PhaseSubmissionSubmit, [167](#)
 Point of View
 components for, [189](#)
 persisting past current session, [189](#), [198](#)
 povString, [376](#)
 preferences, user, [53](#)
 changing, example, [28](#)
 Private, [289](#)
 process management
 constants for actions, [433](#)
 constants for review levels, [434](#)
 methods for, [135](#)
 process units, [24](#)
 processCtrlDispCols, [259](#)
 ProcessFlowGetInfo, [241](#)
 ProcessFlowGetInfo2, [243](#)
 ProcessFlowGetInfoAsXML, [246](#)
 processing instruction, HFMwMbrSel object, [194](#)
 Promote, [169](#)
 Promote2, [171](#)
 Publish, [172](#)
 Publish2, [173](#)

R

Read, [385](#)
 ReadLine, [385](#)
 RegisterApplicationCAS, [49](#)
 Reject, [175](#)
 Reject2, [176](#)
 RemoveAllJournalGroups, [368](#)

- RemoveDocument, [286](#)
 - RemoveDocumentAt, [287](#)
 - RemoveFile, [56](#)
 - RemoveJournalGroup, [368](#)
 - RemovePOVByTag, [56](#)
 - RemoveUserSetting, [56](#)
 - reports
 - journal, creating from column, filtering, and sorting criteria, [371](#)
 - working with, [265](#)
 - ResetAll, [372](#)
 - ResetColumnContents, [372](#)
 - ResetFilters, [372](#)
 - Resource Manager, [392](#)
 - resourceManager, [60](#)
 - ReverseUnicodeByteOrder, [383](#)
 - review levels, constants for, [434](#)
 - ReviewLevelFilter, [259](#)
 - ReviewLevelFilterCondition, [260](#)
 - ReviewLevelSortOrder, [260](#)
 - rowDimensions, [260](#)
- S**
- SaveDirtyCellsOnLastGridPage, [246](#)
 - SaveDirtyCellsOnLastGridPageFromXML, [247](#)
 - SaveDocument, [277](#)
 - SaveDocument2, [278](#)
 - SaveDocumentEx, [279](#)
 - SaveJournal, [368](#)
 - SaveJournal2, [368](#)
 - SaveLastGridPage, [247](#)
 - SavePhaseGroupInfoUsingXML, [248](#)
 - SaveSecurityClassRights, [308](#)
 - SaveTemplate, [369](#)
 - SaveUIStateUsingXML, [248](#)
 - SaveUIStateUsingXMLFromRequest, [248](#)
 - scenario, [340](#), [376](#)
 - default frequency, returning, [74](#)
 - HFMwScenarios component, [73](#)
 - scenarios (property), [70](#)
 - ScriptString, [197](#)
 - SecClass, [289](#)
 - security, [376](#)
 - HFMwJournal component, [340](#)
 - HFMwTemplate component, [347](#)
 - security agent, Web
 - logging on, [44](#)
 - testing whether enabled, [47](#)
 - security class, Default, constant for, [468](#)
 - security classes, enumerating, [293](#)
 - security identifier, obtaining, [307](#)
 - security, HFMwSecurity component, [291](#)
 - selected, Point of View attribute, [460](#)
 - selectedMemberIDs, [261](#)
 - selectedParentIDs, [261](#)
 - session, [60](#)
 - SetCurrentModule, [330](#)
 - SetCurrentModuleEx, [330](#)
 - SetLogonInfo, [49](#)
 - SetLogonInfoSSO, [50](#)
 - SetNumberOfLineItemEntries, [335](#)
 - SetNumberOfLineItems, [335](#), [343](#)
 - SetOrgByPeriodFilteringInfo, [249](#)
 - SetPhaseGroupInfoAsJavaScript, [249](#)
 - SetPOVByTag, [57](#)
 - SetProcessManagementFilterAndSortOptions, [249](#)
 - SetRowandColumnCount, [250](#)
 - SetTextCellLineItems, [132](#)
 - SetTextCellsLineItems, [134](#)
 - settings, custom
 - getting, [55](#)
 - setting, [57](#)
 - SetUserSetting, [57](#)
 - SetWebSession
 - HFMwCalculate component, [187](#)
 - HFMwData component, [135](#)
 - HFMwDataGrid component, [251](#)
 - HFMwManageDocuments component, [281](#)
 - HFMwManageJournals component, [369](#)
 - HFMwManageProcess component, [177](#)
 - HFMwMetadata component, [66](#)
 - HFMwQueryDef component, [372](#)
 - HFMwSecurity component, [309](#)
 - HFMwSharesCalc component, [389](#)
 - HFMwSystemInfo component, [330](#)
 - HFMwWorkspace component, [287](#)
 - setWebSession, [192](#)
 - Shared Services
 - returning provisioning project names, [37](#)
 - SharesCalculation, [390](#)
 - ShowNames, [289](#)
 - showsDescriptions, [61](#)
 - SID, obtaining, [307](#)
 - SignOff, [177](#)

- SignOff2, [178](#)
- single sign-on
 - logging on, [50](#)
 - token, getting or validating, [51](#)
- singleEntity
 - HFMwJournal component, [340](#)
 - HFMwTemplate component, [347](#)
- singleParent
 - HFMwJournal component, [340](#)
 - HFMwTemplate component, [348](#)
- SiteMinder
 - logging on, [44](#)
 - testing whether security agent support enabled, [47](#)
- Size, [385](#)
- SortedOnly, [197](#)
- source transactions, getting, [238](#)
- Start, [179](#)
- Start2, [180](#)
- StartConsolidation, [251](#)
- status, [341](#)
- statusFilter, [377](#)
- statutory applications
 - destination transactions, getting, [224](#)
 - source transactions, getting, [238](#)
 - transactions, getting as XML string, [240](#)
- StopConsolidation, [252](#)
- StopRunningTasks, [331](#)
- stored data, returning, [221](#)
- strings, methods for, [381](#)
- stylesheet, [377](#)
- subcubes, [23](#)
- submissionPhaseFlag, [71](#)
- Submit, [182](#)
- Submit2, [183](#)
- SubmitProcessStateChangeFromXML, [252](#)
- SubWorkspace, [289](#)
- suppressNodataColumns, [261](#)
- suppressNodataRows, [261](#)
- suppressZeroRows, [261](#)
- supressZeroColumns, [262](#)
- SwapDocPosition, [288](#)

T

- task audit
 - ClearAuditTasks, [311](#)
 - EnumAuditTasks, [315](#)
- task lists

- adding to other task lists, [289](#)
 - overview, [281](#)
- TaskFlow, [289](#)
- tasks, adding to task lists, [282](#)
- templateType, [348](#)
- thousandsSeparator, [61](#)
- timestamp, [71](#)
- token, external authentication, getting or validating, [51](#)
- topMemberID, [262](#)
- totalColumnCount
 - HFMwDataGrid component, [262](#)
 - HFMwQueryDef component, [377](#)
- totalFilteredRowCount, [262](#)
- totalRowCount, [262](#)
- transaction status
 - constants for, [425](#)
- transactions
 - as XML string, getting, [240](#)
 - destination, getting, [224](#)
 - source, getting, [238](#)
- Translate, [187](#)
- triplets, Value dimension, [23](#)
- Type, [290](#)
- type
 - HFMwJournal component, [341](#)
 - HFMwQueryDef component, [377](#)
 - HFMwTemplate component, [348](#)
- typeFilter, [378](#)

U

- Unicode, extracting files in, [59](#)
- UpdateUserAppPreferences, [51](#)
- useList, [263](#)
- user preferences, [53](#)
 - changing, example, [28](#)
- userName
 - HFMwManageApplications component, [52](#)
 - HFMwSession component, [61](#)
- usesApplets, [61](#)
- UTF-16, extracting files in, [59](#)

V

- val, Point of View attribute, [460](#)
- ValidateLineItems, [369](#)
- ValidateTokenOnClusterSSO, [51](#)

- ValidationAccount, [71](#)
- ValidationAccountByPhase, [71](#)
- value, [341](#), [378](#)
- Value dimension triplets, [23](#)
- valueDimension, [348](#)
- ValuedOnly, [197](#)
- values, [71](#)
- vFilteredPageCount, [263](#)
- views, [72](#)
- visible, Point of View attribute, [460](#)
- VisibleOnly, [198](#)
- vPageCount, [263](#)

W

- WarnUsersForShutDown
 - HFMwManageApplications component, [51](#)
 - HFMwSystemInfo component, [331](#)
- Web security agent
 - logging on, [44](#)
 - testing whether enabled, [47](#)
- Width, [290](#)
- Write, [387](#)
- WriteBinary, [387](#)

X

- xml
 - HFMwQueryDef component, [378](#)
 - HFMwWorkspace component, [290](#)
- XML strings, reference of, [473](#)
- XmlDom, [198](#)
- XmlString, [198](#)

Y

- year, [341](#), [378](#)
- years, [72](#)

A B C D E F G H I J K L M N O P R S T U V W X Y